UNIVERZA NA PRIMORSKEM
FAKULTETA ZA MATEMATIKO, NARAVOSLOVJE IN
INFORMACIJSKE TEHNOLOGIJE

ZAKLJUČNA NALOGA
(FINAL PROJECT PAPER)

SISTEM ZA IZMENJAVO NAVODIL ZA POPRAVILO ELEKTRONSKIH NAPRAV
(KNOWLEDGE SHARING SYSTEM FOR ELECTRONIC DEVICE REPAIR INSTRUCTIONS)

TRAJCHE BELEVSKI

2022
ZAKLJUČNA NALOGA (FINAL PROJECT PAPER)
BELEVSKI

UNIVERZA NA PRIMORSKEM
FAKULTETA ZA MATEMATIKO, NARAVOSLOVJE IN
INFORMACIJSKE TEHNOLOGIJE

Zaključna naloga
(Final project paper)

Sistem za izmenjavo navodil za popravilo elektronskih naprav

(Knowledge sharing system for electronic device repair instructions)

Ime in priimek: Trajche Belevski
Študijski program: Računalništvo in informatika
Mentor: doc. dr. Peter Rogelj

Koper, avgust 2022

Ključna dokumentacijska informacija

Ime in PRIIMEK: Trajche BELEVSKI

Naslov zaključne naloge:
Sistem za izmenjavo navodil za popravilo elektronskih naprav

Kraj: Koper

Leto: 2022

Število listov: 54 Število slik: 29 Število tabel: 7

Število referenc: 11

Mentor: doc. dr. