Editor: Dana Simian

Proceedings of The International Conference on Applied Informatics

ICDD

IMAGINATION, CREATIVITY, DESIGN, DEVELOPMENT

May 5 - 7, 2022, Sibiu, Romania

Editor DANA SIMIAN

IMAGINATION, CREATIVITY, DESIGN, DEVELOPMENT

Proceedings of the International Conference
on Applied Informatics
ICDD

May 05th – 07th, 2022
Sibiu, Romania

Lucian Blaga University of Sibiu

Lucian Blaga University of Sibiu, 2022

Editor Dana Simian

All papers in this volume were peer review by two independent reviewers

ISSN-L 2069 – 864X

Associated Editor Laura Florentina Stoica

Proceedings of the International Conference on Applied Informatics, ICDD
May 05th – 07th, 2022, Sibiu, Romania

Preface

This volume contains refereed papers presented within the International Conference on Applied Informatics, “Imagination, Creativity, Design, Development“ - ICDD 2022, which was held between May 05th – 07th, at the Faculty of Sciences, Lucian Blaga University of Sibiu, Romania.

The conference is mainly addressed to young researchers from all over the world. The conference gives the participants the opportunity to discuss and present their research on informatics and related fields (like computational algebra, numerical calculus, bioinformatics, etc.). The conference welcomes submissions of original papers on all aspects of informatics and related fields ranging from new concepts and theoretical developments to advanced technologies and innovative applications. Specific topics of the conference included but are not restricted to: Algorithms and Data Structures, Graph Theory and Applications, Formal Languages and Compilers, Cryptography, Modelling and Simulation, Computer Programming, Computer Vision, Computer Graphics, Game Design, Data Mining, Distributed Computing, Artificial Intelligence, Service Oriented Applications, Networking, Grid Computing, Mobile Operating Systems, Scientific Computing, Software Engineering, Bioinformatics, Robotics, Computer Architecture, Evolutionary Computing, Multimedia Systems, Internet Communication and Technologies, Web Applications.

The conference has brought together participants from 4 countries (Germany, Romania, Serbia and USA).

We thank all the participants for their interesting talks and discussions. We also thank the members of the scientific committee for their help in reviewing the submitted papers and for their contributions to the scientific success of the conference.

September 2022

Dana Simian
Conference Chair

Motto:

“There are no limits, only your imagination”

Scientific committee

Kiril Alexiev - Bulgarian Academy of Sciences, Bulgaria
Vsevolod Arnaut - Moldova State University, Republic of Moldova
Galina Atanasova - Angel Kanchev University of Rousse
Alina Barbulescu - Ovidius University of Constanta, Romania
Arndt Balzer - University of Applied Sciences, Würzburg-Schweinfurt, Germany
Lasse Berntzen - Buskerud and Vestfold University College, Norway
Peter Braun - University of Applied Sciences, Würzburg-Schweinfurt, Germany
Amelia Bucur - Lucian Blaga University of Sibiu, Romania
Stelian Ciurea - Lucian Blaga University of Sibiu, Romania
Nicolae Constantinescu - Lucian Blaga University of Sibiu, Romania
Daniela Danciulescu - University of Craiova, Romania
Lyubomyr Demkiv - Lviv National Polytechnic University and Robotics Lead at SoftServe, Ukraine
Oleksandr Dorokhov - Kharkiv National University of Economics, Ukraine
Dmytro Dosyn - Institute of Computer Science and Information Technologies, Lviv Polytechnic National University
George Eleftherakis - The University of Sheffield International Faculty, City College Thessaloniki, Greece
Michael Emmerich - Leiden Institute of Advanced Computer Science, Leiden University
Calin Enachescu -University of Medicine, Pharmacy, Science and Technology of Targu Mures, Romania
Ralf Fabian - Lucian Blaga University of Sibiu, Romania
Tobias Fertig - University of Applied Sciences Würzburg-Schweinfurt
Stefka Fidanova - Bulgarian Academy of Sciences, Bulgaria
Ulrich Fiedler - Bern University of Applied Science, Switzerland
Adrian Florea - Lucian Blaga University of Sibiu, Romania
Teresa Gonçalves - University of Evora, Portugal
Andrina Granić - University of Split, Croatia
Katalina Grigorova - University of Ruse, Bulgaria
Piroska Haller - University of Medicine, Pharmacy, Science and Technology of Targu Mures, Romania
Daniel Hunyadi - Lucian Blaga University of Sibiu, Romania
Saleema JS - Chris University, Bangalore, India
Milena Lazarova - Technical University of Sofia, Bulgaria
Lixin Liang - Tsinghua University, Beijing, China
Suzana Loskovska - "Ss. Cyril and Methodius" University in Skopje, Republic of Macedonia
Rossitza S. Marinova - Concordia University of Edmonton, Canada
Gabriela Moise - Petroleum-Gas University of Ploiesti, Romania
G.Jose Moses - Raghu Engineering College Visakhapatnam, Andhra Pradesh, India

Mircea Musan - Lucian Blaga University of Sibiu, Romania
Mircea Iosif Neamtu - Lucian Blaga University of Sibiu, Romania
Elena Simona Nicoară - Petroleum-Gas University of Ploiesti
Grażyna Paliwoda-Pękosz - Cracow University of Economics, Poland
Camelia Pintea - Technical University Cluj-Napoca, Romania
Antoniu Pitic - Lucian Blaga University of Sibiu, Romania
Alina Pitic - Lucian Blaga University of Sibiu, Romania
Cristina Popirlan - University of Craiova, Romania
Anca Ralescu - University of Cincinnati, United States of America
Mohammad Rezai - Sheffield Hallam University, United Kingdom
Cosmin Sabo - Technical University of Cluj-Napoca - North University Center Baia Mare
José Saias - University of Evora, Portugal
Abdel-Badeeh M. Salem - Ain Shams University, Cairo, Egypt
Livia Sangeorzan - Transilvania University of Brasov, Romania
Soraya Sedkaoui - Khemis Miliana University, Algeria
Andreas Siebert - University of Applied Sciences Landshut, Germany
Ioan Silea - Politehnica University of Timișoara
Dana Simian - Lucian Blaga University of Sibiu, Romania
Petrica C. Pop Sitar - Technical University Cluj-Napoca, Romania
Lior Solomovich - Kaye Academic College of Education, Israel
Ansgar Steland - RWTH Aachen University, Germany
Florin Stoica - Lucian Blaga University of Sibiu, Romania
Laura Florentina Stoica - Lucian Blaga University of Sibiu, Romania
Detlef Streitferdt - Ilmenau University of Technology, Software Architectures and Product Lines Group, Germany
Grażyna Suchacka - University of Opole, Poland
Jolanta Tańcula - University of Opole, Poland
Milan Tuba - Singidunum University of Belgrade, Serbia
Eva Tuba - Singidunum University of Belgrade
Anca Vasilescu - Transilvania University of Brasov, Romania
Dana Vasiloaica - Institute of Technology Sligo, Ireland
Sofia Visa - The College of Wooster, United States

Contents

Sixth International Conference on Applied Informatics
Imagination, Creativity, Design, Development
ICDD 2022, May 5-7
Sibiu, Romania

Videos as Measure for Information Security Awareness at Fraunhofer Institute for Silicate Research

Hakan Arda, Kristin Weber

Abstract

The Fraunhofer-Gesellschaft is the world's leading organization for application-oriented research. For such an organization, information and its supporting systems represent important assets. Appropriate measures must therefore be developed and deployed to protect these assets. For this reason, this work introduces videos as an information security awareness measure at Fraunhofer ISC. In this way, a new method of conveying information was tested and prepared for future measures. The paper does not address the change in security awareness among Fraunhofer ISC employees. This would require a before-and-after study. Instead, a subjective opinion survey was conducted, which helped to record the interest in such media. Thanks to the numerous responses to the survey, a forecast for future video portfolios is presented that reflect the content desired by the employees.

1 Introduction

Life in the 21st century is characterized by the penetration of information and communication technology. Hardly any business process can do without IT support anymore. With increasing digitization, horizontal and vertical networking continues to grow. At the same time, the high level of penetration increases dependency and vulnerability to the continuously growing security threats, for example in the context of cyber security. Attackers often use the simplest methods to break into corporate IT systems. According to a study by Kaspersky Lab [8], most attackers use the obvious means, such as malware, phishing mails, or social engineering. In those attacks employees unintentionally give out data or information, or attackers use vulnerabilities and errors in software, to penetrate a company's internal systems. Because employees are the most chosen target for attacks, increasing awareness can increase a company's information security. For example, if employees recognize phishing e-mails, they can quickly report them to the responsible persons in the company. To raise this awareness, measures must be taken. In order to be effective, awareness videos must be targeted, i.e., adapted to the target group, e.g., in terms of age, language and level of knowledge [5]. They must also be entertaining and encourage people to deal with the issue of information security themselves. This paper examines the extent to which videos are suitable for conveying information security topics in a vivid way; the focus of the study is on creating videos in such a way that users find them informative and helpful. Fraunhofer Institute for Silicate Research (ISC) serves as a use case. The aim of this research was to create videos as an information security awareness measure for Fraunhofer ISC. The trigger for the video production was the creation of an information security manual. The guidelines and measures resulting from the information security manual should be made available to the employees in an understandable and entertaining way. The videos themselves were developed and produced using the PDCA cycle (Plan, Do, Check, Act). At the beginning of the paper, the basics of information security and security awareness are described. The current state of research is presented, and some technical terms are explained. Chapter 3 then describes the methodology. Here, the Fraunhofer ISC as a research institution and its staff are presented. The methodology also describes the approach to video production, by using the PDCA Cycle. In the next chapter, the creation of the awareness videos is elaborated in more detail by explaining the design decisions underlying the video production. For this purpose, the video "Detect phishing emails" is presented step by step. In Chapter 5, the results of the online survey are presented, evaluated, and critically examined. The last chapter finalizes this paper with a conclusion about the gained knowledge. In addition, future perspectives for the video portfolio and the possible subject areas are presented. It was not part of this work to measure or evaluate the competence of employees in terms of their security awareness. The feedback from the online survey is only intended to capture and reflect the subjective assessment of the viewers.

2 Information Security Awareness

For an adequate protection of information and IT systems, information security is based on the three principles of confidentiality, integrity, and availability [7]. Compliance with confidentiality means that unauthorized access to information by persons or processes is not permitted. Confidentiality is breached, for example, if a cyber-attack results in data being leaked to unauthorized third parties. Integrity measures ensure the completeness and correctness of values. Measures in this area therefore protect information from manipulation, such as inserting, deleting, or duplicating messages. Availability means that all information or values are available to authorized users when they are needed. For example, in the event of a server failure, the data stored in backups or similar must be available again as quickly as possible. Humans play an important role in ensuring these three principles of information security. Information security awareness (ISA) is the level of sensitivity in an organization or an individual [11]. It describes the degree to which individuals securely handle information and thus fulfill the three principles of information security. Information Security Awareness can be increased with the help of awareness measures. By increasing awareness, the handling of information becomes more secure and the risk of a violation of the protection goals for the company is reduced. To increase awareness, three steps are needed. First, all stakeholders in an organization must understand that there are threats to information security [1]. These threats need to be identified, e.g., through labels or warnings to raise awareness or attention. Then, it must be made clear that everyone can protect themselves from these threats. This can be done with the help of specific use cases or by introducing new security measures. The final step is to adapt the behavior of users. Only by changing behavior information security can be increased in the long term. Furthermore, the organization itself is an additional aspect of security awareness. It is responsible for ensuring that all employees could behave in a manner that is compliant with information security. The existing IT landscape should, for example, allow for sending encrypted e-mails or to organize documents according to their confidentiality classification. The interplay of these four aspects forms the level of awareness in a company. Security awareness – with the exception of the aspect of organization – is a product of cognitive and affective processes. For example, measures are often influenced by feelings, knowledge, or perception, and in turn influence future behavior. To positively change behavior, habit and salience are important in addition to the three aspects of organization, knowledge, and intention [11]. Salience describes a stimulus that is perceived by a person. It describes how strongly a stimulus stands out from its context and depends on how consciously it is perceived. Habits describes the subconscious performance of measures, which can be performed thoughtlessly through multiple repetition, similar to driving a car or riding a bicycle [4]. In the case of security awareness measures, it is necessary to determine which aspect of ISA the measure shall improve. For example, employees may already know that it is a high risk to send confidential documents unencrypted, but still do so by habit. In this case, it would be a good idea to point out that a confidential document must be sent encrypted by automatically displaying a reminder shortly before users send e-mails with attached documents. Another example is the use of secure passwords. Many users already know that a secure password is recommended, but they don't know how to create or change one. For this, detailed instructions on how to create secure passwords and where to change passwords would be helpful. The instructions explain step-by-step how the password change process works and what it is good for. By deciding which ISA aspect to improve, employees can focus on the desired behavior and improve it.

3 Methodology

3.1 Fraunhofer ISC

At Fraunhofer ISC, situated in Würzburg, Germany, individual material solutions, for example compostable food packaging [3], are developed for various applications. The IT department at Fraunhofer ISC consists of nine colleagues. The team takes care of all IT systems in the institute and provides first-level support. For the Fraunhofer-Gesellschaft, information and the processes, IT systems and communication networks that support it, represent important business assets. These business assets must be adequately protected and controlled. For this reason, the Fraunhofer-Gesellschaft has adopted and implemented an information security management system (ISMS) that applies to all its units including Fraunhofer ISC, based on the international standards DIN EN ISO/IEC 27001 and 27002. The establishment of a consistently structured and uniform information security manual is a key success factor of the ISMS: It defines technical and organizational measures, documents the degree of implementation, and provides a tangible concept for operational practice to guarantee the secure handling of information. The information security manual and the resulting set of guidelines and measures are the foundation for awareness measures. The measures to promote information security can take place via various media. Videos are a form of information transfer that is still largely unused at Fraunhofer ISC. To fulfill the processes and the execution of individual creation steps, an audit team consisting of the institute information security officer (ISO), the deputy ISO, the IT manager, and an auxiliary scientist in the field of information security was established. This team was also responsible for the evaluation of the videos in the PDCA cycle.

3.2 PDCA Cycle

The PDCA cycle was developed and disseminated as a basic principle for a continuous improvement process by William E. Deming in the 1950s [2]. This improvement cycle starts with the reflection of an initial situation (plan) and leads to the planning of solutions. These are then implemented with the help of concrete measures (do), which in turn are checked after execution and compared with the initially planned goals. This comparison makes it possible to find out what has worked as planned and what needs to be improved. Finally, problems are eliminated, and well working processes are expanded (act). The results and insights are included in the next improvement cycle, which starts over. The goal of this model is to examine all processes of an organization for opportunities for improvement, to eliminate waste through regular reflection on work processes, to collaboratively identify challenges, to develop creative solutions to problems, to consistently implement measures, and to continuously track results. Therefore, the PDCA model is particularly well suited as a control measure for the development of the ISMS and the awareness videos [1]. In the next chapters, the individual process steps of the PDCA cycle are examined in more detail and presented by means of an example.

3.3 Process of Video Creation

The content of the videos was to be based on the ISMS guidelines. To decide on the priorities for the upcoming measures, the audit team discussed and assessed the set of information security guidelines and measures. The discussion showed that the areas

- password security,
- phishing e-mails,
- encryption of e-mails, and
- behavior in the event of an IT emergency,

should be prioritized. Therefore, the video portfolio should cover those four most important topics relating to information security-compliant work at Fraunhofer ISC. Video production proceeded according to the PDCA cycle (cf. Fig. 1) with the aim to constantly improve the quality of the videos and to keep the production process efficient. Each cycle began with the preparation of the video material. The respective content was taken from the set of guidelines. A video script was created that provisionally summarized the content. This script was enhanced and compared with information from different sources, such as the BSI or Fraunhofer headquarters. Subsequently, the individual videos and the corresponding texts were recorded. This raw video material was processed using the video editing program Da Vinci Resolve and edited into a first version. This first version was used to check whether all the intended content was included and as a test sample for the audit team. The video was presented to the team, which expressed its wishes and suggestions for improvement. Aspects of the videos that had already been successfully implemented were emphasized and carried over into the next video cycle. As a result, the quality of the individual videos increased continuously. After the portfolio of four videos was completed, it was ready for institution-wide publication.

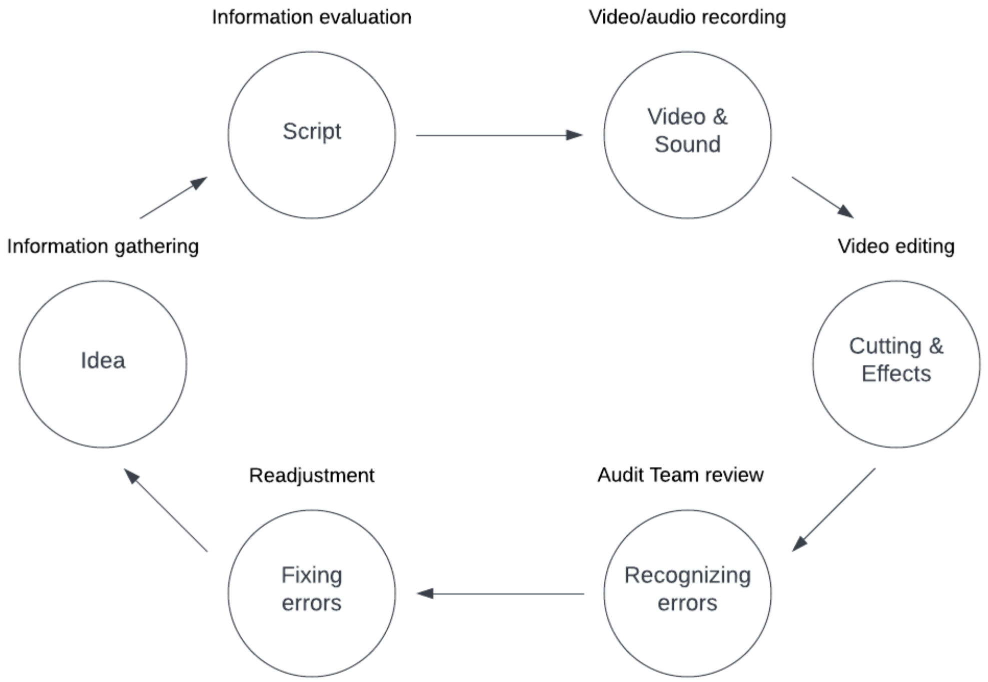


Fig. 1: PDCA cycle for creating the awareness videos

4 Creation of Awareness Videos

4.1 Design Decisions

The use of videos in teaching and learning processes is a didactic decision that follows the definition of teaching and learning objectives in the context of the methodological considerations. The use of instructional videos pursues a specific goal. For example, instructional videos can primarily aim to impart knowledge or to generate action. Another goal can be the promotion of empathy and emotional involvement and the activation of the learners. For the achievement of the objectives a didactic design of the videos is necessary. Each awareness video can be evaluated according to certain quality criteria. In a study at the University of Leipzig, it was found that the technical and creative (formal) criteria are considered more important by the users than the content. With respect to technical-creative criteria, aspects such as presentation quality, professionalism, and length of an instructional video are valued by the users. In terms of content, the main focus is the richness of the information, and comprehensibility [10]. This means that the videos have to be compact and yet informative. Due to the didactic design, the content of the videos is adapted to the needs of the users. That is why the videos use visual examples from around a typical Fraunhofer ISC workplace. With that, users are able to transfer the behavior shown in the videos directly into their everyday work. A lot of emphasis was also placed on comprehensibility. All viewers should be able to understand and reproduce the content, even if they only have basic IT knowledge. The overall aesthetics of the video were simple and served only as visual support. Previous videos at Fraunhofer ISC were criticized for being too long and for the pronunciation being too unclear sometimes. Therefore, the awareness videos were designed to be much shorter (they have a length of four to seven minutes) and to have a much clearer pronunciation. The videos were produced with the ulterior motive of being usable in the long term. Therefore, the content was not recorded from the subjective view of the speaker, but from a purely objective point of view. This approach meant that the content did not contain any opinion but was based purely on facts and figures. In the next section, we will take a closer look at the production of one of the four videos.

4.2 Video on detecting phishing e-mails

Phishing attacks on the information of Fraunhofer ISC are becoming more and more frequent as digitization increases. For this reason, “detecting phishing e-mails” was one of the four prioritized security topics for video production. The video was produced based on the PDCA cycle as follows.

Plan-Phase

All employees have access to a Fraunhofer e-mail address and an e-mail inbox. Users must use their Fraunhofer e-mail address when registering on websites for business purposes. Unfortunately, it may happen that these e-mail addresses are selected as spam recipients or as potential phishing victims. As a result, all employees are subject to a general risk regarding phishing. The aim of this video was to present a procedure for detecting a fraudulent e-mail and how to further deal with it. In doing so, the video explains the process of a phishing attack with the help of an example e-mail. The focus for this video was to explain why such phishing attacks are dangerous for the company and how such mails can be unmasked.

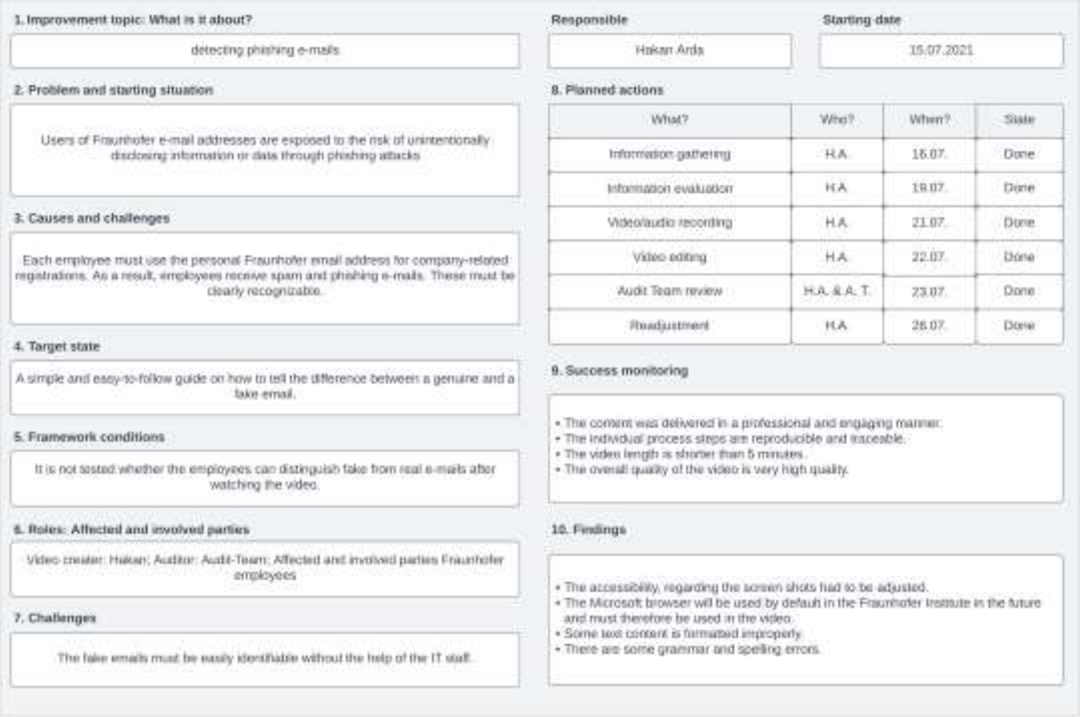
Do-Phase

For the production of the video “detecting phishing emails”, these measures were defined:

- Information gathering,
- Information evaluation,

- Video/audio recording,
- Video editing,
- Audit team review and
- Readjustment,

Each of these measures was provided with a completion date and the current status (cf. Fig. 2). The A3 report contains all important results of the plan phase, e.g., success control, standardization, and results, which will be important for the next two phases. With the help of this document, all basic information is written down on one sheet of paper and can be communicated more easily. First of all, the e-mail infrastructure and guidelines from the Fraunhofer ISC were examined. Users are given a personal e-mail address when they join the company. Every e-mail sent to such a Fraunhofer address first reaches an e-mail server. On this server, each e-mail is checked, and risky e-mails are blocked without disturbing or endangering the user. However, it can still happen that dangerous e-mails get through, because the system did not recognize them as dangerous.



1. Improvement topic: What is it about?

detecting phishing e-mails

Responsible

Hakan Arda

Starting date

15.07.2021

2. Problem and starting situation

Users of Fraunhofer e-mail addresses are exposed to the risk of unintentionally disclosing information or data through phishing attacks

3. Causes and challenges

Each employee must use the personal Fraunhofer email address for company-related registrations. As a result, employees receive spam and phishing e-mails. These must be clearly recognizable.

4. Target state

A simple and easy-to-follow guide on how to tell the difference between a genuine and a fake email.

5. Framework conditions

It is not tested whether the employees can distinguish fake from real e-mails after watching the video.

6. Roles: Affected and involved parties

Video creator: Hakan; Auditor: Audit-Team; Affected and involved parties Fraunhofer employees

7. Challenges

The fake emails must be easily identifiable without the help of the IT staff.

8. Planned actions

What?	Who?	When?	State
Information gathering	H.A.	16.07.	Done
Information evaluation	H.A.	19.07.	Done
Video/audio recording	H.A.	21.07.	Done
Video editing	H.A.	22.07.	Done
Audit Team review	H.A. & A. T.	23.07.	Done
Readjustment	H.A.	26.07.	Done

9. Success monitoring

- The content was delivered in a professional and engaging manner.
- The individual process steps are reproducible and traceable.
- The video length is shorter than 5 minutes.
- The overall quality of the video is very high quality.

10. Findings

- The accessibility, regarding the screen shots had to be adjusted.
- The Microsoft browser will be used by default in the Fraunhofer Institute in the future and must therefore be used in the video.
- Some text content is formatted improperly.
- There are some grammar and spelling errors.

Fig. 2: A3 report for detecting phishing e-mails

Check-Phase

In the check phase, the conformity, effectiveness, and efficiency of the measures for the video were checked. The measures were all executed and documented accordingly. The completion date was firmly adhered to for each measure, resulting in a smooth run. In addition, there was no deviation with regard to the sequence or execution of the individual measures. Therefore, the video detecting phishing e-mails and its measures are compliant with regard to the plan phase. For effectiveness, the intended goal of this video was reviewed. The goal was to present a method that makes it easier to detect phishing attacks. The content should be understandable and reproducible even without prior knowledge. The reproducibility was realized with the help of a system-oriented presentation. First, the Windows desktop was presented and then a phishing attack was unmasked in the e-mail program with the help of a demonstration example. For this purpose, the detection features sender address, content, links, and attachment were explained. In addition, the content was more accessible, because only the relevant areas of the e-mail were shown instead of the complete screen. In that way, it is easier for users to follow what is happening. In terms of comprehensibility, some aspects had room for improvement. For example, the so-called “who area” [9], the part of an e-mail that indicates which sender it came from, was not made clear in sufficient depth. Furthermore, although it was explained that website links can be faked, there was a lack of explanation how to detect these links. Overall, the effectiveness of the first version of the video was rated as sufficient and had to be improved within the act-phase.The production of this video took 20 hours, from that about 6 hours for video recording and about 5 hours for the video editing.

Act-Phase

In the Act phase, the suggestions of the audit team were realized. For example, in the first version of the video, there was no screenshot of the e-mail application, which made the video section look monotonously and uninteresting. In the readjustment, a new graphic was added to help viewers understand how an e-mail is first scanned on the server and then arrives in their inbox. Some audio recordings were also re-recorded or added in the readjustment. Because there was a lack of explanation and clarification in the first version. For example, it was added that phishing e-mails are often phrased in a threatening way or with an urgent need for action, because this way the victims get into a state of stress and act rashly. Thus, users have more clues and can better recognize such features. With these changes and other minor adjustments, the final version of the videos was much more effective than before.

4.3 Publication of Videos

All videos were uploaded to a public video streaming platform and provided with attractive thumbnails (cf. Fig. 3). The videos are only accessible to Fraunhofer ISC employees on the platform, after logging in using the Fraunhofer e-mail address.

Fig. 3: password security, detecting phishing e-mails, encryption of e-mails, behavior the event of IT security incidents [6]

Each video is accessible via a URL on the video platform. The links to the videos were sent to all Fraunhofer ISC employees via e-mail (approximately 450 people). The e-mail also contained the link to the evaluation survey. To reach even more employees, a post was published on the Fraunhofer ISC intranet informing them about the videos. The videos can then also be forwarded to the users by the internal IT hotline. This forwarding process is done according to the evaluation of the IT staff, whether the video is helpful for the problem or not. For the video production itself, all documents were stored on a secure network folder. With the help of these documents, such as the A3 report (cf. Fig. 2), new videos with various other topics can be produced even more efficiently in the future. All IT employees have access to this network folder. This ensures transparency of production and errors can be detected more easily and quickly.

5 Evaluation

To obtain initial feedback from the employees in order to derive improvement measures for the videos a simple opinion poll was conducted. The survey served to check whether the videos have aroused the interest of the employees and whether there is any criticism or further wishes. In this way, it was possible to evaluate the future of the video portfolio and to make content more suitable. The survey was conducted anonymously and on a voluntary basis. Depending on which videos the users watched, they were free to rate all videos individually. There were also open questions regarding topics and change requests. The opinion poll ran from August 15 to September 01, 2021. A total of 66 colleagues took part. Fifty-two questionnaires were completed in full and 14 partially. The videos were viewed a total of 227 times (as of 02.09.2021, 15:15). Thus, about 57 of approximately 450 employees have clicked on and viewed the videos. Each video was rated individually in the survey. However, for this work, the results of the survey were summarized. The videos were perceived as comprehensible and understandable (cf. Fig. 4). These characteristics have been the goals of the video production and have now been confirmed by the customers. The overall impression of the individual videos was also rated as “very good” on average. The decisive factor was the speed with which the content was conveyed. If the videos were shorter, some explanations would have been missing. Whereas with longer videos, interest and attention would have been lost quickly. All videos have an average length of approx. 5:30 min. and are thus exactly as long as the employees would like them to be. The overall style of the videos. including graphics, texts and symbols used, was also accepted by the employees. However, there was one major point of criticism in the open questions. The background music was partly perceived as disturbing and annoying. This might be due to the fact that the same soundtrack was used in all four videos and the music volume was sometimes too high. There was a lack of variation and subtlety here. In future, videos could be produced with and without music. The ones with music would need a sound check to ensure that everything is understandable. This way, users could decide whether they watch videos with background music or without.

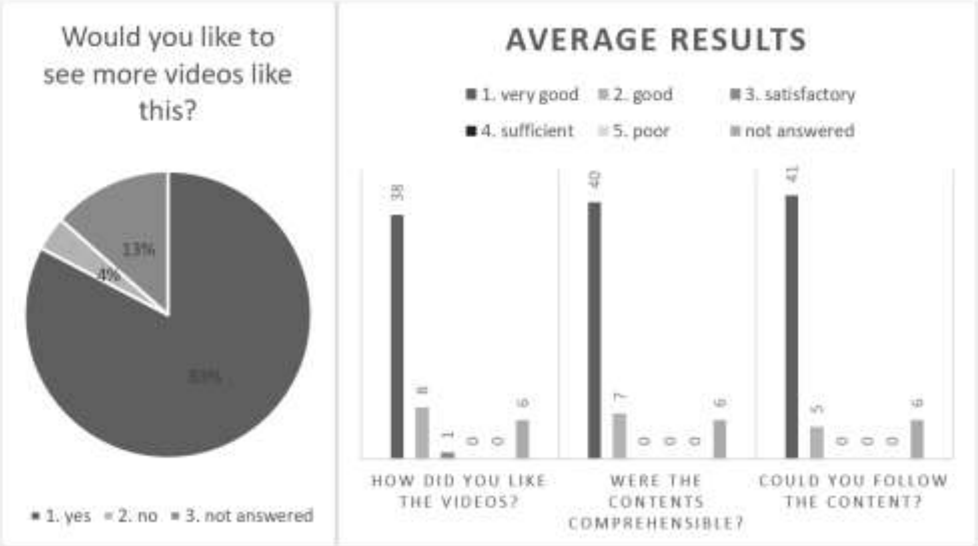


Fig. 4: Results of the survey

Of these 57 viewers, the majority agreed when asked whether more videos were desired (cf. Fig. 4). The conclusion can be drawn that there is a general interest in videos as a security awareness measure at Fraunhofer ISC and that this method could be used for future measures. In addition, some colleagues have already suggested topics. For example, how to open an e-mail header, how to surf the net securely, and how a password manager works. However, there were also topic suggestions outside of IT security, such as general instruction on how to use a computer. All suggestions will be evaluated together with the audit team and prepared for future video projects. Because the evaluation of this video portfolio was very positive, the method of conveying information in the form of videos will be continued in future.

6 Conclusion

With the work on this topic, it became clear that there is a great deal of interest in information security awareness videos at Fraunhofer ISC. In the opinion survey, the videos scored very well on average. This may be due to the fact that emphasis was put on comprehensibility and understandability. The videos were accessible without in-depth IT knowledge and have been informative. However, clear objectives had to be defined and documented beforehand. The achievement of these goals were checked by an audit team and the videos were constructively criticized. In this way, the videos became better and more efficient as production progressed. Working in the audit team had additional advantages. On the one hand, the team had a lot of expertise, which was very helpful for taking technical decisions. On the other hand, the individual interim results were always evaluated honestly and acknowledged accordingly. These characteristics were clearly emphasized by the transparent work philosophy of the continuous improvement process and subsequently put into practice by means of the PDCA cycle. The entire process and its measures developed in this way, can be reused for later video projects. Before the production of each video, the audit team jointly and clearly determined content and goals. The joint work has increased the importance of the goals achieved and the recognition of the work done. Additionally, the motivation for an even better work performance increased during the production of the following videos. This natural process automatically leads to a cycle of improvement. The orientation towards customer requirements not only improves the quality of the videos, but also their acceptance. There were also some issues that negatively impacted this work. There was a lack of resources to determine if the videos also provided an increase in Information Security Awareness. For this, a measurement of the awareness level before and after the consumption of the videos has to be done. It should be checked whether the shown contents have been understood and put into action by the viewers. For this reason, it is planned to conduct a follow-up study based on the results of this work. With the help of this study the addressed examination of the Security Awareness should take place and be determined whether individualized videos are suitable also for other enterprises. By examining the current state of IT in other companies, the results of this work can be almost effortlessly adapted to other companies. Another point would be to check whether videos are also worthwhile for the company from an economic point of view. For this purpose, it is advisable to carry out a cost-benefit analysis. Furthermore, the portfolio can also be used for purposes other than security awareness. For example, the question how and where to change passwords is regularly asked the IT hotline. The video “Password Security” could be sent to users as a solution suggestion. The platform, on which the videos were published, is also available for future video releases, so the portfolio can be expanded as desired. In conclusion, the method of introducing videos as an information security awareness measure works if the videos are specific to the company. The content must be accessible to the users and presented on their working environment. With that realistic scenario the employees recognize their role as a pillar of information security. They must be appropriately prepared for this role and then reaffirmed. Only in this way measures to increase information security awareness can develop their full potential.

Acknowledgement: This work was supervised by Professor *Kirstin Weber*, from *University of Applied Sciences*.

References

[1] Brenner, M.; Felde, N.; Hommel, W.; Metzger, S.; Reiser, H. & Schaaf, T. (2019). *Abdruck der DIN ISO/IEC 27001*. 10.3139/9783446462762.009.

[2] Deming, W.E. (1986) *Out of the crisis. Quality, productivity, and competitive position.* Cambridge: Cambridge Univ. Pr. ISBN 0-911379-01-0.

[3] Emmert, K.; Amberg-Schwab, S.; Braca, F.; Bazzichi, A.; Cecchi, A. & Somorowsky, F. (2021). bioORMOCER®—*Compostable Functional Barrier Coatings for Food Packaging*. Polymers. 13. 1257. 10.3390/polym13081257.

[4] Fertig, T.; Schütz, A. & Weber, K. (2022). A*utomated Measuring of Information Security Related Habits*. 10.24251/HICSS.2022.925.

[5] Findeisen, S.; Horn, S. & Seifried, J. (2019). *Lernen durch Videos – Empirische Befunde zur Gestaltung von Erklärvideos*. MedienPädagogik: Zeitschrift für Theorie und Praxis der Medienbildung. 2019. 16-36. 10.21240/mpaed/00/2019.10.01.X.

[6] Flaticon Webseite. (2021) *free icons for videos* URL: https://www.flaticon.com/de/

[7] Hanschke, I.. (2019). *Herausforderungen in Informationssicherheit und Datenschutz*. 10.3139/9783446459731.001.

[8] Kaspersky Lab. (2020*) Damage control: The cost of security breaches it security risks special report series*. URL: https://media.kaspersky.com/pdf/it-risks-survey-report-cost-of-security-breaches.pdf

[9] KIT SCC. (2021) *Betrügerische Nachrichten*. URL: https://www.scc.kit.edu/dienste/13034.php

[10] Mehner, C. & Bischof, S. (2015). *Förderung von Videos in der Hochschullehre: Begleitstudie untersucht Sicht der Studierenden*. 10.24251/HICSS.2022.925.

[11] Schütz, A.; Weber, K. & Fertig, T. (2020). *Analyze Before You Sensitize: Preparation of a Targeted ISA Training*. 10.24251/HICSS.2020.800.

Hakan ARDA
University of Applied Sciences
Würzburg-Schweinfurt
Fraunhofer-Institut für Silicatforschung ISC
GERMANY
E-mail: hakan.arda@student.fhws.de

Kristin WEBER
University of Applied Sciences
Würzburg-Schweinfurt
GERMANY
E-mail: kristin.weber@fhws.de

Sixth International Conference on Applied Informatics
Imagination, Creativity, Design, Development
ICDD 2022, May 5-7
Sibiu, Romania

The Simulation of Natural Behavior in Games Using Artificial Intelligence. Imagine Processing and Smart Input

Ramona-Antonela Bondă, Andrei-Daniel Mihai

Abstract

This article presents research in the game development field, a new input method for 3D games based on facial recognition and movements, using image processing and artificial intelligence. Its purpose is to make the virtual world available to more individuals, taking the gaming experience to a whole new level. Using an individual camera, users now have the opportunity to explore the gaming universe through various facial movements, in a manner similar to classic commands entered through the console or keyboard. It also allows them to better identify with their character, as the movements are more natural and similar to the human behavior, thus engaging in the activity of role-playing.

1 Introduction

This article responds to the need for development and innovation of our fast-paced world; therefore, it aims to bring something new to society and have an impact on the everyday life of many individuals that currently cannot experience nor enjoy the complex world of 3D video games.

Many recent studies have proven the benefits of video games. As opposed to the initial status quo, when video games were considered more of a waste of time, we now have proof that they can actually improve the cognitive abilities of an individual, enrich their problem-solving skills and logic, provide a more effective way to learn multi-tasking, as well as enhancing the prosocial behavior, helping develop a greater attention to detail and increasing coordination [1]. Another study suggests that games can help the players deal with negative emotions, while also engaging in multi-cultural experiences, crossing vast geographical distances, closing the gap between different generations and surpassing the language barriers [2]. Furthermore, there are also medical studies which highlight the benefits of engaging in computer playing. One such article suggests that games could have a serious purpose and help out patients who suffer from cerebral palsy. The study participants showed some improvements regarding their motivation and happiness. However, games are not the absolute solution, but were recommended as being complementary to the main traditional rehabilitation sessions. It also required a high diversity and encouragement to engage the participants in following the process [3].

That being said, it is to be acknowledged the importance of searching for new ways to improve the gaming experience so as many people as possible get to enjoy their abstract worlds and perfect different skills.

Regardless of the type of game, the goal was always to better identify with the main character, and so, during the past few years most of the work was put into perfecting the design, producing exceptional storylines, offering complex moral choices and enriching the communication between people. Given the hastened swiftness of all these past refinements, it is to be noted that one of the areas that longs for an update is the physical interaction between the players and the games. Needless to say, there are different ways already implemented to accomplish this connection, like the classical keyboard with mouse, controllers, consoles, and VR equipment. The new direction that does not yet benefit from full awareness is the interaction using different detections, due to the fact that its development only started around the year 2014 [4]. Most of the focus regarding it, however, went into games dedicated for smartphones with just a few facial or hand-gestures detections. Some such games to be considered are FaceRun, RainBrow and Nose Zone which use detections only for the eyebrows and the nose.

When it comes to 3D games, few steps have been made forward, allowing space for even more innovation. If the goal is to better identify with the main character of a game, one idea worth exploring is how to control the character in a more natural way, similar to the way our body moves, without the need of additional hardware, making use of the possibilities offered by artificial intelligence. Therefore, this article aims to bring forward one such method of control input for simulation of natural behavior in games, through the outstanding performance of artificial intelligence.

The direction this paper follows is controlling this type of game using facial recognition and detection only, and as it demands accessibility for every single individual, the only physical equipment it requires is a video camera. The idea itself not only encourages the overall progress and excites the regular player to develop new skills, but it also comes in aid of those less fortunate who cannot experience the contentment of the virtual world, on account of lacking the bodily abilities to interact with it. With this in mind, the solution is straightforward and affordable on a large scale. The user only has to place a regular camera in front of the screen they're using, the mechanics being a part of the software development and therefore suitable as an addition to any existing game.

The current world is leaning towards inclusivity, developing structures and algorithms which allow people to use regular equipment despite their disabilities. As the main focus of this article addresses the AI based input and control mechanisms for games, it proves why it is adding to this flow and how exactly it allows people to fully enjoy the virtual world. Currently there is not enough research done to provide the less fortunate people with accessible ways to play games. A paper from 2017 suggests that one of the current ways someone with impairments gets to enjoy this activity is by purchasing a WESH technology that allows the gamers to operate the controller with their muscle movements, sending signals to specific sensors which allows them to control the game characters [5]. However, it does not cover the specific cases where the users don't have control over their muscles either. And such, the need for a different type of input was raised, and this article in a response. A video camera is affordable and already owned by most people, therefore a software implementation such as the one presented in this paper brings a light over the future of human-computer interaction.

The title of this work emphasizes the fact that one can generate in the virtual world a behavior as natural as possible and known to the human abilities in the virtual world. There are two points of view from which the title can be explained. One of them emphasizes the already existing ways of interaction and the possibility of expanding them, while the other brings focus on the translation of our desires into the virtual world, a movement mechanism similar to that of everyday life. The first of these viewpoints strives to expand the same area of possibilities with the help of artificial intelligence. There are various combinations of keys to produce animations that allow the mobility of the main characters in games, such as: moving forward, backward, jumping or selecting various objects, and these can be done in different ways than those implemented so far, through artificial intelligence. These components can then be processed and given attributes to react in the exact same ways of interaction as before, without limiting the experience of complex games. The second point of view complements those specified, since exactly those classic gestures can be programmed to add to the current experience. For example, a slight lean of the body or face towards the screen can make the protagonist of the game move forward, a lateral tilt can have a same similar effect, pushing the character towards the sides. Thus, the character in the game moves naturally, imitating the instinctive gestures of the user.

The article is organized as follows: Section 2 contains descriptions of the apps and used algorithms, as well as a detailed explanation for the connection between them; and Section 3 presents future developments and Section 4 presents the conclusions reached while pursuing this project.

2 Design and Concepts

As proof of concept this project required two main workspaces: a detection app and a game developed in Unity3D. The first one was created to help detect the face and its mimicry while the second represents a simple environment to test different movements based on the detections. The connection between them is server-client based and it is to be described in the following chapters.

2.1 The Detections App

This application was created using the Python programming language as it provides a fast development of any type of algorithm and also allows working with the OpenCV, Dlib and NumPy libraries which are some of the most important pieces to allow image processing.

2.1.1 Artificial Intelligence (AI)

Over the past few years, AI has quickly become the new definition of the future, being one of the most supported areas with an annually collected income of over 20$ billion dollars from the tech giants of the world [6]. Due to its fast development, it ensures a safer future and supports the growth of different fields like: transportation, health care, education, customer service, manufacturing and media. It’s believed to be one of the most important aspects of the world today with no limit on its advancement and importance. Therefore, the desire to include AI algorithms in the future development of games is easy to explain.

The segment of this project inherently linked to working with the AI is rooted in the detection process of the face. Such algorithms have been trained on thousands of pictures and have had constant data provided to them to successfully identify different features in pictures and videos [7]. The Dlib library is a perfect example of a suchlike trained AI algorithm as it has the ability to identify different landmark points on a person’s face as it’s shown in Fig. 1.

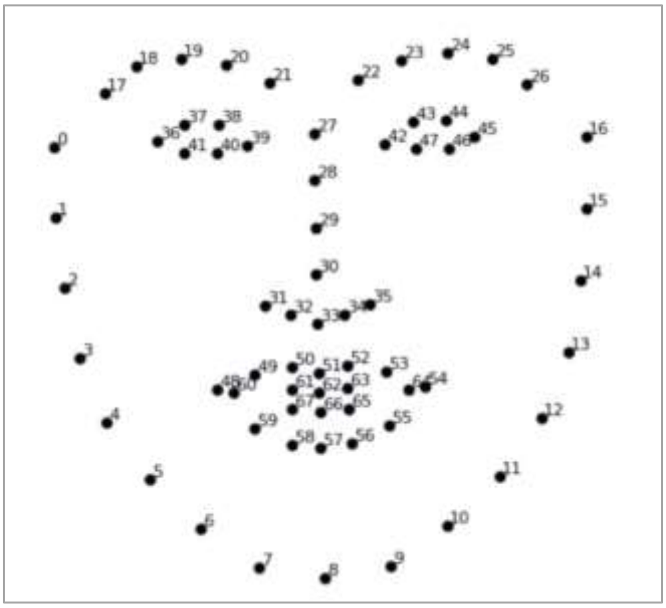


Fig. 1: 68 landmark points

OpenCV is a huge open-source library dedicated to visualization, machine learning and image processing techniques. It can process videos and images to identify a variety of objects, faces and gestures. Combined with NumPy which is a library optimized for numerical calculations, and with Dlib, the possibilities expand to create any type of detection-based application in a fast and efficient manner.

2.1.2 Components

The Detections App is split into 5 scripts:

- CompleteDetection.py
- Face_detection
- Eye_detection
- UdpComms
- Server

CompleteDetection is the component concentrating all of the detections and in which the video camera is initialized, thus being the main source of the app. It initializes the two haarcascade components in the OpenCV library, namely the AI parts built for Face_detection and Eye_detection. The app uses only the frontal face, as it only takes facial mimicry to control the app, which is why there was no need to use a much more evolved AI algorithm. A second reason that determined this choice is the efficiency in terms of execution speed, the 68 landmark points being sufficient for this kind of detections.

In this script, after initializing the camera and running the detections on each frame, a string vector is built, which is reset with each frame and embodies the result passed by each function from the Face_detection and Eye_detection scripts. Both of these scripts return strings according to the detected mimicry.

Once all movements are analyzed, the vector is passed to the server component, which then sends the string to a client script programmed in C#, tied to the game.

Face_detection is a script that currently contains a total of 5 functions dedicated to the detection of different mimics and poses.

The face_landmarks function is an algorithm that displays the landmark points on the user's face. They are detected on the missing color information variant (gray) of a frame, to aid efficiency and processing, then drawn on the colored version of the images on the camera that the user also sees.

Face_proximity is the function that creates two fixed borders around a person's face. The first border coming from face_landmarks is flexible and it changes its size according to the position of the player. The other two boundary boxes are initialized according to this first border and remain fixed, although they follow the user's face, but do not change their size after being initialized [8]. The purpose of these boundary boxes within the application is to transmit to the game forward and backward movements. Due to the fact that one of these borders is slightly larger than the original one, if the latter exceeds the former in size, the function can transfer the information to the server that the player has approached the camera. Analogously, a smaller border is placed, and once overcome by tilting the position of the head back, a message is transmitted that can be interpreted as backward movement within the game.

The way this function was implemented is by obtaining the angle of the diagonal created by the main border with the help of sine and cosine mathematical functions. Their sizes are dictated by some parameters given by the programmer, after dozens of attempts and under different conditions. These borders are displayed on the screen so that the user can observe them and more accurately understand the area in which they have to move to trigger the two movements.

The mouth_open function dictates the appearance of the user's mouth, which within a game and including the one in this project, can be used to execute actions such as jumping. If the player opens their mouth, the algorithm processes the information and passes the signal further to the game.

The way this function was implemented is by calculating two hypotenuses, one vertical between the landmark points 51 and 57, and the other horizontal between the landmark points 48 and 54, as illustrated in Fig. 2.

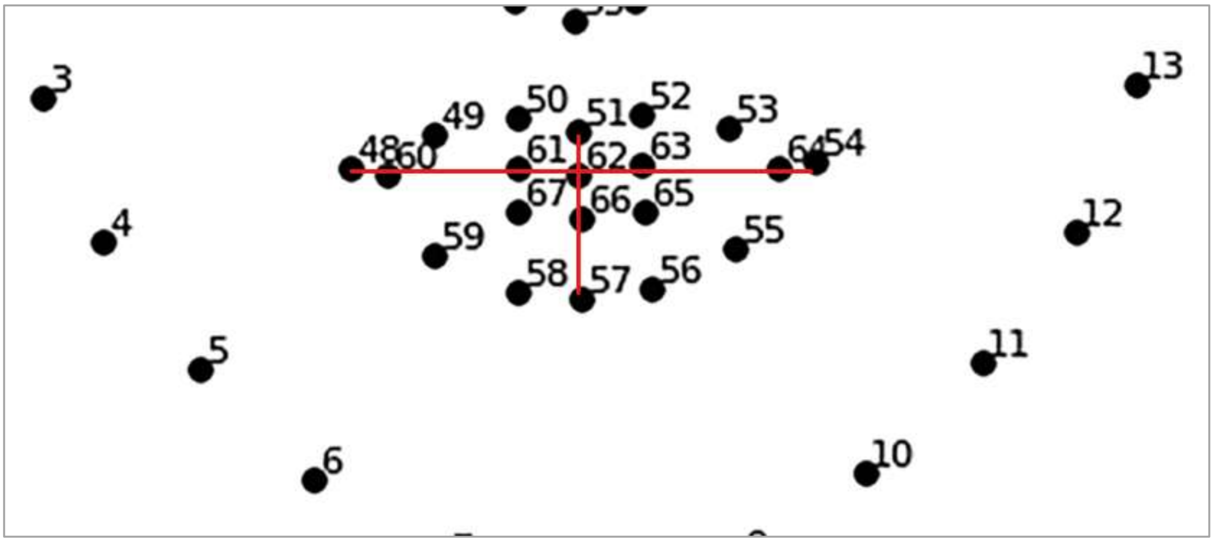


Fig. 2: hypotenuses used in mouth_open detection

Given the dimensions of these two lines, during the development of the algorithm the programmer can obtain the standard ratio resulting from dividing the horizontal line by the vertical one. The programmer decides then over a limit, compares them to the original value and depending on ratio, a corresponding message is sent to the server. If the change is detected, the message “mouth_open” will be sent, otherwise an empty string will be returned.

Nose_pointing_sideways is a function implemented to detect whether the user is turning their head left or right. Such an algorithm can be used for any type of movement, but within the current project, it was used to dictate to the game whether the character should move to the left side or to the right side. The algorithm was developed by calculating an additional point on the tip of the nose, placed between the landmark points 30 and 33. After creating this point, two more points were calculated on a horizontal line according to the flexible border of the face. The left and right halves are taken into account and two different hypotenuses are created, both starting from the newly created point. The programmer then retrieves the initial values of these two horizontal lines then decides over a threshold required for the detection. The function returns a message according to this threshold: if by subtracting the left hypotenuses from the right one, the value obtained is greater than that standard, the message “turn_left” is returned. Correspondently, if by subtracting the right hypotenuses from the left one the obtained value is greater than the standard threshold, the message “turn_right” is returned. Otherwise, the server will receive an empty string.

Nose_pointing_up_down is a function created in a similar way to the previous one, using the same newly created point on the middle of the nose, only in this case the hypotenuses are vertical. The programmer creates a new threshold, but in addition to the previous algorithm it should be taken into consideration that the distance from the nose to the middle of the top border is much greater than the distance from the tip of the nose to the middle of the bottom border. The possible messages to be retrieved from this function are: “look_up”, “look_down” or an empty string message.

The **Eye_detection** script was created separately from the previous one, even though the eyes also belong to the facial features. The eyes include many more components for detections like gaze or blinking and it uses a different type of haarcascade, thus justifying the decision to create a separate, specialized script for them.

One of the main functions of this script, get_blinking_ratio, works on two different groups of landmark points, as represented in Fig. 3: the group of numbers from 36 to 41 belonging to the left eye, and the group from 42 to 47 belonging to the right one.

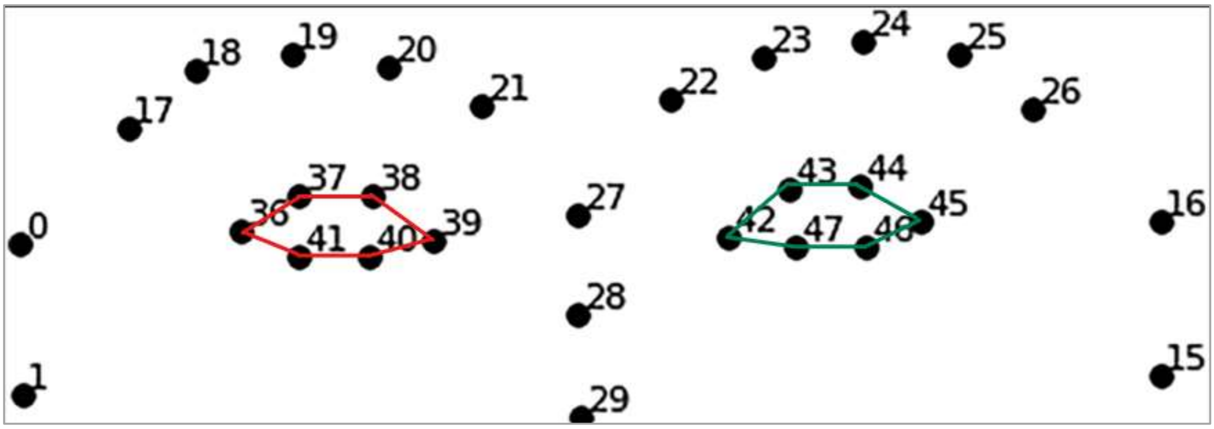


Fig. 3: the groups of landmark points used for eye detection

As an example, for the left eye, similar to the algorithm used to detect the movement of the mouth two hypotenuses are drawn, one horizontal and one vertical. The horizontal one is drawn between points 36 and 39, while for the vertical one 2 additional middle points are created between the landmark points 37 and 38, respectively 41 and 40. Having these two lines drawn, all this function does is return the division between the horizontal and the vertical hypotenuse, so that the result can be used in another function.

The previous described function is best used to detect blinking, but in order to avoid accidental and unwanted maneuvers in the game, the results that dictate the blinking must be consistent over 5 frames [9].

The second major function within this script is get_gaze_ratio, generally designed to be used on both eyes, which tells the algorithm the direction in which the user is looking. The algorithm works as follows: the area formed by the 6 landmark points is cropped, enabling the program to interpret only the information in that area, then the color information is removed and a mask is applied. The purpose of this mask is to be able to see the whole area in black and white. Given that the sclera of the eye is white, and the iris is colored, for the latter the mask will receive the black color. The two pieces of information can be used to detect the direction of a person’s gaze. The eye is then divided vertically into two halves, following which the two are compared by counting the pixels. The direction in which the user is looking is the one in which the number of black pixels is higher [10].

One aspect that it to be considered when running the application is that nose_pointing_sideways and get_gaze_ratio may overlap, thus creating false detections and unwanted actions. To avoid this, the condition must be set that if nose_pointing_sideways is detected, the second function will not be activated.

All other functions inside this script are used to streamline programming, without the need to specify the landmarks points within the main program.

UdpComms function can be used by Python both as a server and as a client, but in the case of this project it was only used to transmit data by The Detection App to Unity. The main function in this class has 5 important parameters, an IP that must coincide with that of the client, in this case 127.0.0.1, two ports: 8000 to transmit data from Python to Unity, and 8001 to receive data from the game, a Boolean variable that decides whether data can be received or not, and another variable of the same type that helps to troubleshoot the code. All other functions inside the Class deal with initializing the socket, closing it, sending data in the form of bits, encoding and decoding, receiving messages, and displaying errors in the event that certain variables or functions could not be initialized correctly. [11]

The only purpose of the **server** script is to send data, retrieve the strings with detections from each frame from the main program and send them to the C# client using the UdpComms script.

2.2 Unity3D – Lost

Lost is a first-person shooter type of game, created within Unity3D game engine. It tells the story of Jason, a man who woke up one day and realized that all of the people from his village were gone. In search for them, he ends up in a dungeon full of zombies and spiders and after passing that specific passage of the game he finally meets some villagers. After a short conversation with them, he realizes that he’s been thrown back in time and has to search for a way to return to his loved ones, hence the name of the game: Lost. Some screenshots from the game are presented in the Fig. 4 and Fig. 5, and they’re relevant to the paper as a proof that the game is personal and indeed created in 3D mode. Also, the game is designed for experimental purpose only, allowing the use of the detections and new implemented movements to be tested.

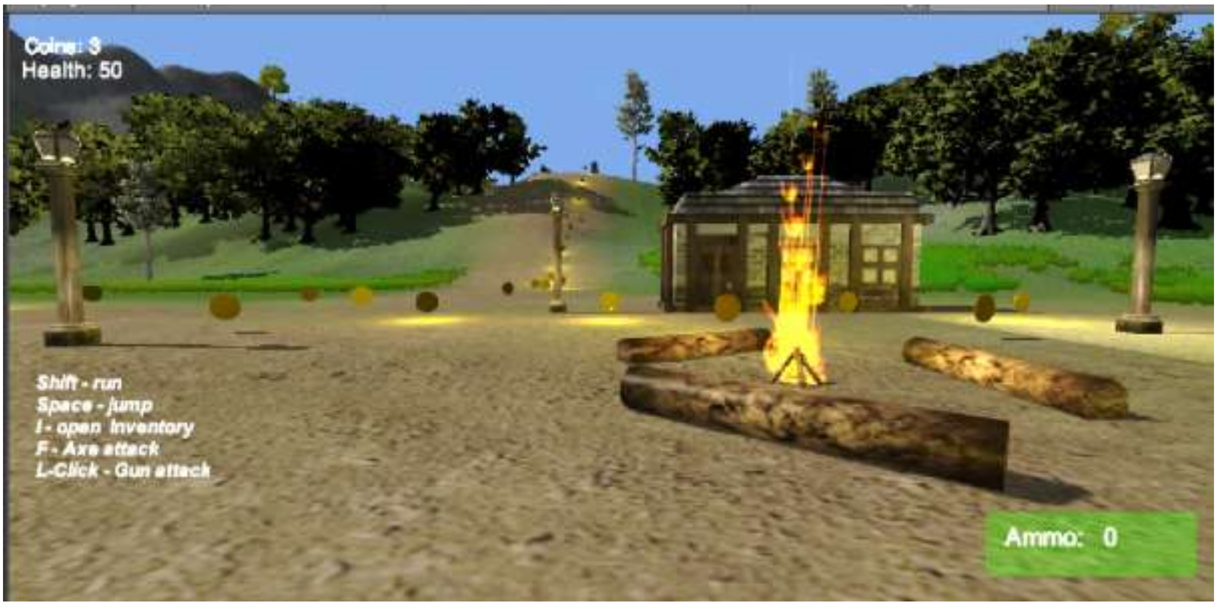


Fig. 4: The village of the protagonist

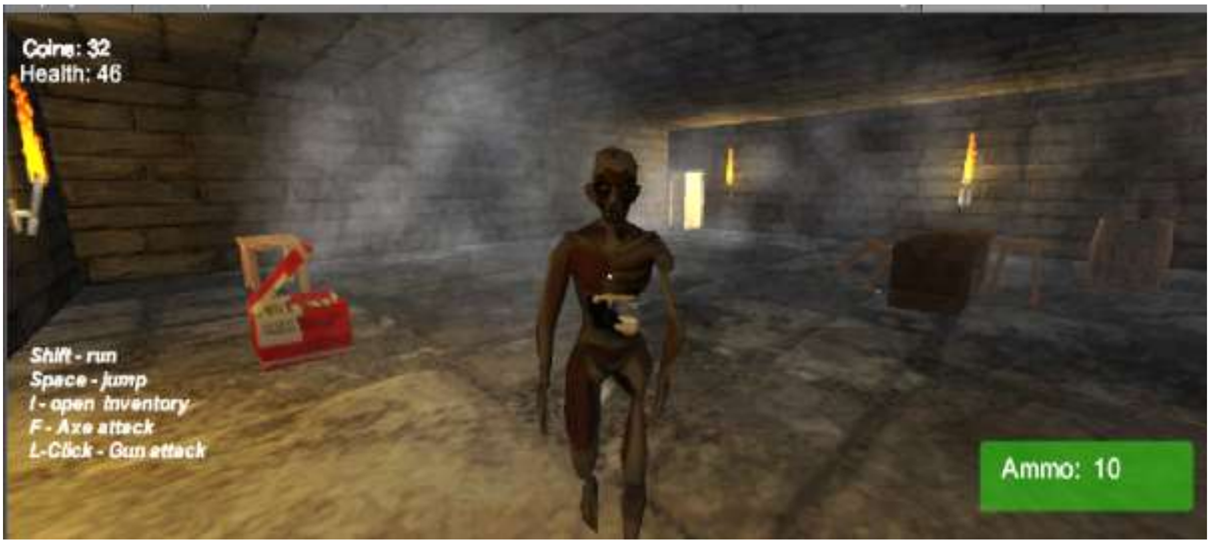
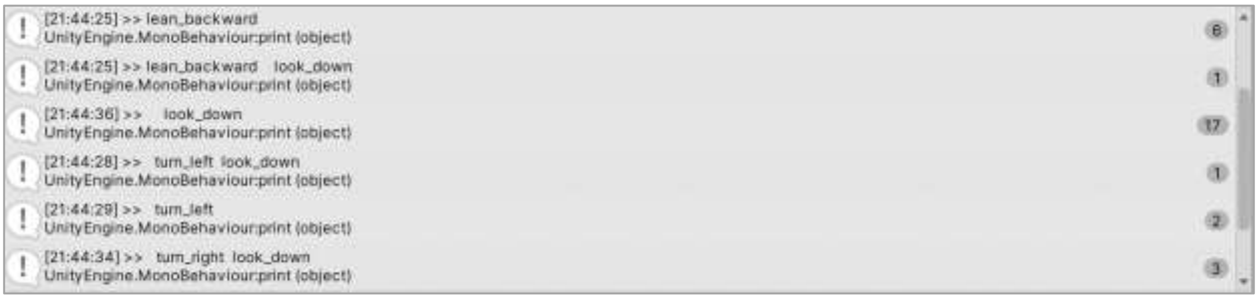


Fig. 5: First encounter with an enemy in the tomb

2.2.1 Components

The game contains a sum of 43 scripts, of which truly relevant for this paper are only 2 of them: UdpSocket and FaceRecognitionMovements. All the others assure the functionality of the game, given the animations, UIs and movements.

UdpSocket is a script which works in a similar way to UdpComms script in the Python application. It has both functionalities for receiving and transmitting data, but in the current version it's been used as a client only, having the same IP 127.0.0.1 and ports. It is implemented to receive data dynamically and constantly, with each frame processed by the app in Python. All the received data is initially gathered in a thread, to prevent the application from getting blocked. Afterwards, each string is decrypted so that it can be read in the same way as it was sent, not bitwise, then displayed within Unity3D, as shown in the Fig. 6.



```
[21:44:25] >> lean_backward                                   6
UnityEngine.MonoBehaviour:print (object)
[21:44:25] >> lean_backward   look_down                       1
UnityEngine.MonoBehaviour:print (object)
[21:44:36] >>    look_down                                    17
UnityEngine.MonoBehaviour:print (object)
[21:44:28] >>   turn_left  look_down                          1
UnityEngine.MonoBehaviour:print (object)
[21:44:29] >>   turn_left                                     2
UnityEngine.MonoBehaviour:print (object)
[21:44:34] >>   turn_right  look_down                         3
UnityEngine.MonoBehaviour:print (object)
```

Fig. 6: Received strings in Unity

Once a string is received and decrypted, it is sent forward to the FaceRecognitionMovements script, which analyzes and processes it according to the programmer's needs.

Within this script, multiple detections can run concurrently, without creating an overlap or run-in problems. The algorithm identifies specific pieces of string that are then interpreted to activate certain movement. For instance, if the algorithm finds the expression "lean_forward “, analogous to pressing the "w" key in a regular game, the character will move forward.

In the current stage of development, the algorithm executes movements according to the following strings:

- "lean_backward" – the character in the game moves backwards
- "mouth_open" – the character in the game jumps
- "turn_left" – the character moves to the left side
- "turn_right" – the character moves to the right side
- "gaze_left" – the character turns to the left
- "gaze_right" – the character turns to the right

These movements have been implemented up to this point as proof that the facial detections are fast and reliable enough, therefore could be used in all of the games as an alternative to the input hardware. Of course, other detections related to blinking, or facial emotions like smiling or frowning can be added to complete the gaming experience. There are a lot of detections that can help with the weapon selection, interacting with the enemies, opening doors and everything else, depending only on the programmer's imagination.

2.3 The connection between the apps

Human-Computer Interaction (HCI) is a concept that emerged in the 1980s when the need for better communication between the two parties presented itself. Along with the popularization of the personal computer, people also realized the lack of methods to minimize the interaction cost and the amount of physical and mental effort was daunting. The computers were no longer built just for experts, but were also presenting a more friendly interface which allowed access to more potential users [12].

Unlike most software, games are not made to support tasks defined by users, but encourage them to engage in their own defined activities which players perceive and adopt [13]. There is also a need for understanding the behavioral implications of game play, as they have a direct correlation to the cognitive, physical, emotional and motivational process during gameplay. Monitoring the health issues of individuals is imperative for the process of developing any game before it becomes public [14]. For the current paper these specified attributes were not analyzed on a vast amount of people, as it presents just a proof of concept.

At this stage of development, the human-computer interaction requires to install PyCharm to be able to run the Python script, as well as the game engine Unity3D. To begin the process or properly playing the game, as previously stated in this paper, it is necessary to place a video camera above or in front of the screen running the game, thus being able to follow the action that is happening in the game and allowing the Python application to run properly.

Once these settings are accomplished, it is recommended that The Detection App is ran first, since it works as a server and the data should already be available to the client at the time the game is run.

Once the Python application is started, the user will be able to observe the detections highlighted on their face. They're important to better understand the correct positioning relative to the camera since one of the scripts controls the forwards and backwards movements, depending on the boundary boxes. The user should position themselves so that the mobile border is halfway between the other two fixed borders, as shown in Fig. 7.

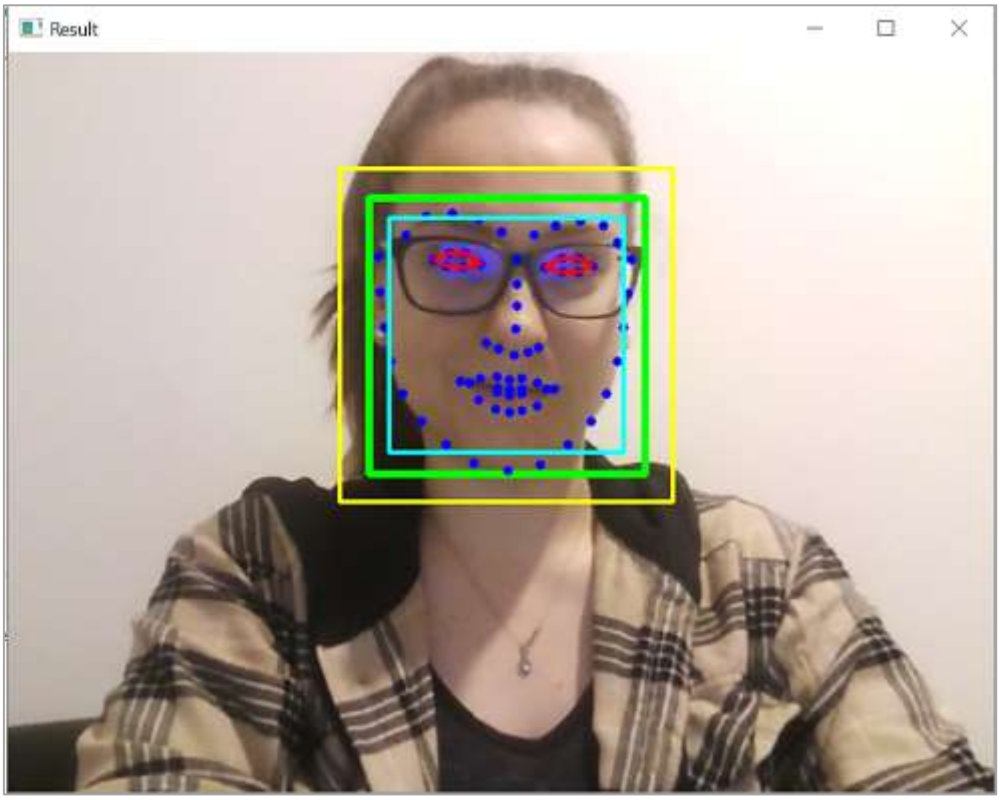


Fig. 7: Correct positioning within The Detections App

The detections are color coded to facilitate their understanding, as follows:

- Dark blue – the landmark points shown above in Fig. 1
- Red – the groups of landmark points used for eye detections, highlighted in Fig. 3

- Green – the main, mobile border created around the landmark points
- Yellow – secondary border, fixed and larger than the main one, created so that when the mobile one exceeds its size, the algorithm can transmit the information “forward”
- Light blue – secondary border, also fixed and smaller than the main one, created so that when the mobile one becomes smaller in size, the algorithm can transmit the information “backward”

The following step after the correct positioning in front of the camera is to start the game "Lost", leaving The Detections App to run in the background. After that, there are no more steps to be followed nor modified, and the player can enjoy the game.

3 Future developments

A significant plus to this work would be brought by a parameter calibration app. For current detections, only a few dozen calibration attempts were used and only on the face of 1 person, in a standard position and with certain light information. So far, no obvious problems have arisen, but in order to be able to use this application on a large scale and to make it available to anyone, an app trained with an AI script that calculates the ideal distances on its own would be required. Not all people have the same setup, not all have enough light information to read their eyes correctly, and certainly not all have the same space to provide ideal mobility. Thus, a calibration system to dictate the distance of the two fixed borders and mathematically calculate the ideal distances is currently on the list of future directions.

Another idea for the future is programming an aimbot. There are a lot of games currently using it to help beginners explore the world of a game in a relaxed way, made for story lovers. In the case of this project, such a script would be helpful because it is not easy to quickly and efficiently guide the character to face the enemies with only the facial detections, so a quick target focus on them would complement the experience of any first-person shooter game.

On the list of future directions are also the hand gesture detections. This would be an interesting detail to add when the app will also have an interface so that the user can decide whether they want to play classic, with facial detections or with gestures.

Another important detail within the application would be to make a direct connection between Python and Unity, without having to start the applications separately and in a certain sequence, which would make the interaction between the player and the game more efficient, intuitive and attractive. An in-game menu set to activate the camcorder would be required to fully integrate the two apps. Also, in the same game menu, commands could also be made available to the player. The user could decide for themselves what kind of detections to use and for what and in the more distant future an AI script would allow them to create their own detections, which would make the app all the more attractive.

4 Conclusions

We believe that the described application has potential and throws a ray of light towards what can be achieved in the future. Although at the moment this is only a prototype and the components mentioned in the future directions part would bring it a significant boost, it is already at a fairly evolved stage, taking into account the fact that we have been working in parallel on a game developed on our own. The results are not always absolute, but we think that the reason behind this problem is that the parameters in Python have to be trained to work under different light sources and face angles. We intend to work on this detail in the near future, in the hope that it will also optimize the client-server connection between these two different programming languages. That being said, we also view this app as a move forward toward inclusivity, as a solution to problems that ordinary people tend to ignore. It is a voice that draws attention to the need to develop certain software that will provide new experiences and make everyone's work easier, and this is called progress.

Acknowledgement: This work was supervised by Lecturer PhD Ralf Fabian, from Lucian Blaga University of Sibiu.

References

[1] Video games for well-being: https://www.frontiersin.org/articles/10.3389/fpsyg.2018.02127/full (retrieved 18.0.2022)
[2] The benefits of playing video games: https://www.apa.org/pubs/journals/releases/amp-a0034857.pdf (retrieved 18.0.2022)
[3] A systematic review of interventions in patients with cerebral palsy: https://www.fronti ersin.org/articles/10.3389/fpsyg.2018.01712/full (retrieved 18.0.2022)
[4] A Face Recognition Based Multiplayer Mobile Game Application: https://link.springer.com/chapter/10.1007/978-3-662-44654-6_21 (retrieved 18.0.2022)
[5] Technology which allows people to play video games without hands or arms: https://www.eletimes.com/technology-will-let-play-video-games-without-arms-hands#:~:text=A%20technology %20has%20been%20developed,will%20send%20signals%20to%20sensors. (retrieved 20.06.2022)
[6] The future of AI: How Artificial Intelligence Will Change the World: https://builtin.com/artificial-intelligence/artificial-intelligence-future (retrieved 20.06.2022)
[7] How transfer learning works: https://www.mygreatlearning.com/blog/transfer-learning/ (retrieved 20.06.2022)
[8] Fixed borders around the user’s face: https://learnopencv.com/playing-chromes-t-rex-game-with-facial-gestures/ (retrieved: 02.04.2022)
[9] Blinking algorithm idea: https://www.youtube.com/watch?v=VWUgkcX_KoY&t=1s&ab_channel=Pysource (retrieved 15.11.2021)
[10] Gaze algorithm idea: https://pysource.com/2019/01/14/eye-gaze-detection-1-gaze-controlled-keyboard-with-python-and-opencv-p-3/ (retrieved 15.11.2021)
[11] Server-Client connection idea: https://github.com/Siliconifier/Python-Unity-Socket-Communication (retrieved 13.04.2022)
[12]Human-Computer interaction: https://www.interaction-design.org/literature/topics/human-computer-interaction (retrieved 18.07.2022)
[13]Human-computer interaction and games: https://www.sciencedirect.com/science/article/abs/pii/S095354380 6001159 (retrieved 18.07.2022)
[14]Human-Computer Interaction Serious Games as Behavioral Change Moderators: https://www.frontiersin.org/research-topics/22086/human-computer-interaction-serious-games-as-behavioral-change-moderators (retrieved 18.07.2022)

Ramona-Antonela BONDĂ
Lucian Blaga University of Sibiu
Faculty of Science
Department of Mathematics and Informatics
5-7 Dr. Ratiu Str, Sibiu 550012
ROMANIA
E-mail: ramona.bonda19@gmail.com

Andrei-Daniel MIHAI
Lucian Blaga University of Sibiu
Faculty of Science
Department of Mathematics and Informatics
5-7 Dr. Ratiu Str, Sibiu 550012
ROMANIA
E-mail: andrei_mihai_23@yahoo.ro

Sixth International Conference on Applied Informatics
Imagination, Creativity, Design, Development
ICDD 2022, May 5-7
Sibiu, Romania

Automated System for Home Greenhouses

Sorin Ionut Conea

Abstract

The greenhouse is the structure that helps to increase the growth of crops, plants, fruits, vegetables, etc. Naturally, the plants can be grown only in a certain region, being necessary to meet certain conditions, such as weather, humidity, temperature, and season. The greenhouse helps the accelerated and safe growth of the plants by controlling some parameters. Conventionally, the cultivation of plants using greenhouses is difficult as it requires changing the parameters of plant growth with the help of human intervention. These methods are less effective because the constant and continuous involvement of humans leads to an increase in the workforce need and can often cause errors. This paper proposes an autonomous greenhouse model with remote control using an Arduino microcontroller. This project monitors and controls the climate, without the need for human intervention. For this, we use different sensors to be able to measure the environmental parameters according to the needs of the plant.

By connecting this system to the Internet and the Blynk Cloud, we can access data remotely, eliminating the need for physical plant monitoring. The Cloud server allows data processing and application of actions in the remote greenhouse.

1 Introduction

Internet of Things (IoT) refers to a network of interconnected objects that allow their remote control. The model presented has the role of favouring the development of plants, regardless of the climate in which they are planted or can be used for the development of plants at home in the absence of a human being. The greenhouse can be placed anywhere, both inside and outside the house.

Firstly, a special microclimate must be created, in which the plants are provided a temperature as close as possible to the optimal conditions. At the same time, in such a protected space, there is better control of the levels of humidity, light, and air circulation. To get satisfactory results, the right cultivation technology for the chosen species needs to be followed.

To monitor and ensure the main factors of microclimate (ambient temperature, light, soil moisture, and ambient humidity) in an artificial environment using an automated system is the goal of this paper.

2 Related Research

Many researchers have studied the influence of various factors for plant development in greenhouses. Their documents show that controlling the parameters of plant ridges helps to develop it. Plant production and development is directly influenced by climatic conditions, mainly by temperature and humidity. The two criteria can be controlled in automated greenhouses in which growers have control over the automatic system for monitoring and regulating the important characteristics necessary for plant development. [1].

The effect of light-emitting diode (LED) supplemental lighting on the folate and polyphenols contents as well as antioxidant properties of Valerianella locusta (Lamb's lettuce) cultivated in a greenhouse in autumn and winter seasons was studied in [2]. Six LED-lighting combinations were used for the experimental set. Four of them differed in percentage share of red (R) and blue (B) in emission either warm 4700K+2700K) or cool white light. Control plants were grown under high-pressure sodium (HPS) lamps.

In [3] was shown that chlorophyll and starch levels were increased when music it is used in plant growth so usic not only accelerates growth but also significantly influences the concentration of various metabolites.

To scale the project by adding more sensors from different locations requires an increase in processing power and storage, therefore the Cloud technology becomes an implementation option. Cloud computing is a smart technology that connects multiple servers to a cloud platform to share data collected by sensors. In [4], the scalability of IoT is highlighted using Cloud technology, as this technology not only connects servers but also quickly processes information from sensors, thus providing feasible solutions.

According to [5] inside the greenhouse, the crop growth is influenced by PAR radiation, temperature and CO2 level, so controlling this parameters using a combination of sensors and modules will make the plant grow faster.

Wireless ZigBee modules can be used to control and monitor the environment inside the station. In [6] a system uses this module to send the data obtained by the sensors to the microprocessor to be processed and then the relays will be turned on in order to maintain the climatic parameters in the predefined interval.

A web platform and three nodes can be used to control the inside climate in the evening. In [7] the 3 nodes are presented: sink node, wireless sensor node, and wireless climate control node. The 3 nodes work simultaneously to collect, analyze and process the information so that the climate can be controlled based on the data collected.

In order to display the comfort level in the greenhouse, 3 physical LEDs can be used to illuminate depending on the conditions. In [8], the temperature sensor sends data to the Arduino Microprocessor, which commands one of the 3 LEDs depending on the temperature (WARM, COLD, MODERATE).

3 Proposed System Architecture

The proposed architecture requires minimal human intervention, thus saving time and money. To have a fast and safe growth of the plants, we measure the temperature, the soil humidity, the ambient humidity, and the light level using dedicated sensors. To interconnect these sensors, we use a microcontroller that takes over the values provided by the sensors and sends them to the Blynk Cloud server. After evaluating the received data, the microcontroller operates various components such as the fan, the water pump, the heat source, and the lights, periodically adjusting various parameters to facilitate plant growth. All through this process, the collection of data from the sensors and their processing is done at Cloud level, so that we can access the greenhouse system remotely. Being connected to the internet, this system reduces the cost of workforce and facilitates the control of various parameters.

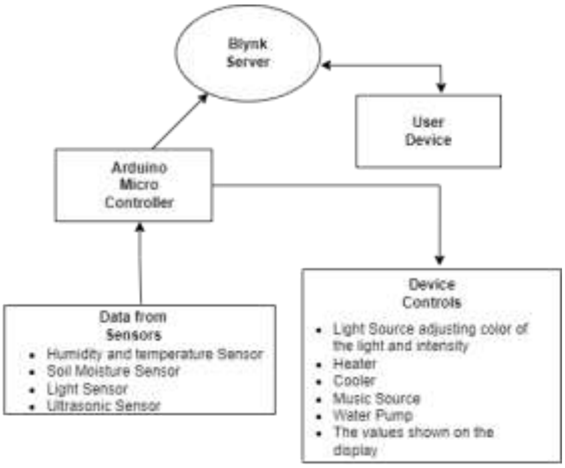


Fig. 1: Automated system for home greenhouse architecture

Fig. 1 shows how the data is taken over and transmitted to the microcontroller and then to the Blynk Cloud, the actions corresponding to the plant's needs following to be ordered.

4 Operating the system

4.1 Local System

The temperature and humidity sensor constantly checks the two parameters it sends to the microcontroller. If the reference values set by the user are exceeded, the fan is automatically operated to lower the ambient temperature and reduce the humidity level. When the reference value is reached, the fan will turn off automatically. On the other hand, if the temperature is too low, the heater device will start automatically and will operate until the needed temperature is reached.

The soil moisture sensor periodically sends the data to the microcontroller, and if the indicated value is lower than the reference value, the water pump is activated. Watering is carried out at a low speed to allow a slow and even absorption of water into the soil. When the reference value is reached, the water pump is automatically switched off.

The role of the light sensor is to measure the intensity of the light from the greenhouse and to send these values to the microcontroller. If the amount of light is not enough for the development of the plant, the LED strip is started and the colour Ultraviolet RGB (95,75,139) is set and will be periodically changed in colour Blue RGB (0,0,255) and Red RGB (255,0,0) to favour the flowering.

In [3], the effectiveness of classical music in plant growth is proven. As a result, the proposed model integrates a player equipped with a speaker that will play music during the day.

The role of the GSM module is to facilitate the data transfer to the Cloud Server in areas where there is no Internet signal, thus only the GSM signal is needed.

The display indicates in real time the values read by the sensors, thus facilitating easy access to data without the need of a third device.

The red button is used to put the greenhouse in maintenance mode, when only the values read by the sensors are indicated without any automatic action being applied.

The Ultrasonic sensor measures the height of the plant and sends the data to the microcontroller, which will send a notification when the plant reaches the desired height.

4.2 Hardware Requirements and usage

The Arduino Mega 2560 R3 development board is used to monitor the used modules. The following sensors and other peripherals are used to collect and create the optimal conditions for plant development.

“Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The boards feature serial communications interfaces, including Universal Serial Bus (USB) on some models, which are also used for loading programs from personal computers. The microcontrollers are typically programmed using a dialect of features from the programming languages C and C++. In addition to using traditional compiler tool chains, the Arduino project provides an integrated development environment (IDE) based on the Processing language project.” [9]

DHT 22 is a temperature sensor which follows DHT 11, but now with a higher measurement accuracy and it is used to measure the humidity and temperature in the greenhouse. The reading frequency is 0.5 Hz; the accuracy for air temperature in [-40°C, 80°C] is 0.5%; the accuracy for air humidity is less than 5% [10].

Ultrasonic Sensor - HC-SR04 is the sensor used to measure the height of the plant. It works by emitting sound waves at a frequency too high for a human being to hear and waits for the sound to be reflected, thus calculating the distance based on the required travel time.[11]

The Light Dependent Resistor (LDR, photoresistor) is a special kind of resistor. The resistance of a photoresistor depends upon the intensity of the incident light [12].

The display is a 16x2 device (16 characters per line, 2 lines) that requires the connection of all the pins to the Arduino development board, this restricting the possibility of adding as many sensors and modules as possible.

QuadBand GPRS-GSM SIM800L is a small quad-band GSM / GPRS module equipped with a quad-band antenna that can send text messages and make calls. The most advantageous thing is that it facilitates internet access to send data to the Cloud.

DFPlayer - Mini MP3 Player is an Mp3 module equipped with a Micro SD card slot that can emit sounds directly to a speaker without the need for an amplification system.

SparkFun Soil Moisture Sensor is used to monitor soil moisture, the sensors consists of two probes that conducts electricity and returns values from 0 (completely dry) to 1023 (completely wet) according to the soil moisture.

Relay Module - is a relay module with separation boards, being used to isolate the components and create a sub-circuit in the main circuit with a different voltage. The relay was used to be able to start a water pump with a higher supply voltage which is supplied using an external power source such as a socket.

The submersible water pump used to irrigate the plants has a watering capacity of 240 l / hour and a maximum hose of 3m can be attached to it. These types of water pumps are used in aquariums

NeoPixel RGB LED STRIP - addressable LED strip that uses a data pin so that the colours can be controlled by the microcontroller but it can be used an normal RGB LED Strip and just mix the RGB colors in order to obtain de desired color.

The computer fan is used for ventilation and for lowering the temperature and humidity in the greenhouse and can reach 2000 RPM.

Parameter that is controlled	Model of sensor/module	Working Voltage
Temperature and humidity	DHT 22	3V DC
Relay Board	1 Channel	5V DC
Display	16x2 I2C Display	3.3V DC
Internet connection	QuadBand GPRS-GSM SIM800L	5V DC
Plant Height	HC-SR042	5V DC
Sound	DFPlayer – Mini	5V DC
Soil Moisture	SparkFun Soil Moisture Sensor	5V DC
Lights	NeoPixel RRB Strip	5V DC
Cooler	Generic PC Fan	12V DC
Heater	Generic Heater	UP to 220V DC
Microcontroller	Arduino Mega 2560 R3	12V DC

Table 1: Deployment Parameters

Table 1 shows the parameters controlled by the sensors, their model and the supply voltage. In order to supply the circuit with all the components, a charger is needed to supply the circuit with 220V (for the heater) and 5V for microcontoller and the other modules.

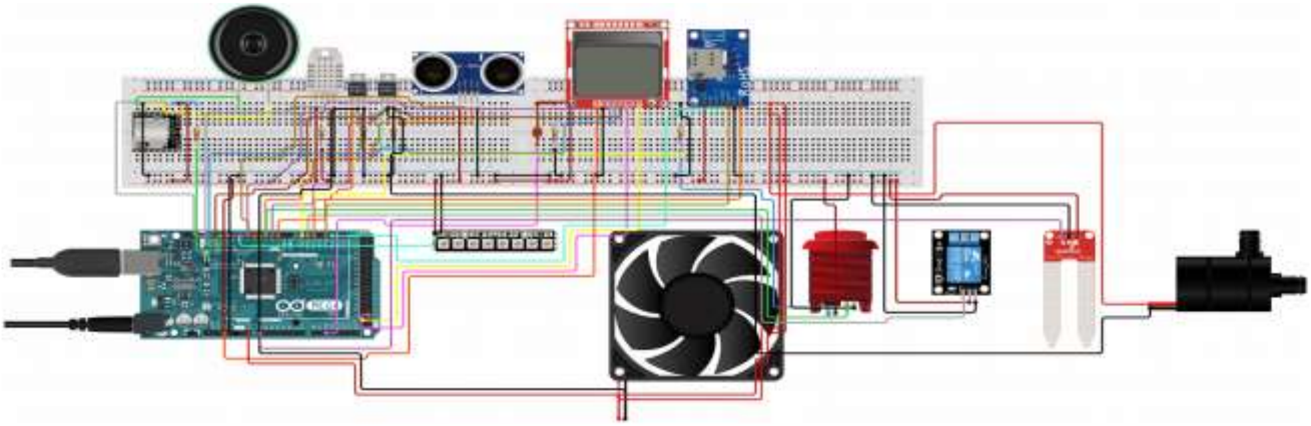


Fig. 2: Wiring the system

Figure 2 shows how the components were connected on the breadboard, the simulation was made using Circuito which is an Web based app that allows users to create simulations for their projects using ready-made components.

4.3 Software Requirements

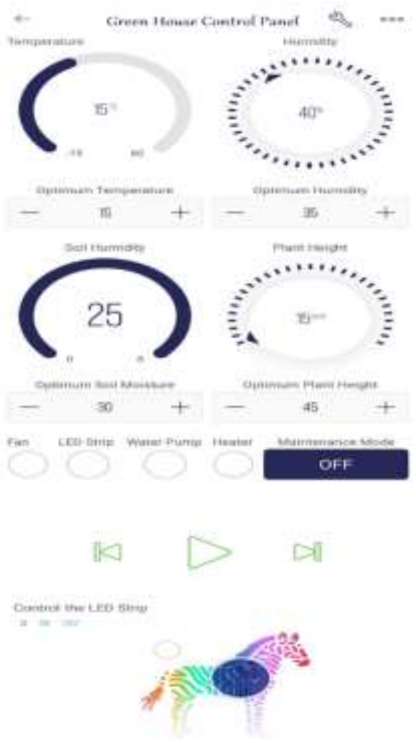


Fig. 3: Blynk Control Interface

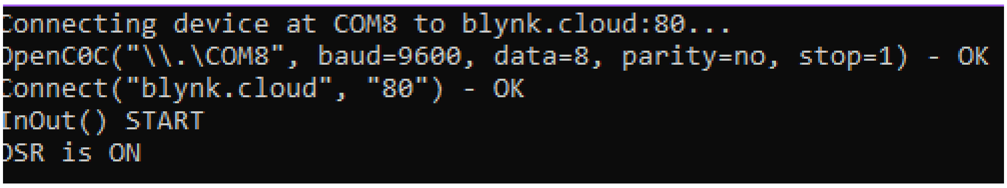
Blynk is an open-source electronics prototyping platform that is a modified version of wiring/Arduino IDE with iOS and Android apps to control Arduino and the likes over the Internet. It is a digital dashboard where a graphic interface for any project can be built by simply dragging and dropping widgets. The final control interface is shown in Fig 3.

5 Prototyping

Since a GSM module can be purchased at affordable prices from China, I chose to make the internet connection using the serial port of the Arduino development board. This required the installation of the Blynk library in the Arduino IDE and the opening of port 80 in the router settings. After performing the presentation steps, the blynk-ser.bat file available in the Blynk library had to be modified to send data to blynk.cloud and use the port assigned to connect the computer development board.

```
COMM_PORT=COM8
set SERV_ADDR=blynk.cloud
set SERV_PORT=80
```

After running the script from the Blynk library, I got an internet connection that sends the datato the Blynk cloud server.



```
Connecting device at COM8 to blynk.cloud:80...
OpenC0C("\\.\COM8", baud=9600, data=8, parity=no, stop=1) - OK
Connect("blynk.cloud", "80") - OK
InOut() START
DSR is ON
```

Fig 4: Connection to the Blynk Server using the serial port

The connection between the Blynk server and the development board is made by assigning data such as the authentication key indicating the device that has been configured, Template ID, indicating the dashboard used and Device Name indicating the name of the device used. All this data is defined in the code header as follows:

```
#define BLYNK_TEMPLATE_ID "TMPLyw2_Xes5"
#define BLYNK_DEVICE_NAME "GreenHousesz"
#define BLYNK_AUTH_TOKEN "JI8X73KXMGsgxy2J6nl8FPItOA-OuIsl"
```

Sensor	Arduino PIN	Declaration
Humidity and Temperature sensor (DHT11)	2	#define DHTPIN 2 #define DHTTYPE DHT11 DHT dht(DHTPIN, DHTTYPE);
Soil Humidity (Spark fun)	A0	int soil_humidity_pin = A0;
Relay used for Water Pump	6	int water_pump=6;
Relay used for Fan	13	int fan_pin=13;
Light Sensor	A3	int light=A3;
Button used for maintenance mode	11	int button=11;
Ultrasonic Senor	3 and 4	#define echoPin 3 #define trigPin 4

Table 2: Connecting the sensors to the microcontroller

To read the data from the sensor I created a method that read the data in loop. It reads data from DHT11 sensor and sends the data to Blynk Server.

```
void sendSensor()
{  float h = dht.readHumidity();
  float t = dht.readTemperature();
  if (isnan(h) || isnan(t)) {
    Serial.println("Error for reading data from sensor!");
    return;
  }
  Blynk.virtualWrite(V5, h);
  Blynk.virtualWrite(V6, t); }
```

In the prototyping phase, I faced an anomaly. After reading the soil moisture if the soil was drier than the optimum value entered in the Dashboard then it was necessary to start the water pump. In my case the water pump was started by the relay but could not be stopped. This problem has been solved by adding a 1n4007 doid between the VCC and GND pins to allow the electric current to flow in one direction only.

The distance was computed using the duration and the sound speed divided by 2 for traveling back and forth (emission and reception).

```
distance = duration * 0.034 / 2;
```

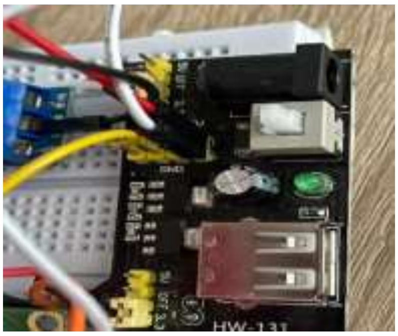


Fig. 5: HW-131 power supply

The power for all components of the development board using the serial port that connects the development board to the computer could not be realized because the power supply port of the computer has a current of 100 mA. To solve the problem, a HW-131 power supply was used; it powers the fan and the water pump using a 9V battery or it can be directly plugged in into a socket using a phone charger. [Fig 5]

The retrieval of the data read by the sensors is done by calling the corresponding subprogram but using the delay function; the reading of the other sensors is affected and the sending of the data to the Blynk server is no longer performed. The solution found was to start a timer and call the functions in setup() , timer.setInterval (2000L, soil_humidity); where the first parameter is the data reading frequency, and the second is the subroutine for data reading.

As the reference value is a fixed number and parameters such as temperature can fluctuate rapidly this would continuously operate the relays and could damage them.

Hysteresis is the deviation of the humidity allowed by the microcontroller before switching on or off the relay that operates the water pump. A low hysteresis value ensures lower fluctuations in soil moisture but can lead to too frequent relay starts. To minimize this risk, it can be used Hysteresis [13], by using that method we can set an threshold value in order to turn on the pump when is necessary.

```
 if(value>=op_soil+50 && value>=op_soil-50)
 {
  WATER.on();
  digitalWrite (water_pump, LOW);
 }
else    {    digitalWrite (water_pump, HIGH);
WATER.off();
 }
```

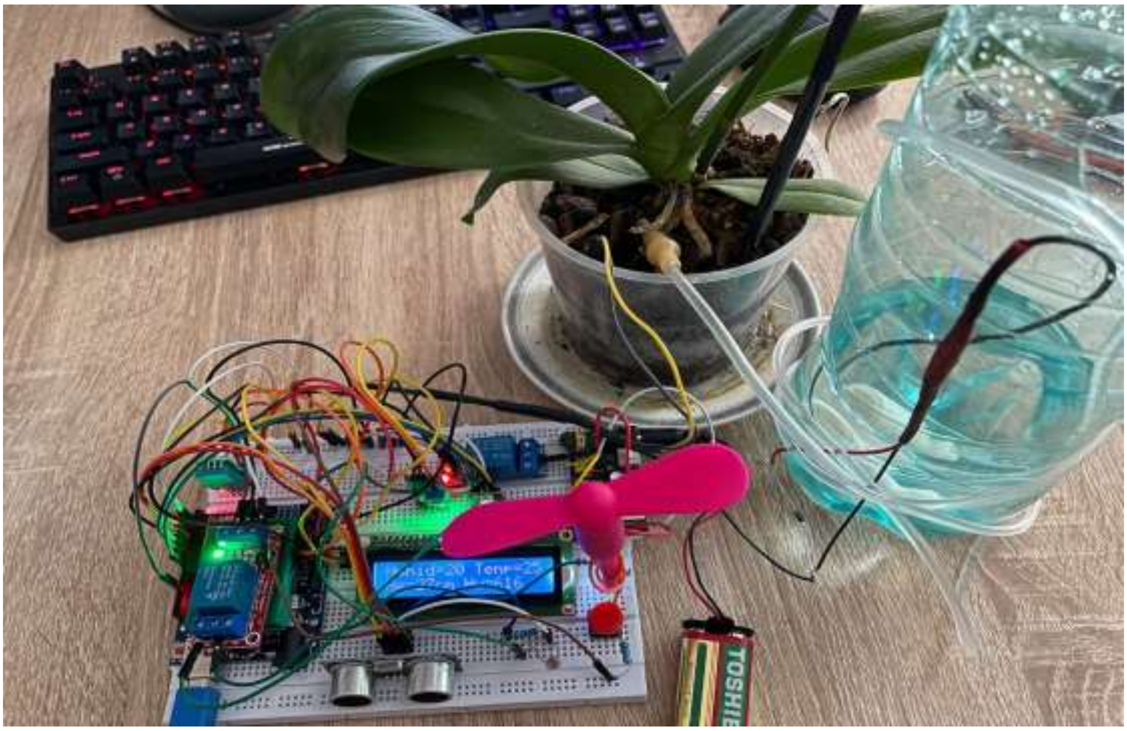


Fig. 6: Current state of prototype

The cost of the project is $ 140 but can be reduced by using a cheaper development board such as the Arduino Uno R3 and ordering components from China. The total price can be halved by buying components from China. Comparing the project presented with the micro greenhouses present in the market [11], [12] is around 250$-400$ this project is at a much lower price and can be scaled according to the needs of everyone from domestic to industrial use.

6 Conclusion and future work

The technology can be used to our advantage so that we can automate certain processes that require wrkforce and time. IoT technology allows the control and monitoring of greenhouses remotely. Such a system not only brings economic benefits, but also accelerates the process of plant growth. The system can be used both for domestic use but can be scaled up, so that it can be used in the agricultural industry.

In the future, this system may have a photovoltaic system so that it becomes completely autonomous and a set of cameras that monitor the plants and transmit the video flow to an artificial intelligence system in order to detect certain diseases. In fact, a wireless ESP 01 module (which is small and has a low energy consumption) can be added to each sensor. This would make the sensors independent and will send the data to the Cloud, following to be processed and displayed in the Control Panel and on the integrated Display. Adding all te components to the project will be mandatory.

Acknowledgement: This work was supervised by Professor *Crișan Gloria Cerasela*, from *Department of Mathematics & Informatics, Vasile Alecsandri University of Bacău.*

References

[1] J.delSagrado, J.A.Sánchez, F.Rodríguez, M.Berenguel, N*etworks for greenhouse temperature control*, Journal of Applied Logic, vol. 17, pp.25–35, September 2016.

[2] Olga Długosz-Grochowska, Anna Kołton, Renata Wojciechowska, *Modifying folate and polyphenol concentrations in Lamb's lettuce by the use of LED supplemental lighting during cultivation in greenhouses*, Journal of Functional Foods, vol. 26, pp.228–237, October 2016.

[3] Anindita Roy Chowdhury, Anshu Gupta, *Effect of Music on Plants, An Overview*, Journal of Science Innovation Technology, Vol. IV Iss 6, 28 December 2015

[4] N V Titovskaia, T S Titovskaya, S N Titovski, *Application of the IoT technology in agriculture*, OP Conference Series: Earth and Environmental Science, 2020

[5] J.delSagrado, J.A.Sánchez, F.Rodríguez, M.Berenguel, *Networks for greenhouse temperature control*, Journal of Applied Logic, vol. 17, pp.25–35, September 2016.

[6] Al-Adwan and M. S. Al-D, *The use of zigbee wireless network for monitoring and controlling greenhouse climate*, International Journal of Engineering and Advanced Technology (IJEAT), vol. 2, pp. 35-39, 2012.

[7] G. Qiang and C. Ming, *Research and design of web-based wireless sensor network management system for greenhouse*, Computer and Electrical Engineering, 2008. ICCEE 2008. International Conference on, 2008, pp. 657-661.

[8] Mihai Bogdan, *Greenhouse. Monitoring and Control of Temperature and Soil Moisture* ,https://www.researchgate.net/publication/337440140_Greenhouse_Monitoring_and_Control_of_Temperature_and_Soil_Moisture

[9] S. R. Kakade, R. G. Butale, S.V. Gupta, *Automation of Greenhouse Using Arduino and GSM Module*, Trends in Biosciences 11, 2020.

[10] Lady Ada, *DHT11, DHT22 and AM2302 Sensors*, 2021, https://cdn-learn.adafruit.com/downloads/pdf/dht.pdf

[11] How HC-SR04 Ultrasonic Sensor Works & Interface It With Arduino, *https://lastminuteengineers.com/arduino-sr04-ultrasonic-sensor-tutorial/#:~:text=HC-SR04%20Hardware%20Overview&text=The%20one%20acts%20as%20a,the%20distance%20the%20pulse%20travelled*.

[12] A. Snigh Chib, A. Ahmed and S. Arya, *"Semiconductor Photoresistors"* in I.M.I Ahamed, R. Boddula and T. Altalhi, Optical Properties and Applications of Semiconductors (1st ed.). CRC Press, 2022, 16 pages.

[13] Anton Van der Ven, Kimberly A. See, Laurent Pilon, *Hysteresis in electrochemical systems*, *https://www.researchgate.net/publication/359290790_Hysteresis_in_electrochemical_systems*

Sorin Ionut CONEA
Vasile Alecsandri University of Bacău
Department of Mathematics & Informatics
Calea Mărăşeşti, nr. 157, Bacău, 600115
ROMANIA
E-mail: coneasorin@outlook.com

Sixth International Conference on Applied Informatics
Imagination, Creativity, Design, Development
ICDD 2022, May 5-7
Sibiu, Romania

Smart Bicycle Parking Monitoring System

Luca-Dumitru Drîndea, Bogdan-Ioan Oprișiu, Mircea-Alexandru Răuțoiu

Abstract

The article presents the process of developing and prototyping a smart monitoring system for parking spaces on any rack. The final product is based on a hardware and a software component. Physically, the system consists of several Arduino development boards (slaves) that send information about vacancies through sensors and an Arduino board with a Wi-Fi module (master) that receives the information and transmits it further in a database. The software component consists in processing the information received by the sensors from the environment. The information is taken by the mobile application from the database, an application where users can easily view the availability of parking spaces for personal bicycles.

1 Introduction

In this article, we will detail the steps we took during the research and development of our project prototype, as well as its technical specifications.

Cycling is a beneficial activity for the health of the individual, as well as for the environment, by removing automobiles from the road. This also has the indirect benefit of reducing congestion in inner cities, thus further reducing pollution.

Our aim with this system is to ease the finding of bicycle parking spots. Our users will be able to find free parking spots using our mobile app. This project is unique in Romania. We believe that this project will help dramatically in the development of Romania bicycle infrastructure and reduce our carbon footprint. We tested our project in the front yard of our high school with moderate success and with some improvements it will be ready to succeed in the free market.

The system can be separated into two parts: the mobile app and the “SmartRack”. “SmarRack” is the name we have given our system based on arduino boards and ultrasonic sensors that will be found on a normal bicycle rack. Every system consists of a master arduino and its slaves.

Technologies used: Kotlin, C++, Firebase, Android

2 Functionality

When the ultrasonic sensors detect the presence of a bicycle, they send the information to a slave which sends it further to the master. The master transmits the information to our Firebase database by using an integrated Wi-Fi module. The mobile app is then able to use the information provided by the master and display how many spots are free on any SmartRack. We equipped the master with a screen to be able to display the free spots without checking the mobile app called BrukBike. BrukBike offers a map of all regular bicycle racks as well as the smart Racks. The mobile app offers the possibility of adding any existing rack on the map by the app's users. This way we aim to create a map of all bicycle parking spots using the power of the community. To limit the abuse of this power we have conceived a reputation system. Only reputable users will be able to add new parking spots.

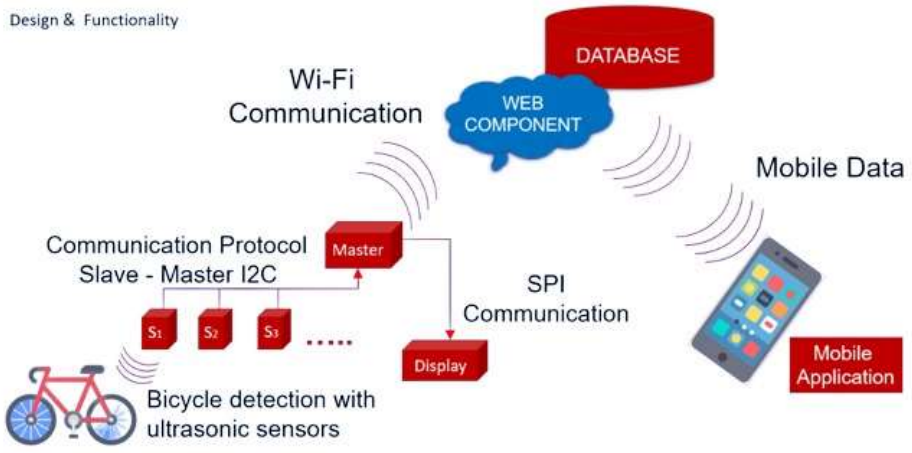


Fig. 1: Design and Functionality

3 SmartRack-Hardware

The master is an Arduino ESP8266. We have chosen have chosen this model, because we need the master to be permanently connected to the internet and to send information to our data base. Alongside the master there is found an LCD display on which the amount of free parking spots will be displayed.

Fig. 2: Master - Arduino ESP8266

Fig. 3: Slave - Arduino uno

The master is connected to the slaves using the i2c protocol [1]. This protocol offers the possibility to have up to 256 interconnected devices, making possible the theoretical use of 255 slaves alongside one master on a single bicycle rack. Each slaves can have up to 6 sensors, making it possible to have a maximum of 1530 different parking spots, in theory. This is the absolute limit of our current technology, but we have not gotten even close to these limits in our tests, so in a situation like this we will most likely encounter signal problems in the cable. One solution might be the wireless transmission of data from slave to master, but this will most likely require a lot more testing and development in the future.

The slave is an Arduino Uno. We have chosen the model because of its excellent price to quality ratio. This board has 13 digital and 6 analog pins at a reasonable price, making it perfect for large scale projects like this one. Each sensor needs two digital ports to function and 2 analog pins are occupied by the cables that enable the i2c transmission.

To detect bicycles, we use ultrasonic sensors [2]. The sensors send an ultrasonic signal in front of them and measure the time it took for the sound to return to the sensor as seen in the diagram below. By knowing the speed of sound and the time we can extrapolate the distance between the sensor and an object in front of it.

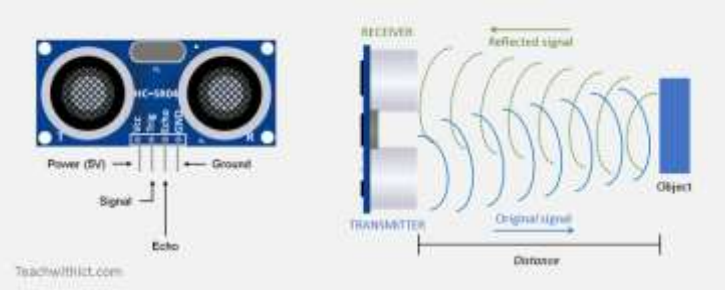


Fig. 4: HC SR-04P Sensor

To protect the boards and the sensors we have designed custom cases which we have 3D printed, cases that protect parts of the connecting cables as well. We have temporarily mounted the cases on the rack double adhesive duct tape and glue but, in the future, we plan to use screws, as a more permanent solution. The master case has a built-in place to allow the installation of an LCD screen. To protect this screen from the elements we have mounted a piece of Plexiglas on top of it.

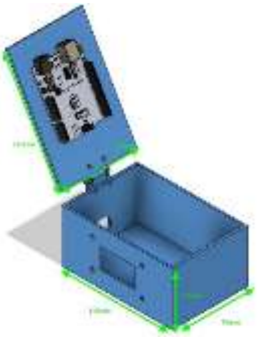


Fig.5: Master case

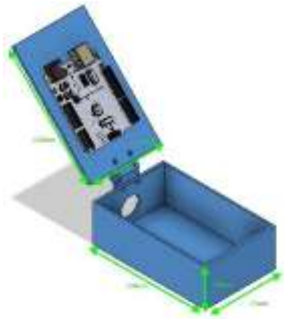


Fig.6: Slave case

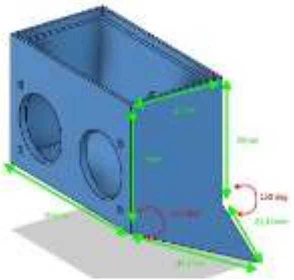


Fig.7: Sensor case

The sensors are the most fragile part of the hardware part of the system. This is the reason that we have opted to make a separate case for each sensor. We have designed the case to be set perpendicular with the rack on the loves of the bicycle wheel. The weird shape of it is given by the rack in our high school's courtyard where we have tested the prototype.

Every sensor has 4 cables that connect it to the master and every slave has two cables that connect it to the master. As we cannot protect all these cables in the cases, we have opted to encapsulate them in a flexible plastic tub.

Fig. 8: Car battery

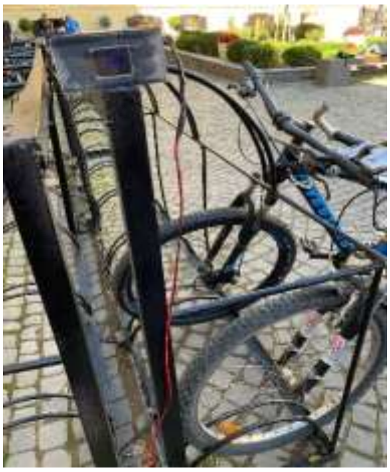


Fig. 9: The installed system

To power our project, we are using a 12-volt car batterie. This batterie can power our system in its current form for approximately two weeks until it must be recharged manually.

4 SmartRack-Software

4.1 Slave Component

To figure out the availability of free parking spaces on the rack, we have connected 6 HC-SR04 sensors on each Arduino board which we refer to as a slave. The code architecture is modular, meaning that it is not necessary to rewrite a different code for each slave, it's only necessary to change the sensor count and the Arduino board's ID through which the master finds from which slave it receives information.

The **UltrasonicSensor** class has functions in which, after every iteration of the program, the distance to the nearest object in front of the sensor is calculated and in which, through a list, it is saved which sensor detects or not something. If one of the sensors detects something we add a 1 to the list and if not a 0. The sensor transmits through the **_loop** function an ultrasonic impulse in front of it and waits for it to come back, computing each time the distance that the impulse needed to come back. The maximum distance for which we are looking is predefined beforehand the compilation of the code. In our case 20 centimeters was sufficient.

```
class UltrasonicSensor{
  public:
  //(Constructor) sets trigger pin and echo pin
  UltrasonicSensor(int trigPin, int echoPin){
    this->trigPin = trigPin;
    this->echoPin = echoPin;
  }
   //initialize trigger pin and echo pin
  void _setup(){
    pinMode(trigPin, OUTPUT);
    pinMode(echoPin, INPUT);
  }
   //computes distance to object
  void _loop(){
    //clear the trigPin:
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);

    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);
    //gets the sound wave travel time in microseconds
    duration = pulseIn(echoPin, HIGH);
    //calculate the distance:
    distance = duration * 0.034/2;
  }
  //returns whether there is an object inside the set distance Threshold
  bool _getResult(){
    if(distance <= distanceThreshold)
      return 1;
      else
        return 0;
  }
  int trigPin;
  int echoPin;
  private:
  long duration;
  //distance to object
  int distance;
  //maximum accepted distance for Bike wheel
  static const int distanceThreshold = 20;
};
```

The following functions are used for the transmission of the information to the master through an I2C protocol. The **initSlave** function is used only one time and informs the master of the number of sensors that the respective slave has. The list of free parking spaces is transmitted by the slave thorugh the **sendData** function. This list of 0's and 1's lets us know which parking spaces are free or not, from each slave being able to know in real time the state of 6 parking spaces.

```
//Tell the master how many sensors there are on this slave
void initSlave(){
  firstRun = false;
  Wire.write(sensorCount);
}
//send the result array
void sendData(){
  //Wire.write(results);
  for(char result : results){
    Wire.write(result);
  }
}
```

This way we can know in real time from a multitude of Arduino boards (slaves) information about the availabilty of parking spaces for bicycles on various racks.

4.2 Master Component

This component consists of an Arduino ESP8266 board with a Wi-Fi module which makes possible the transmission and saving of information to the Firestore Database because of the connection to it.

The **connectToWiFi** function connects in first phase the master to the internet by searching for a SSID, meaning the name of the internet network to which we want to connect. To do this we need to have the password and name of the internet network declared previously.

```
void connectToWiFi() {
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("connecting");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
  Serial.println();
  Serial.print("connected: ");
  Serial.println(WiFi.localIP());
  Serial.println();
}
```

Through the **setupFirestore** function we can connect to the database with the added help of some special libraries made by a contributor on GitHub for such purpose [3]. For this connection, we need a secret key named API_KEY offered by the database and an email address as well as a password so that the Arduino board can log in to this database through a Google account.

```
void setupFirestore() {
  config.api_key = API_KEY;
  auth.user.email = USER_EMAIL;
  auth.user.password = USER_PASSWORD;
  config.token_status_callback = tokenStatusCallback; //see addons/TokenHelper.h
  Firebase.begin(&config, &auth);
  Firebase.reconnectWiFi(true);
}
```

After all the connections have been made, we are able to send in real time the information about each parking space from a SmartRack as a list to the database. We send this information to the Firestore Database with the help of a function called **sendDataToFirestore** which has elements from the libraries mentioned previously.

```
void sendDataToFirestore(vector <bool> spots) {
  if (Firebase.ready()) {
    Firebase.reconnectWiFi(true);
    FirebaseJson content;
    String documentPath = "Racks/0";
    content.set("fields/occupiedSpots/integerValue", String(totalOccuSpots).c_str());

    for(int i = 0; i<spots.size(); i++){
      String temp = "fields/Spot_";
      temp += i;
      temp += "/integerValue";
      content.set(temp, String(spots[i]).c_str());
    }
    Serial.print("Update a document... ");

    if (Firebase.Firestore.patchDocument(&fbdo, FIREBASE_PROJECT_ID, "" /* databaseId can be
(default) or empty */, documentPath.c_str(), content.raw(), "occupiedSpots,Spot_0,Spot_1,Spot_2,Spot_3" /*
updateMask */))
      Serial.printf("ok\n%s\n\n", fbdo.payload().c_str());
    else
      Serial.println(fbdo.errorReason());
  } }
```

4.3 The Firestore Database

Firestore Database, an extension of Firebase, is a database provided by Google for the development of mobile and web applications with the purpose of saving information for later use [4]. It is a NoSQL database with a powerful interrogation engine, with live and synchronization and an offline mode, reasons for which we chose to use it for the development of our application.

4.4 Security

The security consists of some rules written in Firestore in which we specify that only our Arduino board has the permission to save data from various SmartRacks, connecting to it with an email address and a password, so that no one else can change the information in the database. The exclusive permission of the Arduino board to write in the database is given by the following rule:

```
match /Racks/{
  allow write: if request.auth.token.email == “luca.drindea@gmail.com”;
           allow read: if true;
           }
```

5 The Mobile Application

We chose to use the Android Navigation Component to achieve a simple and efficient organization of our application. The Navigation Component allows us to divide our app into standalone, but interconnected fragments, the most important one being the **mapFragment.** As the name suggests, the **mapFragment** has a map with all the useful information.

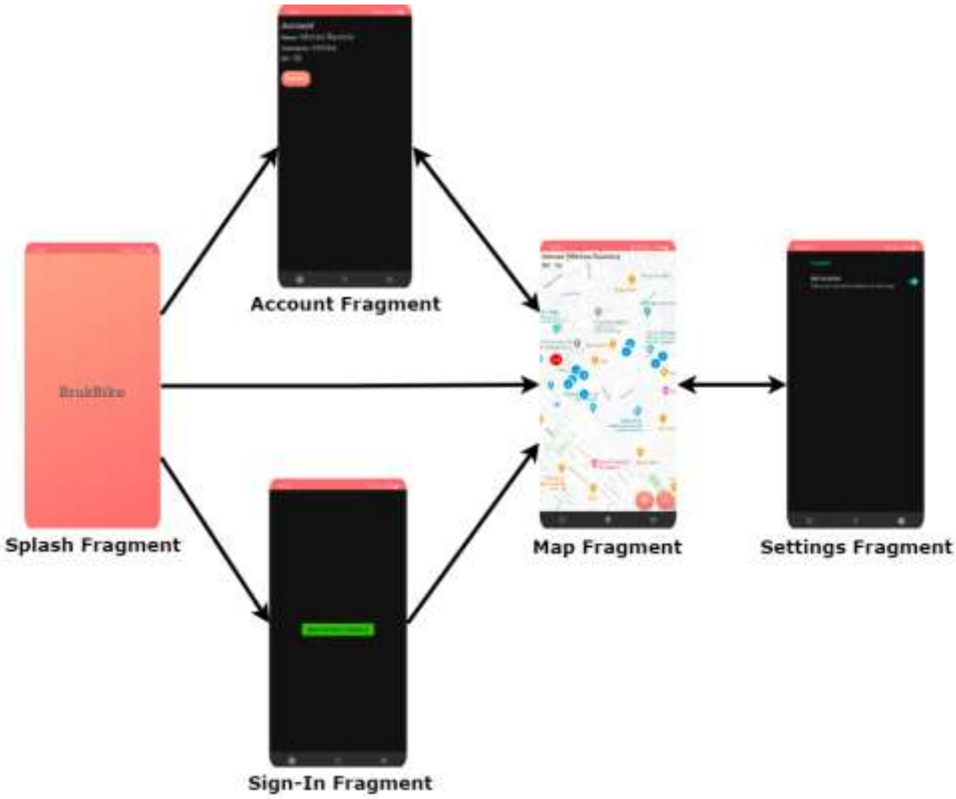


Fig. 10: Fragments scheme

The mobile application uses multiple APIs from the Google Workspace suite to fetch, send and display the necessary data. BrukBike would be pointless without an intuitive way of presenting the information to the user, so the first API that the application uses is:

5.1 Google Maps API

Obviously, the **Google Maps API** is used in the **mapFragment**. The most important method of this fragment is **onMapReady**, a function that gets called when the map is fully loaded and ready to be used. If the user grants the application permission to use the device's location, the map will initialize in a useful perspective, showing the racks in the vicinity of the user. The **mapFragment** is also responsible for data validation. Users can sign in via a Google account, functionality provided by the **Firebase Auth API**.

```
override fun onMapReady(map: GoogleMap) {
   this.map = map

   //check location permission
   if (ContextCompat.checkSelfPermission(
         requireContext(),
         Manifest.permission.ACCESS_FINE_LOCATION
      ) == PackageManager.PERMISSION_GRANTED && usesUserLocation
   ) {
      locationPermissionGranted = true
      map.isMyLocationEnabled = true
   }
   //camera to user position
   panToMyLocation()

   initMap()
   map.uiSettings.isMyLocationButtonEnabled = false
   map.uiSettings.isCompassEnabled = false
   addSmartRackMapMarkers()
   addUserRackMapMarkers()
}
```

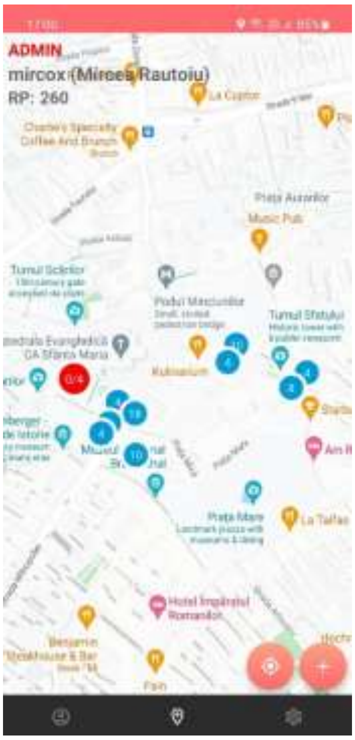
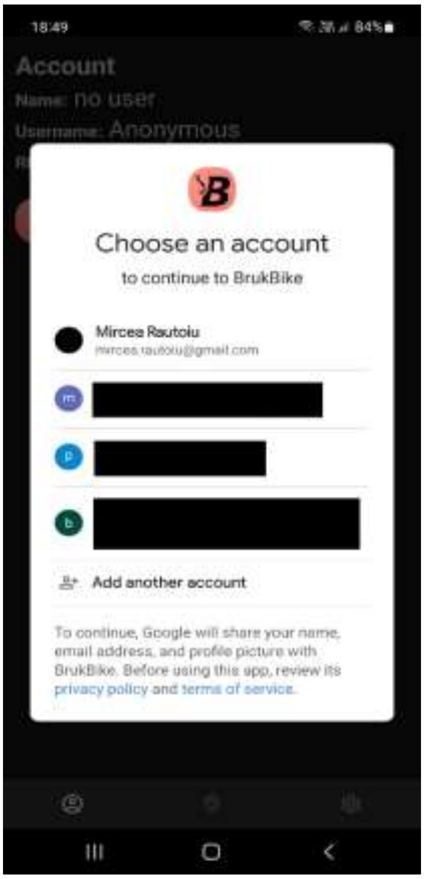


Fig. 11: Map

5.2 Firebase Auth API

The **AuthHelper** class contains all the methods used to authenticate the users. By using an object of the type **GoogleSignInOptions** we can set all the appropriate parameters and the type of authentication, in this case via a Google account. Finally, we create the **activityResultLauncher**, better known as the sign in pop-up that prompts you to choose an account. [5]

```
class AuthHelper(
    val fragment: Fragment,
    private val onAuthCompleteListener: OnAuthCompleteListener
 ) {
    private val gso = GoogleSignInOptions.Builder
(GoogleSignInOptions.DEFAULT_SIGN_IN)
        .requestIdToken("xxxx.apps.googleusercontent.com")
        .requestEmail()
        .build()

   private val mGoogleSignInClient: GoogleSignInClient =
       GoogleSignIn.getClient(fragment.requireContext(), gso)
    private val mAuth = FirebaseAuth.getInstance()

   private val activityResultLauncher = fragment.registerForActivityResult(
       ActivityResultContracts.StartActivityForResult()
    ) { result ->
       if (result.resultCode == Activity.RESULT_OK) {
           val task =
GoogleSignIn.getSignedInAccountFromIntent(result.data)
                            try {
                           val account = task.getResult(ApiException::class.java)!!
              Log.d(TAG, "firebaseAuthWithGoogle:" + account.id)
              firebaseAuthWithGoogle(account.idToken!!)
          } catch (e: ApiException) {
              // Google Sign In failed, update UI appropriately
              Log.w(TAG, "Google sign in failed", e)
          }
       }
    }
         }
```


18:49 84%

Account
Name: no user
Username: Anonymous

Choose an account
to continue to BrukBike

Mircea Rautoiu
mircea.rautoiu@gmail.com

[redacted]

[redacted]

[redacted]

Add another account

To continue, Google will share your name, email address, and profile picture with BrukBike. Before using this app, review its privacy policy and terms of service.

Fig.12:Authentification pop-up

5.3 Cloud Firestore API

This API enables the interfacing between the mobile application and the **SmartRack** systems through the Firestore cloud database. In addition to storing the real-time S**martRack** data, the Firestore database helps us retain further information about our users, such as **RP (Reputation Points)** and the racks added by them. The mobile application interacts with the database by using three classes: **FirestoreSmartRackDB, FirestoreUserRackDB and FirestoreUserDB**. The aforementioned classes function in a similar manner, but we chose to separate access to the different data categories to accomplish a modular design. The **getRacks** function of the **FirestoreSmartRackDB** is presented below. This method fetches data from **Firestore**, updating the application asynchronously. Fortunately, the **Firestore** library for Android stores the last set of information automatically in a local database, so there is no need to create a separate database for the cases when an internet connection is not available.

```
/**Fetches data from Firestore when called and listens for any updates*/
private fun getRacks() {
   rackReference.addSnapshotListener { snapshots, e ->
      if (e != null) {
         Log.w(TAG, "listen:error", e)
         return@addSnapshotListener
      }

      for (dc in snapshots!!.documentChanges) {
         when (dc.type) {
            com.google.firebase.firestore.DocumentChange.Type.REMOVED -> {
                synchronized(this) {
                    racks.removeIf { it.id == dc.document.id.toInt() }
                }
            }
            com.google.firebase.firestore.DocumentChange.Type.MODIFIED -> {
                synchronized(this) {
                    for (rack in racks)
                        if (rack.id == dc.document.id.toInt()) {
                            val modifiedRack =
                                getRackWithId(
                                  dc.document.toObject(SmartRack::class.java),
                                   rack.id
                                )
                            racks[racks.indexOf(rack)] = modifiedRack
                            Log.i(TAG, "getRacks: MODIFIED")
                        }
                }
            }
            com.google.firebase.firestore.DocumentChange.Type.ADDED -> {
                val id = dc.document.id.toInt()
                val rack =
                            getRackWithId(dc.document.toObject(SmartRack::class.java), id)
                synchronized(this) {
                    if (rack.title != "null") {
                        racks.add(rack)
                        Log.i(TAG, "getRacks: ADDED = $rack")
                    }
                }
            } } }
        onDataChangeListener.onDataSetChanged()
    }}
```

6 Implementation costs

The most expensive part of our project is the hardware component, which consists of the Arduino boards, the ultrasonic sensors, the boxes in which these are found, the cables which connect everything, the source that powers the system and the display which is connected to the Arduino master. We have included these elements in the following table.

Components	Price (lei)	quantity	Subtotal(lei)	Total (lei)
Arduino	30	4	120	500
Arduino esp8266	60	1	60	
Ultrasonic sensor	5	24	120	
Other (battery, cables)	200	-	200	

Table 1: Cost of components

This table describes the components used in our prototype rack. It has lasted in its current form for about two months. This rack has 24 parking spots and access to the school's Wi-Fi. Most of our costs on this prototype come from the car battery that powers it. In the future we plan to use either photovoltaic panels or connect the system directly to the power grid, thus eliminating the cost of the battery. Both variants have their respective benefits and downsides. On one hand the photovoltaic panels are ideal for places where access to the grid is limited and the sun is plentiful, unlike shady areas like the courtyard of our high school. We can conclude that the medium cost of a smart bicycle rack seems affordable, but have no correct measurement counterpart as this concept is unique in Romania.

7 Conclusions and future developments

In conclusion, we will continue to mention further functionalities which we want to implement to the future SmartRacks. We plan to add a system of electronic locks on the racks, thus cutting the need to use a personal lock. These locks will enable us to add a system of online reservation directly from the mobile app, in other words a parking space for a surcharge. Another improvement of the system could be the replacement of ultrasonic sensors with other types of sensors, such as infrared sensors, these being more precise and cutting noise pollution, noise that can affect different animals. Solar panels are another facility which would allow us to power the monitoring system with green energy. This system could in the future have a wider reach, monitoring not just bicycles but also electric scooters or even cars.

Aknowledgement: This work was supervised by Professor Delilah Florea, from Samuel von Brukenthal National College of Sibiu.

References

[1] I2C: *https://docs.arduino.cc/learn/communication/wire*
https://www.circuitbasics.com/basics-of-the-i2c-communication-protocol/
[2] Ultrasonic sensor: *https://lastminuteengineers.com/arduino-sr04-ultrasonic-sensor-tutorial/*
[3] Firestore libraries: *https://github.com/mobizt/Firebase-ESP-Client*
[4] Firestore Database:

- *https://firebase.google.com/docs/firestore*
- *https://firebase.google.com/docs/firestore/query-data/listen*
- *https://firebase.google.com/docs/firestore/query-data/queries*
- *https://firebase.google.com/docs/firestore/security/rules-structure*

[5] Firebase Auth API: *https://firebase.google.com/docs/reference/rest/auth*

Luca-Dumitru DRÎNDEA
Samuel von Brukenthal National College
Mathematics-Informatics
Piața Albert Huet 5, Sibiu
ROMANIA
E-mail: luca.drindea@gmail.com

Bogdan-Ioan OPRIȘIU
Samuel von Brukenthal National College
Mathematics-Informatics
Piața Albert Huet 5, Sibiu
ROMANIA
E-mail: bogdan.oprisiu2003@gmail.com

Mircea-Alexandru RĂUȚOIU
Samuel von Brukenthal National College
Mathematics-Informatics
Piața Albert Huet 5, Sibiu
ROMANIA
E-mail: mircea.rautoiu@gmail.com

Sixth International Conference on Applied Informatics
Imagination, Creativity, Design, Development
ICDD 2022, May 5-7
Sibiu, Romania

On Designing and Implementing a Tomato Shape App for Android

Maxwell Hosler, Craig Akiri, Mircea Ionescu, Esther van der Knaap, Sofia Visa

Abstract

The design and React Native implementation of an Android app for tomato shape classification is presented here. The app functionality includes taking pictures of tomato fruits, segmenting the fruit, classifying the fruit into one of the nine shape categories, and storing the fruit image and its classification into a Google Firebase database.

1 Introduction

Identifying genes that regulate size, shape, taste, and other characteristics of tomato fruit is **important for breeding new tomato varieties** selected for specialized uses. For instance, rectangular tomatoes with high flesh content are desirable for canning and tomato sauce; in addition, their shape prevents them from rolling over from conveyer belts, thus their processing can be easier to automate. Small, colorful, and juicy varieties like yellow pear tomatoes are mostly produced for fresh consumption in salads and they are often found in groceries stores and fresh markets.

The work presented here is part of a larger collaborative project that investigates the **interaction between three known fruit shape genes,** OFPs (Ovate Family Proteins), TRMs (TONNEAU1 Recruiting Motif) and SUNs [1-2,7-9]. This larger project investigates the cell division patterning at various developmental time points to better understand the onset of the fruit shape and how it evolves. This research will be carried first on tomato fruits and then extended to other crops like peppers, beets, carrots, onions. Hoverer, the concern of the work presented here is automating the collection of tomato fruit pictures and the classification of their shape.

Since this work involves modeling large genetic and shape datasets to discover links and correlations between them, automating the process of collecting some of this data is useful. For instance, DNA and images of thousands of fruits are collected from a variety of tomato that are cultivated in several replicas. Thus, an app that takes pictures of tomato fruits, classifies their shapes, and uploads this data on a cloud database, will be very helpful to this research. Here we present the design and implementation of such an app.

2 Design and implementation

The app for shape categorization of tomato fruits is built for the Android operating system using React Native components and Java programming. In addition, the images and their corresponding labels are stored on a Google Firebase database. To model rules for the nine shapes shown in Fig. 1 [3], we use MATLAB and decision trees.

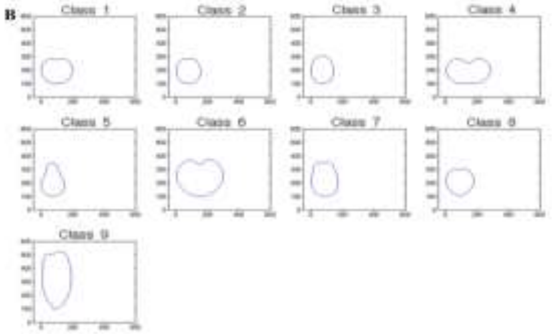


Fig. 1: Nine tomato fruit shape classes [3]

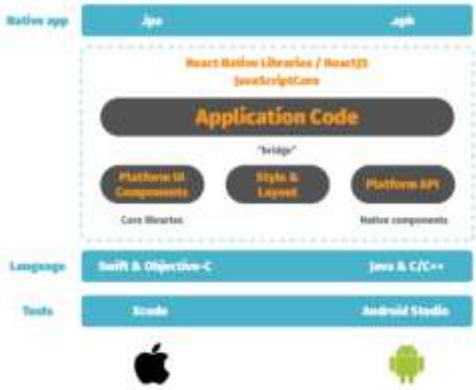


Fig. 2: The React Native libraries allow for code portability between Android and iOS [5]

By design, the app is **partitioned into two parts**: the **classification** of tomatoes is handled by a Java module built on top of the OpenCV library [4]; the rest of the app functionality is handled by a React Native [5] frontend programmed in TypeScript. Storage is handled principally through Firebase while the internal storage is used as an intermediary step caching photos until Internet access is gained. The images and their associated metadata are also stored within the locally synced Firebase database. Fig. 2 shows that React Native libraries support the portability of apps between Android and iOS operating systems. While this app is currently running on Android platforms, by rewriting from Java to Objective-C (or Swift) the OpenCV classification part from the Application Code layer shown in Fig. 2, this app will also be working on iOS systems.

3 Building the app functionality via React Native

When a **picture is taken,** the system saves it to a temporary location and uses a URI (Uniform Resource Identifier) to refer and identify this image. The URI of this location is passed to the app, which generates a unique ID for it and then moves it to a permanent location, under the name {imageID}normal.jpg. The URI of this location is, in turn, passed to the Java/OpenCV backend which in turn segments the image by discovering which pixels are part of the tomato and thus creating a mask – this part of the app is illustrated in Fig. 3. This mask is then classified using if-then rules discovered outside the app, as explained later in this paper.

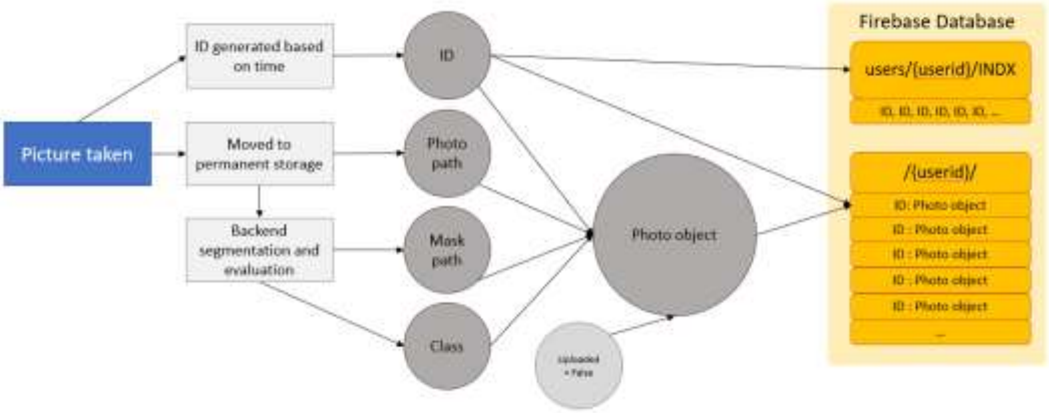


Fig. 3: The process that occurs whenever a photo is taken with the app. Light gray boxes are processes, dark circles are data, and the orange box is the online database.

The classification label and a base64 representation of the mask are then returned to the React Native frontend, which saves the mask as a PNG image in the same local directory as the original photo, under the name {imageID}mask.png. The ID (identification), paths to both the photo and mask, and the classification are bundled into an object, in addition to a thumbnail field (which is simply the path to the original photo) and an 'uploaded' flag, which is set to false. All of these are added to the Firebase database, indexed under the user's UID (user ID) and the photo's ID. In addition, this photo ID is added to the list in the field users.{UID}.INDX, which contains a list of all image IDs for a given user, as illustrated in Fig. 3.

Next, the app will attempt to upload to Firebase; if this fails, uploading will have to be done manually later when an Internet connection is available. After a successful upload, the 'uploaded' flag of the corresponding tomato object is set to true.

3.1 Storing images and metadata into the Firebase database

Uploading to Firebase is a relatively simple process. The function uploadImageFirebase receives an instance of the image data structure Photo and checks the upload flag; if it is true, the image is already uploaded, and the function terminates. Otherwise, it uploads both the photo and the mask into the folder users/{UID}/{imageID}/ with the names normal.jpg and mask.png, as shown in Fig. 4. Next, a copy of the image data object is made, and the fields for its photo and mask URIs are replaced with a generated Firebase download link. In addition, new fields are added indicating their location within the Firebase storage file structure. The uploaded flag is set to true, and this new object replaces the old image object.

All of this is done within a try/catch block, so if any of the steps preceding the database update fail, the database will not be updated. Even in the case that one or more files are uploaded before an error occurs, this will not cause much of an issue as they will be overwritten when the image is successfully uploaded, and until then they will sit in the database, entirely unreferenced. The only issue that may occur is in case of an upload failure with the image being deleted from the local storage. Then, these images would be 'orphaned,' unreferenced with no way of deleting them. However, this will not cause problems on the user end, and all it will do on the admin end is take up space. It would be relatively easy to write a script that deletes such unreferenced images should that become necessary.

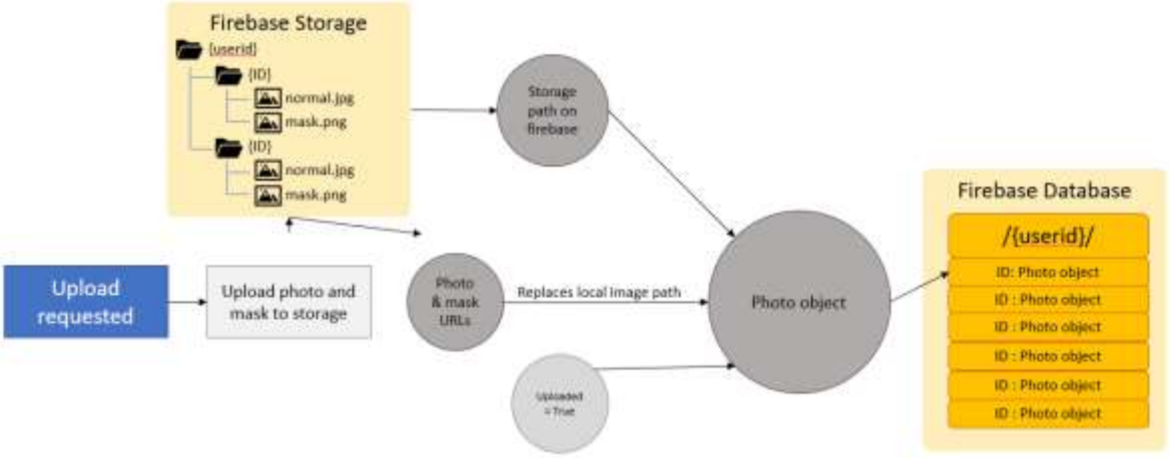


Fig. 4: An upload request uploads to Firebase database all photo objects with the 'upload' flag set to False.

3.2 The graphical user interface (GUI) design

The **GUI of the app is** relatively simple. There are two ‘tabs,’ a camera tab and a gallery tab shown in the navigation bar of Fig. 5. The **camera tab** generates a screen containing only the camera viewfinder and a button to capture the image. The **gallery screen** illustrated in Fig. 6 is a standard image gallery, showing image thumbnails in a grid format, and allowing the user to tap any image to see it up close. However, unlike in a standard image gallery, images that have failed to upload are greyed out. In the single picture view of the gallery screen, the image mask is overlayed onto the image, and its classification is displayed above it as shown in Fig. 7. This single picture view of the gallery screen includes a black bar at the bottom containing two buttons: one button takes the user back to the main gallery view, while the other button brings up a menu with options to delete or upload that image. Tabs can be swapped at any time using a tab bar at the bottom.

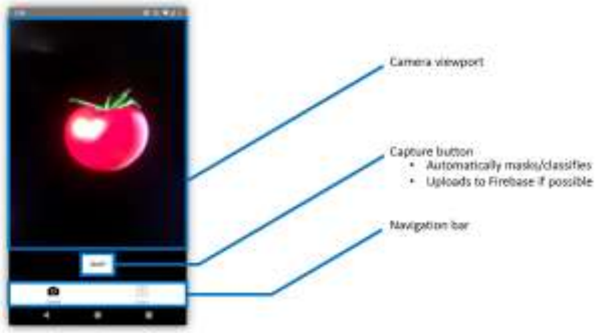

Camera viewport
Capture button
• Automatically masks/classifies
• Uploads to Firebase if possible
Navigation bar

Fig. 5: The camera tab includes a button for capturing the picture and an app navigation bar.

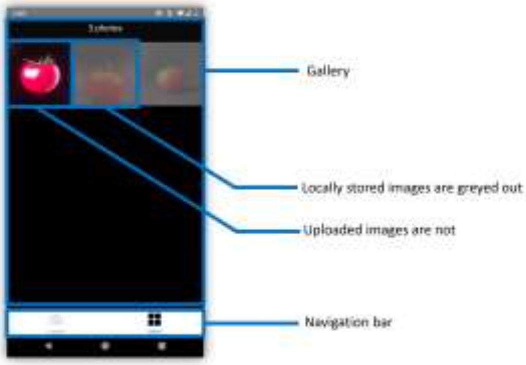


Fig. 6: The gallery tab shows captured images in a grid and includes an app navigation bar.

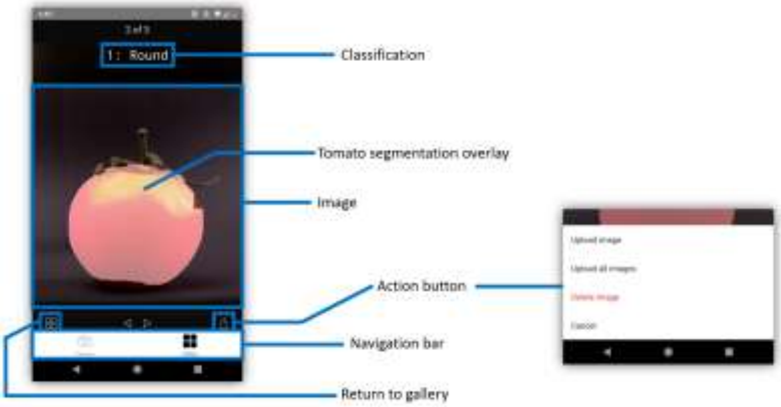


Fig. 7: Single picture view of the gallery screen with two buttons: return to gallery button and upload/delete action button.

3.3 Image segmentation using OpenCV to identify tomato fruits

This section describes the process of identifying tomatoes within an image from staging the photograph to generating a mask of the tomato to latter on be classified. To experiment with parameters and with Open CV’s capabilities for modeling shapes within an image we used Jupyter Notebook along with matplotlib, NumPy, and OpenCV libraries. Thereafter, the code was translated to Java to be ported into the app. These two steps are necessary as it will be inefficient to write and test this image recognition and classification module directly into the app. One of the inefficiencies stems from the need to re-compile and start up the app numerous times while developing the segmentation and classification code. Next, we detail the steps taken to create a robust mask of the tomato fruit image.

1. **Taking the images.** To create an accurate mask the source image must meet certain requirements. First, the tomato should be the subject of the image. Next, a uniformly colored background is needed because the segmentation process uses color information to identify the tomato. We have found that using a black paper/cloth produces a finer mask, while glossy or white backgrounds create reflections which negatively affect the final mask, as shown in Fig. 8. Shadows should also be minimized by using soft lighting and by avoiding flash. When placing the tomato in the scene, it should be held by the stem to avoid holding the tomato. Fig. 9 includes several recommended ways for taking a photo of the tomato fruit

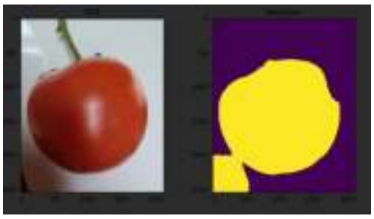


Fig. 8: White or glossy backgrounds create reflections or shadows that negatively affect segmentation.

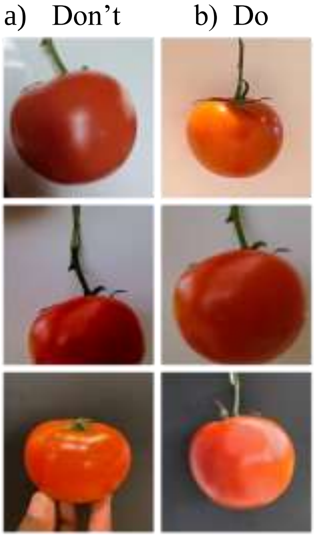


Fig. 9: Column a) shows ill ways of taking a photo; the recommended way is sown in the bottom picture of column b).

2. Image Segmentation. When the Snap button is pressed, the app saves locally the image of the tomato fruit as jpg, and passes it to the segmentation and classification modules. The segmentation function takes in the jpg as an rgb (red, green, blue) image, and then uses OpenCV calls to convert it into a HSV (hue, saturation, value) color model. The first mask (or segmentation) is then generated by extracting a range of colors from the image i.e., the reds and yellow hues.

The following OpenCV Java command returns a mask from an hsv image passed as the first parameter to the inRange() function.

```
mask1 = cv2.inRange( hsv, np.array([0, 110, 50]), np.aray([20, 255, 255]) )
```

The mask (or segmentation) created at this stage is a 2-dimensional array containing 0 and 1 values, with 1 signifying cells belonging to the object. The two parameter arrays pass to the function red-ish and yellow-ish thresholds. For instance, the inRange()call from above keeps in the mask only pixels in the specified range, i.e. with hue, saturation, and value in the ranges 0-20, 110-255, and 50-255, respectively. Fig. 10 shows the initial jpg images of the tomato fruits and their corresponding segmentations after an inRange() function call. However, this color thresholding is imperfect, as it picks up pixels outside of the target object, as shown in Fig. 11, where the yellow spots circled in red belong to the background not the fruit.

a) Original jpg image

b) Segmented mask after inRange() call

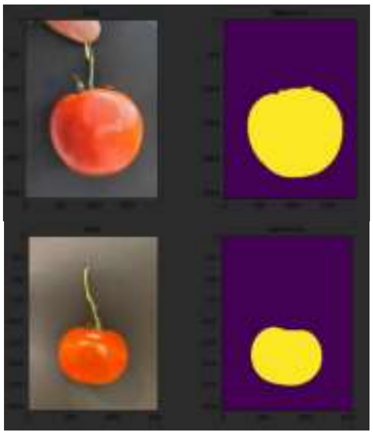


Fig. 10: Second column shows the segmentations of the original jpg images shown in the first column.

To remove these small imperfections that can occur in the first mask we apply an opening morphology transformation. This process firsts erodes then dilates all the objects in the mask to remove any specks that can cause errors when calculating the area of the mask [4]. In a nutshell, this process runs a 3x3 pixel kernel throughout the image and replaces the center pixel of the kernel with the minimum value of the kernel – this is the erosion step which basically shrinks any object by one pixel because all the pixels on a perimeter will be assigned to the background. Next, via the dilation step we grow the objects to the original size via a 3x3 kernel in which the center pixel is replaced by the max pixel value in the kernel. This effectively removes any spots of size one; if we apply this process with a 5x5 kernel, we will be removing larger spots consisting of two pixels, and so on. This process also helps in removing isolated small objects that belong to the stem of the tomato fruit. In sum, Fig. 12 illustrates how this process removes the small specks which do not belong to the tomato fruit.

Next, we calculate the largest area among all objects (or spots) detected in the mask shown in the left image of Fig. 12. These objects are “blobs” of various sizes, and by keeping only the largest blob, and by assigning the remaining ones to the background, we remove the imperfections in the mask.

a) Original jpg image b) Segmented mask after c) Segmented mask after
inRange() call eroding and dilating

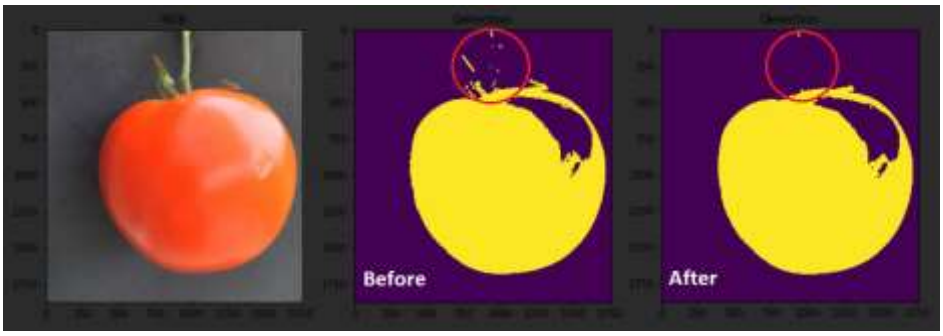


Fig. 11: Small specks are removed by repeated eroding and dilating steps.

Sometimes images have shiny spots on the tomato area which are not within the red and yellow parameters of the *inRange()* function, and thus those spots are classified as background instead of fruit. These spots are visible in Fig. 12 as “holes” in the mask. To address this, we fill in all holes with area smaller than the largest object with the same value as the neighboring pixels. This uses the built in floodFill() method from the OpenCV library. The final mask shown in the right picture of Fig. 12 is returned as the output and is passed on for classification.

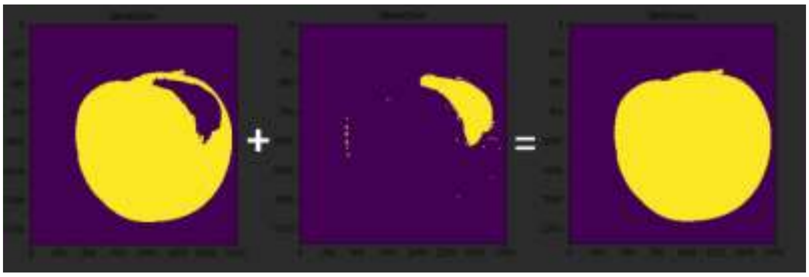


Fig. 12: The flood-fill technique eliminates “holes” in the mask.

4 Building the tomato classification module

To classify the segmented image (or mask) illustrated in Fig. 12, we use several mask properties computed via the OpenCV library. Using these properties for fruit representatives from all nine shape classes, we have built a decision three [6] to extract rules for shape classification. The properties are listed below in the order of their importance in classification. The last two properties (roundness and convexity) were not identified by the tree modeling as useful in classification and thus are not included in the app.

1. **Aspect_ratio = w_ rotated_box / h_ rotated_box** This OpenCV property computes the ratio between the shortest and longest edge of the rectangle enclosing the mask. As shown in Fig. 13 it considers the rotated box if the tomato is not vertically aligned. While for a long tomato such as the one shown in the figure this correspond to the ratio between the width and high of a tomato, on the flat shape (i.e. tomatoes shaped like a pumpkin) it measures the ratio between the height over the width of the tomato because the width is the longest edge.
2. **Min_circle ratio = object_area / area_of_circle_enclosing_object** This ratio measures how close is the object to a circle; if object is exactly a circle, the ratio is 1.
3. **Solidity = object_area / convex_area** This property is used to measures deformities in respect with convexness. For instance, the pattern of a hand with fingers spread and closed

together have a solidity of about 0.4 and 1, respectively. Clearly, in the first case the solidity detects protrusion created by the fingers. This measure will hopefully identify classes having indents in their shape such as classes 4 Flat and 6 Oxheart.

4. **Min_triangle ratio = object_area / area_of_triangle_enclosing_object** This measurement returns 1 if the object is a triangle. This property is useful in distinguishing the Oxheart class which resembles an upside-down triangle – see Table 1.
5. **Compactness = object_area / area_of_circle with same perimeter** This property will measure deformities caused by non-convex shapes.
6. **Extent = object_area / rotated_box_area** This ratio measures how close is the object to a rectangle and thus equals 1 if the object is a rectangle. For instance, in Fig. 13 this property computes the ratio between the fruit (red) area and the area of the blue rectangle, the enclosing rectangle with the longest edge.
7. **Ellipse_ratio = object_area / area_of_the ellipse that best fits the contour points** For a tomato that perfectly fits an ellipse, this ratio is 1.
8. **Roundness = object_area / area_of_circle** This ratio is 1 when the tomato is a perfect circle, but the decision tree did not pick up this measurement as useful in classification and thus it is not used in the app.
9. **Convexity = convex_perimeter / perimeter** Even though we expected this property to help in classifying non-convex shapes such as Flat and Oxheart (see Table 1), the decision tree modeling did not pick up this property and therefore it is not used in our implementation.

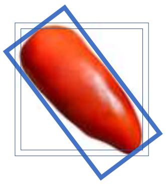


Fig. 13: A rotated tomato objected enclosed in a rotated rectangle.

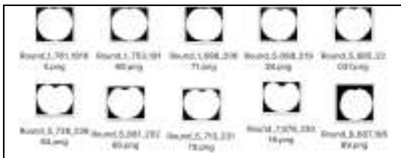
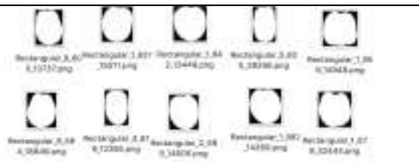
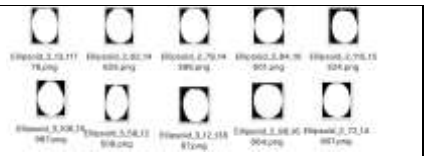
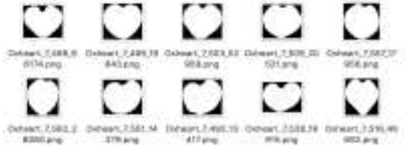
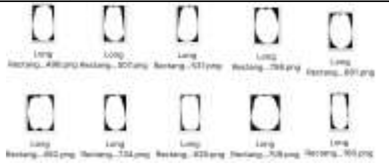
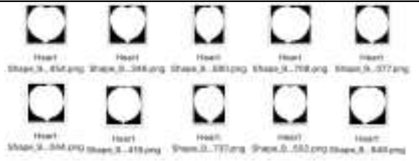
		
a) Class 1 Round	b) Class 2 Rectangular	c) Class 3 Ellipsoid
		
d) Class 4 Flat	e) Class 5 Obovoid	f) Class 6 Oxheart
		
g) Class 7 Long Rectangular	h) Class 8 Heart	i) Class 9 Long

Table 1: Each of the nine classes are represented by ten training images.

Table 1 shows the ten images for each of the nine classes used in the decision tree modeling, which was done in MATLAB outside of the app. For each of the 90 images we compute the corresponding nine properties described above and then model the tree for the nine classes. This produced the 10 classification rules shown in Table 2. Each of these rules uses thresholds for the seven properties described above to define each of the nine classes in a similar way that the following rule describes class 9 Long:

If Min_triangle_ratio < 0.69 and Compactness < 0.80 and Min_circle_ratio < 0.52 then class 9 Long.

These rules were ported into the app and were tested on about 20 fruits. From this limited testing we observed that classes Obovoid and Heart are not as easy to recognize as classes Flat and Long, for instance. But in our future work, with more testing data, we will be able to evaluate the performance of these rule and adapt them for better performance.

5 Conclusions and future work

We present the design and implementation of an Android app for automating data collection for a larger research project. This app allows for taking pictures and classifying tomato fruits according to their shapes. It also stores these images into the cloud for later research such as linking specific genetic markers to fruit shapes. Currently, the app is functioning as envisioned, but needs improvements on the shape classification module. Further, with a few changes, the app can be adapted to be portable into iOS phones.

Challenges in this project include installing all the dependencies for app and creating a proper segmentation of the fruit from the background. A proper segmentation without holes, indents, and protrusions is essential for a successful shape recognition. The next step includes a thorough testing of the app on hundreds of tomato fruit of various shapes, and then improve the app into a new version. In future iterations of the app, we will be refining the segmentation and the shape classification modules. For this work, more testing is required to assess current performance and future improvements. The app requires Android 6 or higher and is freely available for download at GitHub *github.com/mircea-ionescu/tomato-shape*

1 - if min_triangle_ratio < 0.69 and compactness >= 0.80 and aspect_ratio >= 1.02 and min_circle_ratio >= 0.84 then class 1 round.

2 - if min_triangle_ratio < 0.69 and compactness >= 0.80 and aspect_ratio >= 1.02 and min_circle_ratio < 0.84 and ellipse_ratio >= 1.01 and if extent >= 0.81 then class 2 rectangular.

3 (Ellipsoid) - if min_triangle_ratio < 0.69 and compactness >= 0.80 and aspect_ratio >= 1.02 and min_circle_ratio < 0.84 and ellipse_ratio < 1.01 then class 3 ellipsoid.

4 (Flat) - if min_triangle_ratio < 0.69 and compactness < 0.80 and min_circle_ratio >= 0.52 and solidity < 0.97 and aspect_ratio >= 1.3 then class 4 flat.

5 (Obovoid) - if min_triangle_ratio >= 0.69 then class 5 obovoid.

6a (Oxheart) - if min_triangle_ratio < 0.69 and compactness < 0.80 and min_circle_ratio >= 0.52 and solidity < 0.97 and aspect_ratio < 1.3 then class 6 oxheart.

6b (Oxheart) - if min_triangle_ratio < 0.69 and compactness >= 0.80 and aspect_ratio >= 1.02 and min_circle_ratio < 0.84 and ellipse_ratio >= 1.01 and extent < 0.81 then class 6 oxheart.

7 - if min_triangle_ratio < 0.69 and compactness < 0.80 and min_circle_ratio >= 0.52 and solidity >= 0.97 then class 7 long rectangular.

8 - if min_triangle_ratio < 0.69 and compactness >= 0.80 and aspect_ratio < 1.02 then class 8 heart.

9 - if min_triangle_ratio < 0.69 and compactness < 0.80 and min_circle_ratio < 0.52 then class 9 long.

Table 2: Classification rules obtained from the decision tree analysis.

Acknowledgement: This work is funded from the research grant NSF DUNS ID: 004315578 and was supervised by Professor Sofia Visa from the College of Wooster.

References

[1] G.R. Rodriguez, S. Munos, C. Anderson, S.C. Sim, A. Michel, M. Causse, B. McSpadden Gardener, D. Francis, and E. van der Knaap. *Distribution of SUN, OVATE, LC, and FAS in the Tomato Germplasm and the Relationship to Fruit Shape Diversity*. Plant Physiol 156: 275-285. 2011.

[2] NSF Grant for *" Robust organ patterning by OFPs, TRMs and SUNs in plant morphogenesis"* (PI: E. van der Knaap from Univ. of Georgia; Co PI: I. Goldman from Univ. of Wisconsin-Madison, S. Visa, from College of Wooster), 2020-2023.

[3] S. Visa, C. Cao, B. McSpadden Gardener, E. van der Knaap. *Modeling of Tomato Fruits into Nine Shape Categories using Elliptic Fourier Shape Modeling and Bayesian Classification of Contour Morphometric Data*. Euphytica, 200:429-439. 2014.

[4] G. Bradski, *The OpenCV Library*. Dr. Dobb’s Journal of Software Tools. https://opencv.org/ [Accessed on July 2021].

[5] V-V. Helppi. *Diverse Test-Automation Frameworks for React Native Apps*. Smash Magazine. https://www.smashingmagazine.com [Accessed in July 2021].

[6] T. Mitchell, *Machine Learning*. McGraw Hill, 1997.

[7] S. Wu, J. P. Clevenger, L. Sun, S. Visa, Y. Kamiya, Y. Jikumaru, Blakeslee and E. van der Knaap. *The control of tomato fruit elongation orchestrated by sun, ovate and fs8.1 in a wild relative of tomato*. Plant Science 99-104. 2015.

[8] J. Clevenger, J. Van Houten, M. Blackwood, G. Rodriguez, J. Yusuke, Y. Kamiya, M. Kusano, K. Saito, S. Visa and E. van der Knaap. *Network analyses reveal shifts in transcript profiles and metabolites that accompany the expression of SUN and an elongated tomato fruit*. Plant Physiology 168: 1164-1178. 2015. http://dx.doi.org/10.1104/pp.15.00379 [Accessed in July 2021].

[9] E. Sierra-Orozco, R. Shekastaband, E. Illa-Berenguer, A. Snouffer, E. van der Knaap, T. Geon Lee, S.F. Hutton. *Identification and characterization of GLOBE, a major gene controlling fruit shape and impacting fruit size and marketability in tomato*. Hort. Res. 138, 2021.

Maxwell HOSLER
College of Wooster
Mathematical and Computational
Sciences Department
Wooster OH
USA
E-mail: mhosler22@wooster.edu

Craig AKIRI
College of Wooster
Mathematical and Computational Sciences
Department
Wooster OH
USA
E-mail: cakiri23@wooster.edu

Mircea IONESCU
Senior Data Scientist
Assurant Labs
Cleveland OH
USA
E-mail: mircea.ionescu@assurant.com

Esther VAN DER KNAAP
Department of Horticulture
University of Georgia
Athens GA
USA
E-mail: esthervanderknaap@uga.edu

Sofia VISA
College of Wooster
Mathematical and Computational Sciences
Department
Wooster OH
USA
E-mail: svisa@wooster.edu

Sixth International Conference on Applied Informatics
Imagination, Creativity, Design, Development
ICDD 2022, May 5-7
Sibiu, Romania

Using GANs to Innovate Creative Industries: Fashion and Textile Design

Felix Husac

Abstract

The fashion industry is a business worth over 180 billion dollars every year. Due to the Covid pandemic lockdowns, many people shifted focus from physical jobs to online related industries. New opportunities arose from online marketing, dropshipping, and work-from-home consultancy related businesses. People started creating wearable artworks in the form of print-on-demand textile products from providers like Teespring, or selling hand crafted art on platforms like Etsy, Facebook Marketplace etc. But imagining and developing new fashion and artwork designs is a laborious, time consuming process. This paper aims to study the possibility and efficiency of integrating Artificial Intelligence in the process of creating new textile designs. The proposed application gathers large amounts of pictures from online shopping platforms via a web scraper and gives the end user the opportunity to sort through and choose the pictures they'd like to use as a base for their new creations. The selected pictures will then be used to train a GAN, expecting similar designs to be created effortlessly and automatically.

1 Introduction

Fashion is an integral part of everyday life, even if we acknowledge it or not. Fashion doesn't mean only haute-couture, designer brands, but every piece of clothing, fabric or textile material used to create clothes and accessories for wearing or decoration purposes.

The fashion industry is projected to reach revenue levels of over 100 billion USD in 2022, with companies that are implementing AI and machine learning technologies to their product development and sales chain bringing in most of the profits. [1-2]

The old-fashioned way of designing and running a clothes brand has remained mainly unchanged since the first Industrial Revolution. The use of templates, moulds and reference designs has been the way to come up with new creations that were then fed to steam powered looms - similar to today’s automatic weaving and cutting robots.

Every tipping point ever reached in human history has been defined by the creation of a new technological tool that revolutionises a certain aspect of everyday life. The invention of the printing press, the steam engine and subsequently the Great Industrial Revolution, the invention of the computer and the advent of the Internet Age have shaped human existence over the past centuries. These changes also came at the cost of jobs lost due to being replaced by the technology that helped our world evolve. A similar tipping point is presumed to be reached in the near future with the help of Artificial Intelligence and Deep Learning algorithms. This will have a significant impact on our current ways of life, both in terms of convenience and ease of use, but also, on the flip side, the dangers of jobs being replaced by sentient algorithms or machines. [3]

Recently, the fashion industry has begun adopting technology in the way it operates. From using machine learning to make predictions about what users will buy next, to keeping track of inventory, sales, earnings and the metadata generated by online sales, to lately, and still in an early experimental stage, designing and matching clothing styles and coming up with new design ideas. [4-5]

This rise of consumer involvement has also empowered the average person to adapt and even create new fashion items for a financial incentive. Hand crafted clothing items can be sold on platforms such as Etsy or Ebay and thus provide a secondary stream of income. Graphic designs, logos and more complex patterns can also be transformed into actual clothing items with the help of on-demand printing and shipping services, such as Teespring, that give a proportion of the sales price back to the designer as a commission. Fashion combines practicality with aesthetics, is highly customisable and it can easily be developed into a work-from-home business. [6]

The innovation that allowed machine learning algorithms to create new images from scratch is the Generative Adversarial Neural Network, invented by Ian Goodfellow. At its core, a GAN involves two neural networks that compete against each other, by trying to improve each other's loss function. In essence, the Discriminator (the neural network designed to tell if the input is a real world example or fake data) fights the Generator (a neural network that creates fake data, hoping to fool the Discriminator) in something that is called a zero sum game. The aim of the game is to find a saddle point, or to find the point where the probability that the image received by the Discriminator for review has 50% chances of being real or fake. The Discriminator gives the Generator feedback on its progress by assessing the samples sent by the Generator and giving them a probability score. The Generator's aim is to fool the Discriminator into believing that the images it creates are real-world data, and the Dsicriminator's aim is to correctly distinguish between the real dataset and the generated, fake image. [7]

The ways of synthesising data up to this point have relied on traditional means, such as asking a human to manually create the necessary data, capturing an unfolding event or scientific observation, using genetic algorithms to create variations of the same input and more recently, variational autoencoders. One main drawback of a variational autoencoder is that the latent space, or the space in which the learned concepts are represented, is distribution dependent, meaning that any other statistical distribution apart from the Normal distribution will yield completely false results. Also, the size and quality of the images produced are low both in pixel count and in details captured. One of the main advantages of GANs is that they don't suffer from these problems. GANs are distribution agnostic, understanding the underlying distribution of the dataset as they learn. GANs also offer a higher image fidelity and can achieve higher resolutions, but with increased computational power demands and longer training times. [8]

Since their introduction, GANs have taken the machine learning community by storm. With lower cost processing power and better tech every year, the GAN research field is expanding rapidly, with over 360 papers published about GANs per month. [9]

Yan LeCun, chief AI scientist at Facebook, considers GANs as "the most interesting idea in the last 10 years in Machine Learning". [10]

GANs are often used for creating data where there isn't enough real world data to begin with, the quality of the gathered data is poor, or where data can't be legally acquired (due to copyright, GDPR protections etc.). [8]

The aim of this paper is to create a deep learning model capable of coming up with new textile product designs, by using the GAN architecture. The proposed application consists of three parts: a web scraper, an evolving dataset and the GAN network. The web scraper will be used to download the images that will then make up the dataset. To create and maintain the dataset, they will need to be rescaled, filtered and chosen on a more or less subjective basis by the end user. The main component of the application is the GAN network that will create new images based on what it learned from the dataset.

The rest of the paper will describe several state-of-the-art examples of using GANs for creating fashion items on demand in the first chapter, and a proposed method for reaching similar results in the second chapter. The third chapter will deal with the preliminary results obtained by using the method described in the previous chapter.

The conclusion and future development opportunities deal with the current state of this research and the possibilities of expanding it to multiple areas of not only the textile market, but the automotive upholstery industry for customised car interiors, the gaming industry and much more.

2 Brief overview of state-of-the-art approaches

Machine Learning algorithms are used in many aspects of the textile industry. Many prolific brands such as Dior, Macy's, Nike and Nordstrom for example have been using AI to facilitate a better overview of their resources, such as inventory and marketing strategies. These technologies that use metadata collected by large social media corporations are put in place in order to create a better customer experience and individualised customer care. [4]

Some other fashion brands are using AI to determine their choice of clothes and style in advance, and tailor their experience to the clients needs. [11]

By integrating machine learning in their business model, fashion brands can now earn even more money. "As per the data from McKinsey, 20% of the global fashion brands using AI are generating 144% of the industry profits." [2]

Lately, with the appearance of GANs in 2014 and the interest shown in this research field since 2018, many large companies have decided to invest in research and development using the GAN approach. Most notably, Amazon has created a GAN model capable of creating high resolution images of clothing items based on text provided by the end user. The algorithm used, called ReStGAN, creates the images via a series of steps. The user's text input is converted directly into a low resolution image via a preliminary GAN network, and then it is upscaled to a higher quality by a secondary GAN. This approach is based on an existing system, called the StackGAN. [12]

Another interesting approach, chosen by Linlin Liua et al. explores the possibility of matching clothing items based on style. The end result is a series of images containing two or more pieces of clothing that the AI thinks would match harmoniously. The complicated task of teaching a neural network concepts regarding fashion styles, types of textile items, colour and shape combinations was achieved by using two discriminator networks and manually labelled data. Their approach is called Attribute-GAN because it uses annotated data for training the networks. [5]

One of the most interesting approaches comes from a Berlin startup that specialises in fashion and technology, called Zalando. The Zalando team have created a GAN model based on nVidia's StyleGAN architecture that can copy and paste virtual fashion models and poses for displaying different clothing items. The style transfer network was trained on four nVidia V100 GPUs and it took four weeks to train. The results are comparable to high end catalogue photography, the GAN model being able to create realistic images with correct poses and various fashion items. They decided to take two approaches, using both a conditional and an unconditional GAN architecture, but the unconditional GAN yielded better results. [13]

The innovations in the field of fashion and textile design are endless, and there are many approaches that can be used to create realistic designs. The main shortcoming of such techniques is that in order to create a high resolution, high quality and realistic image, the user needs a large dataset of curated images as a base for training, and expensive hardware to run the computations needed. For a simple home user without the necessary resources, these approaches can seem daunting. The end user lacks the financial and physical resources to create their own architecture or experiment with their own dataset, due to the prohibitive cost of renting servers, buying graphics cards or even gathering and organising pictures in a dataset.

3 Contributions and proposed approach

3.1 General design

The purpose of the proposed approach is to empower home users to create and sell their own textile creations and design ideas in an easier and ideally more cost effective way. To achieve this goal, the application will be constructed out of three main components, namely the web scraper, the dataset and the GAN. The dataset, consisting of images necessary for running the application, will be gathered from Amazon.com's clothing section of the website. It will consist of pictures of fixed size and bit depth. The GAN architecture will be chosen after trying out multiple variations of hyperparameter values and loss functions for multiple models.

3.2 The web scraper

In the search for the best method of scraping a large and dynamic shopping platform such as amazon.com, multiple avenues were tried and tested. An approach was made by using the Python library named BeautifulSoup. The main drawback of this method is its lack of flexibility. BeautifulSoup is easy to use, but it is most effective on static website pages, as it mostly relies on the HTML tree to find the requested content. [14]

An alternative to BeautifulSoup is Selenium, an automation framework, mainly used for website testing and cross-browser compatibility checking among other things. Selenium's main weak point is that it's cumbersome to adapt to a data mining purpose, and the browser used for running the automated data gathering needs to be granted special permissions. [15]

The approach used in this paper is via the Scrappy Python library. In conjunction with a Google Chrome CSS selector plugin, this approach is the fastest and easiest to implement, run and adapt. The CSS Selector plugin, named SelectorGadget, offers a visual aid when choosing a CSS style element by highlighting it in accordance with the classes specified in its HTML tag definition. It can easily discard items that share a common class by clicking on them until the selection box disappears and it can just as easily add new items to the list. [16] It can then export the class names as a text string that can be copied into the web scraper code that will look for similar styled elements on the selected page.

Scrapy is a powerful yet simple tool for creating automated web scrapers. It allows the user to scrape multiple web pages of the same domain recursively, it can enter subdomains and it can navigate page numbers. It also implements a robots.txt compliant behaviour that protects the end user from lawsuits for illegal scraping if the website prohibits the use of such practices. The Scrapy user agent warns the webserver that it is not a regular user, and the server can decide the request rate and bandwidth allocated to the scraper, thus minimising the impact on performance for regular end users and providing a relatively fast request rate. [17] The web scraper features a data collection script that searches for the CSS selectors previously found with the Chrome Browser plugin and saves the data inside the HTML tags into a python list that can then be printed to the console or saved in a .csv file. The data fields created for this purpose are the product name, the product price and the image link of the product.

3.3 Creating the dataset - challenges

In order to train a GAN successfully, a user needs a dataset with over 50.000 images. [18] Gathering, filtering, cleaning and exporting such a dataset is a daunting task.

The image link .csv file resulting from running the step described above will then be fed into another Python script that will download the pictures into a designated folder on the user's computer. The images will be subject to further analysis, filtering and dataset cleaning, after which they will be standardised to a set image resolution and colour depth.

There are two ways of creating a Google Colab compatible image dataset that can be used to train and assess the performance of neural networks. The user can create another .csv file with all the attributes necessary and manually update it or use a python script to fill in the document fields, or they can upload the images to Kaggle and create the dataset in an automated way.

Kaggle allows users to create their own datasets, but a user can only publish his dataset to the whole platform after he has verified his credentials and proved his intent. For this paper, a private dataset was created. The simple drag-and drop interface is easy to use and the only bottleneck is the user's internet upload speed. After Kaggle is done processing the images, the user can then go and add the metadata necessary to the newly created dataset. In order to then use it, a user token key needs to be generated. The user also needs to follow some simple setup steps to download and mount the dataset into the Google Colab drive space. [19]

3.4 Choosing a GAN architecture

The current paper discusses the pros and cons of using the Simple GAN, as first presented in the 2014 seminal work by Ian Goodfellow [7], the DC-GAN and the Wasserstein GAN variants.

The reasoning behind choosing different GAN architectures is to evaluate the performance and results created by each network, to tweak the hyperparameters in an effective manner and to have a comparison baseline for determining the best approach. The Simple GAN variant was used as a "control" run, to which the other two architectures were compared against.

The dataset used to evaluate the models in a preliminary run is the standard Fashion-MNIST dataset. It contains over 70.000 images, depicting multiple clothing item categories. The images have a 28x28 pixel resolution and are in the grayscale format. [20] The reasoning behind choosing this dataset as the benchmark for evaluating the models is that the lower resolution and bit depth allows for faster processing time, and it has been tried and tested by multiple researchers. The techniques that will yield the best results on this dataset will be adjusted to work on the proprietary dataset, with images of 24-bit RGB bit depth and a higher resolution.

The differences between the Simple GAN and the DC-GAN are the way they construct and upscale the images. The DC-GAN adds kernel operations to upscale a lower resolution base image, and it also adds batch normalisation between hidden layers so as to optimise the training process. The Wasserstein GAN model improves on the architecture by providing a loss function that is tightly coupled to the quality of the output images. [21]

3.5 Limitations

There were multiple choices in regards to the training environment. A server renting service, advertising an nVidia V100 GPU solution with 32GB of RAM costs over 1000 USD per month to run as their basic service, a price far too high to be justified for this paper. [22] Another choice could have been Google Colab Pro or Pro+. but these services are available only to a handful of countries worldwide, and the features they offer over the free version aren’t compelling enough to make the jump to a paid subscription. [23]

The chosen approach was to use Google Colab Free. It is a powerful solution, but it comes with several severe limitations that make it difficult to use sometimes. One of the main strengths of this platform is the ability to request a GPU to run demanding calculations, but it can come off as a double edged sword: on the one hand, the free GPU increases the performance considerably and shortens the execution time, but on the other hand, Google automatically downgrades the service and can even terminate the session if the client is using up too many resources or is sitting idle.

Google Colab Free sessions are also time limited. A session usually lasts up to 12 hours of continuous runtime, but clients can easily get disconnected even after 90 minutes of running their task. Google is also checking for user activity in the browser window, such as to minimise resource wasting, and can also disconnect a session if no user input or activity is detected inside a certain timespan. [24]

4 Preliminary results

4.1 Speed and performance

All models were run for 20.000 epochs on the same configuration (12 GB RAM, nVidia Tesla K80). The average amount of time for a training run is around 5 hours. The fastest GAN training time was observed with the Simple GAN architecture, while DC-GAN and Wasserstein-GAN had very similar training times. A CPU only training run was also tried for the Simple GAN architecture, with a significant increase in training time.

4.2 Visual results

The visual results are promising, with potential for improvement via hyperparameter tuning and longer training times (more training epochs). A superficial ranking, based on the visual results created after multiple runs of the same models with slightly tweaked hyperparameters would place the Simple GAN architecture at the top of the list, with DC-GAN on the second place. The images produced by the Wasserstein-GAN were the least satisfactory of all architectures. Despite the many praises to the efficiency, stability and improved output quality of the Wasserstein-GAN model, more work is needed to streamline and stabilise the current Wasserstein-GAN approach for this particular use case.

Figure 1 shows the ideal expected output (several image examples from the Fashion-MNIST dataset).

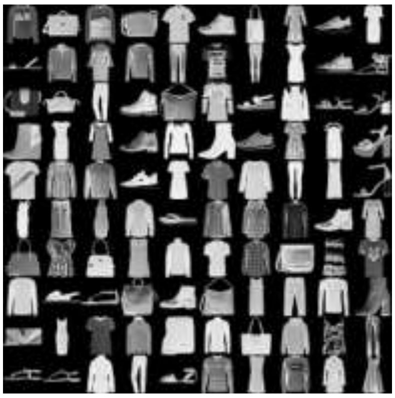


Fig. 1: Grayscale images of different element classes present in the Fashion-MNIST dataset

The results gathered from each method used are summed up in figure 2:

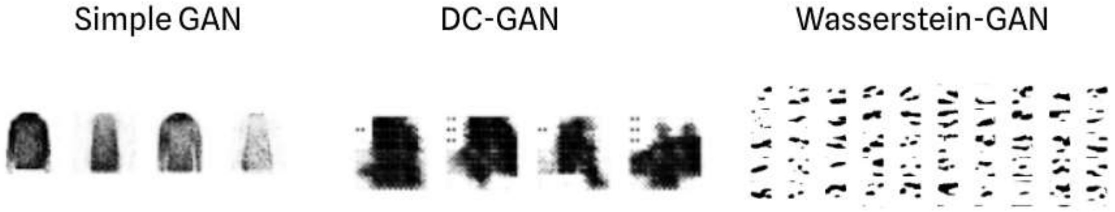


Fig. 2: Inverted Grayscale 28x28 pixel images created by the different GAN architectures, labelled above.

4.3 Problems and planned improvements

A common problem that arises from the underlying antagonic mechanism of GANs is mode collapse. [8] Mode collapse takes place when the Generator learns that by creating the same picture over and over again, usually extremely similar to a real example, it can trick the Discriminator into classifying it as real. Mode collapse limits the capability of a GAN to produce varied samples, because it will never represent all the image classes and subclasses present in the dataset. It is limited to repeating the same example or a handful of examples over and over again. Mode collapse becomes more difficult to detect the larger and more complex the dataset is (the more different objects/features are present, there is no certainty that they will all be represented in an output image). There is yet no mathematical way of proving or discovering mode collapse. [8]

Figure 3 visually represents a mode collapse example for the Simple GAN approach:

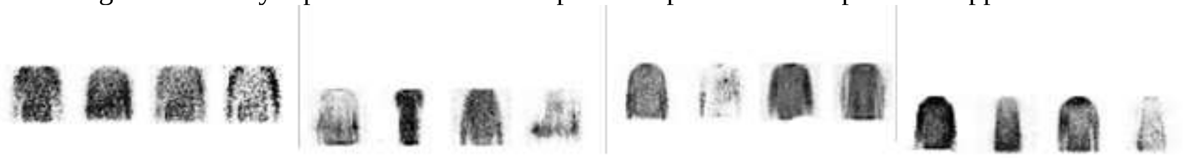


Fig. 3 - These samples were collected over a stretch of 20.000 epochs.
As can be seen, they represent the same clothing items with a small variation in the
6th generated example (from left to right). Three of the clothes shapes repeat themselves across\
all the above examples, demonstrating a case of mode collapse.

Another problem is the short training time. The middle column of figure 2 presents the results of running a DC-GAN for 20.000 epochs. The second to last image from that column looks familiar, and it appears to be representing a long sleeved jacket or half a hoodie. More training time is needed in order to find a balance between underfitting a network and overfitting it, such as to optimise its performance. The images represented here were extracted from the last running epoch, so a longer training time is visibly necessary.

Another planned improvement is to test several hyperparameter combinations. It was observed that for this use case, the most sensitive hyperparameters are the batch size and the number of epochs, and also the activation functions and the kernel size (for the DC-GAN).

5 Conclusion and future developments

This paper presents a simple approach for creating on-demand clothing and textile designs effortlessly and cost-effective. It employs the use of several components, ranging from data gathering, cleaning and standardising the data and lastly implementing the appropriate GAN architectures, all using freely available online tools and working environments.

The results are mainly satisfactory, with the exception of the images produced by the Wasserstein-GAN model. Further testing is needed, and other architectures might be pursued if the results stagnate or the improvement rate is not satisfactory.

A worthy improvement that will significantly increase the speed and hopefully also the performance of the GAN models is finding a new cloud computing platform for running the training sequences on. Free to use platforms are currently ridden with issues and limited support, and many paid options are far too expensive for the benefits being offered.

More work is needed in finding an optimal hyperparameter combination for each model. Hyperparameter tuning is a long standing problem for training and evaluating neural networks, and is not limited to only GANs. It is a time-consuming process with a trial and error type of approach.

The custom dataset will be expanded to contain a larger number of images, spanning over multiple clothing categories. The images inside will have to be rescaled to one or more standard image sizes, and also the image bit depth will need to be altered, such as to create both grayscale and RGB variants of multiple resolution datasets. Such a daunting task is necessary in order to find the best working combination between a well tuned GAN model and a well optimised dataset.

The resulting images, created by the trained GAN models, should represent a starting point for new designs or a drag-and-drop solution for print-on-demand businesses. This sort of approach can also be used in designing car interiors, upholstery design, hotel and spa amenities such as towels and bedsheets and even texture brushes for image editing software. This technique can also be applied for creating and pairing different clothing items and styles for video game NPCs, thus creating a more authentic and immersive experience. Ultimately, this approach aims to bring the end user the possibility of a higher degree of object personalisation, either for branding purposes, such as logo designs, or for home and fashion use and even business.

Acknowledgement: This work was supervised by Professor PhD *Dana Simian*, from *Lucian Blaga University of Sibiu*.

References

[1] *Luxury Fashion,* Statista, accessed on 13.05.2022 https://www.statista.com/outlook/cmo/luxury-goods/luxury-fashion/worldwide

[2] *Impact of AI in Fashion | Benefits of Implementing AI,* Govisetec, accessed on 13.05.2022 https://govisetech.com/ai-in-fashion/

[3] Chris Kolmar, *23+ Artificial Intelligence And Job Loss Statistics,* Zippia, accessed on 18.05.2022 https://www.zippia.com/advice/ai-job-loss-statistics/

[4] Khushboo Malhotra, *AI in the fashion industry,* Prescouter, accessed on 13.05.2022 https://www.prescouter.com/2020/06/ai-in-the-fashion-industry/

[5] Linlin Liu et al., *Toward AI fashion design: An Attribute-GAN model for clothing match, Neurocomputing,* 1-2, 2019

[6] Lynn Jones, *Teespring Business Model: All You Need To Know,* CMSmart, accessed on 13.05.2022 https://cmsmart.net/community/teespring-business-model-all-you-need-to-know

[7] Ian J. Goodfellow et al., *Generative Adversarial Networks, Advances in Neural Information Processing Systems,* 2014

[8] Jajub Langr, Vladimir Bok, *GANs in Action : Deep learning with Generative Adversarial Networks*, Manning Publishing , 2019

[9] Avinash Hindupur, *The GAN Zoo*, DeepHunt, accessed on 13.05.2022 https://deephunt.in/the-gan-zoo-79597dc8c347

[10] Jamie Beckett, *What's a Generative Adversarial Network? Inventor Explains,* nVidia, accessed on 18.05.2022 https://blogs.nvidia.com/blog/2017/05/17/generative-adversarial-networks/

[11] Chatwalee Pingkasan, *How Ai is Changing the Fashion Industry,* IT Chronicles, accessed on 13.05.2022 https://itchronicles.com/artificial-intelligence/how-ai-is-changing-the-fashion-industry/

[12] Kyle Wiggers, *Amazon's AI generates images of clothing to match text queries,* Venturebeat, accessed on 18.05.2022 https://venturebeat.com/2020/03/02/amazons-ai-generates-images-of-clothing-to-match-text-queries

[13] *AI Creates Fashion Models With Custom Outfits and Poses*, Synced, accessed on 20.05.2022 https://syncedreview.com/2019/08/29/ai-creates-fashion-models-with-custom-outfits-and-poses/

[14] Faruque A., *Python Web Scraping Tools: Advantages And Disadvantages Simply Explained*, WScraper, accessed on 18.05.2022 https://wscraper.com/python-web-scraping-tools-advantages-and-disadvantages-simply-explained/

[15] Sri Manikanta Palakollu, *Scrappy Vs Selenium Vs Beautiful Soup for Web Scraping*, Medium, accessed on 13.05.2022 https://medium.com/analytics-vidhya/scrapy-vs-selenium-vs-beautiful-soup-for-web-scraping-24008b6c87b8

[16] Andrew Cantino, Selector Gadget, accessed on 18.05.2022 https://selectorgadget.com/

[17] *Scrapy at a glance*, Scrapy.org, accessed on 18.05.2022 https://docs.scrapy.org/en/latest/intro/overview.html

[18] Isha Salian, *NVIDIA Research Achieves AI Training Breakthrough Using Limited Datasets,* nVidia, accessed on 20.05.2022 https://blogs.nvidia.com/blog/2020/12/07/neurips-research-limited-data-gan/

[19] *Getting Started*, Kaggle, accessed on 13.05.2022 https://www.kaggle.com/getting-started/49766

[20] Zalando, *Fashion MNIST*, Kaggle, accessed on 13.05.2022 https://www.kaggle.com/datasets/zalando-research/fashionmnist

[21] Abhishek Suran, *Deep Convolutional Vs Wasserstein Generative Adversarial Network,* Towards Data Science, accessed on 18.05.2022 https://towardsdatascience.com/deep-convolutional-vs-wasserstein-generative-adversarial-network-183fbcfdce1f

[22] *Linode Compute - GPU*, accessed on 18.05.2022 https://www.linode.com/docs/products/compute/gpu/

[23] Benedikt Droste, *Google Colab Pro+: Is it worth $49.99?*, Towards Data Science, accessed on 20.05.2022 https://towardsdatascience.com/google-colab-pro-is-it-worth-49-99-c542770b8e56

[24] M. Khorasani, *Automate and Supercharge Google Colab with JavaScript,* Towards Data Science, accessed on 13.05.2022 https://towardsdatascience.com/automate-and-supercharge-google-colab-with-javascript-9f494d98489d

Felix HUSAC
Lucian Blaga University of Sibiu
Faculty of Science
Department of Mathematics and Informatics
5-7 Dr. Ratiu Str, Sibiu 550012
ROMANIA
E-mail: felix.husac@ulbsibiu.ro

Sixth International Conference on Applied Informatics
Imagination, Creativity, Design, Development
ICDD 2022, May 5-7
Sibiu, Romania

Model-driven Software Development as the Key to RESTful Web Interfaces and Digitization

Matthias J. Keckl, Patrick B. M. Müller, Peter Braun

Abstract

We propose a model-driven approach for the development of REST APIs. With this approach, higher education institutions with limited development resources are enabled to move forward with digitization through the development of web interfaces for their administrative processes and at the same time enforce the REST compliant implementation of these. We argue that the architectural style Representational State Transfer (REST) is the perfect fit for such web interfaces on the one hand and a model-driven approach on the other hand. With FIWIS, a real-life example is shown that proves our approach works. In this project, 88.4% of the backend, database layer and frontend were generated. The remaining 11.6% include specific business logic such as e-mail notifications and the integration of third-party systems. This significant reduction in development costs is what makes digitization possible in the first place. The guaranteed REST compliance through the model-driven approach and its strict and correct usage of the web protocols (HTTP, URI) ensures longevity and independent extensibility of these web interfaces.

1 Introduction

Roy Fielding proposed a new architectural style for distributed hypermedia systems, called Representational State Transfer (REST), as part of his doctoral thesis [1] in 2000. As Tilkov et al. [3, p. 9 ff.] well illustrate, Fielding, who was also involved in the standardization of the web protocols from the beginning, aims in his work to abstract the concrete architecture of the web and to extract the fundamental architectural concepts. These concepts should serve as the underlying formal architecture and should guide the further development of the web protocols. A protocol that implements this formal architecture is the well-known Hypertext Transfer Protocol (HTTP). It is used not only to serve static websites, but also to implement Web APIs that cover complex business use cases. These Web APIs would obviously benefit if they comply to the underlying architecture (REST), or said differently, one can only use an architecture (REST) optimally if one uses the concrete implementation (HTTP) as intended by the architect (Fielding) [3, p. 10]. Nevertheless, a lot of so-called REST APIs[1] violate fundamental concepts of REST like Fielding himself criticises in his famous blog post from 2008 [4]. With this violation, one loses the advantages that REST provides as an underlying architecture. These include loose coupling between different systems as well as performance and scalability of the distributed system [3, p. 2–4].

To retain the advantages and to ensure REST compliant development, we propose a model-driven approach in this paper based on Schreibmann and Braun [5]. We argue that a Model-Driven Software Development (MDSD) approach can solve the current issues of the design and implementation of REST APIs. With this approach, the source code is generated based on an abstract model [11], or more precisely, based on our meta-model developed in this paper. However, we are not only able to enforce REST compliance, but also to accelerate the digitization process at universities, which we can show with

[1] We call any HTTP-based API that comply with the REST concepts a REST API.

an example from our university[2]. As part of the digitization process, the faculty of computer science developed an information system called FIWIS, which provides a web interface that allows the employees to perform their daily tasks and processes. Due to limited resources available for the development of this web interface, a technique was required for automatically generating source code for the database, the business logic, and the graphical user interface. And it is precisely this requirement that the model-driven approach fulfills. Thereby, we designed our approach in such a way that custom code can be easily injected into the generated software artifact. Thus, the developers can focus on writing the custom code that cannot be produced by the software generator.

The structure of this paper is as follows. First, in Chapter 2, we give a brief overview of the concepts of REST. Afterwards in Chapter 3, we argue why MDSD matches REST and also why this applies the other way round. In Chapter 4, we present our practice-oriented REST meta-model. In Chapter 5, we provide an example of how to create an instance of the meta-model. In the following sixth Chapter, we will look at FIWIS and how many artifacts had to be generated and how many of them had to be implemented manually. Then, we briefly go into how the source code is generated and which frameworks are used. In Chapter 8 we look at some statistics from the FIWIS project. Before we conclude, in Chapter 9, we evaluate whether we have successfully ensured REST-compliant development and enabled the digitization with our approach. We end the paper with a conclusion and an outlook on future research topics.

2 Representational State Transfer

As Fielding notes in his thesis [1, 1, p. 75], REST was developed to provide a model for how the modern Web should work. To retain the principles behind this model, he defined five mandatory architectural constraints. And in fact, only if all constraints are fulfilled, the concrete distributed system gains all advantages of REST like scalability or the loose coupling between different systems. The World Wide Web (WWW), as the biggest concrete implementation of REST, should therefore comply with all these constraints. Adapted from Haupt et al. [13], we will now give an overview of the five constraints and their realization with the specifications and techniques of the Web, such as HTTP [6] or URI [14]. For each constraint we provide the exact passage where Fielding explains the constraint in the thesis.

Client-Server [1, p. 45, p. 78] According to this constraint, communication within network-based applications takes place between client and server. The client makes a request to the server to use its provided service, and the server responds to this request. The communication thereby is always initiated by the client. In the WWW, this constraint is guaranteed to be complied with by the use of HTTP [15].

Stateless [1, p. 78–79] *Stateless* communication requires that each request must contain all the necessary information so that the server can understand it. Thereby, the server is not allowed to use any server-side context, that is held in the memory for example. This constraint has no consideration in the techniques used by the WWW and is the responsibility of the API developers.

Cache [1, p. 79–81] This constraint specifies that the data within each response to a request must be labeled as cacheable or non-cacheable. The client is then allowed to reuse cached data for future identical requests. To define the caching behaviour, the HTTP protocol [8] provides several functionalities. A well-known example is the *Cache-Control* header. Nevertheless, the developers are still responsible for defining the caching behaviour for each response by setting the *Cache-Control* header for example.

Uniform Interface [1, p. 81–82, p. 86 ff.] The *Uniform Interface* constraint, mentioned by Fielding as the central feature that distinguishes the REST architectural style from other network-based styles, prescribes a uniform interface between all components in the distributed system. As stated by Tilkov et al. [3, p. 15], one can think of it as a Java interface, specifying a very small number of functionalities that must be supported by every component in the distributed system. To obtain such an interface, Fielding defined another four architectural constraints that need to be fulfilled.

[2] https://www.fhws.de/en/

Identification of Resources Resources are the key abstraction of information in REST. Thereby, "[a]ny information that can be named can be a resource". Each resource is identified by a unique resource identifier, which should remain the same even if the content of the resource changes. In the context of HTTP, the URI standard [14] is used for this. The developers are responsible to design their API resource-oriented, to define suitable identifier for the resources as well as to ensure that the identifiers stay the same.

Manipulation of Resources through Representations The interaction between client and server consists of sending and receiving representations of resources. If the client wants to create a resource, it sends a representation of it that represents the intended state of the resource. Thereby, the representation is made up of the representation itself as a sequence of bytes and representation metadata that describe these bytes. HTTP provides the so-called media types [16] and corresponding headers [6] (e.g. *Content-Type*) for this purpose.

Self-Descriptive Messages Richardson et al. [2, p. 348–349] give a very good analysis of this constraint. The core aspect of *self-descriptive messages* is that each message contains all the information for the recipient to understand. This includes the obligatory usage of standard methods and media types. Applied to HTTP, one have to use the HTTP methods defined in RFC 7231 [6] according to their definition. Furthermore, the API developers must use the standard media types defined in RFC 2046 [16] and specify the corresponding media type in the headers provided for this purpose [6].

Hypermedia as the Engine of Application State *Hypermedia as the engine of application state*, or HATEOAS, states that the server should guide the client through the application. Links should be included in each response to guide the client to the next possible actions. Thus, the server defines the possible next actions and the client chooses the action it wants to perform. Using the Hypertext Markup Language (HTML), this approach is natural for developers, but when it comes to API development, this constraint is usually ignored. This leads to the situation Fielding criticises in his blog post [4], which is that APIs are called RESTful even though they completely ignore hypermedia. This is also the reason why hypermedia is one of the biggest challenges in the development of REST APIs. Accordingly, developers are responsible for defining the links and sending them to the client with each response.

Layered System [1, p. 82-84] Richardson et al. [2, p. 353] summarize the basic idea of this constraint well. "The layered system constraint is less about proxies and gateways, and more about the fact that adding one between client and server is nearly a transparent operation. A client doesn't know whether it's talking directly to a server, or whether it's talking to a proxy that talks to a proxy that talks to a gateway". Just like the Client-Server constraint, the Layered System one, is guaranteed to be complied with because of the HTTP usage [15].

3 REST and Model-Driven Software Development

We are pursuing two objectives. On the one hand, we would like to provide a web interface based on the REST architectural style and, on the other hand, this interface should be developed with a model-driven approach. The web interface should conform to REST, as this, designed for universities, should be long-living and easily extendible, and REST offers exactly that. Or as Fielding puts it "REST is software design on the scale of decades: every detail is intended to promote software longevity and independent evolution" [4]. Several approaches already exist to implement REST APIs with a model-driven approach. For example, Haupt et al. [13] propose a multi-layer model with which they can enforce the REST compliant development. They argue that by having multiple models, they address all required aspects of REST. We follow the argumentation, that Model-Driven Software Development (MDSD) can counteract the currently existing issues of the REST API development. But we also argue that this is equally true in the other direction. REST is the perfect architecture for generating web applications with a model-driven approach. To summarize in one sentence, MDSD can address the issues of REST API development and

REST is the perfect underlying architecture for a web interface to be generated and thus, the two form the perfect match. In the following two sections, we present our argumentation why these two concepts fit so well together.

3.1 MDSD matches REST

The goals of MDSD are among others manageability of complexity through abstraction and increased development speed [11, p. 13–14]. The first goal addresses two of the problems we experience in the development of REST APIs. Many developers may be unfamiliar to think in a resource-oriented instead of a Remote Procedure Call (RPC) style. Through the higher level of abstraction of MDSD, the developers can be guided and partly forced to think and design resource-oriented much better compared to the level of abstraction of a concrete programming language. The same applies to the second problem we observe, namely that it is very difficult for the average developer to understand Fielding's PhD thesis [2, p. 343]. The developers will not require the same level of deep understanding of the REST concepts, if they work on a higher level of abstraction. The underlying meta-model shows them which concept have to be considered during the design and implementation of a REST API.

Due to the lack of tools for implementing a RESTful interface, its process can be erroneous but is in any case very time-consuming [5]. Thus, the MDSD approach compensates the missing tools making the design and implementation of REST APIs significantly more efficient.

3.2 REST matches MDSD

Considering now the other direction, we have to observe the following situation. An API is to be generated with a model-driven approach. There are several techniques that could be used as the basis for this, including REST, RPC or Simple Object Access Protocol (SOAP). We now argue that REST offers key advantages and thereby, we see three major advantages, which we explain in the following.

1. **REST naturally restricts the domain.** An MDSD approach cannot be used to generate arbitrary software, but to generate software for a specific domain. In our case, the domain is the generation of Web APIs. Although this domain is already clearly restricted, we would benefit from further restrictions, as the smaller the domain, the easier it is to implement the software generator. We now argue that REST and and its strict architectural constraints can restrict the domain even further and is thus the perfect technique for the domain of Web APIs. As we will see in the next section, we apply additional restrictions to REST to minimize the domain even further. With this subset of potential Web APIs and in fact of potential REST APIs, it is now possible to develop a feasible approach to a generator.

2. **Requests are very similar.** Due to the *Uniform Interface*, the interaction with resources is limited to a very small number of possible operations. Referring to the HTTP protocol in concrete terms, this means one can just use the four HTTP CRUD (Create, Read, Update, Delete) methods GET, POST, PUT and DELETE. These methods have a fixed semantic [6] and therefore their processing on the server-side is usually very similar. One can even identify a fixed sequence of processing steps for each of the four methods. This fixed sequence allows to generate large parts of the code for the processing of the requests. This includes, for example, pagination or the handling of conditional requests [9].

3. **Frontend can be generated.** As just mentioned, the *Uniform Interface* ensures a very limited and clearly defined interaction with resources. As a result, not only the server-side processing but also the client-side processing is usually very similar. For example, many web interfaces consist of list views for displaying multiple resources that were requested from the server. If additional constraints are now applied, a frontend can be generated based on the same meta-model used to generate the backend. Both, the additional restrictions as well as the meta-model used for the generator are presented in the next section.

4 A practical Meta-Model for REST

In this section, we present our meta-model seen in Figure 1, serving as abstract syntax for our MDSD approach. The meta-model represents our formal understanding of REST and its architectural constraints. But it is also practical-oriented and heavily influenced by our experiences of designing and implementing REST APIs. Therefore, it also includes a role-based authentication mechanism as well as a database configuration to generate the database layer based on MySQL[3]. Our practical experience is also the basis for the additional constraints we apply to REST. These restrictions mean a trade-off. We can move more to towards enforcing REST-compliant development but we lose the ability to model any arbitrary REST API. We have also considered the frontend in the modelling process which is why our meta-model differs from others (Haupt et al. [13], Schreier [12]). The frontend part of our meta model is based on Müller et al. [20]. Furthermore, a role-based authentication system and the generation of the database layer are taken into account.

A well-designed web application should behave like a finite-state machine [1, p. 109]. Therefore, we identify the states of an application as well as its transitions as two key concepts of the REST API design. The transitions are thereby represented as hyperlinks included in every response sent from the server to the client. As third key concept we identify resources and its representations. Thus, these three concepts are the central elements of our meta-model. The application states are represented by the *AbstractState* class and its subclasses. Each state has an arbitrary number of transitions, where each transition is either a *ContentTransition* or an *ActionTransition*. Each resource has a finite number of attributes, which themselves also represent different attribute types. Besides the attributes, it is required to define the caching behaviour and at least one *ResourceView*. A resource view is among others defined by a subset of the attributes of the corresponding resource, a media type and multiple classes which specify the appearance of the view on the client-side. In combination with the transitions and its relation types, the resource views form the linkage of the server and the client. And in fact, in a REST-compliant application, the client only needs to and is only allowed to know the media and the relation types [4]. In our meta-model, resource views and media types are considered to be equivalent since each resource view has an unique media type and therefore there is a 1:1 relationship.

4.1 Further Restrictions

We now add further constraints to our meta-model to better implement our MDSD approach which, in turn, eases the development of the software generator. Since we are not able to show all additional applied constraints, we present the two that influence the structure of our meta-model the most. One is the correct usage of HTTP verbs, which we enforce very strictly so that there is no possibility for the developers to use them incorrectly. Furthermore, we limit the nesting of URIs to two levels.

4.1.1 Usage of HTTP Verbs

The interaction with resources when using a REST API is restricted to the four HTTP CRUD verbs GET, POST, DELETE, PUT. These verbs have a fixed semantic defined in RFC 7231 [6]. For instance, one defined function of POST is to create a new resource on the server side. We enforce strictly the usage of the HTTP verbs as defined in the corresponding RFC as well as a strong coupling between the semantic of the HTTP verbs and their use in the business logic.

4.1.2 Limitation of Resource Levels

We limit the maximum number of nested URIs to two. We argue that any deeper nested URI can be represented by a URI with a maximum of two levels. Thus, despite this limitation, one can represent any association between resources. We illustrate this circumstance with the following example of a social network. We request all posts of the user via *GET /users/5/posts*. Within this collection resource, a link to each individual post resource is provided. Each of these links points to the corresponding post resource at the primary level. For example, a link in the collection resource to a single post would look

[3] https://www.mysql.com/

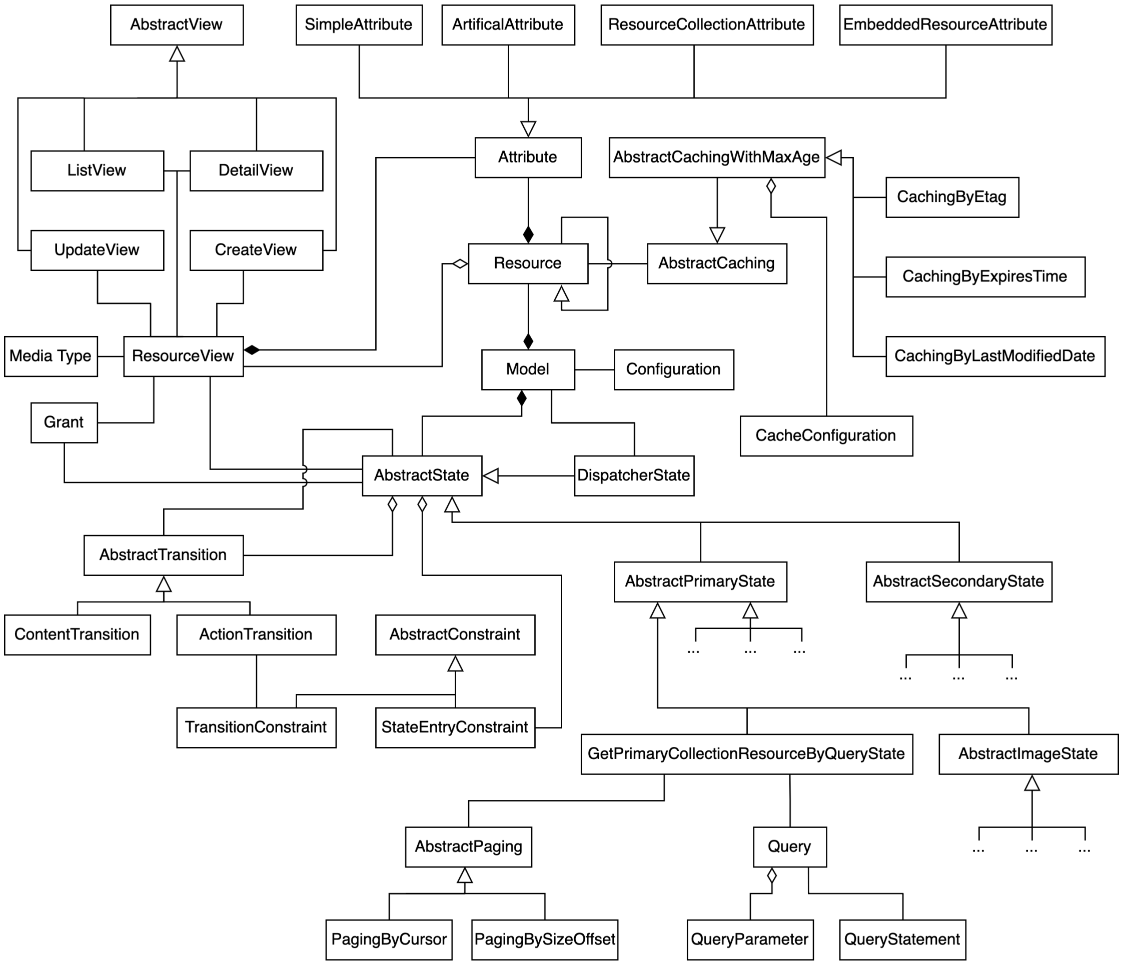


Figure 1: Simplified illustration of our meta-model. (Cardinalities are omitted here for reasons of clarity.)

like this */posts/11*. Now, if we want to request all comments on this post, we can do so with *GET /posts/11/comments*. Because we were directed to the primary level in the previous step, we are now on the secondary level and not the tertiary level. With this mechanism, which is used throughout the application, we can represent any association between resources with only two levels.

4.2 Resources and Resource Views

Since resources are the core of RESTful interfaces, the developers should put a lot of effort into the design of the resources. To enable them to design well-designed resources, we provide a wide range of different attribute types. A simple attribute can, for instance, be a string, integer or one of the new data types we have introduced, with which location and image attributes can be modelled. With the *ResourceCollectionAttribute*, resources can be linked to each other. With the *EmbeddedResourceAttribute*, on the other hand, the other resource is not linked but is contained directly in the representation. To comply with the cache constraint, a caching strategy must be defined. The meta-model offers therefore different caching strategies, including caching by *ETags*, and considers the setting of an arbitrary number of *Cache-Control* directives.

Resource views fulfill two responsibilities. First, they allow to define different views on the same resource, so different views can have different subsets of attributes. Thereby, each resource view may be restricted to specific user roles, which is represented in the meta-model by the connection of the *ResourceView* and the *Grant* element. The second responsibility is that the views define how the view is displayed in the frontend. For each view, a *List*, *Detail*, *Update* and *Create* view must be specified. As can be seen from the names, these elements are used on the client-side to interact with associated resource using the CRUD methods.

4.3 States

The states are the second core element of our meta-model. A state consists of any number of *Transitions* and *StateEntryConstraints*, one *ResourceView* and one *Grant* element. The state entry constraints can be used to forbid the access to a specific application state based on the associated resource state and the current user. Just like with the resource views, the grant element is used to restrict the access to an application state to a specific group of user roles.

There is a strong coupling between the states and the four HTTP CRUD methods. Each state represents the processing of one of the four methods. Thus, our meta-model provides a model for each HTTP CRUD method respectively for the primary and the secondary level. Besides these states, we also provide states to interact with binary data such as images. From the client's perspective, a state is defined by its corresponding resource and its HTTP method and is accessible via a URI.

A special state type is the *DispatcherState*. This state acts as the entry point to the RESTful application and its URI is the only one the client must know. It does not provide any representation but a list of possible transitions to start to interact with the application.

4.4 Transitions

These states are now being linked with transitions. Thereby, one transitions always connects exactly two states. We distinguish between two types of transitions, namely *ContentTransition* and *ActionTransition*. A content transition connects resources with each other and the corresponding link is located in the representation of a resource. An exemplary use case is a collection resource. In this case, a content transition can be used to include a link to each single element of the collection in the representation.

In contrast, an action transition is located in a *Link* header [7]. Additionally, they can be constrained by applying an *TransitionConstraint* to the transition. These are used like the *StateEntryConstraints* and the corresponding transition is only added to the response if the constraint evaluates to true. As a basis for this, the current resource state as well as the current user can be used again.

4.5 Configuration

The meta-model contains a few more configurations that are later needed by the generator to automatically create the database layer and the authentication strategy. This includes the database connection credentials, the dispatcher state URI for the client and the details about the desired authentication strategy. Thereby, our meta-model provides an HTTP basic authentication [17] scheme as a default implementation. If activated, JSON Web Tokens (JWT) [18] are generated automatically.

5 Creating an Instance of the Meta-Model

To create an instance of the just presented meta-model, we provide a Java DSL. This DSL guides the developers through the design process of a REST API and ensures that all aspects necessary for REST-compliant development are taken into account. In the following, we want to present an exemplary usage of the DSL. Therefore, we want to model the *User* resource with the attributes *id*, *name* and *password* and three states to create, to get a single and to get a collection of users. Finally, we want to model the transitions between these three states.

```
.resources()
.defineResourceWithName( "User" )
    .withSimpleAttribute( "name" ).asString()
    .withSimpleAttribute( "password" ).asString()
    .allAttributesDefined()
    .andCachingByValidatingEtags().usingConfiguration( CacheConfiguration.PRIVATE )
    .defineViewWithName( "UserView" ).hideAttributes( "password" ).done()
    .defineViewWithName( "SimpleUserView" ).hideId().done()


.states()
.primaryPost()
    .named( "CreateUserState" )
    .withResource( "User" ).withResourceView( "SimpleUserView" ).done()
.primaryGetSingle()
    .named( "GetSingleUserState" )
    .withResource( "User" ).withResourceView( "UserView" ).done()
.primaryGetCollection()
    .named( "GetAllUsersState" )
    .withResource( "User" ).withResourceView( "UserView" )
    .withQuery().name( "QueryByName" ).path( "queries" ).orderBy( "name" ).ascending()
    .queryTemplate( "{model.name} like {query.q}" )
    .queryParameters().named( "q" ).withDefaultValue("%").asString().doneWithQuery()
    .offsetSize()
        .offsetParameterIsNamed( "offset" ).withDefaultValue( 0 )
        .sizeParameterIsNamed( "size" ).withDefaultValue( 10 )


.transitions()
    .fromState( "GetSingleUserState" ).asHeaderLink().toState("GetAllUsers")
        .usingRelationType("getAllUsers").noConstraint()
    .fromState( "GetAllUsersState" ).asContentLink().toState( "GetSingleUser" )
    .fromState( "GetAllUsersState" ).asHeaderLink().toState( "SimpleUserView" )
        .usingRelationType( "createUser" ).noConstraint()
```

We define two views in addition to the actual resource, which are used to create and read the resource. The collection resource is paginated by an offset-size pagination strategy and filtered by the user's name for which we defined a query template that is later used for accessing the data in the database. In this example, we can also see the different usage of the types of transitions. As we can observe, a content link does not require a relation type because the attribute name is automatically used as the relation type.

6 FIWIS and its Extensions

As the example we have just looked at is very minimalistic and does not represent a realistic use case, we would like to present some insights of the FIWIS project, hat we developed at the faculty of computer science at our university to digitize the administrative processes. For this project **564** states, **353** resources and resource views, **1089** database query statements, **907** transitions and **528** constraints are defined and generated based on our meta-model and software generator. Even though large parts of the business logic can be generated, some logic must be implemented manually via so-called extensions. In FIWIS, **179** states, **13** resources and resource views, **244** database query statement must be added manually. However, the states are not developed completely manually as they only extend the already generated ones. For example, a state for updating a resource can be extended to send an email notification when certain changes are made to the resource. The extended states are injected into the software artifact to replace the original states.

7 How the Generator works

After creating an instance of the meta-model with the DSL, our software generator uses this instance to generate the artifacts. These artifacts are based on Java 8 and Jersey[4] as REST framework in the backend and on Lit Element[5] as web component framework in the frontend. For the generation process, we utilize FreeMarker[6] templates. Below we show exemplarily the simplified template for the generation of the resource classes. Thereby, the list of defined attributes is iterated through and the corresponding getters and setters are generated. As it can be seen, the different types of attributes are distinguished. While for a *SimpleAttribute* only the basic getters and setters are generated, for a *ResourceCollectionAttribute* a link is injected into the resource via the *@InjectLink* annotation.

```
public class ${modelClassName} {

<#list attributes as attribute>
    <#switch attribute.type>
        <#case "simple">

            private ${attribute.javaDataType} ${attribute.name};

            public ${attribute.javaDataType} get${attribute.name?cap_first}() {
                return this.${attribute.name};
            }

            public void set${attribute.name?cap_first}
                ( ${attribute.javaDataType} ${attribute.name} ) {
                    this.${attribute.name} = ${attribute.name};
            }
        <#break >
        <#case "resourceCollection">

            @InjectLink(
                style = InjectLink.Style.ABSOLUTE,
                type = "${attribute.mediaType}",
                rel = "${attribute.rel}",
                value = "${attribute.value}"
            )
            private Link ${attribute.name};

            public Link get${attribute.name?cap_first}() {
                return this.${attribute.name};
            }

            public void set${attribute.name?cap_first}( Link ${attribute.name} ) {
                this.${attribute.name} = ${attribute.name};
            }
        <#break >
    </#switch>
</#list>

}
```

[4] https://eclipse-ee4j.github.io/jersey/
[5] https://lit.dev/
[6] https://freemarker.apache.org/

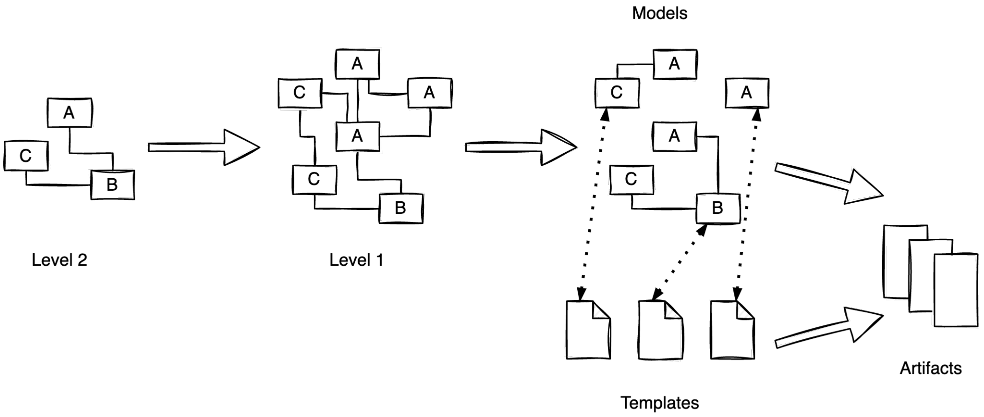


Figure 2: The generation process shown schematically.

Figure 2 shows the entire flow of the generation process. The developers use our DSL to create an instance of our meta-model, which is shown in Figure 1. Such an instance can be seen in Figure 2 at Level 2. In a first step, this Level 2 instance is transformed into a Level 1 instance by a model-to-model transformation. Thereby, as can be seen, one element in Level 2 can result in several elements in Level 1. This is due to the circumstance that some information of the meta-model is needed in multiple parts of Level 1. The most obvious example are the attributes of the defined resources. These are required not only for the generation of the resources, but also for the generation of the database layer as well as the frontend. This behavior is modeled in the figure by the *A* element. In the next step, the elements of level 1 are transformed into the models. Models in this context are the information required for the respective FreeMarker template. For each template there is exactly one model and so there is a *one-to-one* relationship between the models and the templates. In the last step, the models and templates are merged to create the artifacts. The set of artifacts then builds the generated application.

8 Some Statistics of the FIWIS Project

Having explained how the source code is generated, we are now looking at the amount of source code generated in the FIWIS project and how much had to be implemented manually. As a parameter for the measurement, we use Lines of Code (LOC). In Table 1, we can see that there are a total of **220269** lines of Java code in the entire project. 175395 lines or **79.6%** of these lines are generated and only **20.4%** were written manually. In the case of HTML, JavaScript and CSS, even **100%** of the lines were generated. In total, the project consists of **387889** lines of source code of which **88.4%** were generated. However, these 88.4% do not take into account the SQL queries that have to be written manually but are already specified when the meta-model instance is created and are not implemented via extensions and are therefore not included in these statistics.

The parts of the business logic that are common to every generated application were moved to a separate framework, that we call *Norbury*. As mentioned in section 3.2, the processing of requests is mostly very similar. Therefore, the sequences of processing steps can be implemented within Norbury. These sequences are located in the base classes for the different states and are defined using the Template Method [10, p. 325 ff.] pattern. This approach also allows the handling of conditional requests and the creation of pagination links to be implemented in the framework. Other functionalities implemented in Norbury are the handling of database connections, the creation of the database schema and tables and the preparation of the basic SQL queries. To give an idea of the complexity of Norbury, we also provide the LOC metric of it. Norbury consists of **8485** lines of code.

	Java	HTML & CSS & JavaScript
Total Lines of Code (LOC)	220269	167494
Generated LOC	175395	167494
Manually implemented LOC	44874	0
Ratio	79.6%	100%

Table 1: Comparison of generated and manually written LOC.

9 Evaluation

We pursued two objectives with our approach. Firstly, we wanted to develop a technique with which it is possible to generate all three layers of a web interface, thus database layer, business logic and graphical user interface. This approach was necessary to make the digitization of our faculty possible in the first place. With a ratio of 79.6% generated Java code and 88.4% generated code in total, the first objective was achieved. Only specific business logic like semantic checks, e-mail notifications or integration with third party systems must be implemented manually. The success of this approach can also be seen in the successful integration of FIWIS into the daily work routines of the faculty for over 3 years. This proves that our approach is suitable for generating real-world information systems based on the REST architectural style.

Our second objective was to enforce the REST compliant development of web interfaces. To evaluate this objective, we discuss each REST constraint and how our approach ensures the compliance with this constraint. The *Client-Server* and *Layered System* constraints are guaranteed to be complied with through the use of the HTTP protocol. The *Cache* constraint is covered through the caching mechanisms of HTTP [8] and and the mandatory usage of these mechanisms by our meta-model. A caching behaviour can be defined for each resource as one can see in Figure 1. If no behavior is specified, a default caching behavior is automatically set so that the caching constraint is guaranteed to be followed. An application is modeled with our meta-model like a finite-state machine, and therefore the client must follow the transitions to reach the subsequent states, and must send its complete information for each transition. Through this design, we comply with the *Stateless* constraint and at the same time also with the *HATEOAS* one. One must explicitly model the transitions to connect the application states and further URI templates [19] are generated automatically. Our meta-model allows the business logic and business entities to be defined exclusively with resources. It also generates the URI structure based on the defined resources, making each resource uniquely identifiable. Due to these two circumstances, the *Identification of Resources* constraint is also fulfilled. The defined resource views associated with the resources ensure the compliance with the *Manipulation Of Resources Through Representations* constraint. Our generator gives each resource view a unique JSON-based media type with which the client will interact when using the REST API. The last constraint still open is the *Self-Descriptive Messages* one. This is guaranteed to be complied with through the strict and correct use of the four HTTP CRUD methods, more precisely by using the different state types. The use of standardised JSON media types and the HTTP-compliant setting of the headers, like *Content-Type* or *Location*, also ensures compliance with the constraint.

In summary, we can guarantee REST compliant development through a correct usage of the web protocols (HTTP, URI, etc) and through the strict guidance of our meta-model and DSL.

10 Conclusion and Future Work

We proposed a model-driven approach for the REST API development. With this approach, we aim for two objectives. First, we intend to enable higher education institutions with limited resources to move forward with digitization through the development of web interfaces for their administrative processes.

Constraint	**Ensured by**
Client-Server	HTTP
Stateless	Meta-model (States)
Cache	HTTP Caching mechanism & meta-model
Identification of Resources	URI & meta-model
Manipulation through Representations	Meta-model (ResourceViews)
Self-Descriptive Messages	HTTP & meta-model (States, Headers, ...)
HATEOAS	Meta-model (Transitions)
Layered System	HTTP

Table 2: Overview of how the REST constraints are complied with.

These web interfaces, once implemented, must have a focus on longevity and independent extensibility. And it is precisely these two requirements that the REST architectural style can fulfill, as it was developed precisely for such systems [4]. Thus, the second objective is to ensure REST-compliant development. We have also argued that REST is not only a perfect fit for this type of web interface, but also for an MDSD approach. We have identified three key advantages including the fact that REST naturally restricts the domain of the software generator.

The development of the FIWIS project proved that our approach is successful and can be used in practice. In this project, 88.4% of the source code could be generated and thus the developing costs significantly reduced. However, our Java DSL shows still some problems. For large projects with a great number of resources, states and transitions the source code for defining the REST API can become very large. This can cause problems when interacting with the integrated development environment (IDE) that needs to render, parse, and index the source code. For this reason, we are currently working on the possibility of defining the API with the help of YAML[7]. This would make it possible to outsource the definition of the API to several files. It is also possible to offer autocompletion with the help of JSON Schema[8].

Some institutions are tied to a specific tech stack and therefore cannot use the Java artifacts we generate. For this reason, it would be very interesting to work towards an approach that would allow the artifacts to be generated in multiple programming languages. For this, the platform code would have to be written in the respective language for each supported programming language and new templates would also have to be rewritten.

References

[1] Roy T. Fielding. *Architectural Styles and the Design of Network-based Software Architectures. PhD thesis.* University of California, 2000

[2] Leonard Richardson, et al. *RESTful Web APIs. Services for a Changing World.* O‘Reilly, 2013

[3] Stefan Tilkov, et al. *REST und HTTP.* dpunkt.verlag, 2015

[4] Roy T. Fielding, *REST APIs must be hypertext-driven.* 2008. URL: https://roy.gbiv.com/untangled/2008/rest-apis-must-be-hypertext-driven (visited on 24/02/2022)

[5] Vitaly Schreibmann and Peter Braun. Model-driven Development of RESTful API, *WEBIST*, pp. 5–17, 2015

[6] Roy T. Fielding and Julian Reschke. Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content. *RFC Editor*, 2014. URL: https://rfc-editor.org/rfc/rfc7231.txt (visited on 20/02/2022)

[7] Mark Nottingham. Web Linking. *RFC Editor*, 2010. URL: https://rfc-editor.org/rfc/rfc5988.txt (visited on 21/02/2022)

[7] https://yaml.org/

[8] https://json-schema.org/

[8] Roy T. Fielding et al. Hypertext Transfer Protocol (HTTP/1.1): Caching. *RFC Editor*, 2014. URL: https://rfc-editor.org/rfc/rfc7234.txt (visited on 23/09/2022)

[9] Roy T. Fielding and Julian Reschke, Hypertext Transfer Protocol (HTTP/1.1): Conditional Requests. *RFC Editor*, 2014. URL: https://rfc-editor.org/rfc/rfc7232.txt (visited on 18/02/2022)

[10] Erich Gamma et al., *Design Patterns*, Addison-Wesley professional computing series, 1994

[11] Thomas Stahl et al. *Model-Driven Software Development: Technology, Engineering Task Force.*. John Wiley & Sons, Inc., 2006

[12] Silvia Schreier. Modeling restful applications. *WS-REST*. pp. 15–21, 2011

[13] Florian Haupt et al. A Model-Driven Approach for REST Compliant Services. *IEEE International Conference on Web Services*. pp. 129–136, 2014

[14] Tim Berners-Lee et al. Uniform Resource Identifier (URI): Generic Syntax. *RFC Editor*, 2005. URL: https://www.rfc-editor.org/info/rfc3986 (visited on 11/02/2022)

[15] Roy T. Fielding and Julian Reschke. Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing. *RFC Editor*, 2014. URL: https://www.rfc-editor.org/info/rfc7230 (visited on 11/02/2022)

[16] Ned Freed and Dr. Nathaniel S. Borenstein, Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types. *RFC Editor*, 1996. URL: https://www.rfc-editor.org/info/rfc2046 (visited on 11/02/2022)

[17] Julian Reschke. The 'Basic' HTTP Authentication Scheme. *RFC Editor*, 2015. URL: https://www.rfc-editor.org/info/rfc7617 (visited on 17/02/2022)

[18] Michael Jones et al. JSON Web Token (JWT). *RFC Editor*, 2015. URL: https://www.rfc-editor.org/info/rfc7519 (visited on 17/02/2022)

[19] Roy T. Fielding et al. URI Template. *RFC Editor*, 2012. URL: https://www.rfc-editor.org/info/rfc6570 (visited on 24/02/2022)

[20] Patrick B. M. Müller et al. Hypermedia: The Key to RESTful Web Applications. *WEBIST*, pp. 378–387, 2018

Matthias Keckl
University of Applied Sciences Würzburg-Schweinfurt
Faculty of Computer Science and Business Information Systems
Sanderheinrichsleitenweg 20, 97074 Würzburg
GERMANY
Email: *matthias.keckl@outlook.de*

Patrick B. M. Müller
University of Applied Sciences Würzburg-Schweinfurt
Faculty of Computer Science and Business Information Systems
Sanderheinrichsleitenweg 20, 97074 Würzburg
GERMANY
Email: *patrick.mueller@fhws.de*

Peter Braun
University of Applied Sciences Würzburg-Schweinfurt
Faculty of Computer Science and Business Information Systems
Sanderheinrichsleitenweg 20, 97074 Würzburg
GERMANY
Email: *peter.braun@fhws.de*

Sixth International Conference on Applied Informatics
Imagination, Creativity, Design, Development
ICDD 2022, May 5-7
Sibiu, Romania

Mirror's Mansion

Alexandru – Serban Moldovan

Abstract

This game is for people that are able to think out of the box! The idea behind this game is to embark the player into a decent looking, but very interesting and mysterious mansion. The goal of the game is to make your way out of this mansion alive, as you may encounter unusual characters and puzzles. These puzzles are special because you will have to think different in order to finally solve them. While doing so, you will also have to avoid some hostile characters. Mirror's Mansion is developed in Unity Engine and programmed in C#.

1 Introduction

Video games became the best and most used way of entertainment, alongside watching YouTube videos or streaming movies; but when we talk about video games, we do not only watch the gameplay, but we also live it, because we are the ones taking the actions, we are the protagonist going through either a full written story, simply following the course of actions, or sometimes we even get to choose which actions to make.

1.1 Structure of the paper

This paper is structured as follows:

1. "Introduction" presents the first part of the paper, where a short description is given, alongside the reasons for choosing this project and the main sources of inspiration.
2. "About the game" describes additional information about the goal and the story of the game.
3. "Non-Euclidian Theory" contains a short explanation on the difference netween different kinds of geometry, Euclidian vs Non-Euclidian.
4. "Game Engine" provides information about the engine on which the game was built on.
5. "Creating the game" is a more detailed section presenting the process of building the game and it's special puzzles.
6. "Conclusion" presents the current state of the game and tells us more about the future developement.
7. "References" represents information and other helpful resources used in this paper.

1.2 Motivation and inspiration

My main inspiration for this game was the fascinating idea of the non-Euclidian geometry. This specific geometry is referring to the impossible geometry, that cannot be just simply written or demonstrated on a piece of paper, we can only assume it; this is the geometry that we could only imagine. Examples for such are: small houses with interior two times bigger, infinite staircases (that takes you back from where you came from), infinite halls (halls that lead you to the same room again and again), etc.

A recent study conducted in 2020 by RGDA (Romanian Game Developers Association) shows that approximately 8 million Romanians, aged between 15 and 64, are active video game players, most with at least 2 main platforms which I use. The highest percentage of devices are mobile phones (64% of players) and computers (60% of players). Another interesting fact found in this article, is that the majority of these people are playing games mainly for relaxing after a hard day (71%), because it is their hobby (52%), to get rid of boredom (40%), to learn new stuff (30%).

Such a high percentage gave me the motivation to start this project, knowing that in the near future many people may appreciate my work. Since the percentage of first person video game players is quite high (49%) and the percentage of puzzle enthusiasts is quite modest (29%), I decided to combine these 2 properties and add them to the project its own, thus attracting both classes of players. to try my game.

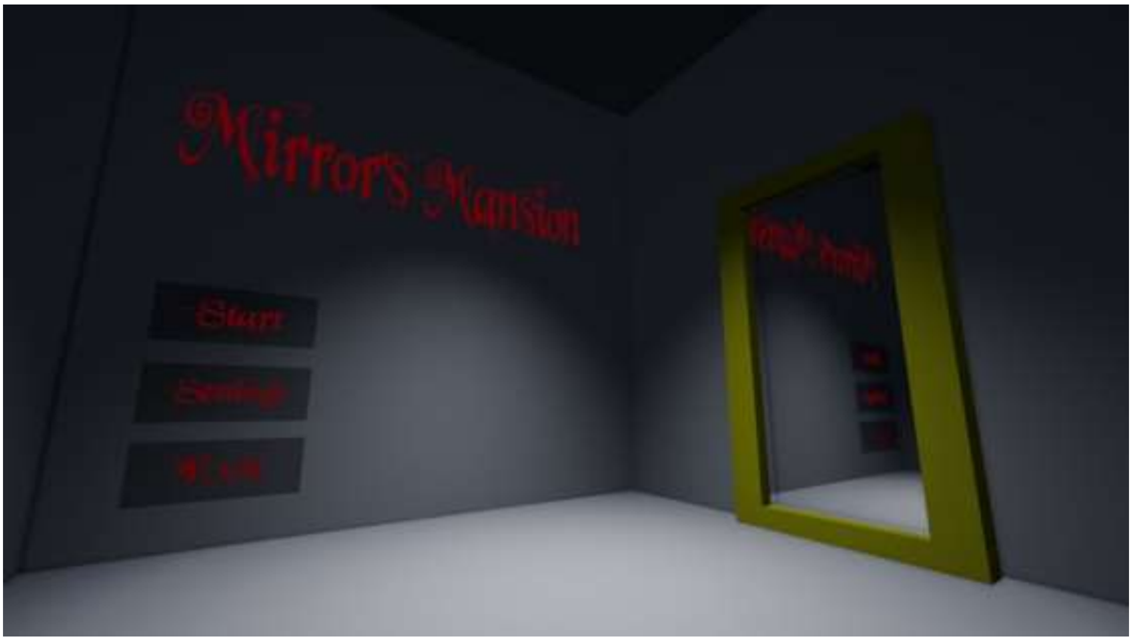


Fig. 1: Main menu screen

2 About the game

My project is the video game "Mirror’s Mansion". This is a first-person 3D computer game in which the player must solve a series of unusually complex puzzles to escape a giant mansion.

It all starts from the moment the main character enters the first room that gathers a restaurant. Here you can hear the sounds of musical instruments coming from the next room. The second room is a discotheque, with an uncomfortable atmosphere of no people, overturned tables and chairs, as well as silhouettes of musical instruments and a mirror that reflects everything about this work. As you approach the mirror, it shatters, revealing a secret entrance that leads down to a dark tunnel that leads to further rooms, which you can exit just by solving a few visual puzzles.

The trick is that you have to think out of the box, to find a solution which you wouldn't normally try. This kind of puzzles keeps you from going deeper inside the mansion. Many others died trying to escape, either by going insane or being killed by the mysterious entity find in within.

3 Non-Euclidian Theory

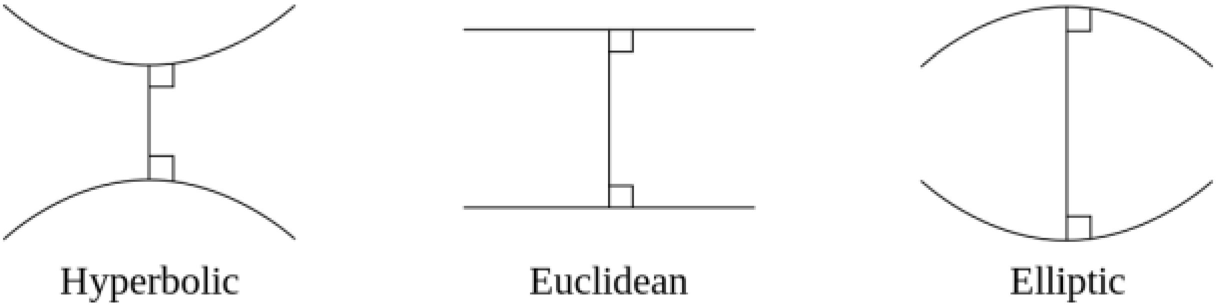


Fig. 2: Non-Euclidian theory representation

When we think about non-Euclidian, we are excluding all the postulates of Euclid's elements. That means it is based on axioms (assumptions of something being true, without a proper demonstration), closely related to those that specify the Euclidian theory.

A good way of describing such kind of logic would be to consider two parallel straight lines indefinitely extended in a two-dimensional plane.

- In the Euclidian space, these two lines remain constant at any distance, because they are parallel, they will not intersect at any point.
- In a Hyperbolic space, these lines curve away from the point we are viewing them. This makes us believe that they go more distant from each other, but in fact if we change our point of view, the distance between them remains the same.
- In an Elliptic space, these lines seem to intersect in the distance. Same as before, if we change our point of view, the distance between the lines, remains the same.

In both spaces, Hyperbolic and Elliptic, we can't distinguish if these lines are actually parallel, only if we change the perspective with which we are viewing this space.

4 Game Engine

Developing such ambitious ideas require a powerful, versatile and easy to use game engine. I have chosen Unity as the game engine that will power my game due to its extended and sturdy framework and user-friendly interface.

Unity is commonly known as the most versatile game engine because of its ability to create 3D and 2D games on all the popular platforms. Its features make it fairly easy for us programmers to bring the experience of our game to multiple platforms such as Windows, Linux, Mac, Android, IOS, PlayStation, Xbox, WebGL and even TvOS. Most of the optimization and conversion is handled directly by Unity, but we still have to make adjustments to the performance and also the controlling input.

Another advantage of this game engine is the ability of easily crating and editing Assets such as textures, sounds, images or shaders. If you are not into creating such things for your game, you can always search the Unity Asset Store on web, to find a lot of tools that you can download and use as written in their license.

Being one of the widely known game engine and due to its ease of use, there are a lot of tutorials on the internet for almost everything you want to achieve with your game, including: how to script actions, events, how to create assets and how to use them.

5 Creating the game

As for designing the first level of the game, I decided to make the introduction in an open space, so the player can see the actual mansion the title was talking about.

We can actually inspect the dimensions and geometry of this mansion before going inside and experiencing a sudden but rather interesting change in its aspect. Its color represents the coldness of the place.

Respecting the subtle horror schematic, I decided to implement a nightly atmosphere where only the moon and some leafless trees can be seen besides darkness. At the first sight this place should not look very promising, but should attract the attention of the player to check out as much details as possible. I also added dark fog effect, so the player seems to be digging through darkness.

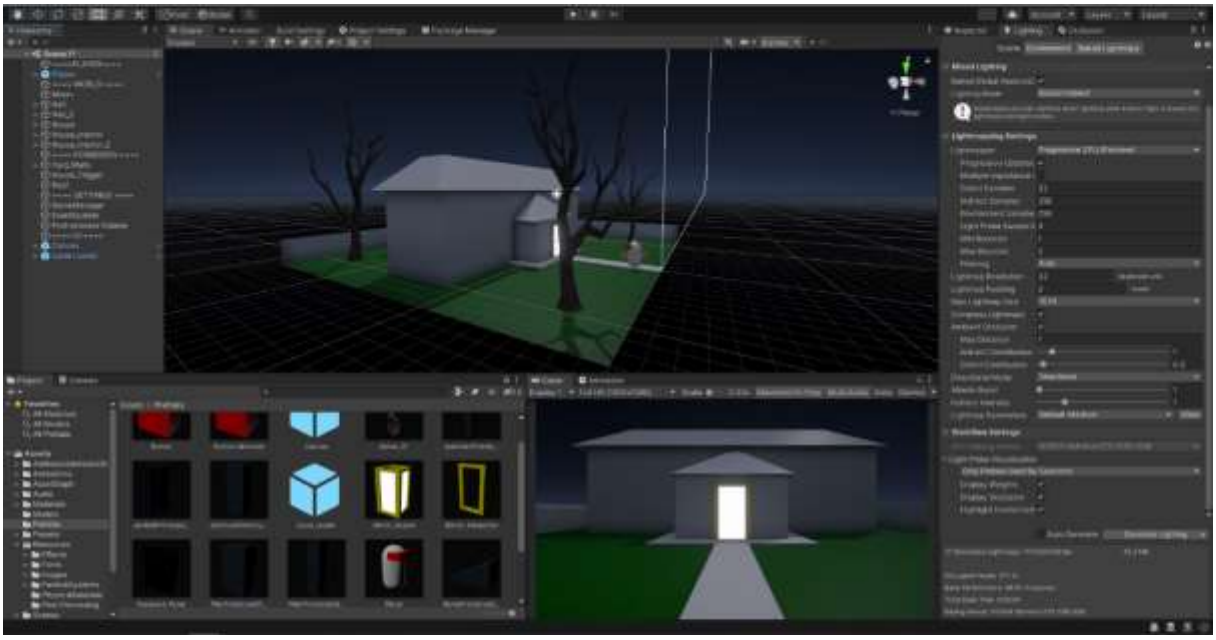


Fig. 3: Creation of the introduction level

As seen in the screenshot above, the entrance in the mansion is white, that is because the interior is rendered outside of this place, thus making a good impact in game's performance and also allowing for an outstanding and intriguing effect of large space inside a small space.

Crossing through the main door teleports the player seamless, far away of this plane, into the mansion's interior. This gives the impression of an impossible space, because the interior will be almost twice as big as the mansion.

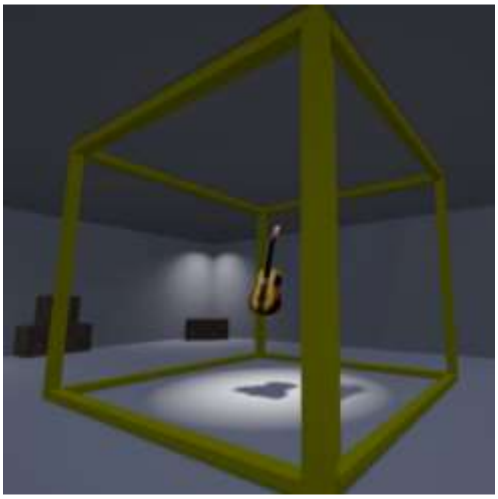


Fig. 4: The impossible cube - Guitar

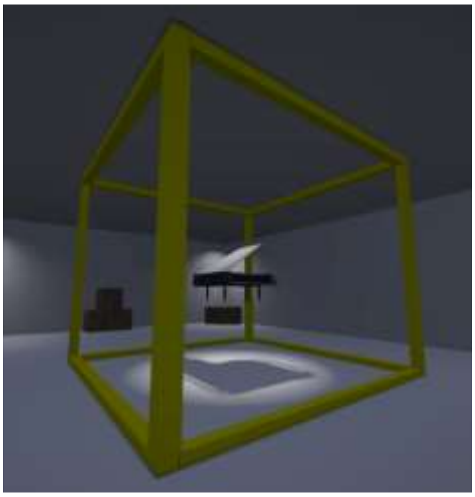


Fig. 5: The impossile cube - Piano

The screenshots above represent another level in the game. The puzzle there is tied to this impossible cube. The idea behind this puzzle is to explore the room found within the cube, searching for an indication about the order of each instrument. This order later will come in handy because the only way out of this room is entering the right password. If you wonder how this cube works; on every side of it, there is a different instrument and a different room. You will be able to cross smoothly between those rooms to find the answer and complete the level. The "impossible" refers to the ability to see multiple objects at the same time, in the same location, but from different perspectives (sides of the cube).

Many scripts were used while creating and designing the game, and because of the project's complexity, a good organization was required.

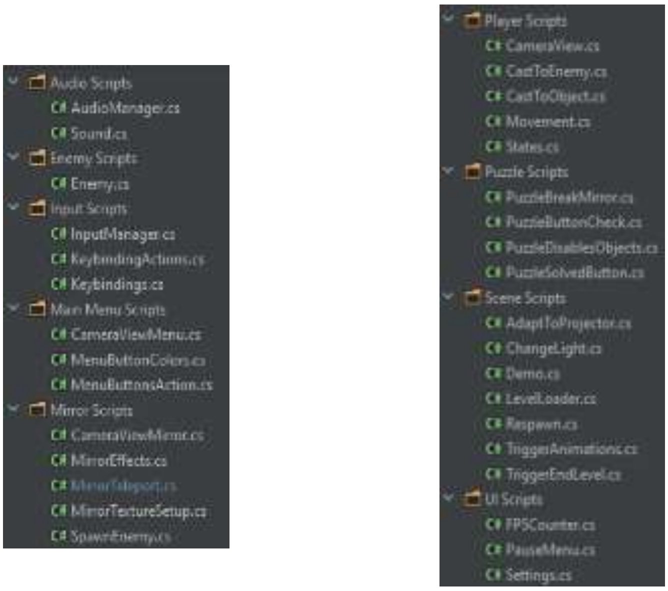


Fig. 6: Scripts hierarchy

The following is a piece of code that makes the seamless teleportation possible between different places:

```
If (dotProduct < 0f)
{
    // Math
    var rotationdiff = -Quaternion.Angle(transform.rotation, otherMirror.rotation);
    rotationDiff += 180;

    // Rotate Player
    _player.Rotate(Vector3.up, rotationDiff);

    // Teleport Player
    var positionOffset = Quaternion.Euler(0f, rotationDiff, 0f) * mirrorToPlayerPosition;
    _player.position = otherMirror.position + positionOffset;

}
```

6 Conclusion

The game is still work in progress. The unique aspect of it comes from the approaching of a new mindset, that makes you develop a new way of thinking and solving puzzles. In the future more levels with puzzles of this styles will be added, making the experience even more overwhelming than it already is.

Some future notes:

- More levels
- Harder levels

- More hostile entities
- Additions to the story
- Embrace more horror atmosphere
- Eventually an android version of the game

Acknowledgement: This work was supervised by Conf. dr. Livia SÂNGEORZAN & Lector dr. Vlad MONESCU from the Faculty of Mathematics and Informatics at Transylvania University, Brașov.

References

[1] Unity home page - https://unity.com/ Last accessed April 2022
[2] Unity manual - https://docs.unity3d.com/2022.2/Documentation/Manual/index.html Last accessed April 2022
[3] Unity scripting manual -https://docs.unity3d.com/2022.2/Documentation/ScriptReference/index.html Last accessed April 2022
[4] More about Non-Euclidian geometry - https://en.wikipedia.org/wiki/Non-Euclidean_geometry Last accessed April 2022
[5] More about the conducted research by RGDA - https://rgda.ro/romanian-video-games-consumer-study-2020/ Last accessed April 2022

Alexandru - Serban MOLDOVAN
Transilvania University Brasov
Faculty of Mathematics and Informatics
Str. Iuliu Maniu, nr. 50, Brasov
ROMANIA
E-mail: alex.moldo.serban@gmail.com

Sixth International Conference on Applied Informatics
Imagination, Creativity, Design, Development
ICDD 2022, May 5-7
Sibiu, Romania

The EU Digital COVID Certificate: A technical standpoint review

Vlad Oleksik

Abstract

The ongoing pandemic situation has faced worldwide entities with the need for a digital, more robust, interoperable system for securely and privately storing medical information of the general population, ensuring that anyone has the ability, provided the consent of the holder, to check the validity of such data, while doing away with the need for a permanent Internet connection or centralised storage. Thus, the present paper aims to provide useful insight regarding the cryptographic, mathematical aspect of the EU Digital COVID Certificate functioning principle, as well as to apply this information in an application developed in Python, serving as a proof-of-concept of the offline certificate verifying process, as well as of the signing process that takes place at the level of the issuing authorities. The program's interface, conceived as a website, has its design based on Bootstrap, and the local server managed using the Flask library of Python. The site also presents a webpage which briefs the user regarding the concept of digital signatures, the functional unit of the application.

1 Introduction

The COVID-19 pandemic has generated an array of problems affecting all social, as well as scientifical aspects, most stemming from the urgent need of a solution to effectively manage the situation from an epidemiology standpoint – the ones affected were not only the authorities, but also various organisations from the cultural, tourism and economic spheres who were required to know the risk level of people they needed to interact with. Thus, at the international scale, several standards for the storage and representation of these data have been put in place, the most commonly used amongst them being the European Digital COVID Certificate (EUDCC) (Fig. 1.).

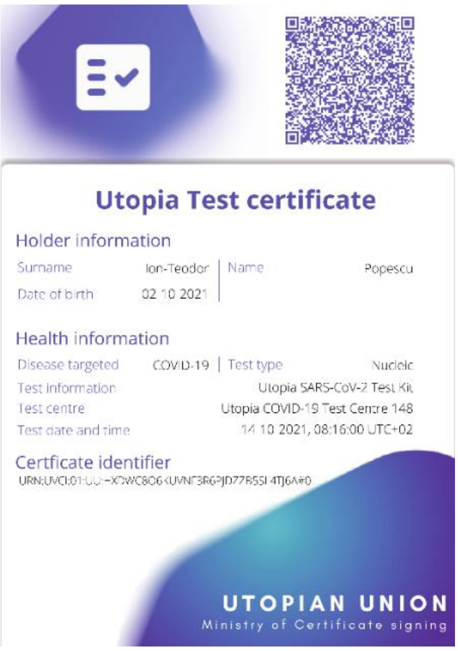
The problem posed by the Green Certificate is, however, related to an older, more general problem: ensuring consensus regarding the origin and/or validity of data, issued by one entity in a centralised/semi-centralised system, but verifiable by anybody. Furthermore, the certificate system must be scalable and robust, protected from outages. Also, its verification process must take place exclusively offline, independently from a communications network [1].

Thus, a brief representation of the EUDCC-specific requirements is as follows:

- The ability of any institution to issue a certificate that would immediately be recognised by every entity.
- The ability of anyone to privately, securely check any certificate presented to him.
- The data contained in a certificate must be stored in one single copy – at the bearer; this way, the privacy is guaranteed by the fact that the holder physically owns their

personal data, which is necessary *and* sufficient enable anyone to verify such a certificate.

- The ability to generate and check any certificate without an Internet connexion.
- The certificate must be safe i.e., nobody should be able to forge a certificate.
- The use of this system must be possible with limited resources, while accounting for most of the world population as its users.



Utopia Test certificate

Holder information

Surname Ion-Teodor | Name Popescu
Date of birth 02 10 2021

Health information

Disease targeted COVID-19 | Test type Nucleic
Test information Utopia SARS-CoV-2 Test Kit
Test centre Utopia COVID-19 Test Centre 148
Test date and time 14 10 2021, 08:16:00 UTC+02

Certficate identifier
URN:UVCI:01:UU:-XDWC8O6KUVNF3R6PJD77R5SI4TJ6A#0

UTOPIAN UNION
Ministry of Certificate signing

Fig. 1: An example of a EUDCC (as per the official standards)

Given these circumstances, there seems to exist a contradiction, from an information theory standpoint, between the inherently private character of such data and the ability of anyone to verify them. In the following chapter, approaches to this generalised problem, as well as their specific advantages and disadvantages, will be presented, along with several best practices considered by the EUDCC standards to increase the security and efficiency of the system.

2 Different approaches

There is a multitude of approaches to privacy-enabled, secure big data management, these presenting differences both regarding the paradigm which is used and in view of their corresponding efficiency and applicability in various situations. Among them, we can distinguish the naïve approach, using centralised (or semi-centralised), permanently synchronised databases, the use of blockchain, as well as the definitive solution represented by the cryptographic procedure of digitally signing a document.

2.1 Using conventional databases

A first approach to the problem of the EUDCC is consists of the use of databases. These could either be centralised (represented by a single database managing the entire dataset corresponding to all existing certificates) or semi-centralised (comprising of a number of databases, each managed by a relevant national or regional institution, each hosting a total or partial subset of the entirety of the EUDCC-related data).

This method of managing the certificate system would immediately present numerous apparent flaws. Firstly, this approach would imply not only maintaining a functioning Internet connexion to enable the issuing authority to generate the certificate, but also a permanent connexion between institutions for synchronising the databases and a working Internet connexion of any person in need to verify any such certificate.

Furthermore, there would be the significant disadvantage of the unrestricted access of an institution to the entire, *human-readable* database. Thus, in compromising a single node, the entire network could be irreparably affected from a security standpoint.

Another important drawback to this approach is the ability of every entity in the system (including people with no official background) to check any certificate against a *public* list of every person and its corresponding medical information – an unacceptable situation privacy-wise and a problem which is hard to ignore.

Given the circumstances, this approach is fundamentally flawed and has not been implemented as a primary, both nationally and officially accepted way for a person to prove its medical status. Nevertheless, secondary or local systems which are based on repeated queries to a central database have been put in place in countries such as Cambodia, China, Russia, Qatar. A slightly modified version of this paradigm has been implemented in Singapore, with the important mention that the personal medical data are encrypted and stored securely, only being accessed when a decryption key is provided along with a certificate. Still, drawbacks related to the robustness and scalability of this system are evident.

2.2 Using Blockchain technology

An alternative to the naïve approach is the use of a blockchain-like structure which would, functionally, follow the model of a public, write-only database, a ledger containing every action, any modification to the database, available in a decentralised manner, in every node, whose security and, inherently, whose security are ensured using the combined processing power of all the participants [2]. Each block containing data is inherently, mathematically linked to its predecessor (Fig. 2.).

Furthermore, for a block to be accepted, it needs to contain a special, unique number with the special property that its hash ends in a specific number of zeroes. As there is no 'optimised' approach to searching such numbers, the only way to generate a new block is by trial and error (*mining*). The necessary number of zeroes of a desired hash is set so that a new number is, on average, found every 30 seconds (depending on the desired volatility of the blockchain). When a block is found, the miner is incentivised to only include accurate, true transactions, the block being, otherwise, invalidated by other nodes and the reward for generating a block, lost. Such an infrastructure is also at the heart of cryptocurrency.

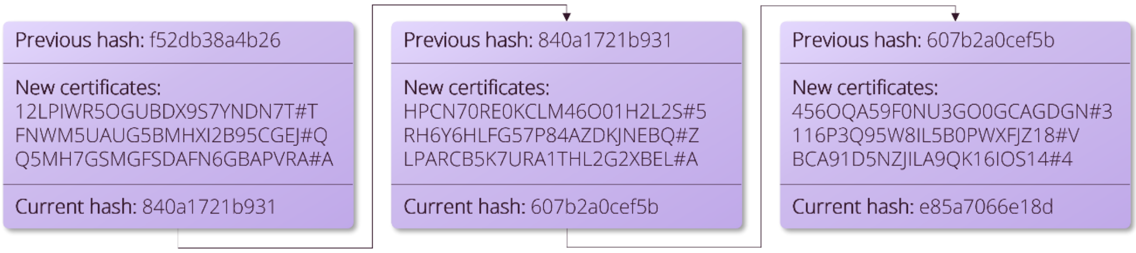

Previous hash: f52db38a4b26
New certificates:
12LPIWR5OGUBDX9S7YNDN7T#T
FNWM5UAUG5BMHXI2B95CGEJ#Q
Q5MH7GSMGFSDAFN6GBAPVRA#A
Current hash: 840a1721b931
Previous hash: 840a1721b931
New certificates:
HPCN70RE0KCLM46O01H2L2S#5
RH6Y6HLFG57P84AZDKJNEBQ#Z
LPARCB5K7URA1THL2G2XBEL#A
Current hash: 607b2a0cef5b
Previous hash: 607b2a0cef5b
New certificates:
456OQA59F0NU3GO0GCAGDGN#3
116P3Q95W8IL5B0PWXFJZ18#V
BCA91D5NZJILA9QK16IOS14#4
Current hash: e85a7066e18d

Fig. 2: The blockchain protocol

This approach also presents functional problems, such as the large processing power needed to keep the system secure, the need for a permanent Internet connexion and the inherently public character of the blockchain. Last, but not least, there is the augmented risk of compromising an institution.

However, several standards have proposed encrypting data on the blockchain to address the privacy concern and using blockchain for a national and even international certificate system. Particularly, the *AOKpass* standard uses pre-existent infrastructure, namely the Ethereum blockchain. Other systems using blockchain can be found in South Korea, as well as New York, with its Excelsior Pass.

2.3 Cryptographic algorithms for digital document signing

In 1977, Ron Rivest, Adi Shamir and Leonard Adleman have published their paper "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems" [3], presenting a solution to the problem of establishing a common secret between two unknown people, while making it impossible for a third party to know the secret. This breakthrough was marking the beginning of asymmetric cryptography.

The **RSA** protocol, defined for the first time in that respective paper, was based on the computing efficiency asymmetry between two, reciprocally inverse, mathematical functions. The functions used by RSA are the multiplication of two (extremely large) prime numbers and, respectively, the prime decomposition of number. In this respect, it is trivial for a computer to multiply two numbers – p and q, albeit of the order of 2^{2048}, but the reverse operation implies dividing n by every prime number which is less than $\sqrt{n}$ before one can expect to find its prime factors p and q. The complexity of such an operation would be so big that such an operation would only be successfully completed once a reliable quantum computer is built (using Shor's algorithm) [4].

This way, we can define a *public* mathematical function, $pub(m) = m^e \, mod \, n$, where e is a standard value, 65537, and m is the secret message. The use of the modulo operator renders the inverse of the public function impossible to directly compute. The result of the public function, c, can be transmitted publicly, securely, as nobody can reconstruct the original m. If we want to feasibly decrypt the message, we have to use information about the prime factors of n, which are *only* known by the owner of the corresponding private function. An essential property of these two functions is:

$$priv\big(pub(m)\big) = m = pub(priv(m)) \qquad (1)$$

The two functions are the inverse of each other and, if using the public functions first makes it possible to send secret messages, by using the private function first, one can generate a special number, which could only have been obtained if the owner of the private key acknowledged the message. It can be stated that $priv(m)$ is the digital signature applied to the message m, as it is publicly verifiable while it offers the guarantee of its acknowledgement by the owner of the key [5].

In practice, there is another asymmetric cryptography protocol, based on an **elliptic curve** - the graph of the implicit equation $y^2 = x^3 + ax + b$. Using a starting point *G*, one can define the addition of two points on the elliptic curve, *pG* and *qG* by drawing the line between the two points (or, if the two points coincide, the tangent to the curve in that specific point), mark the point of its intersection with the curve, then reflect it over the *Ox* axis. The result is the point $(p+q) \cdot G = pG + qG$.

By taking a substantial number, *n*, it is trivial to compute the point *nG* by generating a partition of n and adding together points *kG*, of an increasing order *k*, until we end up with the desired point. The corresponding complexity is usually reasonable enough, but we can choose standard curves with additional properties (NIST curves) that improve the efficiency of this process by a supplemental order of magnitude. When considering the inverse operation, namely, finding *n* given a point *nG* and knowing the generator *G*, one would have to check all values up to *n* to find the desired value, due to the unpredictable character of elliptic curve addition [6].

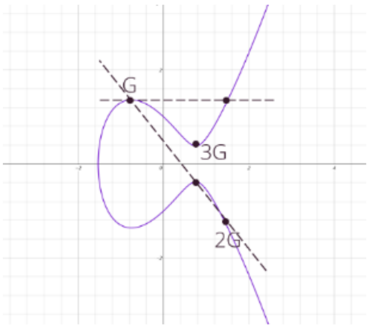


Fig. 3: Point addition on the elliptic curve

This computational asymmetry can also be used to create an encryption/digital signing standard which improves the time efficiency of the private function as described above, while massively reducing the necessary key size when compared to RSA. For instance, a 256-bit elliptic curve key is as computationally secure as a 3072-bit RSA key.

These algorithms allow an institution to sign any information, being verifiable by anyone. Cryptocurrencies such as Bitcoin or Ethereum themselves use elliptic curve algorithms for their security. Several bad implementations of this algorithm have had dire consequences in the security field [8].

Apart from the public key signatures used for defining numerous national and international health certificate standards, another widely used type of certificate infrastructure has been used, namely the one based on the concept of "verifiable credentials" [9]. This concept is theoretically identical to the public key infrastructure, the only practical difference being the implementation of multiple signatures and signing partial messages to provide a more optimised privacy management.

The elliptic curve digital signing algorithm is more robust both memory-wise and in terms of time-efficiency for the signing operation [10], by using particular curves and algorithms such as Pippenger's algorithm for batched scalar multiplication [11].

It is important to mention that, by using such a protocol, all the information necessary to check a EUDCC certificate and obtain medical information is contained in the QR code.

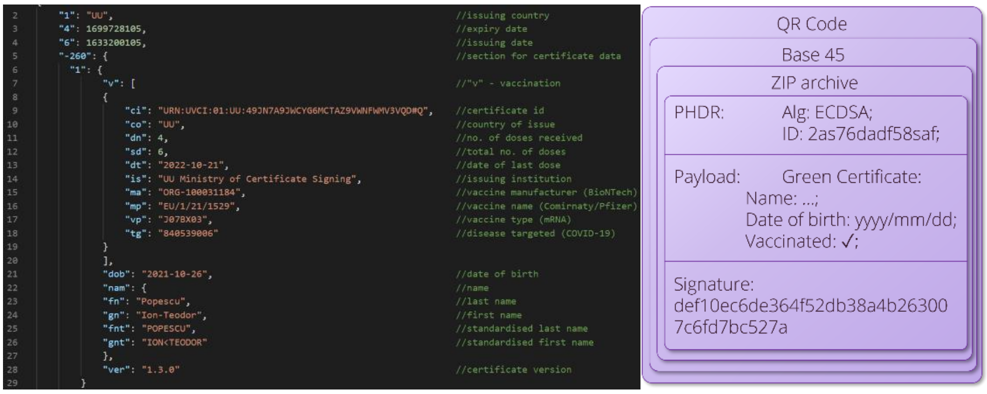
3 The analysis of the EUDCC format

The functional principles on which this standard is based are represented both by the secure off-line validation and by the efficient storage and encoding of the data for increasing convenience and reliability.

An EUDCC certificate is divided into two sections, corresponding to the version of the certificate and its type. The latter, much larger and more significant, is divided into multiple sections for the medical data, the date of birth and the name of the holder, respectively. Along with the exact singing and expiry timestamps in UNIX format, they represent the payload of the certificate.

Thus, the data in a generic certificate are formatted as a JSON object, as in Fig. 4. a.

Starting from these data, the generation process is as follows: a (protected) header is attached to the payload. This section is called 'protected' as it will be signed. The header specifies the signing standard/algorithm (ECDSA/ES256 – recommended by the EU, corresponding to the elliptic curve protocol, or RSASSA-PKCS1-v1_5/RS256 as fallback – corresponding to the RSA protocol [12] described in subsection 2.3.), as well as the first 6 characters of the base-64 encoded public key so as to make it easier for the verifier to find the corresponding key from the list of trusted authorities.



```
    "1": "UU",                                                        //issuing country
    "4": 1699728105,                                                  //expiry date
    "6": 1633200105,                                                  //issuing date
    "-260": {                                                         //section for certificate data
      "1": {
            "v": [                                                    //"v" - vaccination
            {
                "ci": "URN:UVCI:01:UU:49JN7A9JWCYG6MCTAZ9VWNFWMV3VQD#Q",   //certificate id
                "co": "UU",                                           //country of issue
                "dn": 4,                                              //no. of doses received
                "sd": 6,                                              //total no. of doses
                "dt": "2022-10-21",                                   //date of last dose
                "is": "UU Ministry of Certificate Signing",           //issuing institution
                "ma": "ORG-100031184",                                //vaccine manufacturer (BioNTech)
                "mp": "EU/1/21/1529",                                 //vaccine name (Comirnaty/Pfizer)
                "vp": "J07BX03",                                      //vaccine type (mRNA)
                "tg": "840539006"                                     //disease targeted (COVID-19)
            }
            ],
            "dob": "2021-10-26",                                      //date of birth
            "nam": {                                                  //name
            "fn": "Popescu",                                          //last name
            "gn": "Ion-Teodor",                                       //first name
            "fnt": "POPESCU",                                         //standardised last name
            "gnt": "ION<TEODOR"                                       //standardised first name
            },
            "ver": "1.3.0"                                            //certificate version
        }
```

Fig. 4: a) Example of a generic EUDCC payload; b) The structure of an EUDCC

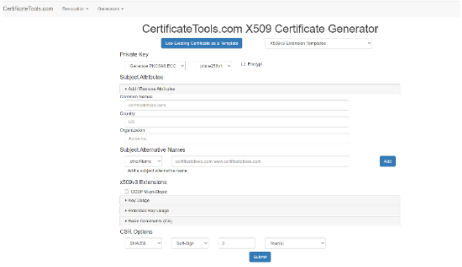
The dictionary thus formed will be passed through a *hash* function, SHA256 – a deterministic function, but very unpredictable and unfeasibly difficult to reverse, used because the result is always of the same length (256 bits) and easier to work with. The result of this function will be signed by encrypting it with the private key of the institution.

Anybody wishing to verify it can apply the public function to this signature and compare the hash of the data with this result. The slightest change will cause the mashes to not match, invalidating the certificate. The signature – the output of the private function – will be attached to the dictionary, resulting in a CBOR Web Token (CWT) which will be zip-archived to optimise the size, base 45-encoded, prefixed with "`HC1:`" and stored as a QR-code to facilitate computer vision readability (Fig. 4. b) [1].

The last component needed for the certificate system to work is the European scale 'Trust framework' comprising of a centralised list of all the trusted keys [13]. As every institution only needs a single public key occupying $\sim 90\,B$, needing to be changed approximately once every 3-5 years, the necessary memory is incredibly small, and the list can even be managed manually.

4 The implementation of the Proof-of-Concept app

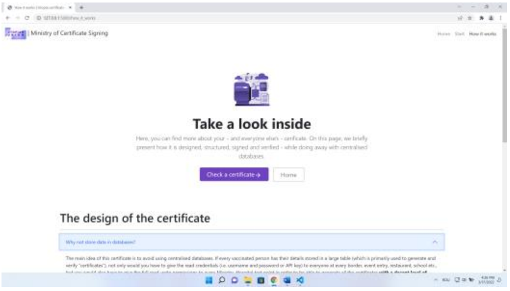
To illustrate the functioning principle of the EUDCC, the present study aimed to develop an open-source web application that would allow scanning a QR-code in an image, decoding the information in the certificate, and validating the medical data by checking it, offline, against a public, periodically updated list of trusted institutions and displaying the result. Also, a second application has been realised, allowing the user to simulate the generation of a EUDCC-standardised certificate starting from a dataset of his choice and a proprietary private key (a X.509 certificate generated using OpenSSL or an online implementation (Fig. 5.) [14], which will obviously not be recognised as trusted, but will confirm the origin of the issuer).



CertificateTools.com
CertificateTools.com X509 Certificate Generator
Use Existing Certificate as a Template
Private Key
Subject Attributes
Subject Alternative Names
Add
x509v3 Extensions
CSR Options
SHA256
Self-Sign
Submit

Fig. 5: The site used for generating a private-public key pair [14]

For implementing this program, the chosen programming language has been Python, as it allows for the straightforward management of complex data, as well as a large variety of modules enabling the integration of a multitude of functionalities into the application, for creating the interface. Thus, the cryptographic operations were performed using the Crypto library, to ensure that the existing standards are properly respected, while the interface was conceived as a website and the Flask module has been used for handling requests from the localhost IP address (127.0.0.1) on port 5000.



Take a look inside

Here, you can find more about your - and everyone else's - certificate. On this page, we briefly present how it is designed, structured, signed and verified - while doing away with centralised databases

The design of the certificate

Why not store data in databases?

The main idea of this certificate is to avoid using centralised databases. If every vaccinated person has their details stored in a large table (which is primarily used to generate and verify "certificates"), not only would you have to give the read credentials (i.e. username and password or API key) to everyone at every border, event entry, restaurant, school etc.,

Fig. 6: Accessing the app from a browser

4.1 The decoding and validation app

For this app, a Python function has been defined, receiving the address of an image with a QR-code, returning a dictionary with the result of the validation, containing all the medical data in the certificate, which is classified as either valid (an official EUDCC), invalid (unknown or inexistent signature), self-signed (only signed by the pair-application, thus recognised, but not officially valid), or error (the data is unreadable or does not respect the standard).

4.1.1 Processing the cryptographic data

To validate a certificate, the data is read from the QR-code using pyzbar.decode() from the pyzbar module. If the resulting string begins with "HC1:", the data is a health certificate and we eliminate this prefix, proceeding to decode the resulting substring (consisting of numbers, capital letters, as well as the symbols ., :, +, -, $, %, *, /, and spaces) from base 45. The resulting byte array is a zip archive, which will be processed using zlib.decompress(), obtaining a signed COSE message. This protocol allows for transmitting *claims*, signed by asymmetric algorithms such as the ones presented in section 2. This message contains a 'payload' – the one in Fig. 4. a), a 'phdr' – a protected header, a 'uhdr' – an unprotected header which is empty, as well as the signature. The application will first find the corresponding key in the trusted list (a JSON file which will be opened from a local address, "EU_keys.json"). An updated version of such a file can be found online [13].

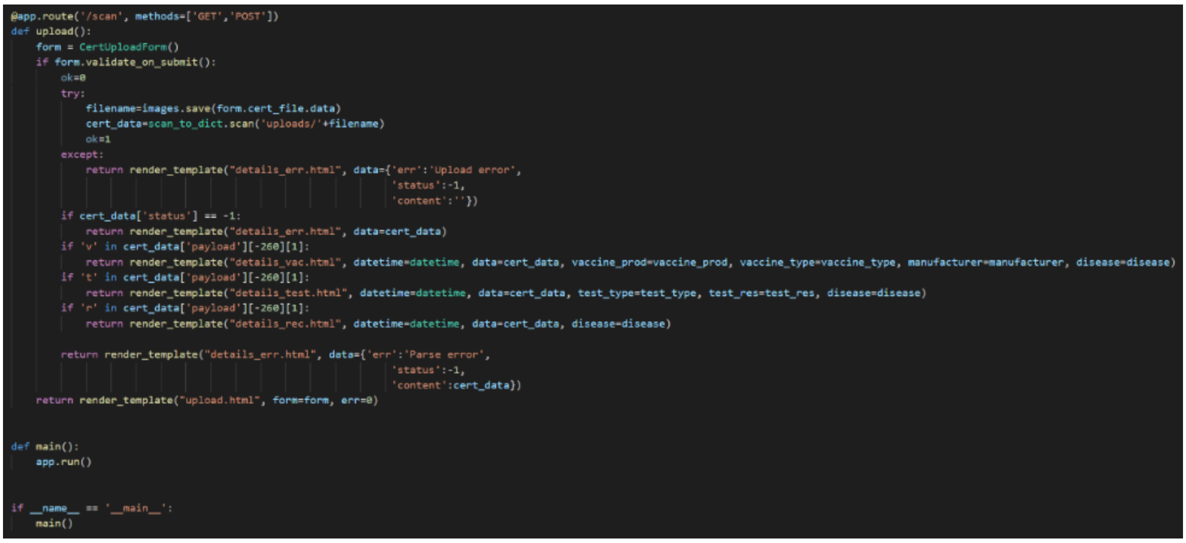
The ID of the public key can be found in the protected header of the certificate, as msg.phdr['KID']. The program iterates through all the trusted keys in the file and finds the correct one, along with information such as the encryption protocol, used to create a COSE signature object to be attributed to `msg.key` before calling verify_signature().

If no EU-trusted key is found to validate the certificate, the program tries the public key used specifically by the generation program described in subsection 4.2 for demonstration purposes. The procedure is similar, but the key data are read from a .pem file. The result determines the status of the certificate.

4.1.2 Running the local Flask server for the web interface

Using the Flask module, the program creates a web server at localhost, on port 5000, delivering the following pages: /home – a homepage presenting the project, /how_it_works – a page presenting the technicalities of the concept, /scan – a page with a field for uploading a certificate to be scanned or, if a file has already been uploaded, a page presenting the data in the certificate.

As one can observe in Fig. 7., when uploading an image, it will be saved on the server filesystem, the resulting address being used as the input for the validation function described in 4.1.1. Based on the resulting dictionary, a success or an error page will be returned from the list of template files on the server, containing placeholder fields which are replaced by actual data regarding the certificate upon rendering the webpage. If no file is uploaded, a file upload page will be displayed.



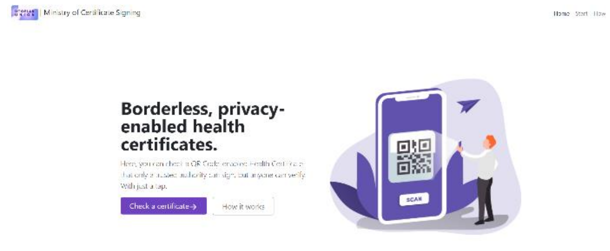
```
@app.route('/scan', methods=['GET','POST'])
def upload():
    form = CertUploadForm()
    if form.validate_on_submit():
        ok=0
        try:
            filename=images.save(form.cert_file.data)
            cert_data=scan_to_dict.scan('uploads/'+filename)
            ok=1
        except:
            return render_template("details_err.html", data={'err':'Upload error',
                                                              'status':-1,
                                                              'content':''})
        if cert_data['status'] == -1:
            return render_template("details_err.html", data=cert_data)
        if 'v' in cert_data['payload'][-260][1]:
            return render_template("details_vac.html", datetime=datetime, data=cert_data, vaccine_prod=vaccine_prod, vaccine_type=vaccine_type, manufacturer=manufacturer, disease=disease)
        if 't' in cert_data['payload'][-260][1]:
            return render_template("details_test.html", datetime=datetime, data=cert_data, test_type=test_type, test_res=test_res, disease=disease)
        if 'r' in cert_data['payload'][-260][1]:
            return render_template("details_rec.html", datetime=datetime, data=cert_data, disease=disease)

        return render_template("details_err.html", data={'err':'Parse error',
                                                          'status':-1,
                                                          'content':cert_data})
    return render_template("upload.html", form=form, err=0)


def main():
    app.run()


if __name__ == '__main__':
    main()
```

Fig. 7: The source code of the scanner application

4.1.3 The interface created based on Bootstrap

For displaying the relevant data as a webpage, multiple template-files have been created. These files are usual HTML pages, also containing adaptive fields which can be replaced with Python variables upon call of render_template() from the Flask module. For the design of these webpages, an open-source collection of CSS and JS files, known as Bootstrap, has been used, allowing the use of a variety of pre-existing, adaptive design elements such as data fields and animated buttons and dialogues by simply adding classes to objects in HTML.

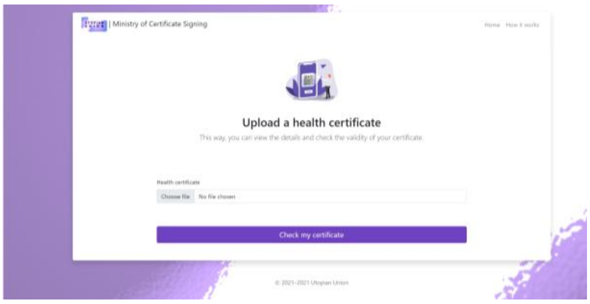
Upon accessing the application, having previously run the Python file starting the server, the homepage is displayed (Fig. 8. a), from which one can access the upload page (Fig. 8. b), which leads the user to upload a certificate and be presented with the corresponding results page (Fig. 8. c).



Borderless, privacy-enabled health certificates.

Here, you can check a QR Code enabled Health Certificate that only a trusted authority can sign, but anyone can verify. With just a tap.

Fig. 8: a) The homepage



Upload a health certificate

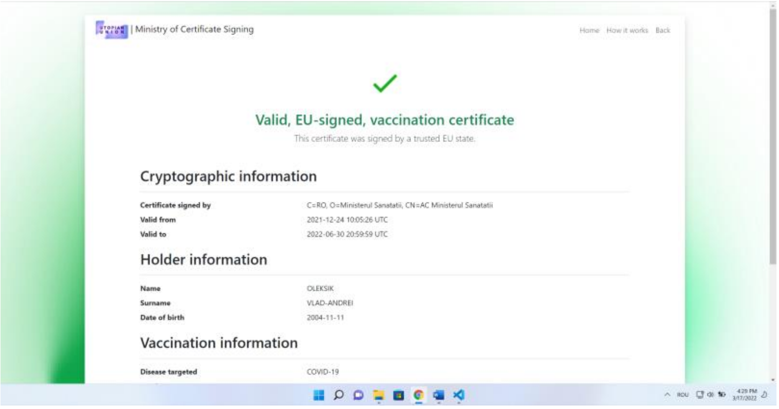
This way, you can view the details and check the validity of your certificate.

Health certificate

No file chosen

© 2021-2021 Utopian Union

Fig. 8: b) The image upload page



Valid, EU-signed, vaccination certificate

This certificate was signed by a trusted EU state.

Cryptographic information

Certificate signed by	C=RO, O=Ministerul Sanatatii, CN=AC Ministerul Sanatatii
Valid from	2021-12-24 10:05:26 UTC
Valid to	2022-06-30 20:59:59 UTC

Holder information

Name	OLEKSIK
Surname	VLAD-ANDREI
Date of birth	2004-11-11

Vaccination information

Disease targeted	COVID-19

Fig. 8: c) The scan results page

4.2 The generation simulation application

The second application realised as a part of the presents study is a ‘complementary’ one, taking a private key file and allowing the user to demonstratively simulate the generation of a QR-code following the EUDCC standards and guidelines, despite only being compatible with the application described in subsection 4.1.

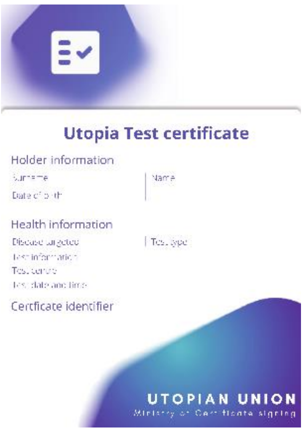
4.2.1 The data signing function

To generate a digital signature corresponding to a dataset, it is firstly necessary to obtain a pair of public and private keys – an ordered set of large numbers, cryptographically used as described in subsection 2.3, stored, according to X.509 standards, as two files with the extensions .pem and .key. Together, they constitute a “document signing certificate”, as the same standard is used to sign PDF documents or information packets when visiting a website over HTTPS. For creating such a key pair, one could use a Linux command-line utility called OpenSSL, as well as similar resources available online [14].

Starting from such a private key file stored in the filesystem of the server, a Python function has been defined, receiving a dictionary structured to contain EUDCC-specific data as input, resulting the local address of a QR-code image file that is generated to de displayed on the webpage.

For this purpose, the generation function constructs the JSON payload of the certificate to be generated depending on the data contained in the input values (vaccination, recovery, test certificate). Input data is processed by eliminating excess spaces and correctly formatting names and timestamps. A random unique certificate identifier is also generated, considering the checksum algorithm used to check its correctness. Medical data are encoded according to international conventions i.e., 'EU/1/20/1528' – Pfizer; '840539006' – COVID-19.

Afterwards, the two key files are read, obtaining the parameters to be used for signing the data thus prepared by being attached to a COSE Message object to which we apply the Sign1Message() method. Following the signing process, the data is archived using the pyzlib module, then converted to base 45, “HC1:”-prefixed and provided to the QR code image generation function form pyzbar. The pillow module (Python image library) is then used to open a template image with the visual format of the certificate, as seen in Fig. 9. a. Using the procedure draw.text, this image is completed with the relevant info and the QR-code image is pasted in place, resulting a certificate such as the one depicted in Fig. 9. b., which is saved and passed to the server Python code displaying it on the user’s screen.



Utopia Test certificate

Holder information

Surname | Name

Date of birth

Health information

Disease targeted | Test type

Test information

Test centre

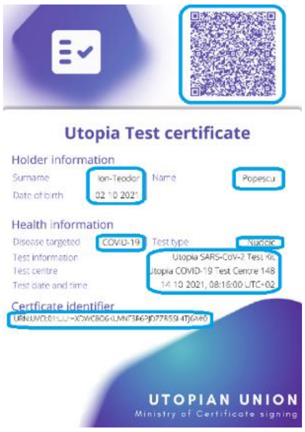
Test date and time

Certficate identifier

UTOPIAN UNION

Ministry of Certificate signing

Fig. 9: a) The template image



Utopia Test certificate

Holder information

Surname Ion-Teodor Name Popescu

Date of birth 02 10 2021

Health information

Disease targeted COVID-19 Test type Nucleic

Test information Utopia SARS-CoV-2 Test Kit

Test centre Utopia COVID-19 Test Centre 148

Test date and time 14 10 2021, 08:16:00 UTC+02

Certficate identifier

URN:UVCI:01:UU-XOWC8O6-KUMNT3R6PJD7ZR5SI-4TJ6A60

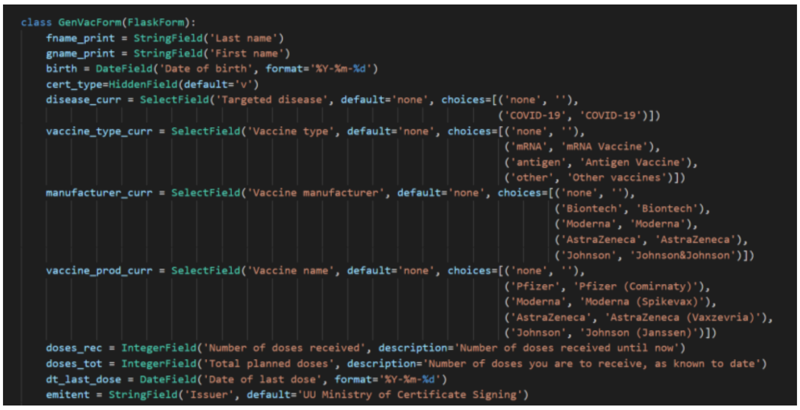
UTOPIAN UNION

Ministry of Certificate signing

Fig. 9: b) The form filled up with data

4. 2. 2. The local Flask server

To manage the user interface, as for the application described in section 4.1, Flask was used for this program as well. The server-side code listens for requests on multiple subdomains: /home, /how_it_works, /generate – comprising links to its subdomains /rec, /vac, /test – for the corresponding types of certificates that can be generated, and a page displaying the result of a generation process, /success. The first three above-mentioned pages are static, while the pages designated for collecting data for generating certificates contain input fields, the content of which will be sent to the server by using the 'POST' method upon pressing the generation button. These forms are handled as Python WTForms objects, structured as in Fig. 10. While waiting for a response, these pages will display a loading animation. When the page with the generated certificate is available, the user will be redirected to it, using the name of the final certificate, generated by the function in sub-subsection 4.2.1 as a variable element in the address. If the input is erroneous, tips will be provided under the text fields.

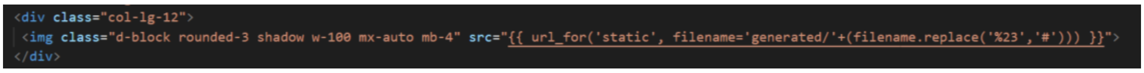


```
class GenVacForm(FlaskForm):
    fname_print = StringField('Last name')
    gname_print = StringField('First name')
    birth = DateField('Date of birth', format='%Y-%m-%d')
    cert_type=HiddenField(default='v')
    disease_curr = SelectField('Targeted disease', default='none', choices=[('none', ''),
                                                                           ('COVID-19', 'COVID-19')])
    vaccine_type_curr = SelectField('Vaccine type', default='none', choices=[('none', ''),
                                                                            ('mRNA', 'mRNA Vaccine'),
                                                                            ('antigen', 'Antigen Vaccine'),
                                                                            ('other', 'Other vaccines')])
    manufacturer_curr = SelectField('Vaccine manufacturer', default='none', choices=[('none', ''),
                                                                                    ('Biontech', 'Biontech'),
                                                                                    ('Moderna', 'Moderna'),
                                                                                    ('AstraZeneca', 'AstraZeneca'),
                                                                                    ('Johnson', 'Johnson&Johnson')])
    vaccine_prod_curr = SelectField('Vaccine name', default='none', choices=[('none', ''),
                                                                            ('Pfizer', 'Pfizer (Comirnaty)'),
                                                                            ('Moderna', 'Moderna (Spikevax)'),
                                                                            ('AstraZeneca', 'AstraZeneca (Vaxzevria)'),
                                                                            ('Johnson', 'Johnson (Janssen)')])
    doses_rec = IntegerField('Number of doses received', description='Number of doses received until now')
    doses_tot = IntegerField('Total planned doses', description='Number of doses you are to receive, as known to date')
    dt_last_dose = DateField('Date of last dose', format='%Y-%m-%d')
    emitent = StringField('Issuer', default='UU Ministry of Certificate Signing')
```

Fig. 10: The structure of a WTForms object

4. 2. 3. The webpage design and Bootstrap

The webpage design is resembling of the verification application, the main difference being represented by the introduction of various input fields, tips, as well as loading animations. The page displaying the resulting certificate contains a dynamic Flask field which integrates the HTML template and is replaced with actual data stored in Python variables upon its rendering. The code for these elements is as it can be seen in Fig. 11., while all the illustrations used in this project are freely available online [15].



```
<div class="col-lg-12">
 <img class="d-block rounded-3 shadow w-100 mx-auto mb-4" src="{{ url_for('static', filename='generated/'+(filename.replace('%23','#'))) }}">
</div>
```

Fig. 11: Dynamic HTML template elements from Flask

5 Security and reliability audit

The application thus developed presents interface elements facilitating the user interaction, pursuing its purely educational role, as it presents, in a high level of detail, the functioning principle of such a safe, yet robust and highly scalable credential management system, but it represents, at the same time, a Proof-of-Concept regarding its possible use for the decentralised management of a vast volume of distinctly personal data.

As the only functional copy of any piece of information stored in this system is held by its holder, the person it is relevant to, and the data contained in a certificate is impossible to forge, making it both necessary and sufficient for gathering necessary information about people interacting on a daily basis in the pandemic context, it can be noted that the total centralised memory necessary for the system to function does not depend on the number of existing certificates “in the wild”. Thus, the need for synchronising databases, one of the hardest computational problems of present data systems [16], is avoided. In other words, a certificate can be permanently deleted after having been received by its holder.

The system only needs to trust that the list of public keys is genuine (all issuing authorities known to the verifier program are the real institutions). Thus, usually **vulnerable points**, such as network vulnerabilities (such as the case of a DDoS), telecommunications controls, or security controls are non-existent due to the simplicity of the offline architecture. Information processing vulnerabilities when generating a certificate are unlikely due to the policy of using prepared statements and considerably restricting user input by all national government applications. Other potential security concerns are the possibility of publicly leaking private keys, as well as manually injecting the verifier code in some locations with altered trust lists. This way, the most concerning aspect of the security of such a system is the human factor and having set national and international management policies to mitigate this risk is crucial. Although one leak has taken place, the impact of such events is temporary and, at most, national.

The system **complies with legislations** set in place across the world as the data stored is in control of and held by the user.

Internal policies for private key management and medical data backup are useful and respected, as the system has been stably functioning over a large period, even facing a never-before seen pressure from numerous ill-willed entities. However, a significant flaw is the lack of training of the certificate verifying personnel, being prone to misinterpret the results – an expired or downright invalid certificate being mistaken for a valid one in spite of numerous software warnings. This has led to applications not displaying information on an invalid certificate to attract attention from the user. Another area of concern is the inability of staff to legally request an ID of a person, a necessary step for ensuring that the valid certificate truly belongs to the one presenting it.

With increasing media attention on this topic, the level of **security training** of everyone involved in certificate verification has significantly risen, strengthening the entire infrastructure to a point of very low risk.

Regarding the **resources needed**, the system only needs to be active upon generation and verification, doing away with the need to store user data or perform inefficient searches and making it reliable and efficient.

Furthermore, the time taken by a verification operation does not depend on the total number of valid certificates since there is no need for searching a list. As the number of institutional public keys is extremely low (<100), the generation of a certificate can practically be done in $O(1)$.

Synchronising these public keys can be done manually, as JSON files or even "on paper", as the size of the trust list allows for it.

The impenetrability of this system is enforced by its "zero-trust" architecture, which enables every node in the system to check every piece of information provided to it before processing it.

The validity of every certificate is, thus, based not on some intrinsic property, but on the acceptance of its corresponding public keys by every validator. Every node has the freedom to choose if it wants to recognize a public key and, indirectly, the authority it represents (Fig. 13.). In case of a private key being leaked (either by ill-will or by institutional failure), it is enough to declare that key invalid for all certificates generated in the specific circumstances it could have been compromised. Afterwards, every certificate affected would have to be re-generated using a new, secure key, compared to other models which could be more severely affected by breaking trust between official entities.

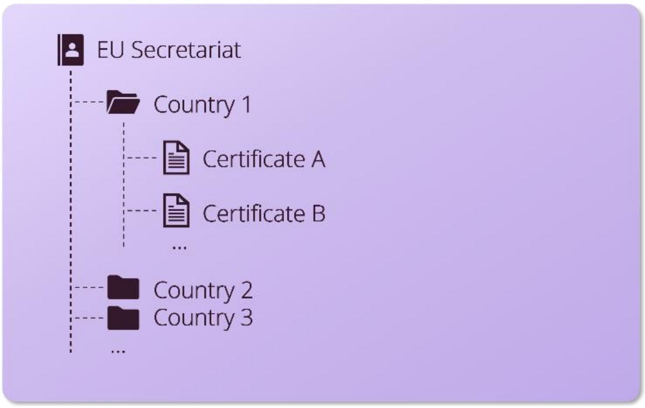


Fig. 12: The EU "Trust Framework"

6 Conclusions

The developed application demonstrates, not only from the educational point of view, but also from an information theory standpoint, the concept of real-life digital signing algorithms, securely and reliably.

The project was realised exclusively using public, open-source resources available online and the files of this project will also be made publicly available in the appendix containing media resources.

The key files are interchangeable with other pairs of files of the same format, without breaking interoperability between the created applications, allowing for multiple node setups without encountering functional problems.

The EUDCC system created based on this standard has enabled a successful handling of the medical data of hundreds of millions of people [17], standing out not only as the most used international health certificate system, but also as a natural continuation of signed, decentralised immunity certificates, the first of which were implemented in the Italy of the 17th century for handling plague epidemics. This system is a potential candidate for modernising the current "Carte jaune" yellow cards used by the WHO from as early as 1933 to prove vaccination against various other infectious diseases existing at a regional level.

Acknowledgement: This study was supervised by Prof. Adina-Mihaela Stănculescu, from the "Gheorghe Lazar" National College of Sibiu.

References

[1] The official technical specifications of the EUDCC, GitHub, https://github.com/ehn-dcc-development/hcert-spec/blob/main/hcert_spec.md

[2] The functioning principle of blockchain, YouTube, https://www.youtube.com/watch?v=bBC-nXj3Ng4

[3] R. Rivest, A. Shamir, L. Adleman, *A Method for Obtaining Digital Signatures and Public-Key Cryptosystem*, Communications of the ACM. **21** (2): 120–126., doi:10.1145/359340.359342, 1978.

[4] P. W. Shor, *Algorithms for quantum computation: discrete logarithms and factoring*, Proceedings 35th Annual Symposium on Foundations of Computer Science, IEEE Comput. Soc. Press: 124–134, doi:10.1109/sfcs.1994.365700, ISBN 0818665807, 1994.

[5] J. G. Urgelles, *Mathematicians, spies, and hackers – encoding and encryption*, RBA, ISBN 978-84-1329-219-9.

[6] D. Johnson, A. Menezes, S. Vanstone, *The Elliptic Curve Digital Signature Algorithm (ECDSA)*, International Journal of Information Security, **1**, 36–63, 2001.

[7] Enumeration of public parameters of curves used in ECC – secp256r1 (P-256), https://neuromancer.sk/std/secg/secp256r1

[8] The security problem caused by the bad implementation of ECDSA in PlayStation 3, YouTube, https://www.youtube.com/watch?v=j6yU9z8mtRE

[9] An overview of the concept of verifiable credentials, https://tykn.tech/verifiable-credentials/

[10] J. López, R. Dahab, *An Overview of Elliptic Curve Cryptography*, Technical Report IC-00-10, State University of Campinas, 2000.

[11] D. J. Bernstein, *Pippenger's exponentiation algorithm*, 2002.

[12] On the algorithms used in digital signatures, https://www.scottbrady91.com/jose/jwts-which-signing-algorithm-should-i-use

[13] Regularly updated list of the public keys for validating a European Digital Covid Certificate, https://raw.githubusercontent.com/lovasoa/sanipasse/master/src/assets/Digital_Green_Certificate_Signing_Keys.json

[14] Graphical interface for using OpenSSL in browser, https://certificatetools.com/

[15] License-free illustration library, https://www.manypixels.co/gallery

[16] H. Adkins, B. Beyer, P. Blankinship, P. Lewandowski, A. Oprea, A. Stubblefield, *Building Secure and Reliable Systems*, O'Reilly Media, Inc., ISBN 978-1-492-08313-9, 2020

[17] G. Karopoulos, J. L. Hernandez-Ramos, V. Kouliaridis, G. Kambourakis, *A Survey on Digital Certificates Approaches for the COVID-19 Pandemic*, IEEE Access, **9**, 138003-138025, doi: 10.1109/ACCESS.2021.3117781, 2021.

Vlad Andrei OLEKSIK
“Gheorghe Lazăr” National College Sibiu
Mathematics-Informatics, intensive Informatics
Gheorghe Lazăr Street 1-3, Sibiu
ROMANIA
E-mail: vlad.oleksik@cnglsibiu.ro

Sixth International Conference on Applied Informatics
Imagination, Creativity, Design, Development
ICDD 2022, May 5-7
Sibiu, Romania

Refining Color Cue Selection for Deep Image Colorization in the Context of Image Compression

Daniel Ostertag, Daniel Habermayr

Abstract

Colorizing a grayscale image is no longer a task reserved merely for artists or children. Putting paint into black and white images can now be done by Artificial Intelligence (AI), too. Considering the skyrocketing amount of pictures and videos being sent across the world, a promising application for such colorful algorithms lies in their combination with image compression of grayscale pictures, since those consist of three times less pixel values than their counterparts from the red-green-blue color space (RGB). This method relies on choosing appropriate color cues from the original image for correct recolorization after decompression. Based on recent work, we present some refined approaches to extract those cues, given an existing deep neural network to colorize grayscale images. One of these techniques lead improved colorizations compared with those of the previous method. Apart from that, we combine this model with a state of the art AI-based image compression algorithm without notable improvements.

1 Introduction

Image colorization of a grayscale image is a highly ambiguous task: many combinations of different RGB channels could correspond to the same luminance. Colorization methods therefore need some other source of information to make better than random ”guesses” about the correct color. Therefore, some form of additional information is required.

Recently, Xiao et al. [1] released their work on the use of Deep Neural Networks (DNNs) for image colorization. In contrast to most previously published methods, their approach allowed for the simultaneous use of global and local color cues. Global cues in this context describe a small set of colors predominantly appearing in an image, a ”theme” so to speak, whilst local cues represent the colors of specific pixels within the picture. In their paper, they focused mainly on making colorization an interactive process, where a user could choose these color hints.

In 2020, Xiao et al. [2] published a follow-up paper about using this method in the context of image compression. Color cues were selected automatically, rather than relying on user input. While Xiao et al. used Simple Linear Iterative Clustering (SLIC) segmentation to choose local cues, we managed to achieve slight improvements, by combining SLIC with our own method of choosing local cues.

While Xiao et al. used Better Portable Graphics [3] (BPG) to compress the input's grayscale version [2], we wanted to combine their net with a state of the art DCS (Deep Compression System), published in 2019 by Agustsson et al. [4]. Their model relies on Generative Adversarial Networks (GANs) to reconstruct parts of the encoded image and highly outperforms traditional compression methods like JPEG [5] at very low bpp (bits per pixel) rates. That inspired us to try to improve the reconstruction quality of Agustsson et al.'s system. Even though this experiment gave mixed results, it nevertheless offered new insights.

2 Related Work

2.1 Deep Colorization

Over the last two decades, many approaches were designed in the field of colorization that deal with the lack of information in the form of a similar, colored reference image. In earlier approaches, these sources had to be chosen by a user [6]. Later, this procedure was automatized through neural networks searching large databases for a suitable reference image [7]. A variation of this data-driven way can be found in works like Iizuka et al. [8]. They too propose fully automatic colorizations with the help of large-scale image data sets, yet instead of trying to find a suitable source image, they use it to train a DNN to colorize grayscale images with likely colors. While these algorithms provide mostly plausible colorizations [1], they still suffer from some ambiguity concerning objects with many distinct possible colors, e.g. cars or clothes, and may colorize them differently from the ground truth or from the user's expectations.

Simultaneously, another branch of methods to circumvent the shortage of information in image colorization developed, which built on letting users directly put colored pixels into certain regions. This approach was first introduced by Levin et al. [9] in 2004, where user-provided color scribbles on a grayscale image were propagated throughout the rest of the image, under the assumption that patches similar in intensity (luminance) should have similar colors.

Since the early 2000s, both of these branches have been improved upon, but in 2017, Zhang et al. [10] sought to combine the best of both worlds. They proposed two variants of combining a users color hints with deep learned priors in the style of 2016 Iizuka et al [8]. First, a "Local Hints Network" gives users the option of sprinkling in some sparse color dots where specific colors are wanted. Second, a "Global Hints Network" provides users with the option of defining a color histogram as theme input. Combined with the deep learned priors, both these methods ensure "realistic" colorization results. However, they cannot be used simultaneously.

In 2019, Xiao et al. [1] presented a novel deep colorization method similar to Zhang et al.'s [10] that allows for the simultaneous use of local as well as global cues, by designing a loss function in a way that it is able to differentiate between the different types of inputs. Being able to incorporate both theme and local hints concurrently does not only make for a pleasant user experience, it also primes this method for recolorization of a previously known ground truth through the use of appropriately chosen cues.

2.2 Colorization for Compression and Cue Selection

Image colorization is not restricted to artistic purposes however. Once sufficient information about its color has been extracted from an image, this information together with the grayscale image provides already a compression effect. Combined with further compressing the grayscale image, a viable overall compression can be reached, as shown by e.g. Horiuchi and Tominaga [11]. They propose a conversion of the ground truth into CIELAB (L*a*b*) colorspace, followed by a grid-like structure for color cue selection from the non-luminance channels, and a discrete cosine transformation to encode the remaining luminance. With this method, they already achieved higher Peak-Signal to Noise Ratio (PSNR) values than JPEG.

While there have been numerous other attempts at utilizing colorization for compression, a prime candidate appears to be Xiao et al.'s method [1] described in 2.1, given the fact it supports simultaneous use of global and local themes, meaning it is not necessary to explicitly save a local cue for every color patch, only for those not already covered by the global colorization. In [2], they create a compression system based on their colorization method. They choose a set of theme colors by computing the 7 main colors of the image, using the K-means clustering algorithm. Local hints are determined by comparing the colorization after only using global cues and the ground truth. The resulting error map gets segmented and for the center of each segment, the corresponding color from the ground truth serves as a local input. This is the process we aim to enhance in section 3. While the authors used BPG to compress the remaining grayscale image, we chose to experiment with a novel GAN-based method in section 4.

2.3 Deep Compression

Image compression is a task that has been worked on for decades by countless researchers. In essence, it aims at minimizing the amount of data required to represent a given quantity of information. The degree to which image compression can be achieved depends on various factors, such as coding or spatial redundancy, and is typically performed in three consecutive steps: transformation, quantization and coding (e.g. JPEG[5]). Depending on the use and level of quantization, the input sequence can either be fully or partially restored. While lossless variants are of particular relevance if the original bitstream has to be recovered, lossy techniques sacrifice some bits of information in favor of a higher compression ratio. Recently, several attempts in building codecs with Deep Neural Nets have been published. Among them is the 2019 released work of Agustsson et al., in which the authors propose a framework for lossy compression within extraordinary low bitrates (< 0.1 bpp), made possible by utilizing GANs to synthesize parts of the image so that they are quite similar to the ground truth, without having to save much information about these parts. Their method runs an input image through several convolutional layers to form a small representation (encoding), followed by a quantizer. To decompress, a generator upsamples the quantized representation back to the original images dimensions. This generator relies on a combination of Mean Squared Error (MSE), Feature Matching and Learned Perceptual Image Patch Similarity (LPIPS) as loss function.

3 Variations on Color Cue Selection

The most important part of all local input driven colorization methods undoubtedly is the selection of the local points. Even with few points, a good colorization result can be achieved if they were picked in good regions. On the contrary, even more points could make the resulting colorization much worse, if they were chosen in bad regions. This would for example be the case, if a point lies exactly on the boundary region of two objects and the color from one propagates to both objects. A good example of this is shown in Figure 1.

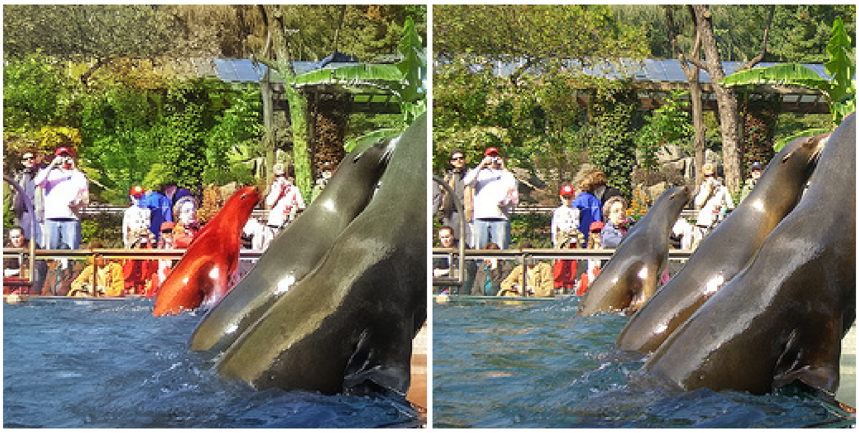


Figure 1: Left: Recolorized using a non ideal method (see 3.2). The red color of the shirt, next to it, propagated into the sea lion. Right: Ground Truth

However, it is quite challenging to find a method that works reliably on a wide range of images. If the colorization method only supports local inputs, the only reference to figure out if a point would contribute meaningfully to a colorization is the original image. Suitable for this task are superpixel segmentation methods, like SLIC [12]. These segment the RGB image into segments of similar color. The centroids of these images then become the color points as input for the colorization. But since the system developed by Xiao et al. [1] simultaneously uses a global theme and local inputs, a more advanced method can be used. In [2] Xiao et al. proposed a system where they chose the points depending on the output of the theme only colorization. This is advantageous, because easy-to-color-regions do not require a lot (or any) points. The sky, or basic vegetation for example can be recolorized quite well using the global theme only.

For this, an error map is being generated, by subtracting the theme recolorized version from the ground truth. This error map can be used in various ways to select local cues at pixels where the theme colorization alone was not able to reconstruct the original color. Some of these will be discussed in the sections below.

All following tests have been run on 1000 test images that the system was not exposed to during training. With the exception of LPIPS, the quality metrics were calculated on the predicted color channels of the Lab color space.

3.1 Segmentation

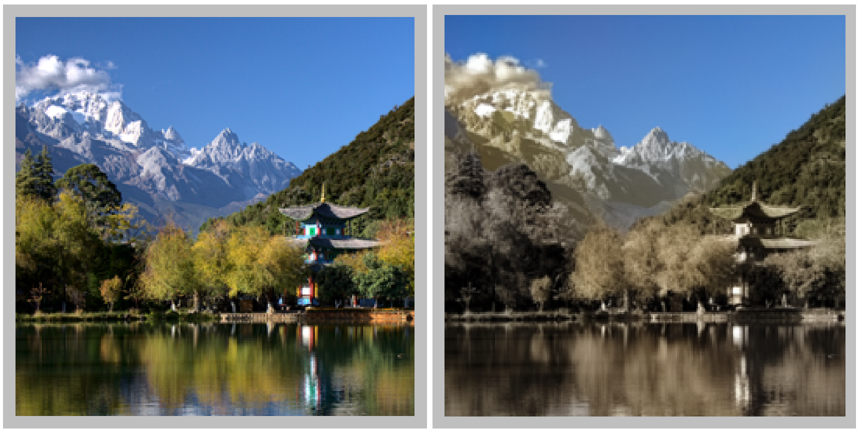


Figure 2: Images used to generate the error map. Left: Ground Truth. Right: Recolorized with theme.

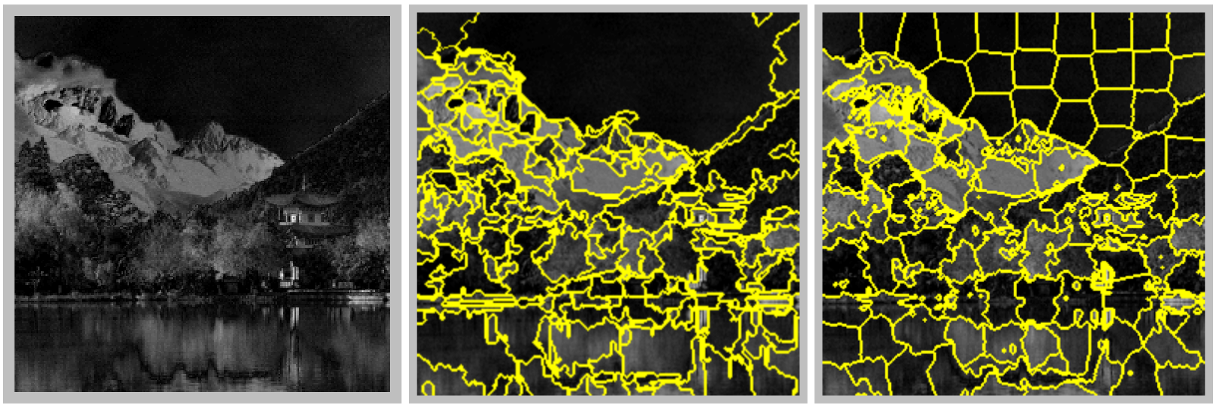


Figure 3: Left: Error map. Middle: Felzenszwalb, 204 segments. Right: SLIC, 100 Segments

In 2020, Xiao et al. [2] used the SLIC superpixel algorithm to segment the error map. Unfortunately, they didn't share details about their implementation, so an exact comparison with theirs is not possible. But we can still compare SLIC - with parameters chosen to the best of our knowledge - to the other methods.

As can be seen in Figure 3 on the right, SLIC still segments large regions of similar values into multiple similarly sized superpixels. For large regions of low error, this is not advantageous, since points are used on an area that is already colorized well. These could be used in other spots, that would benefit more of additional color information.

Therefore, a different segmentation method might perform better. For this we tested a segmentation method introduced by Felzenszwalb and Huttenlocher in [13]. We chose to test it because of its ability to create large segments in areas of similar color and because it adheres better to edges (Figure 3). Though, as the authors of the SLIC method noted in [12], it comes with two drawbacks. It is not possible to control the size of the segments, which can result in long and irregular shapes. These are not ideal to choose individual points for colorization. And it is not possible to control the exact number of segments, and therefore the number of local cues. To combat the latter one and to be able to test and compare it with methods of specific amounts of points, settings will be used that would result in approximately 200 points in most cases. Then, only the segments with the highest mean errors are used, as many as are desired.

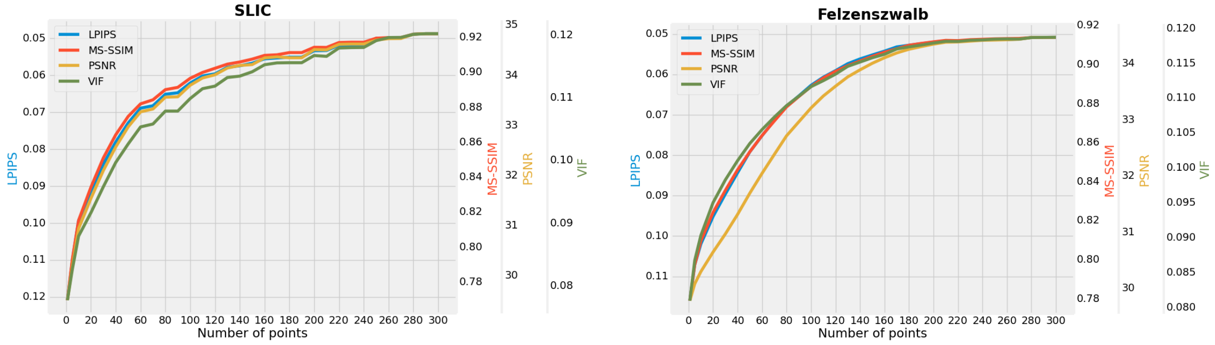


Figure 4: SLIC and Felzenszwalb segmentation methods, compared with different numbers of points.

Figure 4 shows the test results of the colorization with the SLIC and Felzenszwalb segmentation methods applied to our test set. To quantify the outcomes, we utilized PSNR and MS-SSIM (Multiscale Structural Similarity) as widely used and established methods. We also used LPIPS as a modern, state of the art method and VIF (Visual Information Fidelity) due to it aiming at modeling the human eyes visual perception[14]. The number of segments of Felzenszwalb could not be controlled and was fixed for each image. This is why the quality starts to converge towards 200 points. The segmentation method didn't supply more segments. This could have been prevented by calculating the parameters dynamically, depending on the desired number of points. But such a high number of points do not make sense for compression, due to their larger storage demand, but diminishing return in colorization quality.

For a low number of points (about 50), SLIC performs noticeably better than Felzenszwalb. This gap closes with more points at about 100. But Felzenszwalb does not significantly outperform SLIC. This could be different with other parameters for large numbers of points. But these are hardly relevant for compression, so SLIC is the better choice.

3.2 Subtraction of normal distribution

Even though SLIC can create tiny segments, when the `enforce_connectivity` parameter in the scikit-image implementation is disabled, it still often fails to catch small details.
So to achieve better detail reconstruction, we came up with a new method to select points.

Our proposed approach takes the error map as input, just as the segmentation methods. But instead of segmenting the entire map, we search for the point with the highest error and choose a color point at its position. But since the colorization process would spread the color into the surrounding area, another point in the close vicinity is likely not needed any more. With larger distance fades the influence of this

colorization. Therefore, to model this colorization effect, we subtract a bell curve from the error map around this point. (As shown in Figure 5, similar to a normal distribution). This bell curve starts with 1 in the center, to completely remove the error at the newly chosen location. Error points further out are still lowered, to reduce the chance of them getting picked again. Should there however be another point very high in error, it's chance of getting picked remains.
The following process gets repeated until the desired number of points is reached:

1. Search for highest value in error map
2. Subtract normal distribution from the error map around this point.
3. Save the color values from the original image at this point

A resulting error map is visualised in Figure 6. In this illustration, the lowest value was not capped to 0, to make the location of the chosen local cues (the centers of the black dots) more visible.
From now on, this method will be referred to as NorDist, for **Nor**mal **Dist**ribution Subtraction.

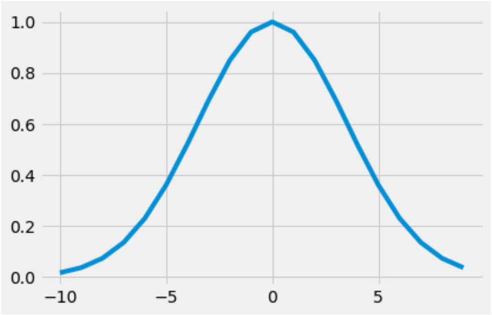


Figure 5: 2D slice from the center of the 3D bell curve, subtracted from the error map.

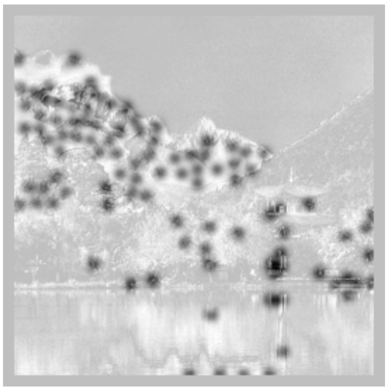


Figure 6: Error map, after choosing 100 points using the "Normal Distribution Subtraction"

3.2.1 Multiple passes

The NorDist method excels at reconstructing small details. But the price for that is the introduction of new, unwanted errors. Approaches relying on segmentation methods would segment the image along color patches and choose the points at the centroids. On the contrary, NorDist might choose a point directly on the edge of such a color patch, if the highest error happens to be there. This can result in gravely wrong colorizations, as seen in Figure 1. This is much less likely when using SLIC, since larger patches tend to get segmented into several regions as seen in 3 on the right. Therefore, lots of additional cues exist and correct potential errors during recolorization. On the other hand, this means that by utilizing SLIC, lots of barely useful local cues are being stored. Another disadvantage of NorDist is, that if a large part of the image is high in error, the points tend to bunch up in these regions, as seen in Figure 6.

So, to address these issues, we chose the points in multiple passes instead. We distribute the total number of desired points evenly across the number of passes. Then for each pass, we run the entire NorDist process and create an intermediate colorization with the newly chosen local cues. This intermediate colorization is used to calculate a new error map, used to choose the next set of points. That way, any introduced wrong colorizations will get fixed. And points will be better distributed throughout the image, since we don't have to completely rely on the guessed colorization effect of the bell curve.

The downside to this is the additional computation time, since for every pass the image has to be recolorized, whereas in the pure SLIC based method, it only has to be recolorized once. Since the recolorization is the most computationally intensive part of the whole process, this linearly increases the run-time with added passes.

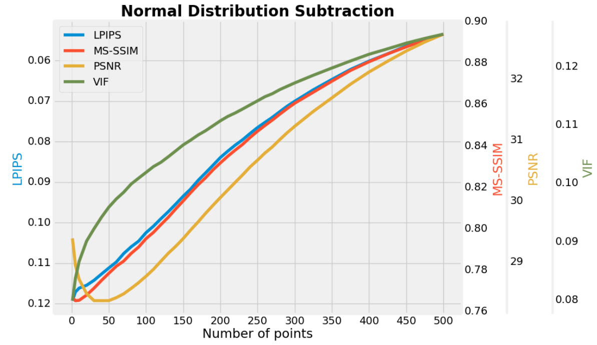


Figure 7: NorDist tested with different numbers of points

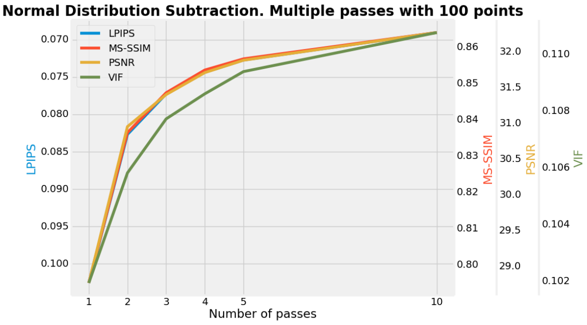


Figure 8: NorDist with a fixed number of 100 points tested over multiple colorization passes.

Figure 7 shows how unreliable the recolorization using the pure NorDist method is. The introduced color errors not only lead to worse results than those SLIC can deliver, they also lead to huge discrepancies between the quality metrics. Especially the PSNR shows a worse quality with fewer points than the theme only colorization achieved.

However, by choosing the points in multiple passes, as described above, the quality can be drastically improved, with the same number of points. Shown in Figure 8. This is a tremendous improvement, over just one pass, but it still does not reach the same level of quality as SLIC does, at least not with reasonable computation time.

3.3 Combination of SLIC and Normal Distribution

To achieve the best of both worlds, the consistently good colorization of SLIC and the superior detail reconstruction ability of NorDist, we decided to combine both methods. For that, the theme and SLIC colorizations serve as a first pass with half of the points. After that, the error map is being regenerated with the SLIC colorization. Based on that error map, the points are being chosen with NorDist. Here again, multiple passes of NorDist are advisable, to achieve the best result.

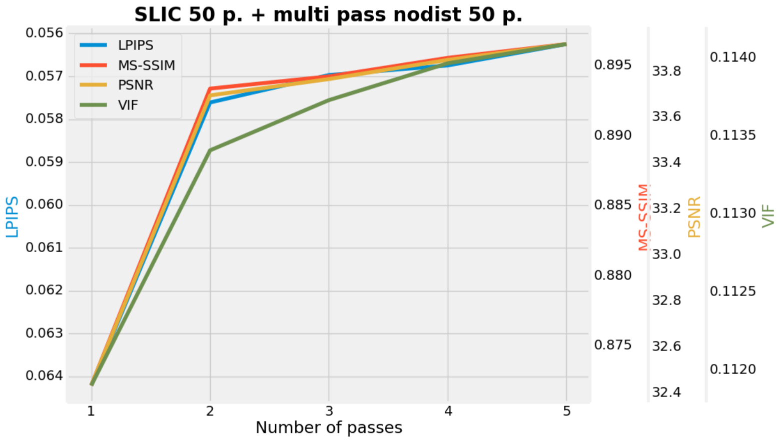


Figure 9: SLIC combined with NorDist at 100 points. Tested over different numbers of NorDist passes.

Figure 9 shows the improvements of multiple NorDist passes. The jump from just one to two passes presents the largest improvement. More could be added to improve the quality further, but it is questionable if the diminishing returns are worth the added computation time. With just one pass a little lower performance than pure SLIC is reached. However, at only 2 passes, this combined method actually outperforms SLIC. This is also better visualized in Figure 10.

3.4 Conclusion about Color Cue Selection

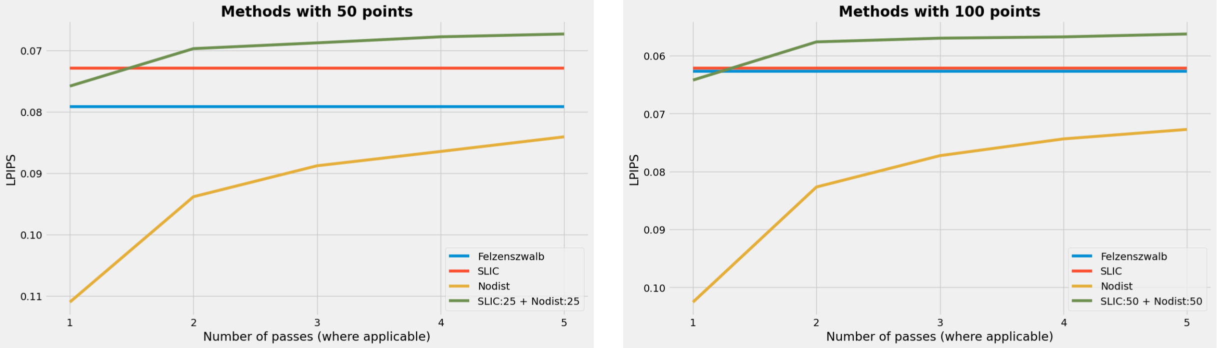


Figure 10: All point selection methods tested on 50 and 100 points and multiple passes. Only tested with LPIPS metric.

As shown above in Figure 10, all our introduced methods performed worse than SLIC, which was used by Xiao et al. [2]. Except for the combination of SLIC and the Subtraction of a Normal Distribution from the Error map (NorDist). SLIC + NorDist slightly outperformed pure SLIC. We tested it with 50 points, because that was the maximum number of points used by Xiao et al. [2] for their training process as well as quality tests. We also tested this with 100 local cues, which is roughly the number of points where diminishing returns in quality make it not worthwhile to use any more.
In terms of computation time, the heaviest computation in this pipeline is the actual colorization of the image itself, used to generate the error maps. Compared to that, all other operations are almost insignificant. The pure SLIC based colorization is therefore the fastest, as it requires one colorization for the encoding and one for the decoding. To achieve a better result with SLIC+NorDist, at least two passes for NorDist are required, which results in 3 colorizations for the encoding and one for the decoding process.

4 Experiments

In this section, we take a look at combining [2] with [4]. For details regarding implementation and training, see section 5. Our setup is as follows: We trained the GAN-compression twice under the exact same circumstances, once for colorful and once for grayscale images. (It should be noted, that the outputs of both compression nets had some weaknesses with several pictures from our test set, yet always achieved very similar results. For the purpose of our demonstration however, we still recognize the nets as valid and limit ourselves mainly to the successful compressions when looking for gained quality through the use of colorization.) We then compared the results of reconstruction after colorful GAN compression and colorization (based on local cues from SLIC+NorDist 2 passes & 100 points) with grayscale GAN compression. Firstly, we computed several distortion metrics between ground truth and both compression systems and averaged over 1000 test images, see table 1 for our results.

This already indicates that the GAN compression is barely enhanced by adding the colorization process, if not worsened. While in some scenarios, color gets represented more accurately through the addition of the colorization (e.g. the flower in Figure 11 at the bottom left or the wolves eyes), in some cases the accuracy of the color even declines (e.g. the flower the butterfly sits on). We assume the latter happened through a too dark luminance reconstruction from the grayscale compression, making the red color cue appear severely darker, too. Similarly, sometimes the GAN compression exeeds in structural appeal, sometimes the comined version. Overall, we come to the conclusion that the GAN-based framework probably allocates most of its disk space towards structure (which we confirmed by

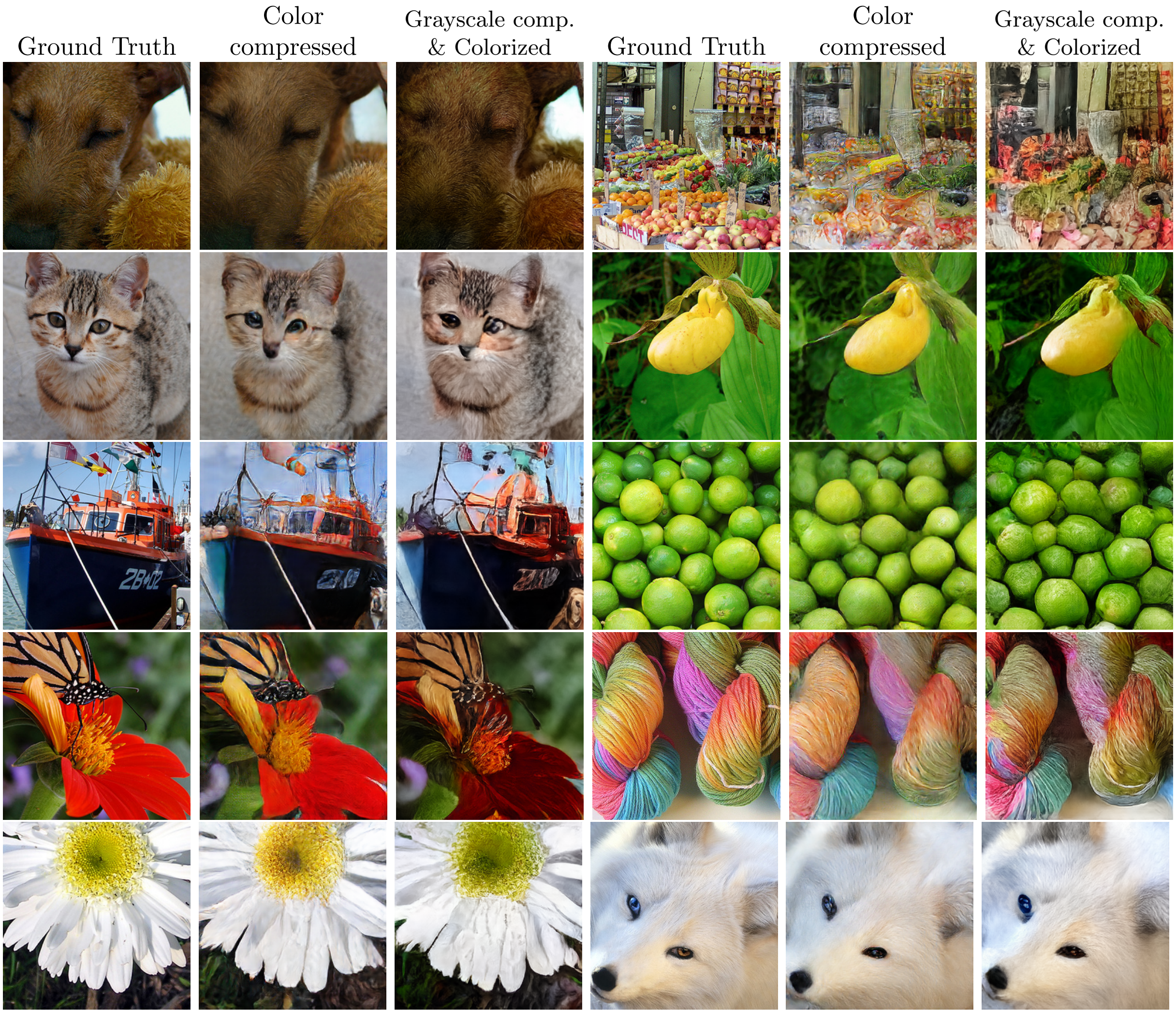


Figure 11: Compression Methods compared. Left: Ground Truth. Middle: Color Compression. Right: Grayscale Compression & SLIC+NorDist 2 passes, 100 points

looking at outputs from colorful compression vs. grayscle compression without the colorization) and is already very much capable of reconstructing color from very little information, making the combination with colorization rather obsolete.

5 Details of our Implementation

All code that we used, as well as trained models, are available on GitHub at github.com/Dakes/Interactive-Deep-Colorization-and-Compression For training of both, the colorization net and both compression nets, we used a subset of the ImageNet Dataset of 100,000 images. These were randomly cropped to 256x256 pixels.

5.1 Colorization

We used almost the same methods and parameters for training of the colorization net as Xiao et al. [2]. However, where our training method differed was the number of local points and theme colors. We used

	LPIPS	PSNR	MS-SSIM	VIF
Color Compressed (Agustsson et al.)	0.19	30.53	0.82	0.037
Gray Comp. & SLIC+NorDist 2p., 100 points	0.22	30.83	0.81	0.047

Table 1: Comparison between the compression methods. Agustsson et al. [4] and ours. For LPIPS a lower score is better.

a random number of points between [10, 100]. For the theme colors we used a fixed number of 6. The generated random points were reused in the first and second training step, but regenerated randomly for the third joint training step.

The colorization network was trained for 300,000 iterations. The residual network, which improves the colorization result, was trained for another 300,000 iterations. The third joint training step ran for 240,000 iterations. The learning rate started at 0.001 for the first and third training step and at 0.0001 for the second training step.

To segment the error maps with SLIC and Felzenszwalb's method, we used the popular scikit-image [15] python image processing library. The fixed parameters for SLIC were: `compactness=0.2, sigma=1, enforce_connectivity=False`. All changed parameters for Felzenszwalb were: `scale=50, sigma=1, min_size=50`.

The quality metrics were calculated using python and the Fast Image Quality Assessment library (FastQA) for MS-SSIM and VIF. For an implementation of FastQA, see `https://github.com/abhinaukumar/fast_qa`. Scikit-image was used for PSNR, while for LPIPS the lpips python package was used, see `https://github.com/richzhang/PerceptualSimilarity`.

5.2 Compression

Both the grayscale version and the colorful version of [4] have been trained on the same subset of 256x256 crops from ImageNet as the colorization network, with bitrates of 0.072 bpp. This bitrate arises through quantizing the encoders output onto 8 levels, as described in [4]. We decided on this, since changes to that parameter leading to lower bitrates resulted in unacceptable image reconstruction quality. We used a batch size of 32 and trained both nets over the course of 40 epochs in a "warm-up" phase, where only the encoder and generator losses were backpropagated, after which losses converged. The training then was followed up by a secondary stage, in which encoder, generator and discriminator losses jointly got optimized over 120 epochs. We employed a learning rate of 0,0002 and the AdamOptimizer, as suggested by [4]. In our choice of loss functions, we diverged from Agustsson et al.'s approach and exerted a combination of MSE with $\lambda = 10$, lpips with $\lambda = 10$ and L2 loss (squared error loss), excluding feature matching loss however. This decision is based on the authors follow up work, see [16] Note that lpips was indeed designed for images with three color channels, yet is still reasonably applicable for grayscale images, if the luminance channel is broadcasted onto 3 identical channels. This was confirmed by Richard Zhang, main author of LPIPS, in a discussion on a GitHub issue about using LPIPS on grayscale images, so we implemented it thusly. At the time of publishing, that conversation can be retraced here: `https://github.com/richzhang/PerceptualSimilarity/issues/23`

6 Conclusion

All in all, we believe to have accomplished a fine-tuning of Xiao et al.'s local cue selection in Colorization for Compression, as apparent in Figures 12 and 13, added in the Annex section. However, there are still more variations to be tried.

Like searching for the highest error in the error map within superpixels retrieved from either the ground truth or the reconstruction in progress. This has the advantage, that it would not mistakenly segment two distinct color patches into the same error segment, which could theoretically happen, when two adjacent different colors have similar error values. Another idea is to finely segment the original RGB image using SLIC and to use these segments as a guide for the subtraction from the error map, instead of a normal distribution. Overall, a better way to model the colorization effect, while choosing pixels, without actually doing a colorization would greatly improve the final quality and decrease computation time. It is also possible to experiment further with our introduced method, by changing the ratio of points between them. i.e. using 75 points for SLIC and 25 for NorDist.

In hindsight, it may have been a little overambitious to want to combine colorization with the GAN-based compression. Nevertheless, it was interesting to see the robustness of the colorization method. Also, the similar output-quality of the two compression nets trained in the same manner gave an opportunity to formulate the hypothesis of the GAN-Framework already implicitly highly prioritizing structure over color, making it an excellent compression framework yet bad candidate for enhancement through colorization. We look forward to combining colorization with other compression methods, especially those developed explicitly for grayscale images.

Acknowledgement: This work was supervised by Professor *Anderas Siebert*, from *University of Applied Sciences Landshut* and M.Sc. *Nikolai Koerber*, who shared with us his implementations of both the colorization network and compression network, which we could build upon. We thank our supervisors for supporting and advising us throughout this project. We also thank the University of Applied Sciences Landshut for giving us access to one of their servers with two NVIDIA A100 graphics cards, which tremendously helped us during the training processes.

Annex

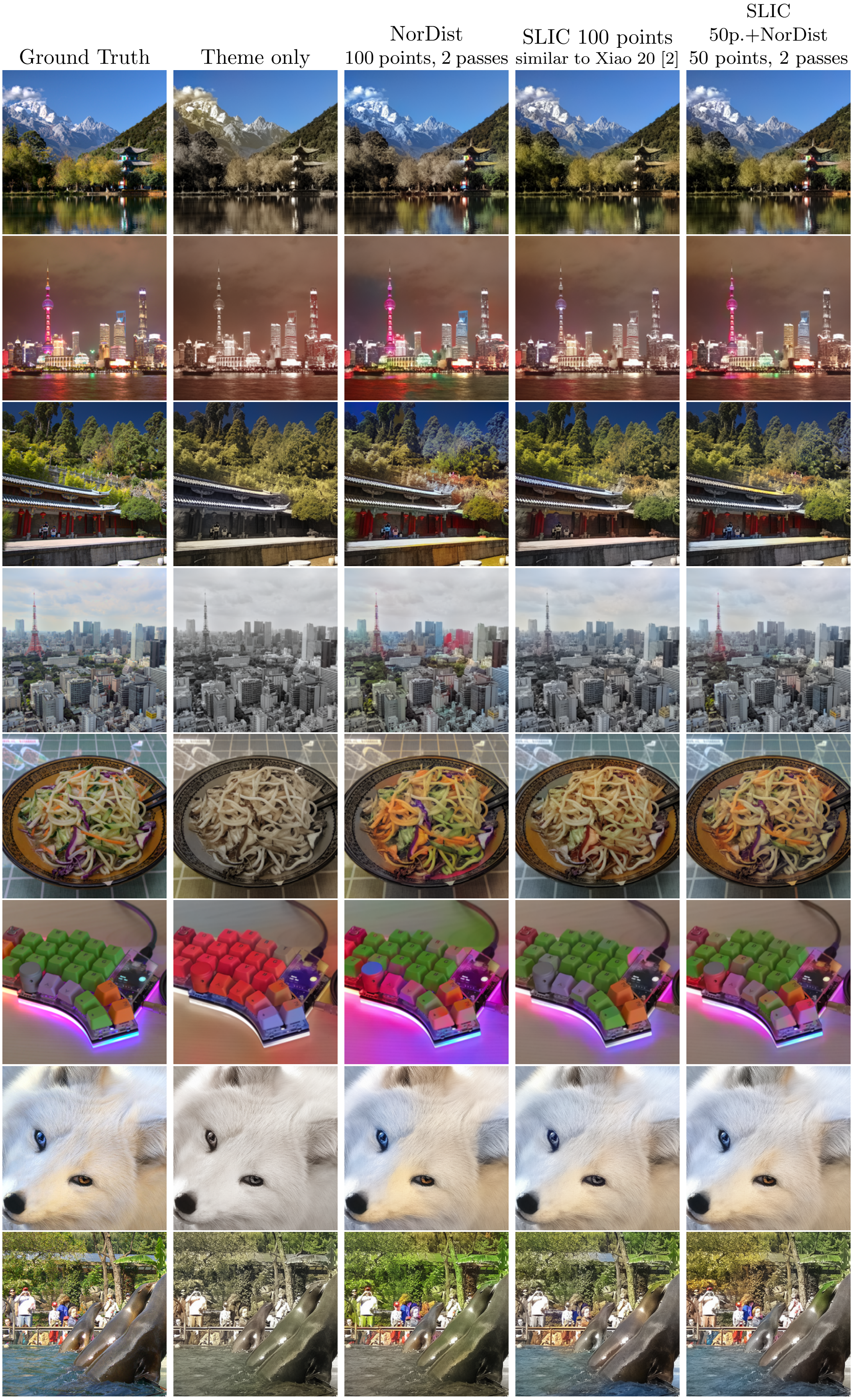


Figure 12: Comparison of different Colorization methods

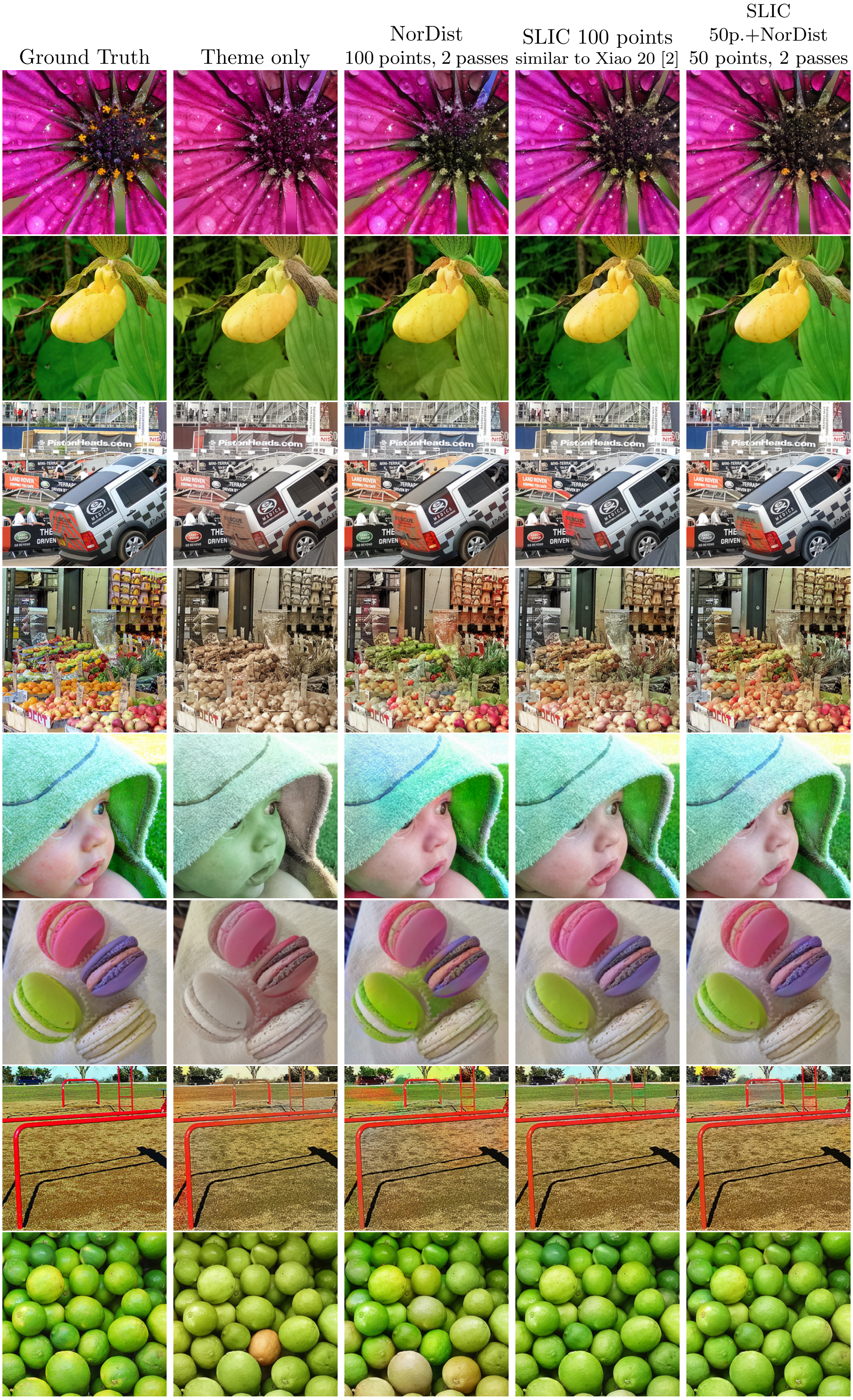


Figure 13: Comparison of different Colorization methods

References

[1] Yi Xiao et al. "*Interactive Deep Colorization Using Simultaneous Global and Local Inputs*". In: *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* 2019, pp. 1887–1891. DOI: 10.1109/ICASSP.2019.8683686.

[2] Yi Xiao et al. "*Interactive Deep Colorization and its Application for Image Compression*". In: *IEEE Transactions on Visualization and Computer Graphics* 28.3 (2022), pp. 1557–1572. DOI: 10.1109/TVCG.2020.3021510.

[3] F. Bellard. "*Bpg image format*". In: *Online reference, last accessed 2022-04-09, https://bellard.org/bpg/* (2017).

[4] Eirikur Agustsson et al. "*Generative Adversarial Networks for Extreme Learned Image Compression*". In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (2019), pp. 221–231. DOI: 10.1109/ICCV.2019.00031.

[5] Alain M. Leger, Takao Omachi, and Gregory K. Wallace. "*JPEG still picture compression algorithm*". In: *Optical Engineering* 30.7 (1991), pp. 947–954. DOI: 10.1117/12.55896. URL: https://doi.org/10.1117/12.55896.

[6] Tomihisa Welsh, Michael Ashikhmin, and Klaus Mueller. "*Transferring color to greyscale images*". In: *Conference on Computer Graphics and Interactive Techniques, 2002, pp. 277–280* (2002).

[7] Zezhou Cheng, Qingxiong Yang, and Bin Sheng. "*Deep Colorization*". In: *2015 IEEE International Conference on Computer Vision (ICCV).* 2015, pp. 415–423. DOI: 10.1109/ICCV.2015.55.

[8] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. "*Let there be Color!: Joint End-to-end Learning of Global and Local Image Priors for Automatic Image Colorization with Simultaneous Classification*". In: *SIGGRAPH 35, 4* (2016).

[9] Anat Levin, Dani Lischinski, and Yair Weiss. "*Colorization using optimization*". In: *SIGGRAPH, 2004, pp. 689–694* (2004).

[10] Richard Zhang, Jun-Yan Zhu, et al. "*Real-Time User-Guided Image Colorization with Learned Deep Priors*". In: *ACM Transactions on Graphics* (2017). DOI: 10.1145/3072959.3073703.

[11] Takahiko Horiuchi and Shoji Tominaga. "*Color Image Coding by Colorization Approach*". In: *EURASIP Journal on Image and Video Processing, Volume 2008, Article ID 158273* (2008). DOI: 10.1155/2008/158273.

[12] Radhakrishna Achanta et al. "*SLIC Superpixels Compared to State-of-the-Art Superpixel Methods*". In: *IEEE transactions on pattern analysis and machine intelligence* 34 (May 2012). DOI: 10.1109/TPAMI.2012.120.

[13] Pedro Felzenszwalb and Daniel Huttenlocher. "*Efficient Graph-Based Image Segmentation*". In: *International Journal of Computer Vision* 59 (Sept. 2004). DOI: 10.1023/B:VISI.0000022288.19776.77.

[14] Hamid Sheikh and Alan Bovik. "*Image information and visual quality*". In: *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society* 15 (Mar. 2006), pp. 430–44. DOI: 10.1109/TIP.2005.859378.

[15] Stéfan van der Walt et al. "*scikit-image: image processing in Python*". In: *PeerJ* 2 (June 2014), e453. ISSN: 2167-8359. DOI: 10.7717/peerj.453. URL: https://doi.org/10.7717/peerj.453.

[16] Fabian Mentzer et al. "*High-Fidelity Generative Image Compression*". In: *arXiv:2006.09965v3 [eess.IV]* (Oct. 2020). DOI: 10.48550/arXiv.2006.0996.

Daniel OSTERTAG
University of Applied Sciences Landshut
Fakultät Informatik
Am Lurzenhof 1, 84036 Landshut
GERMANY
E-mail: *s-doster@haw-landshut.de*

Daniel HABERMAYR
University of Applied Sciences Landshut
Fakultät Informatik
Am Lurzenhof 1, 84036 Landshut
GERMANY
E-mail: *s-dhaber@haw-landshut.de*

Sixth International Conference on Applied Informatics
Imagination, Creativity, Design, Development
ICDD 2022, May 5-7
Sibiu, Romania

Mobile Application Design for Acquiring Financial Skills

Daniela-Nicolia Pătruţ

Abstract

The COVID-19 pandemic brought a forced change of the educational system through a immediate transition to a full-online learning process. The aim of this paper is to provide an overview on how mobile applications have taken advantage of the pandemic and been integrated into the online education process. This analysis uses specific queries for the identification of relevant scientific papers indexed in Clarivate Analytics (ISI Web of Science – ISI WoS). Then, the focus is on reviewing publications based on experiments that show the need for people to access financial services, which is regarded to be one of the most pressing issues confronting communities worldwide sequel to the COVID-19 pandemic. The outcome provided by this research is a set of functionalities that can provide the theoretical basis in the financial domain for students and the appropriate mechanisms to keep them engaged through this learning process. The paper defines the use cases and for a mobile application that aims to create a more dynamic environment that will facilitate the educational process in the financial field for the students of the Politehnica University of Timișoara. Moreover, the paper presents the generated outcome obtained after the development process of the application showing a complete usage scenario of it.

Keywords: COVID-19, online educational platforms, mobile application, learning system, financial knowledge

1 Introduction

The COVID-19 pandemic had a massive impact on our daily activities and it has brought about years of change in the way the educational system works. Everything has changed due to the restrictions and forced lockdowns that the COVID-19 crisis led to. Students and teachers were forced to adapt immediately to online education and the migration to virtual platforms was mandatory [1].

This pandemic context and the students' need to develop financial skills can enhance the development of mobile applications for the financial domain. According to French et al. [2], these applications must keep the user engaged throughout the learning process and provide relevant information to help users "*to keep track of their income and expenses through regularly checking their current account*". The main result of the usage of mobile applications for this purpose is to create a financial behaviour for the users based on "*a greater sense of self-efficacy and a greater confidence in their ability to improve financial decision-making*" [2].

Many of the existing mobile applications for financial education are directed at children [3]. This shows how important it is to have financial skills nowadays. Everyone shall be able to create a personal monthly budget and manage his expenses. As stated by Crișan [4], it is very important to have access to a monthly expense chart and to a monthly cash flow graph, but it is also important to learn about the financial terminology.

This paper contains five sections. After an introduction related to the COVID-19 pandemic changes and the highlight the usage of mobile applications in the educational process related to the financial domain, this paper presents a statistic of current research trends related to the educational process. Section 3 concentrates on the need for financial education and the identification of relevant functionalities for a financial mobile application. Based on this analysis, Section 4 presents the use case definition for a mobile application designed for students from Politehnica University of Timisoara to gain/improve their financial skills. The upshot of the proposed application will be highlighted in the Section 5. The last section presents the conclusions of this study and the recommendations for future work.

2 Research context and related works

The quarantine that came after the global pandemic started, has changed many of our ordinary life processes. The tragedy has been staged across the world and the outbreak forced several universities to swiftly switch into remote mode and implement virtual learning. Even though things are slowly getting back to normal, some of the areas impacted by COVID-19 pandemic are still influenced by the radical changes taken in the last two years. The education industry underwent huge adjustment due to digitalization.

2.1 Research trends – statistics

Learning applications have become increasingly popular among students, and not only, for giving the chance to gain knowledge through an innovative and fun way. The researchers started to give more importance to the emergency learning concept and to the development of new educational online platforms [1]. These research trends can be visible through a scientometric analysis [5] of the recent publications indexed in Clarivate Analytics. This analysis considers the publications about online educational system and mobile as described by the usage of the following queries in Clarivate Analytics:

((TS=(online education OR educational system)) AND TI=(mobile application)) (1)

*((TS=(online education OR educational system)) AND TI=(mobile application))
AND TI=(COVID OR pandemic)* (2)

In order to retrieve information related to people’s perspective over financial knowledge in the last two years, the following query was applied in Clarivate Analytics:

(((TS=(financial education)) AND PY=(2020-2022))) (3)

The queries (1) and (2) described above has provided information about how mobile applications took advantage in educational system in and out of the COVID-19 period. Based on the number of resulting articles we can notice that there are few articles that have dealt with the subject during the pandemic (6), rather than non-pandemic period (188).

Yet, one of the most popular methods of implementing a successful and efficient swap between physical and online learning was introducing mobile learning applications. Technology has created a more dynamic and interactive environment nowadays. Studies show that, due to technology, students are increasingly attracted to autonomy and flexibility rather than austerity [6].

Best mobile learning applications are considered to be the educational mobile games. The studies indicates that games can play a significant role in facilitating student's learning process. The fact that they can learn and develop skills everywhere and anytime, may improve motivation and effectiveness. However, a study from last year shows that, learning through mobile games is not that efficient if students are not equipped with an appropriate learning strategy [7]. The theory is that when students are given the support they need while learning something new, they stand a better chance of using that knowledge independently. Integrated this strategy, called the "scaffolding strategy" in mobile learning games can significantly affect students' perceived enjoyment, usefulness and ease of use.

The results of applying the query (3) in ISI WoS has shown over 5 thousand of relevant publication in the last 3 years. The issue of financial education is addressed in studies all over the world and it has been a field of massive interest lately. Fig. 1 indicates a comparative statistic over the number of papers published in the last 4 years based on this subject. According to this, there is a serious increasing tendance over this time. The increase was exponential from 2019 to 2020 which shows the interest that people gained in the financial area since the COVID crisis started.

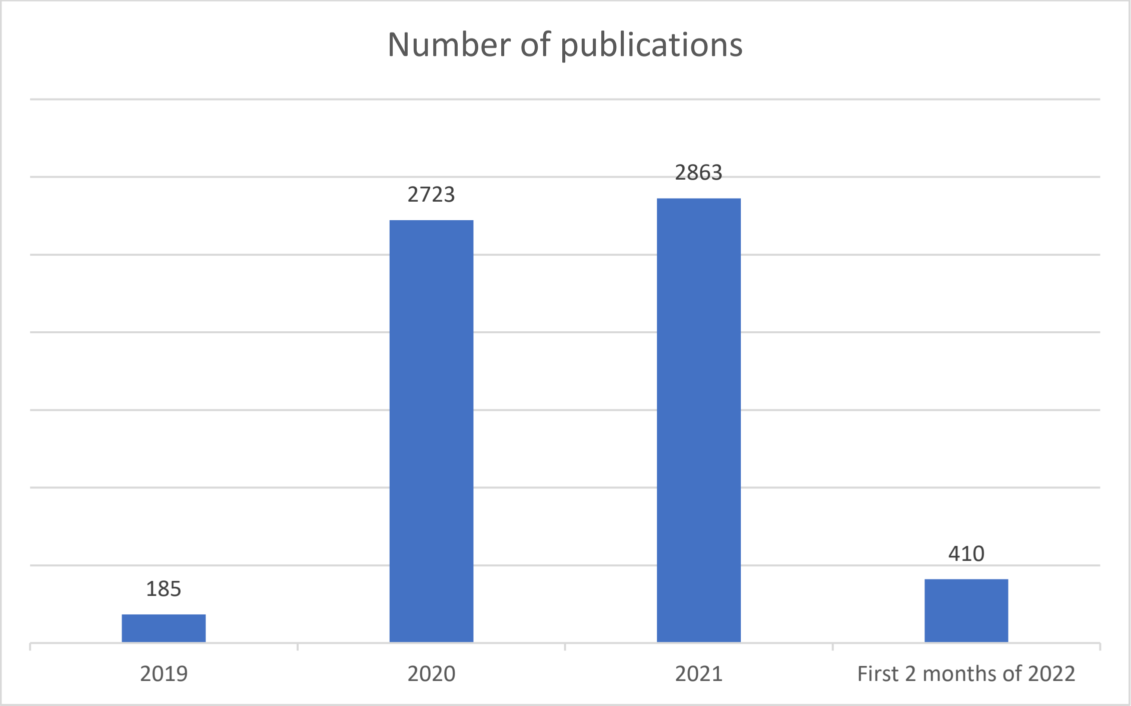


Fig. 1: Publication overview in the past 4 years

2.2 Related works

The popularity of mobile application has raised in the last years and their development impacts not only the educational branch, but also the social one. Technological services, tools and products must be aligned to the needs of people nowadays.

Learning apps have been developed for several years and the covered topic are diverse. One of the most used applications is Duolingo (Fig. 2), a language-learning platform. The application offers an effective and efficient way for gaining communication skills in 40 languages, based on fun challenges and games. Being able to tailor the lessons to the right level of the user, Duolingo has been used as part of the program of studies in two Spanish university courses [8].

Fig. 2: Duolingo

Fig. 3: Kahoot

The technologies used in applications' development are diverse and so are the challenges that can be encountered [10]. The challenge for many organizations can be on releasing application that works both native and on web browsers. The knowledge in programming languages, databases, Software Development Kits (SDKs) and Application Programming Interfaces (APIs) are required in the development of native applications especially [11]. Even though software organizations are following various development strategies and are facing different problems, the user's demands have to be served.

3 Financial education

3.1 Need for financial education

As we seen in the previous chapter, the financial domain has boomed in the last years. Mr. Dr. K. C. Chakrabarty, Deputy Governor, Reserve Bank of India said that "*Including financial education into the school curriculum in an on-going manner would hold the key to making our future generations financially literate*". The studies reviewed has shown that there is an urgent need of growing financial educational programs in schools and universities [12]. Financial knowledge is the key to a responsible money management for consumers in all segments of society.

A study made on students from states colleges and universities in the US, has demonstrate that most of them are facing diverse financial challenges and they don't know how to manage them [12]. Furthermore, the study indicates that the lack of basic financial knowledge a consequence of a lack of personal financial education.

However, the papers said that there are significant differences in the levels of financial satisfaction among various ethnic groups [13-14]. Instead, many of the students seems to be interested in learning about saving, investing, loans or credit cards. Because of this, schools and universities have started to come out with courses that incorporate basic financial notions.

We expect that this tendency of integrated financial literacy courses in universities and colleges will keep growing in the following years in order to fulfill a general financial education requirement among the students.

3.2 Relevant functionalities for a financial mobile application

Considering the review made upon significant publications resulting from the queries mentioned in the second section, we can identify some relevant functionalities that can be integrated in a financial mobile application for students. Designing this kind of application can be very challenging but also a helpful initiative.

Therefore, financial education can easily be an appropriate concept and a useful topic for developing a mobile learning game. Following the above review, there are many chances for students to be interested in educating themselves on a such a needed field nowadays. The essential functionalities required are the following:

- Creating a personal account on the application [15];
- Having a financial literacy curriculum that meets the needs of students and provides eloquent guidance, utility, quality and efficacy, in accordance with country standards [12,16];
- Having a well defined scenario that helps students build a strong and consistent knowledge, based on the dynamic and modular learning strategy [7];
- Integration of self-assessment tests [17-18].

4 Use case definition

Considering the information extracted in this review about online education, mobile applications and financial knowledge, we designed the use case diagram for a financial mobile application (Fig. 4).

Once registered and authenticated, the user will be able to find a various documentation based on different financial subjects. He can also search for a certain topic related to his interest. Other relevant functionalities refer to the possibility of testing the knowledge accumulated after completing a specific module. The practicability of the application is due to the functionalities that allow the user to redo a test at any time he wants. Users can also download the module content, save to favourites, display the test results, edit and update their profiles and log out from the application.

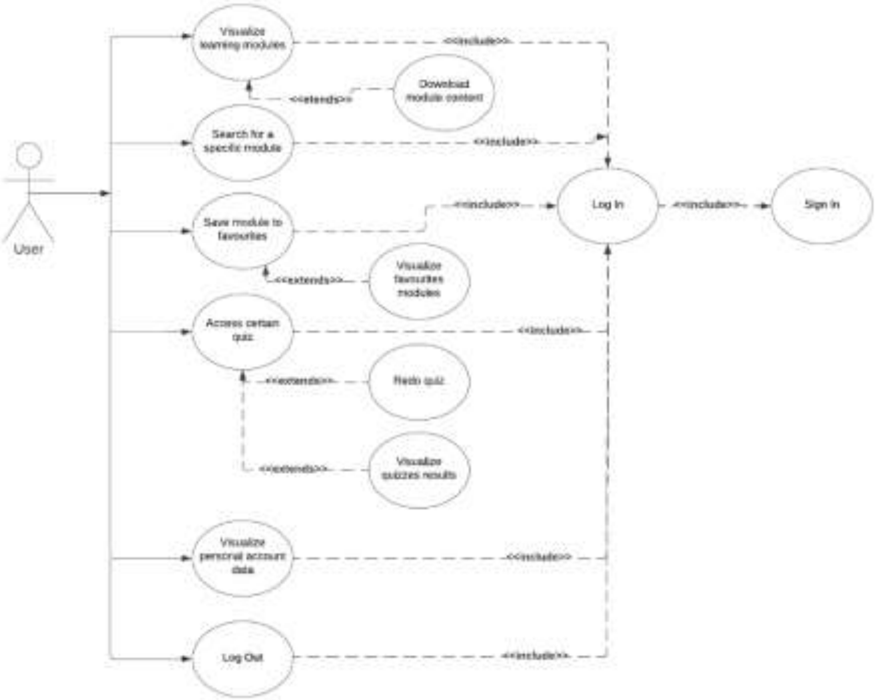


Fig. 4: Use case diagram

5 Results and discussions

Based on the above use cases, this section provides the results of the implemented financial mobile application. The application has been configured to be tested on the Android operating system. The experimental results will be illustrated in the below images and will represent a complete scenario of a user experience through the app. Sign Up and Sign In functionalities are shown in Fig. 5 and Fig 6. When creating the account, the user must complete the registration form, providing a valid username, email and password. If various errors occur, the user will not be able to register or log in to the application and will be alerted by an error message.

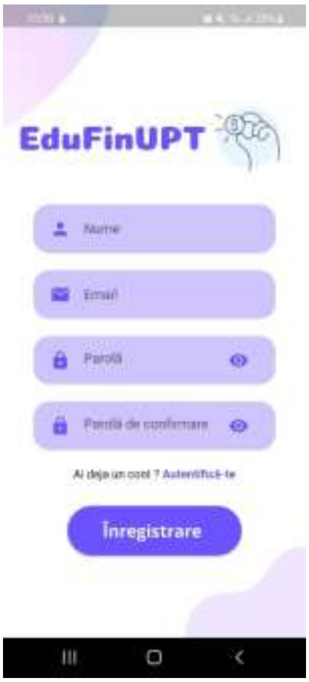


Fig. 5: Înregistrare (eng. Sign Up)

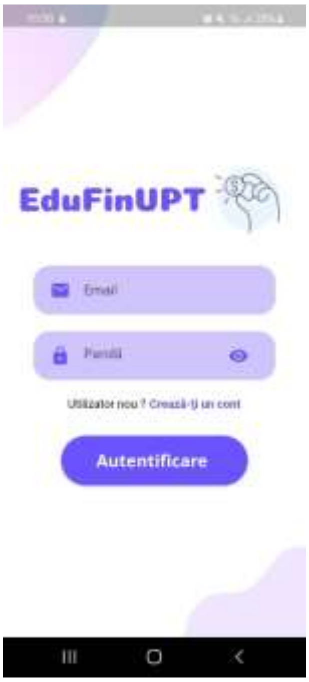
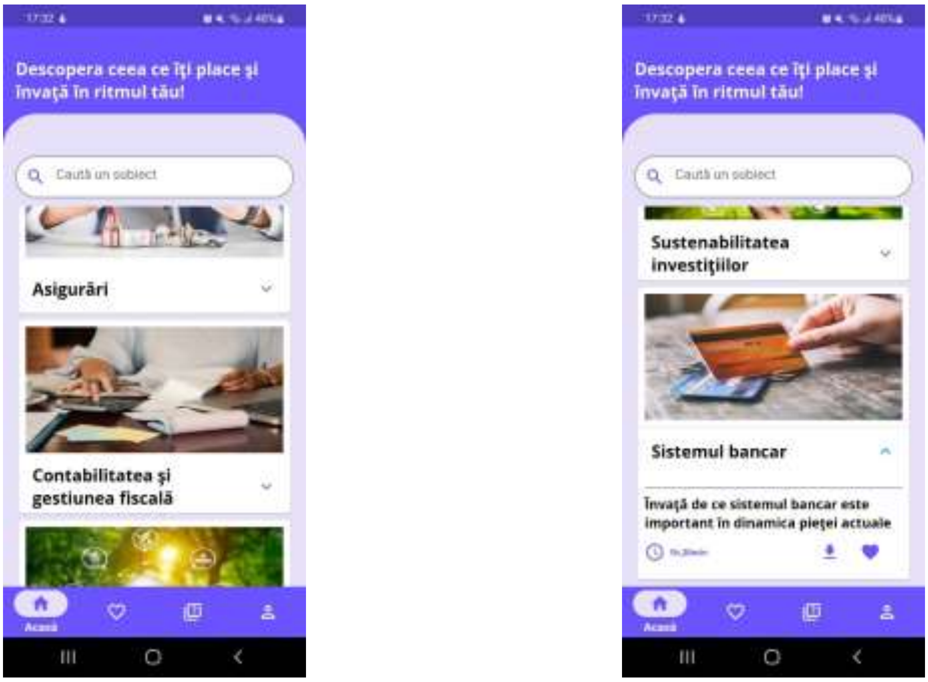


Fig. 6: Autentificare (eng. Log In)

After a successful authentication in the application, the user will be directed to the main page, where all the existing learning modules in the application are visible, as in Fig. 7 (a). The user will be able to extend the card for each module and see some additional details, such as an estimated time for completing the module, an additional description of the module and two buttons through which the user can either download the theoretical content of the module in his own phone or save it in the favourites list (Fig. 7 (b)). The download button only becomes active when the user has opened the theoretical content of the module within the application at least once. The module will be downloaded in the user's phone internal storage as in Fig. 8. Each module has a suggestive title and a representative image. Opening one of the modules will lead to its details page, where the user will be able to read supplementary information about the module and open its entire theoretical content (Fig. 9).



Descopera ceea ce îţi place şi învaţă în ritmul tău!

Caută un subiect

Asigurări

Contabilitatea şi gestiunea fiscală

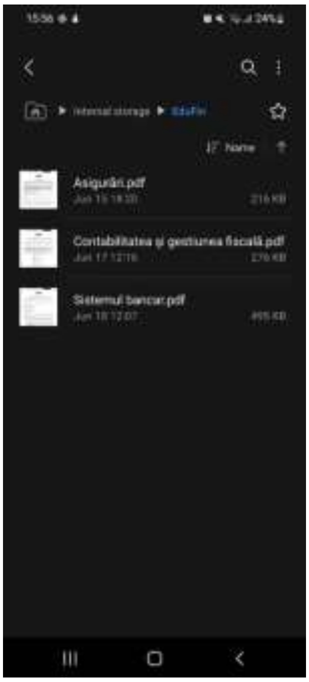
Sustenabilitatea investiţiilor

Sistemul bancar

Învaţă de ce sistemul bancar este important în dinamica pieţei actuale

(a) (b)

Fig. 7: Application interface: (a) main page; (b) module's options



Internal storage ▸ EduFin

Name

Asigurări.pdf
Jun 15 18:30 216 KB

Contabilitatea şi gestiunea fiscală.pdf
Jun 17 12:16 276 KB

Sistemul bancar.pdf
Jun 18 12:07 495 KB

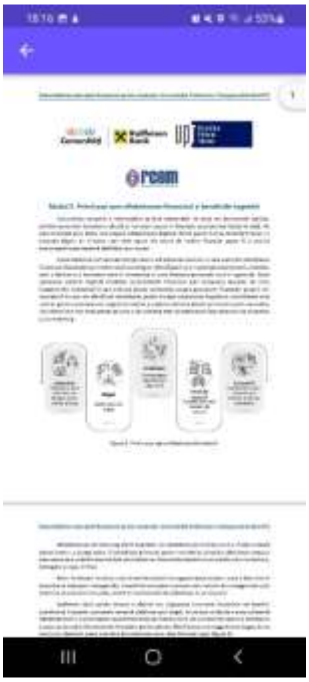
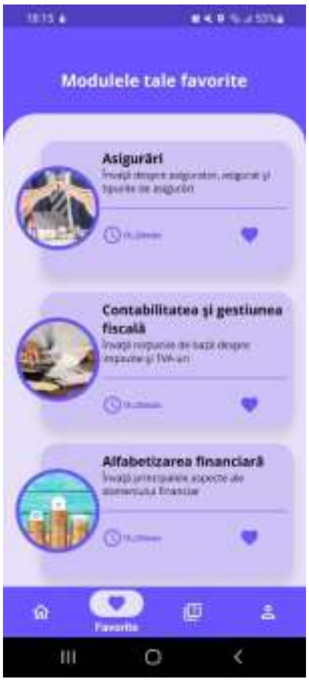


Fig. 8: Downloaded modules

Fig. 9: Theoretical content

When saving a module to favourites, it will be added in the personal favourites list as seen in Fig. 10. The user can also search for a specific subject by introducing a keyword in the search bar. The application will provide only those modules based on the subject of interest (Fig. 11).



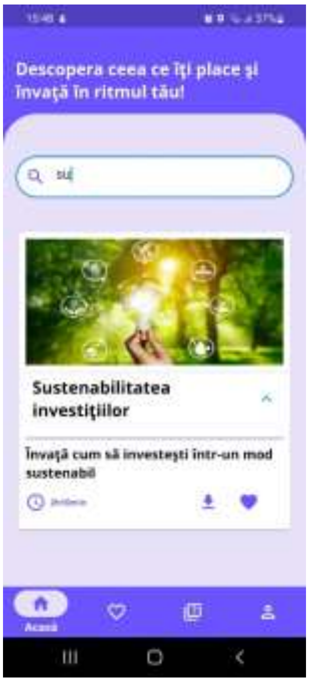
Modulele tale favorite

Asigurări
Învață despre asigurători, asigurați și tipurile de asigurări

Contabilitatea și gestiunea fiscală
Învață noțiunile de bază despre impozite și TVA-uri

Alfabetizarea financiară
Învață principalele aspecte ale domeniului financiar

Fig. 10: Favourite list



Descopera ceea ce îți place și învață în ritmul tău!

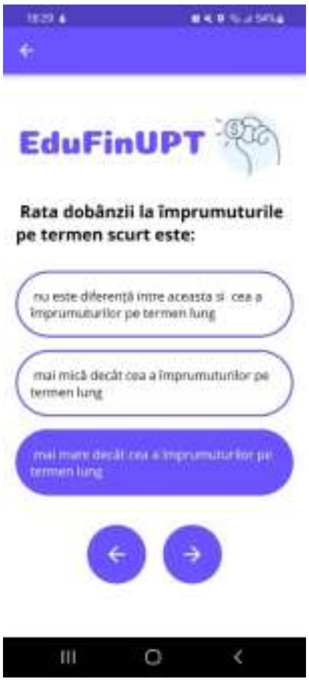
sul

Sustenabilitatea investițiilor

Învață cum să investești într-un mod sustenabil

Fig. 11: Search bar

The assessment test opens when the user presses the "Start Test" button, from the details module page. Following this action, the quiz will be generated automatically, and all test questions will be displayed in a random order (Fig. 12). The score obtained after taking the assessment test will be displayed after its completion. In addition to the current score, the user will be able to see also his best score gained at that quiz. All the taken quizzes will appear in the specific page from the menu, with the possibility of redoing it in case the user has never obtained a maximum score on that quiz (Fig. 13).



EduFinUPT

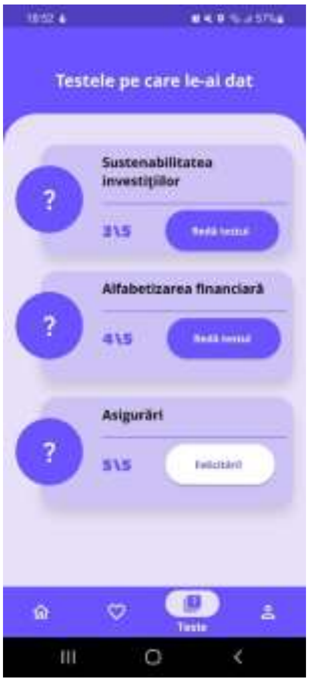
Rata dobânzii la împrumuturile pe termen scurt este:

nu este diferență între aceasta și cea a împrumuturilor pe termen lung

mai mică decât cea a împrumuturilor pe termen lung

mai mare decât cea a împrumuturilor pe termen lung

Fig 12. Quiz



Testele pe care le-ai dat

Sustenabilitatea investițiilor
3\5 Redă testul

Alfabetizarea financiară
4\5 Redă testul

Asigurări
5\5 Felicitări!

Fig. 13 Quiz results

6 Conclusions and future work

The presented paper analyses the trend in the development of mobile applications from the beginning of the COVID-19 crisis to the present in order to see how they have been used and integrated in the educational system. Moreover, the study examines the need of introducing financial education courses in schools and universities.

After reviewing the publications about introducing mobile applications in the online educational system and about the financial knowledge of today's youth, the study presents relevant functionalities for a financial mobile application, which are detailed in the corresponding sections. The result consists in designing a diagram with the most significant features and use cases for the further application that has been implemented and presented afterwards.

The application is a tool through which the transition to a digitalized educational regime can be made in favour of students. The target users are students of the Politehnica University of Timișoara, but it can also be used by anyone who wants to gain financial knowledge. We have created a dynamic context to stimulate the students' interest in accumulating new knowledge, related to a high impact field, namely, the financial field. This application offers the possibility of testing financial capabilities and also learning beyond the existing contexts in classical education system, therefore, learning and measure knowledge in a mobile environment, in which students can use their mobile phones as a method of comfort. The possibility of individual management of their free time for knowledge development can be an advantage.

Future work involves extended the informational base by adding more learning modules. Also, other future developments extend the idea of integrating new features like personalized and adaptive learning, using Artificial Intelligence (AI) concepts.

Acknowledgement: This work was supervised by PhD Assistant Professor *Mădălin-Dorin Pop*, from *Politehnica University of Timișoara, Computer and Information Technology Department (ROMANIA)* and PhD Professor *Larisa Ivaşcu,* from *Politehnica University of Timișoara, Management Department (ROMANIA)*. The application was developed within the project "*Improving financial education for students of the Politehnica University of Timişoara (EduFinUPT)*", financed under the "*Raiffeisen Communities 2021*" call.

References

[1] M. Pop, S. Potra and A. Pugna, *"ROMANIAN STUDENTS EXPECTATIONS FROM EDUCATIONAL ONLINE PLATFORMS IN THE AGE OF COVID-19,"* Online Conference, p. 10372–10381, 2021.

[2] D. French, D. McKillop and E. Stewart*, "The effectiveness of smartphone apps in improving financial capability," The European Journal of Finance*, vol. 26, no. 4-5, p. 302–318, 2020.

[3] Kiplinger, *"My 10 Best Financial Literacy Apps for Kids*," [Online]. Available: https://www.kiplinger.com/article/saving/t065-c032-s014-my-10-best-financial-literacy-apps-for-kids.html. [Accessed 27 03 2022].

[4] G. Crisan, "*Bank connect: How to build a fintech app*," Proceedings of the International Conference on Applied Informatics Imagination, Creativity, Design, Development, pp. 62-71, 2019.

[5] P. Chellappandi and C. Vijayakumar, "*Bibliometrics, Scientometrics, Webometrics/ Cybermetrics, Informetrics and Altmetrics - An Emerging Field in Library and Information Science Research*," 2018.

[6] D. Johnstone and P. Marcucci, *"Worldwide trends in higher education finance: Cost-sharing, student loans, and the support of academic research," Commissioned paper V. Lynn Meek and Dianne Davies,* vol. 81, pp. 1-36, 2007.

[7] A. Al Mulhem and M. Almaiah, *"A Conceptual Model to Investigate the Role of Mobile Game Applications in Education during the COVID-19 Pandemic," Electronics*, vol. 10, no. 17, 2021.

[8] P. Munday, *"The case for using DUOLINGO as part of the language classroom experience,"* RIED: revista iberoamericana de educación a distancia, vol. 19, no. 1, pp. 83-101, 2016.

[9] A. Wang and R. Tahir, *"The effect of using Kahoot! for learning – A literature review," Computers & Education*, vol. 149, 2020.

[10] A. Ahmad, K. Li, C. Feng, S. Asim, A. Yousif and S. Ge, *"An empirical study of investigating mobile applications development challenges," IEEE Access*, vol. 6, pp. 17711-17728, 2018.

[11] M. Joorabchi, A. Mesbah and P. Kruchten, *"Real challenges in mobile app development," Proc. ACM/IEEE Int. Symp. Empirical Softw. Eng. Meas*., pp. 15-24, 2013.

[12] A. R. Woolsey, *"An analysis of first-year freshmen financial literacy and the effectiveness of an online financial education program at small four-year private universities,"* University of La Verne, 2011.

[13] C. Yan, A. Siddik, N. Akter and Q. Dong, *"Factors influencing the adoption intention of using mobile financial service during the COVID-19 pandemic: the role of FinTech," Environmental Science and Pollution Research*, pp. 1-19, 2021.

[14] S. Crain, *"Are universities improving student financial literacy? A study of general education curriculum," Journal of Financial Education*, pp. 1-18, 2013.

[15] S. Kaware, M. Gupta and A. Gupta, *"Mobile Phone Educational Applications: Their Importance in Academic Learning During Covid 19 Pandemic," International Journal of Early Childhood Special Education*, vol. 13, no. 2, 2021.

[16] A. Y. F. Zhu, *"Impact of financial education on adolescent financial capability: Evidence from a pilot randomized experiment," Child Indicators Research*, vol. 13, no. 4, pp. 1371-1386, 2020.

[17] R. Herman, A. Hung, J. Burke, K. Carman, N. Clancy, J. Kaufman and K. Wilson, *"Development of a K-12 financial education curriculum assessment rubric,"* Rand Corporation., 2015.

[18] L. Chalacan, E. Arias, L. Zambrano, A. Paredes and A. Vazquez, *"Mobile applications to improve didactics in times of covid, in technical high schools, from the municipal school que-vedo*, 2020," Revista Conrado, vol. 17, pp. 105-110, 2021.

Daniela-Nicolia PĂTRUȚ
Politehnica University of Timișoara
Faculty of Automation and Computing
Bld. V. Pârvan, No. 2, 300223 Timișoara
ROMANIA
E-mail: daniela.patrut@student.upt.ro

Sixth International Conference on Applied Informatics
Imagination, Creativity, Design, Development
ICDD 2022, May 5-7
Sibiu, Romania

Learning by Playing Video Games

Valentin Rusanda, Adrian Gheorghiță Neagu, Ștefania Adriana Podeanu

Abstract

This paper describes the efficiency of learning real-life concepts by playing simple educational video games. It highlights this concept through a game about electrical safety that aims to show how easy it is to comprehend new information when we are actively involved in experiencing it. The game puts you in a scenario in which you must eliminate the electrical hazards as fast as possible, which adds a sense of urgency and helps cement the information for the player.

1 Introduction

Education in today's day and age is becoming increasingly more versatile as technology advances. As a result of this evolution, entertainment is taking the main role in the life of many people, especially the younger generations. Smartphones make it easy for anyone to access and distribute any kind of material at the press of a button. Because of this, a high percentage of people have become dependent on technology in their daily lives. According to a recent survey [1], Americans check their phones 344 times per day, which would average out to once every 4 minutes. As a result, it is only natural for us to adapt our learning methods according to the trends of technology usage today.

Video games represent a great source of entertainment for many people today. According to Newzoo [2], there were 2.69 billion gamers in the world by the end of 2020 and the numbers are expected to surpass the three-billion mark by the end of 2023. Using video games as a tool for education represents an opportunity for us to expand in a different way than traditional learning. Dr. Mark Griffiths suggests that playing computer games reduces reaction times, improves hand-eye coordination, and has many more benefits [3].

This paper uses a game about electrical safety to highlight the benefits of an interactive environment for learning about certain concepts. It covers the functionality of the game, the technologies behind it, and how it could be used for a better learning experience. As we want to reach as many people as possible with the game, it has been created as an HTML5 game that can be played on both PC and mobile. Paper [4] shows a study related to electrical injuries which states that there are around 1000 deaths per year as a result of electrical injuries and at least 30,000 shock incidents that are non-fatal, all of these in the United States alone. It also states that approximately 20% of these injuries occur in children.

2 Related Work

In this section, we will present other works related to the use of video games in education. There has been a huge number of studies done on this matter (e.g. [3,11,12]) and they tend to lean towards video games being a great tool for education.

For example, paper [5] suggests that there are two major types of games called "problem games" and "world games". It states that problem games focus on solving a problem or a set of problems, while world games focus on simulating a bigger world in which the player can solve various types of problems. The paper also states that games can have elements of both categories, so the distinction between them is not necessarily air-tight. The author of the paper compiled a list of features found in video games that are closely related to learning and that help develop certain abilities: interactivity, customization, strong identities, well-ordered problems, games are pleasantly frustrating, games are built around the cycle of expertise, and "deep" and "fair". Out of all these features, Electrical Safety has a combination of interactivity, minimal customization (custom player name), well-ordered problems, and a hint of pleasantly frustrating that comes from the high score system, as players often tried to beat the highest score on the leaderboard. Moreover, we think our game sits in the middle of the "problem game" or "world game" distinction because it contains various types of problems the player can solve that are hidden throughout a small world, in our game's case a house full of electrical hazards.

Another example of a study that supports video games as a tool for learning is paper [6], which suggests that video games boast many benefits for the mental health of the players and also help in the development of social skills. The paper categorizes the different genres of games and where they are situated through a graph that ranges from non-social to social and from simple to complex. They also suggest that video games can not only help with social skills, but also take advantage of their rewarding nature to boost motivation in the players and help with emotional stability.

To our knowledge, educational video games related to electrical safety have not been created before, so in this aspect, our game seems to be unique.

3 Game Summary

The game was created in Godot Engine (Fig.1), a powerful open-source game engine with flexible and easy-to-use tools and its own programming language called GDScript, which is similar to Python. Godot's scene hierarchy is organized as a tree of nodes, which are the engine's equivalent to game objects with specialized functions.

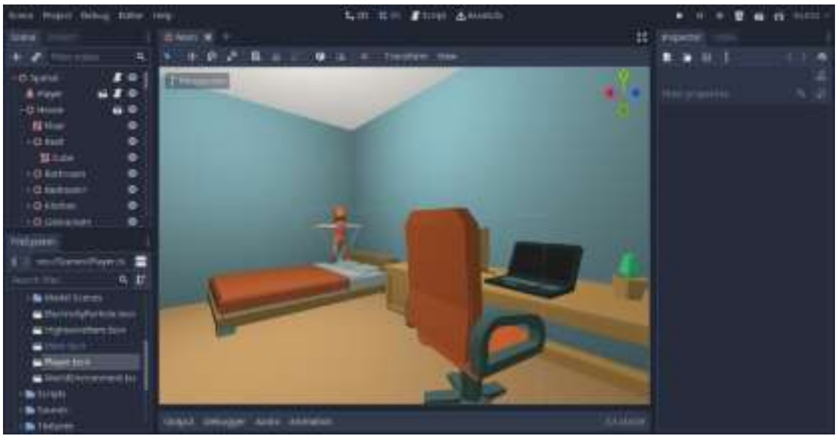


Fig. 1: Godot Engine editor

3.1 Gameplay

The main goal of the game is to find and eliminate the electrical hazards that are lurking around a house. The player is controlled from a first-person perspective by using the WASD keys and the mouse on PC, or by using the touch controls on mobile. The player can move freely around the house and find the hazards at their own pace. Once a hazard is found, a prompt will show up that will describe the hazard and the solution (Fig.2). To eliminate the hazard, the player has to point the camera at it and press the interact button, which will bring up a progress bar that fills up as the hazard is being taken care of. As soon as all the hazards are found or the game goes on for more than three minutes, the game will calculate the player's score based on the time it took them to reach the end and the number of hazards they have found, then submit it to an online leaderboard.

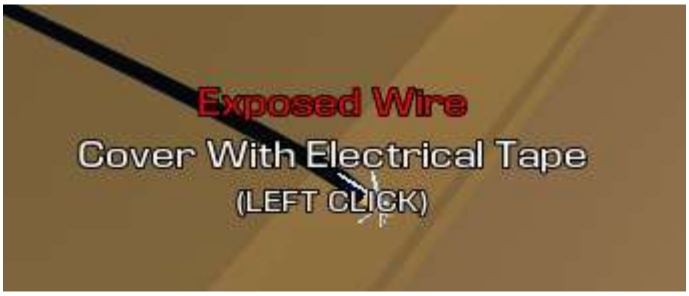
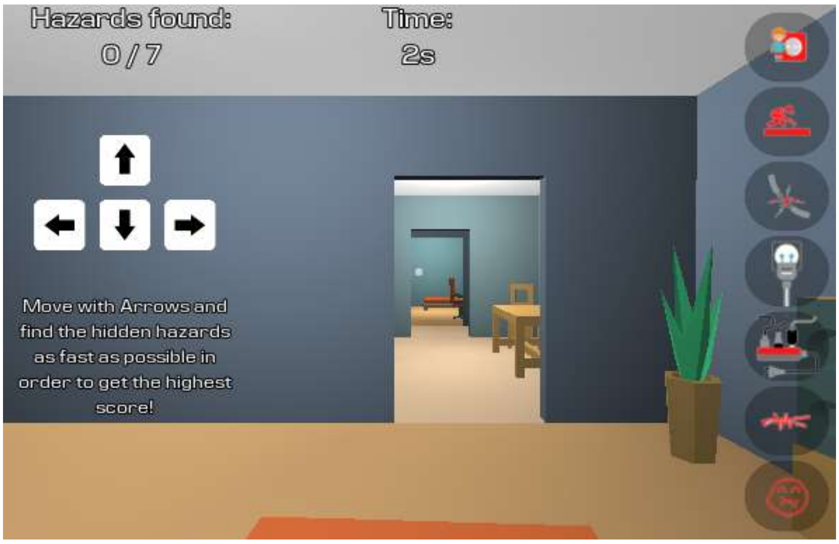

Exposed Wire
Cover With Electrical Tape
(LEFT CLICK)

Fig. 2: Hazard prompt example

3.2 User Interface

The user interface consists of simple elements. There is an introduction message at the start of the game, a form asking you the player for a name, gameplay instructions, a counter for the hazards that have been found, a time counter, and logos for each hazard (Fig.3).

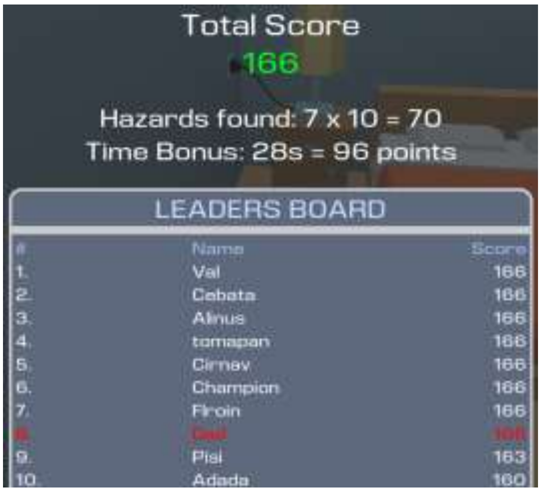

Hazards found:
0 / 7

Time:
2s

Move with Arrows and find the hidden hazards as fast as possible in order to get the highest score!

Fig. 3: User Interface

The interface is easy to understand and provides proper feedback as the player progresses through the game. Every time a hazard is eliminated, the logo corresponding to it will change color and a "+1" text prompt will float towards the "Hazards Found" counter. There is also a leaderboard that will show at the end of the game documenting the highest scores players have achieved (Fig.4).



Total Score

166

Hazards found: 7 x 10 = 70

Time Bonus: 28s = 96 points

LEADERS BOARD

#	Name	Score
1.	Val	166
2.	Cebata	166
3.	Alinus	166
4.	tomapan	166
5.	Cirnav	166
6.	Champion	166
7.	Firoin	166
8.	[illegible]	[illegible]
9.	Pisi	163
10.	Adada	160

Fig. 4: Leaderboard

3.3 Graphics and Performance

As an HTML5 game, Electrical Safety was created to perform efficiently on both mobile and PC. The game uses simple, low-poly models to represent the environment and various hazards around the house and pre-baked lighting to optimize the performance for mobile, as well as older computers. The scripts have been optimized for maximum performance and the shaders used are lightweight and simple.

4 Code Summary

4.1 Engine

The code was written entirely in GDScript, Godot Engine's programming language using the built-in script editor. The engine has many useful features, but one to note here is the "signals" system [7]. Signals work by connecting nodes to one another and calling a function when the signal is emitted. The main advantage of this system is that one or more objects can react to another object's changes without referencing one another. This system is used in many features throughout the game and is a convenient and clean way of writing code.

4.2 Player

The player is represented as a KinematicBody, which is a 3D node in Godot that is mainly used for simulated motion [8]. The movement is handled using multiple Vector3 structures [9]. The input is stored in a Vector3 depending on which key is being pressed and then multiplied to the forward direction of the camera and stored in another Vector3 called direction. This direction is then added to another Vector3 called velocity and multiplied by the speed of the player, after which we apply friction. After calculating the velocity, we then use the KinematicBody function "move_and_slide" with the calculated velocity to modify the player's position and check for collisions.

```
direction += cam_transform.basis.z * input_movement.z
direction += cam_transform.basis.x * input_movement.x
velocity.x += direction.x * speed - FRICTION * velocity.x
velocity.z += direction.z * speed - FRICTION * velocity.z
```

The camera rotation is handled by detecting mouse or touchscreen drag movement and rotating the camera's X and Y axes by the relative position of the mouse.

```
rotate_x(deg2rad(event.relative.y * -sensitivity))
rotate_y(deg2rad(event.relative.x * -sensitivity))
```

Interaction with the hazards is handled by using RayCasting [10]. A ray is cast from the player's position in the direction that the camera is facing. If the ray hits a hazard's collider, the hazard information and solution prompt is shown. Once the ray is no longer pointing at the hazard, the prompt is hidden. If the player presses the interact button while the prompt is showing, the player's "interact" function is called, which disables movement, connects the hazard's "finished" signal to the player, and calls the hazard's "interact" function. As soon as the interaction is complete, the hazard emits the "finished" signal and the player movement is once again enabled.

```
if interact_ray.is_colliding():
        var hit = interact_ray.get_collider()
        if hit is Hazard:
                # Show hazard info
        else:
                # Hide hazard info
```

4.3 Hazards

The Hazard class was created as a base for every hazard to inherit from. It contains properties for the name of the hazard, its solution, duration, and ID, which corresponds to a slot on the logos UI. It also has two base methods: interact and queue_free. The "queue_free" function is a built-in method in Godot that is called to delete the object from the scene. The Hazard class overrides the method to emit the "finished" signal and play a sound before it is deleted from the scene. The "interact" function emits a signal called "interacted", which will be used by the GameManager, and then waits for the hazard's duration in seconds before calling the "queue_free" function.

```
func interact():
        emit_signal("interacted", duration)
        yield(get_tree().create_timer(duration), "timeout")
        queue_free()
```

Each hazard's code inherits from this base class and overrides the "interact" and "queue_free" functions to fit the visual needs of the hazard in question. For example, the hazard of the cable sitting in a puddle of water overrides the "interact" function to make the puddle gradually disappear by using the Tween, which is a Godot object used for smoothly animating a node's properties [11].

```
func interact():
        .interact()
        var material = mesh.get_surface_material(0)
        tween.interpolate_property(material,    "shader_param/dissolved",    0,    1,    duration,
Tween.TRANS_LINEAR)
        tween.start()
```

4.4 Game Manager

The Game Manager is the script that stores the player information and handles time, score, as well as user interface updates. It stores references to most of the UI elements and has methods that initialize and update them when needed. Moreover, the Game Manager is responsible for managing the game flow, handling the hazard-emitted signals, as well as starting and ending the game as necessary. Once the end of the game is reached, the Game Manager disables the player's movement, makes the mouse cursor visible, and calculates the score achieved by the player. After that, the leaderboard is initialized with the highest scores of the previous player, along with the score achieved by the current player, which is then shown on screen.

```
func end_game():
        running = false
        player.frozen = true
        Input.set_mouse_mode(Input.MOUSE_MODE_VISIBLE)
        # Calculate Score
        # Animate UI
        leaderboards.initialize(player_name, score, score_id)
```

4.5 Leaderboard

The leaderboard is achieved by using a free online service for Godot called SilentWolf [12]. It provides backend services for this engine and allows for custom online leaderboards without the need for hosting. The service provides an add-on for Godot which we configured by entering a private API key supplied by the SilentWolf website upon registration. After that, we created a simple leaderboard interface and set up the game to submit the score to the leaderboard once the player reaches the end.

```
yield (SilentWolf.Scores.persist_score(player_name, score), "sw_score_posted")
```

We then populated the leaderboard interface with scores of the previous players, as well as the current player's score, which is highlighted in red.

```
yield (SilentWolf.Scores.get_high_scores(max_items), "sw_scores_received")
```

5 Uses for the game

Considering the simplistic nature of the game, it could be used in many ways to reach a wide audience and educate them about the dangers of electrical hazards and how to avoid them. A possible approach would be having an electrical company upload the game on their website as a simple introduction to electrical safety. Having an interactive environment to learn from is sure to captivate the player and give a better representation of what the hazards look like and how to identify them. In fact, paper [13] suggests that people who play more video games have better problem representation skills, as well as a more efficient metacognitive process than people who do not play video games. When asking our players what was the most appealing aspect of the game, most of them said it was the leaderboard system, as they kept trying to develop strategies to beat the highest score on the leaderboard. This shows us that the players were willing to look for faster routes to get to the end of the game so they could get a higher score, which is a way for them to practice their problem-solving skills.

This application is just one example of what video games could represent in the field of education. Other games could be utilized to enhance the learning experience of the player in certain fields. For example, paper [14] shows a study related to educational games which showcases several examples of video games used for learning. It concluded that educational games are highly beneficial in the process of learning because of the interactive and entertaining elements present in video games.

6 Conclusions

Video games represent a promising tool in the future of education, combining both entertainment and a dynamic learning experience into a powerful medium suited to personal development. The application showcased in this paper applies these principles to inform the player of the many dangers of electricity in a household, with the aim of reaching as wide an audience as possible through the multi-platform support enabled by HTML5. The flexibility of the virtual medium allows for various technologies and applications to be used to create these educational games, examples of which are Unity, Unreal Engine, and Godot Engine. Seeing as technology keeps on going forward, it would be a shame not to exploit this medium for training and educating the people who are going to shape our future.

Acknowledgement: This paper was supervised by Professor *Gabriela Moise*, from *Petroleum-Gas University, Ploiesti, Romania.*

References

[1] Trevor Wheelwright, *2022 Cell Phone Usage Statistics: How Obsessed Are We?*, https://www.reviews.org/mobile/cell-phone-addiction/

[2] Newzoo, *2020 Global Games Market Report*

[3] Dr. Mark Griffiths, *The educational benefits of videogames*, http://irep.ntu.ac.uk/id/eprint/15272/1/187769_5405%20Griffiths%20Publisher.pdf (2002)

[4] Michael R. Zemaitis; Lisa A. Foris; Richard A. Lopez; Martin R. Huecker, *Electrical Injuries*, https://www.ncbi.nlm.nih.gov/books/NBK448087/ (2021)

[5] James Paul Gee, *Are Video Games Good for Learning?*, https://www.idunn.no/doi/epdf/10.18261/ISSN1891-943X-2006-03-02 (2006)

[6] Isabela Granic; Adam Lobel; Rutger C. M. E. Engels, *The Benefits of Playing Video Games* (2014)

[7] *Using Signals in Godot Engine*, https://docs.godotengine.org/en/3.4/getting_started/step_by_step/signals.HTML5

[8] *Kinematic Body in Godot Engine*, https://docs.godotengine.org/en/stable/classes/class_kinematicbody.HTML5

[9] *Vector3 in Godot Engine*, https://docs.godotengine.org/en/stable/classes/class_vector3.HTML5

[10] *Raycast in Godot Engine*, https://docs.godotengine.org/en/stable/classes/class_raycast.HTML5

[11] *Tween in Godot Engine*, https://docs.godotengine.org/en/stable/classes/class_tween.HTML5

[12] *SilentWolf Backend Services*, https://silentwolf.com

[13] Xue-Min Zhang; Zijiao Shen; Xin Luo; Chunhui Su; Jiaqi Wang, *Learning from Video Game: A Study of Video Game Play on Problem-Solving* (2009)

[14] Peña-Miguel Noemí; Sedano Hoyuelos Máximo, *Educational Games for Learning (2014)*

Valentin RUSANDA
Petroleum-Gas University of Ploiești
iTIMF
Bulevardul Bucureşti 39, Ploieşti
ROMANIA
E-mail: valetinrusanda23@yahoo.com

Adrian Gheorghiţă NEAGU
Petroleum-Gas University of Ploiești
iTIMF
Bulevardul Bucureşti 39, Ploieşti
ROMANIA
E-mail: adrianneagu309@gmail.com

Ştefania Adriana PODEANU
Petroleum-Gas University of Ploiești
iTIMF
Bulevardul Bucureşti 39, Ploieşti
ROMANIA
E-mail: podeanu.adriana2001@gmail.com

Sixth International Conference on Applied Informatics
Imagination, Creativity, Design, Development
ICDD 2022, May 5-7
Sibiu, Romania

Rating Based System for Optimizing the Learning Path

Marin-Eusebiu Șerban

Abstract

The purpose of this article is to present a rating system that provides potential students with information that will help them choose their future learning path. The application provides a place for all students to share their thoughts on various topics related to university life. Students must log in using their institutional accounts to make the platform more trustworthy. We employ both artificial intelligence and human decision-making tools to approve the post for publishing (moderator of the platform). Our app also allows you to chat with other users. The ability for users to customize the UI is one of our application's strengths. We use single-page application technology, and there are numerous ranking factors that may be used at the same time.

1 Introduction

This article responds to the need for a better way to manage the future decisions of the future students. Nowadays, most high school students who desire to pursue a higher level of education probably end up attending a college that does not provide them with the impression that they have accomplished something significant with their time spent in that school unit.

On a broad scale, all students who wanted to take this step had to search hard to acquire information from third parties in order to make a convincing decision that will create the foundations for a new life, a student life, which needs to define them some theoretical conceptions even better.

This application aims to change the long-term process that high school students go through in order to make such an important decision, because if a student chooses a specific faculty, it can often not find relevant information, and must rely on the words of others, and this aspect is interpretable from various perspectives. If we consider a conscientious student who studies every day, he will believe that the faculty is good because he gets to learn something from everything that is taught to him, however, if he is a student who does not learn everything and is not attentive to all of the courses, he will believe that the faculty or teachers do not know how to teach.

That being said, the issue of selecting a faculty is far more complicated, and the fact that most people discuss it online is a perfect way for current students to publicly express their viewpoint given that there is presently no platform specifically dedicated to students. There are now just social media platforms available to accomplish this, which frequently end in altering the perception of those seeking information.

Because the application is highly depended on the security of its users, a student who posts a review on the platform will find the review to be anonymous as long as the people viewing the reviews are not authenticated. Furthermore, in the fact that the program was built from scratch, we were able to use the newest and greatest technology on the market in order to make the application as simple, quick and safe as possible.

2 Application design

Firstly, the application is built on the Angular architecture, a Javascript framework that employs the "One Page Application" technology, which provides fluidity, stability, and speed in the program's activities, and the fundamental reason for this improvement is that when a user changes the current location of the page, the static elements are retained by the application and only the selected component is updated, which means that the application no longer needs to be reloaded.

The application is organized into components, which not only helps the programmer keep the code clean, but also keeps various functionalities within a single component, so there are no intersections between them, as a result, SCSS and TypeScript remain isolated from the rest of the components, allowing you to reuse the names of variables, classes, or ids from other components without risk of their conflicting.

The database was created with MySQL, which provides for considerably more extensive table management and is now ranked second in the world in terms of database management.

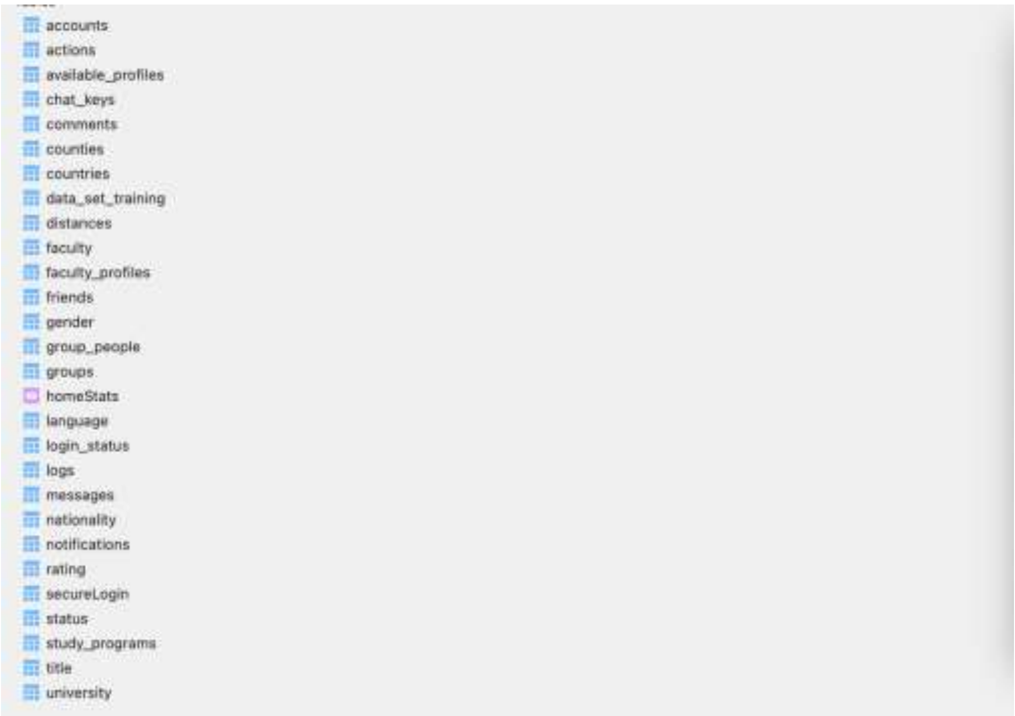
SQL, as a popular language at a very high level, has begun to integrate many more relevant features, which increases its use. The database is designed in such a way that the relationships between the tables are as explicit and manageable as possible.

2.1 Database design

The figure one represents application database, which contains in this moment 24 tables.

1. accounts – stores all the informations about a certain user after the process of registration is completed properly.
2. Actions – there are saved values for actions made to a faculty.
3. Chat_keys – in this table are saved the keys between two users.
4. Comments – there are saved all the reply's made to a review.
5. Counties – represent a list with all the counties from Romania
6. Countries – represent a list with all the countries available in the application
7. Data_set_training – will be used in the future for an AI to predict what could want an user to find.
8. Distances – is a list with all distances between all counties available, this would help in the future to the prediction AI.
9. Faculty – represent a list with all available faculties on the platform.
10. Friends – is a list with all users that are friends on the platform.
11. Gender – there are saved all genders ids.
12. Group_people – contain a list with all the users that have joined in a certain group
13. Groups – store the information about groups.
14. homeStats – represent a materialized view that contain stats that will be placed on home
15. login_status – contein a list with all login status
16. logs – there will be saved all logs.
17. Messages – contain all messages between
18. Available_profiles – there are saved all profiles that could be found in any faculty
19. Study_programs – contain the available study programs(ex: Bachelor,Master)
20. SecureLogin – is used in validation ip process, there are saved all information about an ip and user
21. Title – is used to set the title for user based on him activity on the platform

22. Faculty_profile - there are saved all faculties profiles with the study program
23. Rating – a list with all reviews placed on website
24. Language – used to save all current languages available on platform
25. Nationality – a list with all nationalities



accounts
actions
available_profiles
chat_keys
comments
counties
countries
data_set_training
distances
faculty
faculty_profiles
friends
gender
group_people
groups
homeStats
language
login_status
logs
messages
nationality
notifications
rating
secureLogin
status
study_programs
title
university

Fig. 1: Database Structure

All tables in the application serve a specific purpose. The majority of them store information about the user, while the other side stores the data that the user will select during the connection process, with these data linked to each other using Foreign Keys to make the connection as explicit as possible.

As previously said, some of the tables are reliant on one another, but the majority of the tables are independent and primarily supply information for relational tables.

The main table in the application is that of accounts, it is directly related to the majority of the tables in the program, and it attempts to specify the user who takes an activity in the application on a given branch.

2.2 Application Functionalities

The application has the following functionalities.

1. Login using an institutional account from Google or Microsoft
2. A system of IP validation with a code send on email.
3. Place reviews for a specific faculty.
4. Search for faculty, university or friends using the searchbar.
5. Social platform.
6. Chat with other students.
7. Anonymous posts when placing a review
8. Security using RSA encryption.
9. Toxic posts cannot be displayed on the site based on an AI for processing
10. Searching optimization using the newest technologies.

The first functionality is the most essential because the application is intended for a particular niche, students, and an authorization on the platform can only be made using that institutional account. A separate mechanism will be introduced in the future for those who do not have such an institutional account.

The second functionality is designed to provide future students with trustworthy information about the quality of a faculty.

As a student and public information application, it has developed a search bar through which you may search for faculties, universities, and users, with the restriction that a user who is not logged has access to only the two categories indicated above.

This method was created within the program to be as fast and simple to use as feasible for all users who want to search for certain information.

The application is considered a social platform. It features an integrated chat method so that students can communicate without worry in the application, and an end-to-end encryption technique has been built to keep your data secure.

When a student decides to publish a review on the application, he has the option of making it public or anonymous. If he chooses anonymity, the review will be placed as anonymous and no one will know who posted it.

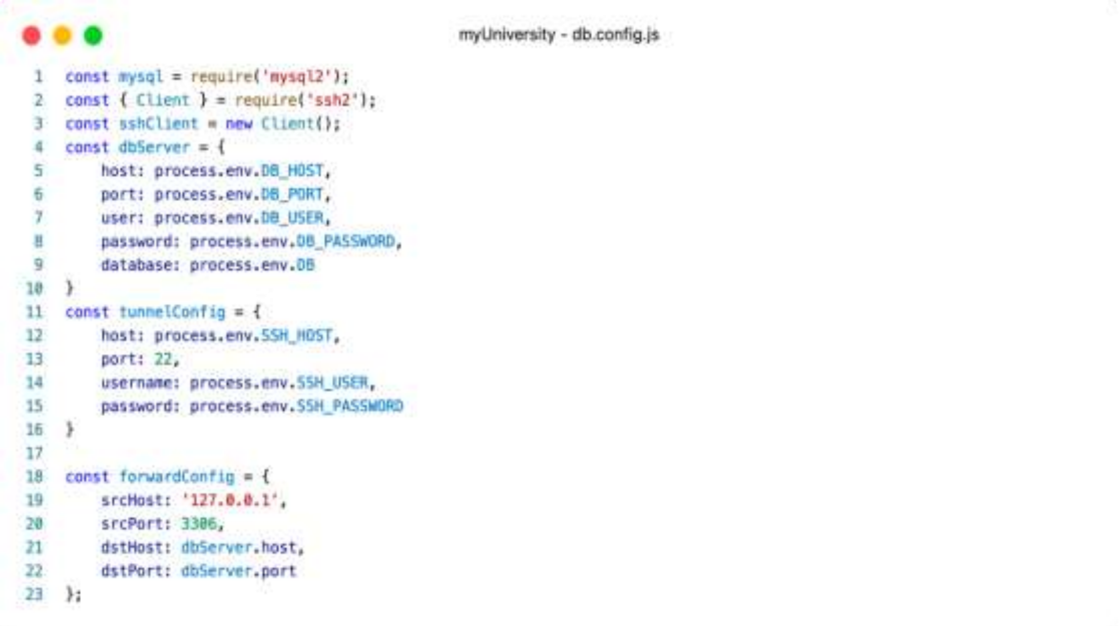
To retain common sense in speech, an AI specially created to detect the toxicity of the language is employed to keep the decency as much as possible within the review platform.

2.3 Technologies

I chose to use the Angular framework for the application, which is a library based on the well-known Javascript language. This framework has the advantage of higher performance and fluency due to the use of "One Page Application" technology.

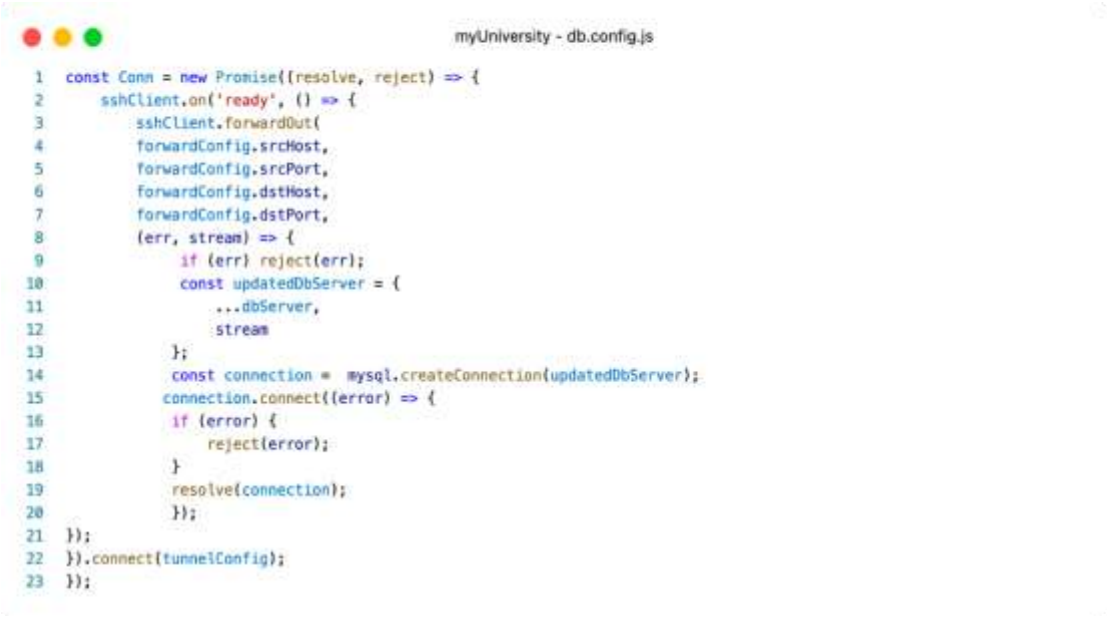
The application is divided into two sections: the client area, which is entirely rendered by Angular, and the server side, which is built with NodeJS with libraries like as EXPRESS, HTTPS, Websocket.io, mysql2 and SSH.

In Figure 2 is showed a method to config a database connection when using ssh and in Figure 3 is showed the actual connection.



```
const mysql = require('mysql2');
const { Client } = require('ssh2');
const sshClient = new Client();
const dbServer = {
    host: process.env.DB_HOST,
    port: process.env.DB_PORT,
    user: process.env.DB_USER,
    password: process.env.DB_PASSWORD,
    database: process.env.DB
}
const tunnelConfig = {
    host: process.env.SSH_HOST,
    port: 22,
    username: process.env.SSH_USER,
    password: process.env.SSH_PASSWORD
}

const forwardConfig = {
    srcHost: '127.0.0.1',
    srcPort: 3306,
    dstHost: dbServer.host,
    dstPort: dbServer.port
};
```

Fig. 2: Database configuration



```
const Conn = new Promise((resolve, reject) => {
    sshClient.on('ready', () => {
        sshClient.forwardOut(
        forwardConfig.srcHost,
        forwardConfig.srcPort,
        forwardConfig.dstHost,
        forwardConfig.dstPort,
        (err, stream) => {
             if (err) reject(err);
             const updatedDbServer = {
                 ...dbServer,
                 stream
            };
            const connection =  mysql.createConnection(updatedDbServer);
           connection.connect((error) => {
            if (error) {
                reject(error);
            }
            resolve(connection);
            });
});
}).connect(tunnelConfig);
});
```

Fig. 3: Creating Database connection

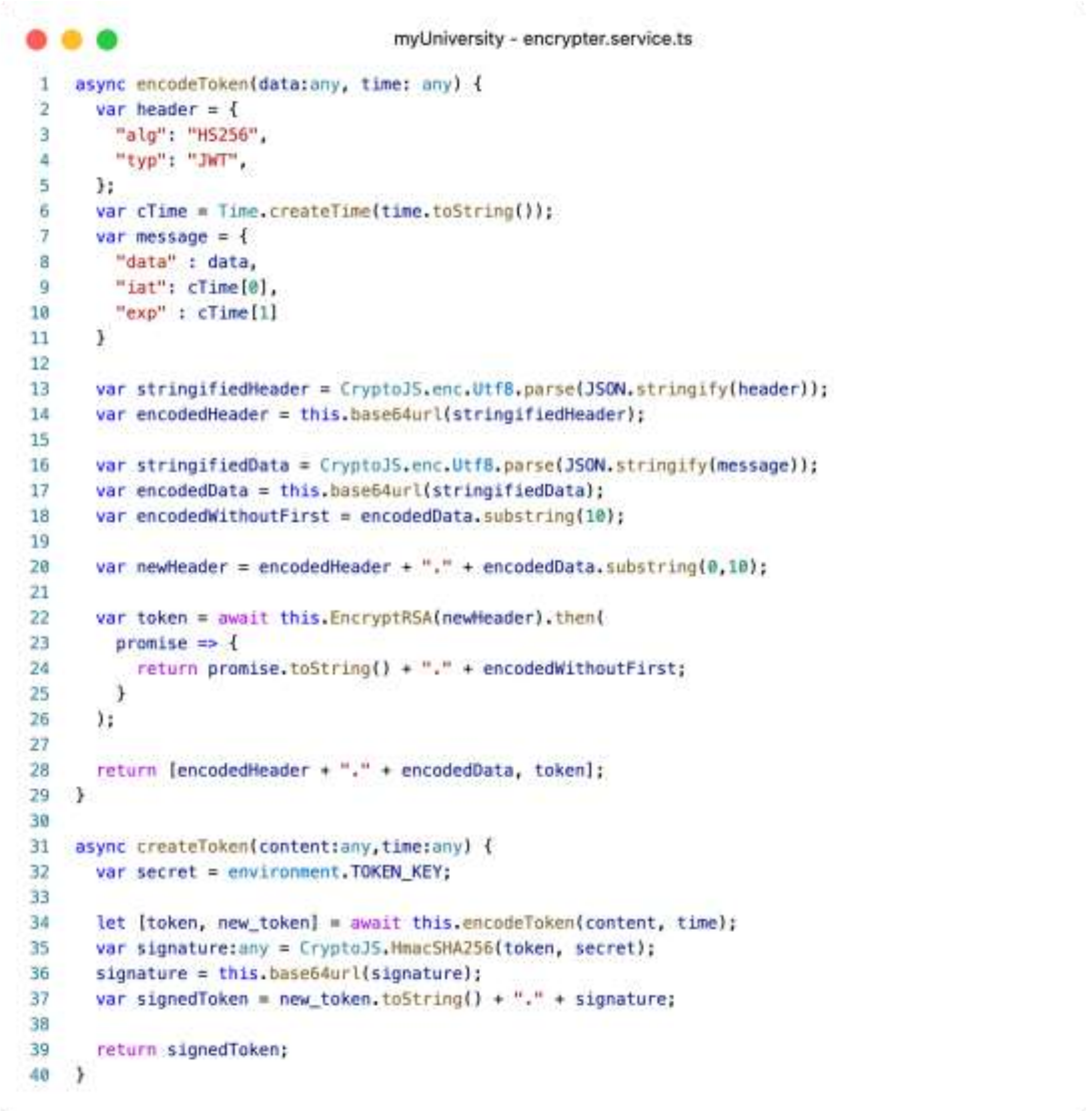
2.4 Security

The application's security was prioritized throughout the development process. This implementation was required primarily because, while all data normally leaves the HTTPS tunnel encrypted, it is not on the network, and anyone listening with a sniffer on the network can find that data.

The security module works as follows: each request is encoded with the help of JsonWebToken in order to be signed and have a time limit in which that signature can be accessed; once created, the token is divided into two fragments, the first of which is encrypted using the asymmetric RSA encryption method.

When the token arrives at the server, it is decrypted and validated to ensure that the signature connection is complete. In Figure 4 is presented the method of encryption created.

As previously stated, the data to be encrypted and signed enter the function and are transmitted to the encryption function, where a token signed with a time limit is created and signed using an RSA key. These types of encryption are performed using the Forge library, which implements RSA encryption methods.

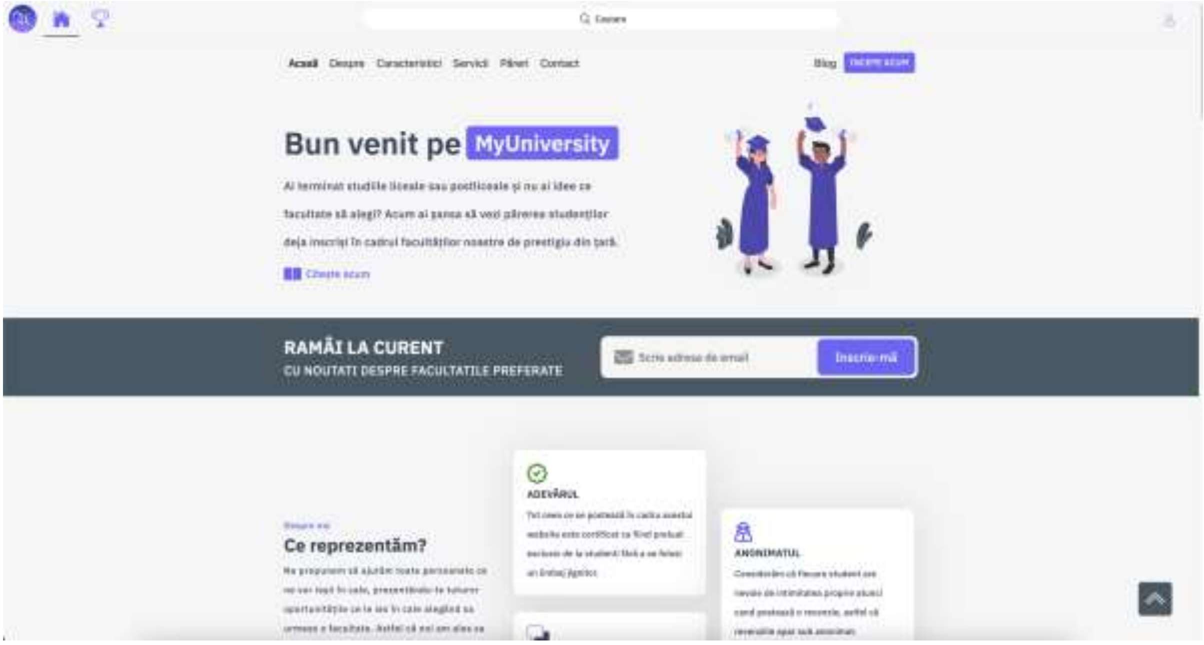


```
async encodeToken(data:any, time: any) {
  var header = {
    "alg": "HS256",
    "typ": "JWT",
  };
  var cTime = Time.createTime(time.toString());
  var message = {
    "data" : data,
    "iat": cTime[0],
    "exp" : cTime[1]
  }

  var stringifiedHeader = CryptoJS.enc.Utf8.parse(JSON.stringify(header));
  var encodedHeader = this.base64url(stringifiedHeader);

  var stringifiedData = CryptoJS.enc.Utf8.parse(JSON.stringify(message));
  var encodedData = this.base64url(stringifiedData);
  var encodedWithoutFirst = encodedData.substring(10);

  var newHeader = encodedHeader + "." + encodedData.substring(0,10);

  var token = await this.EncryptRSA(newHeader).then(
    promise => {
      return promise.toString() + "." + encodedWithoutFirst;
    }
  );

  return [encodedHeader + "." + encodedData, token];
}

async createToken(content:any,time:any) {
  var secret = environment.TOKEN_KEY;

  let [token, new_token] = await this.encodeToken(content, time);
  var signature:any = CryptoJS.HmacSHA256(token, secret);
  signature = this.base64url(signature);
  var signedToken = new_token.toString() + "." + signature;

  return signedToken;
}
```

Fig. 4: Encryption and Token signing method

3 Application interface

When you initially launch the application, you will view the home page, which contains the application's presentation as well as all of the distinguishing components that highlight the platform's functionality. Figure 5 represent the home page.



Cautare

Acasă Despre Caracteristici Servicii Păreri Contact

Blog ÎNCEPE ACUM

Bun venit pe MyUniversity

Ai terminat studiile liceale sau postliceale și nu ai idee ce facultate să alegi? Acum ai șansa să vezi părerea studenților deja inscriși în cadrul facultăților noastre de prestigiu din țară.

Citește acum

RAMĂI LA CURENT
CU NOUTATI DESPRE FACULTATILE PREFERATE

Scrie adresa de email

Inscrie-mă

Despre noi

Ce reprezentăm?

[illegible]

ADEVĂRUL

[illegible]

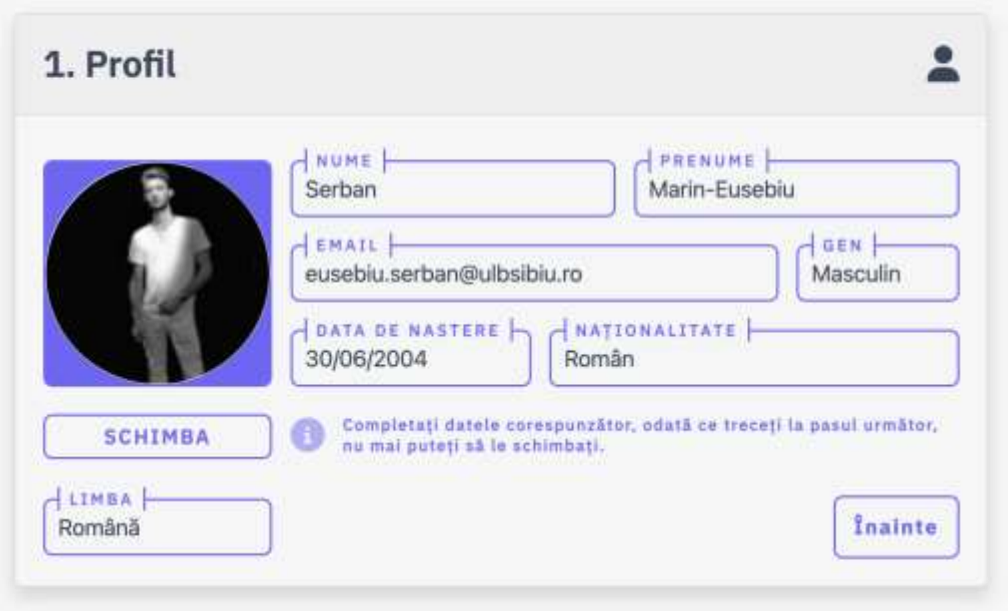
ANONIMATUL

[illegible]

Fig. 5: Application UI

The platform is connected via the institutional Google and Microsoft accounts provided by the institutions upon request. When the authentication is conducted, if the user is registering for the first time on the platform, he is required to go through two phases of account setup.

In Figure 6 is presented the first stage of the registration process and in Figure 7 is presented the second step.



1. Profil

NUME: Serban

PRENUME: Marin-Eusebiu

EMAIL: eusebiu.serban@ulbsibiu.ro

GEN: Masculin

DATA DE NASTERE: 30/06/2004

NAȚIONALITATE: Român

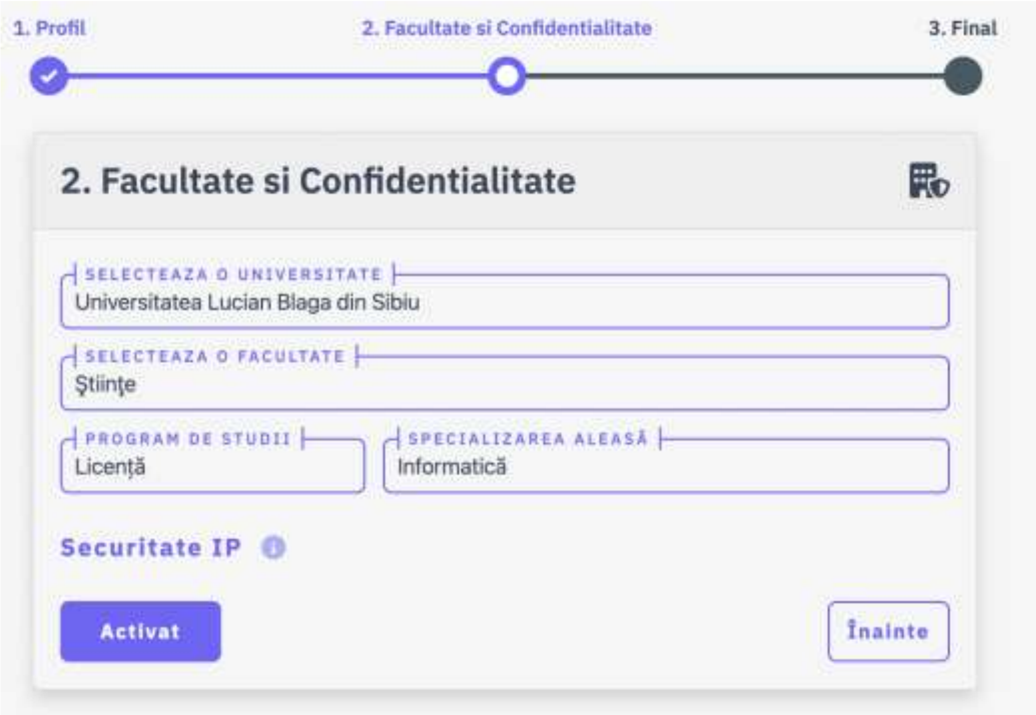
SCHIMBA

Completați datele corespunzător, odată ce treceți la pasul următor, nu mai puteți să le schimbați.

LIMBA: Română

Înainte

Fig. 6: Step one of registration process



2. Facultate si Confidentialitate

SELECTEAZA O UNIVERSITATE: Universitatea Lucian Blaga din Sibiu

SELECTEAZA O FACULTATE: Ştiinţe

PROGRAM DE STUDII: Licenţă

SPECIALIZAREA ALEASĂ: Informatică

Securitate IP

Fig. 7: Step two of registration process

When this account setup page was created, a profile photo change was provided, along with the option to crop the image so that it is better centered to optimize memory space. a technique for compressing photos that are utilized
Figure 8 displays the section with responsibility of adding profile images.

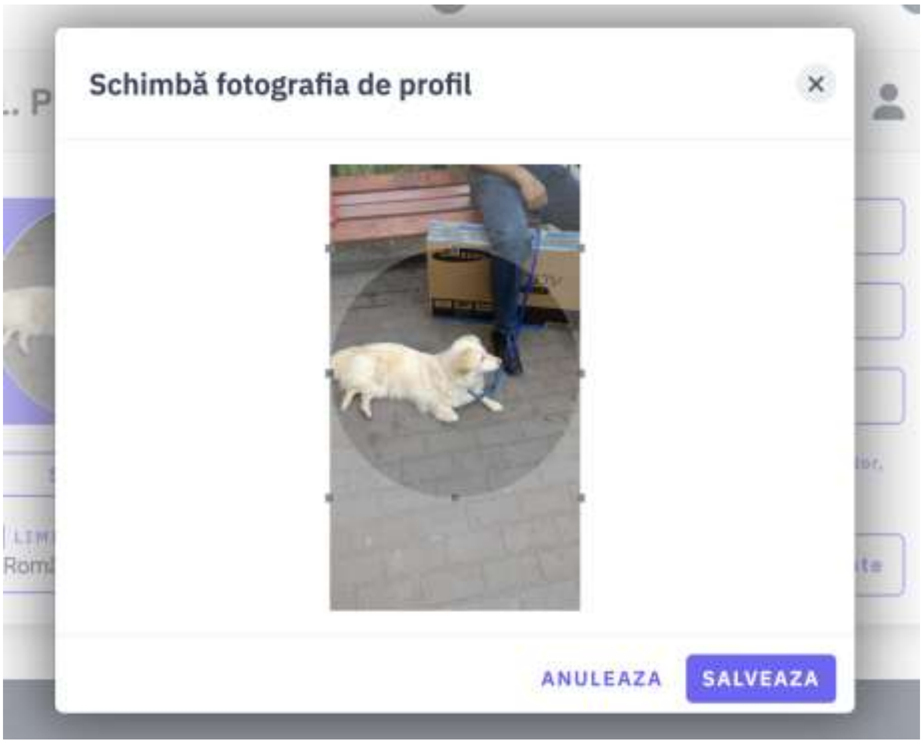
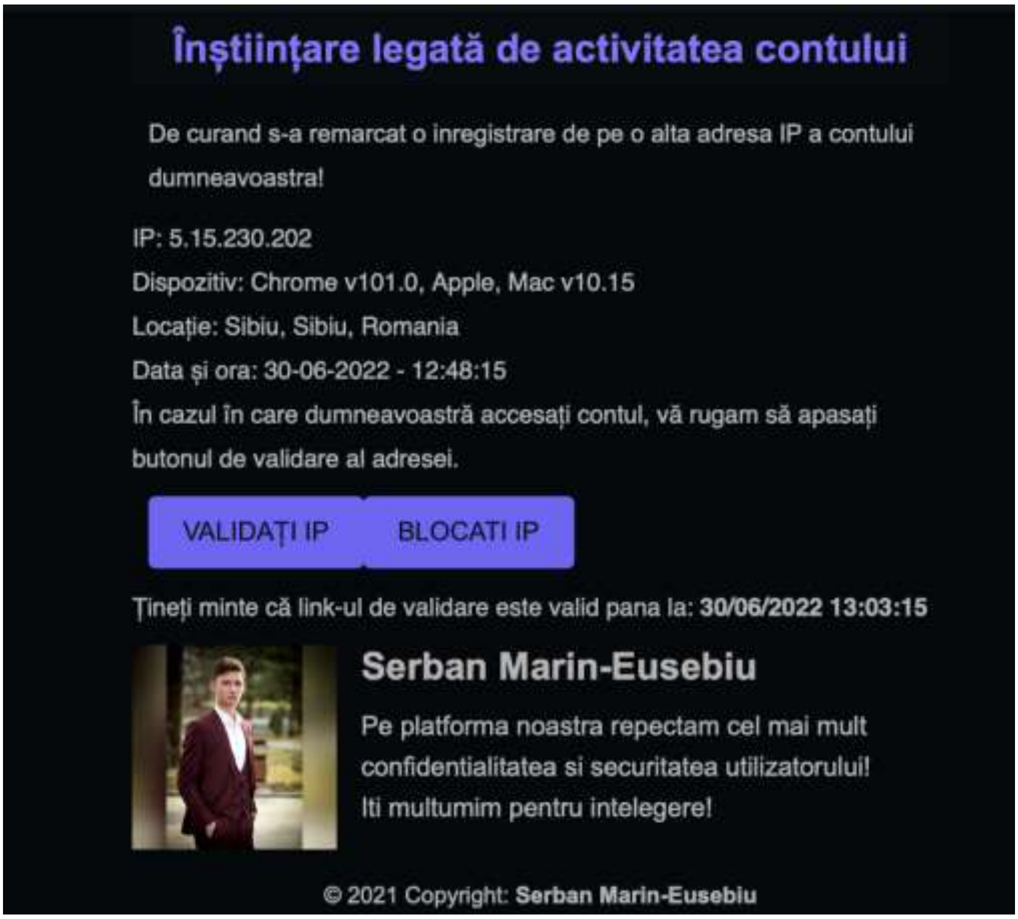

Schimbă fotografia de profil

Fig. 8: Select profile picture

When selecting the secure login option, the person's IP will be determined during each new connection, and if that IP is not authorized, a waiting screen will appear with the option of resending an activation email in case the standard one could not be sent, and the user will have to enter the email to allow access to the platform, Figure 9 shows the email interface with the methods that the user can access.



Înștiințare legată de activitatea contului

De curand s-a remarcat o inregistrare de pe o alta adresa IP a contului dumneavoastra!

IP: 5.15.230.202
Dispozitiv: Chrome v101.0, Apple, Mac v10.15
Locație: Sibiu, Sibiu, Romania
Data și ora: 30-06-2022 - 12:48:15
În cazul în care dumneavoastră accesați contul, vă rugam să apasați butonul de validare al adresei.

Țineți minte că link-ul de validare este valid pana la: **30/06/2022 13:03:15**

Serban Marin-Eusebiu

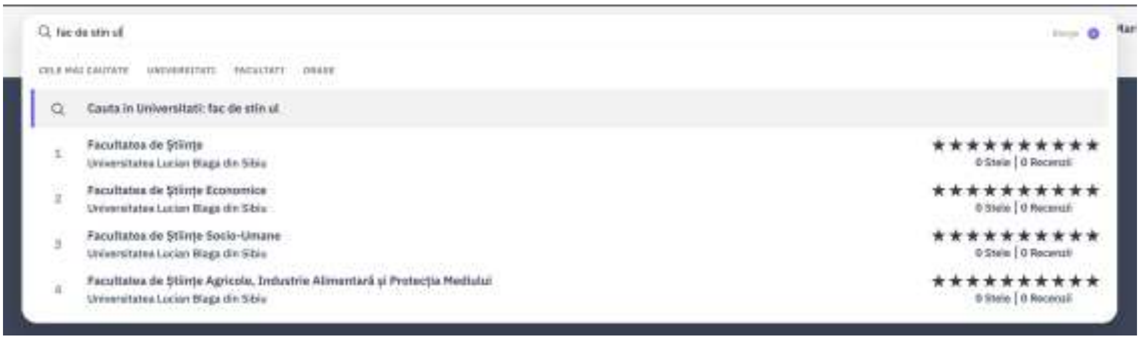
Pe platforma noastra repectam cel mai mult confidentialitatea si securitatea utilizatorului! Iti multumim pentru intelegere!

© 2021 Copyright: **Serban Marin-Eusebiu**

Fig. 9: Validation IP email

As can be seen, the location, time, device, and IP address from which the connection was attempted are preserved in the corresponding email, with the option to deny the respective connection attempt or authorize the connection.

The search bar has been designed in such a way that it can be used quickly, following the Google model, with all searches accessible via the up, down, and enter keys. To make a search as fast and accurate as feasible, the algorithm produces a forecast based on the beginnings of words. In Figure 10 is presented the UI of the searchbar.



Cauta in Universitati: fac de stin ul

1. Facultatea de Științe
Universitatea Lucian Blaga din Sibiu
★★★★★★★★★★
0 Stele | 0 Recenzii

2. Facultatea de Științe Economice
Universitatea Lucian Blaga din Sibiu
★★★★★★★★★★
0 Stele | 0 Recenzii

3. Facultatea de Științe Socio-Umane
Universitatea Lucian Blaga din Sibiu
★★★★★★★★★★
0 Stele | 0 Recenzii

4. Facultatea de Științe Agricole, Industrie Alimentară și Protecția Mediului
Universitatea Lucian Blaga din Sibiu
★★★★★★★★★★
0 Stele | 0 Recenzii

Fig. 10: Searchbar UI

4 Technical Details

The application is designed using the most efficient methods to ensure that the application runs as quickly as possible, so that inside the application, the highest performing platforms and sustainable in terms of future developments are as follows:

- Angular, this framework has been well-developed to the point where all components make development considerably easier.
- Mysql, which is extensively used by most businesses, is a sql database management system that manages to create quick and simple storage methods.
- Nodejs, a framework library as well as a server constructor powerful enough to assist programmers in swiftly creating applications in direct relation to the front end application

The platform is composed of several components that connect with one another via the Angular framework; it employs normal HTML implementation methods with only a few modifications.

The server part is created and implemented using NodeJS and express; this library aids in the creation of a strong and secure API through the algorithms defined in the application. The API interacts directly with the database and the server to fulfill user requests.

5 Future Developments and Conclusions

The presented application implements many proprietary technologies, but they are not yet the most powerful and optimal. I wanted this application to be secure and fast, but at this time the application will perform several changes in the future that are intended to improve performance and usefulness, including the following:

- Creating a personal page where the user can publish whatever he wishes.
- Create a page dedicated to meditations.
- Designing a contemporary messaging interface
- Include video and voice calling capabilities.
- Added the ability to send pictures and videos in discussions, while optimizing the space utilized so that the quality is not compromised.
- Implementation of an AI to better comprehend what is sought on the site, as well as a recommendation system based on the user's location and preferences.
- Include responsive elements to ensure that the app looks normal on all devices.
- Enhancing animation and design.
- The application's implementation of a visual aspect management system.

- An artificial intelligence capable of detecting a stamp and the academic year for students who do not have an institutional account.
- Moving the database to a technology that allows for data caching.
- Native mobile application development in Koltlin for Android and Swift for iOS, respectively.

Acknowledgement: This work was supervised by Professor PhD Dana Simian, from Lucian Blaga University of Sibiu.

References

[1] Nate Murray, Felipe Coury, Ari Lerner and Carlos Taborda, *ng-book The complete book on Angular* 5, CreateSpace Independent Publishing Platform, 2018

[2] Simon Holmes, *Getting MEAN with Mongo, Express, Angular and Node*, Manning, 2019

[3] Christof Paar and Jan Pelzl, *Understanding Cryptography*, Springer-Verlag Berlin Heidelberg, 2010

[4] Eric Enge, Stephan Spencer and Jessie C. Stricchiola, *The Art of SEO: Mastering Search Engine Optimization*, O'Reilly Media, 2015

[5] Baron Shwartz, Peter Zaitsev and Vadim Tkachenko, *High Performance MySQL, 3rd Edition*, O'Reilly Media, 2012

Marin-Eusebiu ȘERBAN
Lucian Blaga University of Sibiu
Faculty of Science
Department of Mathematics and Informatics
5-7 Dr. Ratiu Str, Sibiu 550012
ROMANIA
E-mail: eusebiu.serban@ulbsibiu.ro

LIST OF AUTHORS

Craig AKIRI

College of Wooster
Mathematical and Computational Sciences Department
Wooster OH
USA
E-mail: cakiri23@wooster.edu

Hakan ARDA

University of Applied Sciences
Würzburg-Schweinfurt
Fraunhofer-Institut für Silicatforschung ISC
GERMANY
E-mail: hakan.arda@student.fhws.de

Ramona-Antonela BONDĂ

Lucian Blaga University of Sibiu
Faculty of Science
Department of Mathematics and Informatics
5-7 Dr. Ratiu Str, Sibiu 550012
ROMANIA
E-mail: ramona.bonda19@gmail.com

Peter BRAUN

Würzburg-Schweinfurt
Faculty of Computer Science and
Business Information Systems
Sanderheinrichsleitenweg 20, 97074 Würzburg
GERMANY
E-mail: peter.braun@fhws.de

Sorin Ionut CONEA

Vasile Alecsandri University of Bacău
Department of Mathematics & Informatics
Calea Mărăşeşti, nr. 157, Bacău, 600115
ROMANIA
E-mail: coneasorin@outlook.com

Luca-Dumitru DRÎNDEA

Samuel von Brukenthal National College
Mathematics-Informatics
Piața Albert Huet 5, Sibiu
ROMANIA
E-mail: luca.drindea@gmail.com

Daniel HABERMAYR

University of Applied Sciences Landshut
Fakultät Informatik
Am Lurzenhof 1, 84036 Landshut
GERMANY
E-mail: s-dhaber@haw-landshut.de

Maxwell HOSLER

College of Wooster
Mathematical and Computational Sciences Department
Wooster OH
USA
E-mail: mhosler22@wooster.edu

Felix HUSAC

Lucian Blaga University of Sibiu
Faculty of Science
Department of Mathematics and Informatics
5-7 Dr. Ratiu Str, Sibiu 550012
ROMANIA
E-mail: felix.husac@ulbsibiu.ro

Mircea IONESCU

Senior Data Scientist
Assurant Labs
Cleveland OH
USA
E-mail: mircea.ionescu@assurant.com

Matthias KECKL

University of Applied Sciences
Würzburg-Schweinfurt
Faculty of Computer Science and
Business Information Systems
Sanderheinrichsleitenweg 20, 97074 Würzburg
GERMANY
E-mail: matthias.keckl@outlook.de

Andrei-Daniel MIHAI

Lucian Blaga University of Sibiu
Faculty of Science
Department of Mathematics and Informatics
5-7 Dr. Ratiu Str, Sibiu 550012
ROMANIA
E-mail: andrei_mihai_23@yahoo.ro

Alexandru - Serban MOLDOVAN

Transilvania University Brasov
Faculty of Mathematics and Informatics
Str. Iuliu Maniu, nr. 50, Brasov
ROMANIA
E-mail: alex.moldo.serban@gmail.com

Patrick B. M. MÜLLER

Würzburg-Schweinfurt
Faculty of Computer Science and
Business Information Systems
Sanderheinrichsleitenweg 20, 97074 Würzburg
GERMANY
E-mail: patrick.mueller@fhws.de

Adrian Gheorghiță NEAGU

Petroleum-Gas University of Ploiești
iTIMF
Bulevardul Bucureşti 39, Ploieşti
ROMANIA
E-mail: adrianneagu309@gmail.com

Vlad OLEKSIK

Gheorghe Lazăr National College Sibiu
Mathematics-Informatics, intensive Informatics
Gheorghe Lazăr Street 1-3, Sibiu
ROMANIA
E-mail: vlad.oleksik@cnglsibiu.ro

Bogdan-Ioan OPRIȘIU

Samuel von Brukenthal National College
Mathematics-Informatics
Piața Albert Huet 5, Sibiu
ROMANIA
E-mail: bogdan.oprisiu2003@gmail.com

Daniel OSTERTAG

University of Applied Sciences Landshut
Fakultät Informatik
Am Lurzenhof 1, 84036, Landshut
GERMANY
E-mail: s-doster@haw-landshut.de

Daniela-Nicolia PĂTRUȚ

Politehnica University of Timișoara
Faculty of Automation and Computing
Bld. V. Pârvan, No. 2, 300223 Timișoara
ROMANIA
E-mail: daniela.patrut@student.upt.ro

Ștefania Adriana PODEANU

Petroleum-Gas University of Ploiești
iTIMF
Bulevardul Bucureşti 39, Ploieşti
Romania
E-mail: podeanu.adriana2001@gmail.com

Mircea-Alexandru RĂUȚOIU

Samuel von Brukenthal National College
Mathematics-Informatics
Piața Albert Huet 5, Sibiu
ROMANIA
E-mail: mircea.rautoiu@gmail.com

Valentin RUSANDA

Petroleum-Gas University of Ploiești
iTIMF
Bulevardul Bucureşti 39, Ploieşti
ROMANIA
E-mail: valetinrusanda23@yahoo.com

Marin-Eusebiu ŞERBAN

Lucian Blaga University of Sibiu
Faculty of Science
Department of Mathematics and Informatics
5-7 Dr. Ratiu Str, Sibiu 550012
ROMANIA
E-mail: eusebiu.serban@ulbsibiu.ro

Esther VAN DER KNAAP

Department of Horticulture
University of Georgia
Athens GA
USA
E-mail: esthervanderknaap@uga.edu

Sofia VISA

College of Wooster
Mathematical and Computational Sciences Department
Wooster OH
USA
E-mail: svisa@wooster.edu

Kristin WEBER

University of Applied Sciences
Würzburg-Schweinfurt
GERMANY
E-mail: kristin.weber@fhws.de

SPONSORS (in alphabetical order)

AUSY Technologies Romania

ASTRA National Museum Complex

Asociația BIT

CodexWorks technologies

***Ardeleana Academy* Foundation**

Global Solutions for Development

Keep Calling

Nagarro

NTT Data

PAN FOOD

Omeron Technologies, Romania

ProIT

ROPARDO

Top Tech

Wenglor

VISMA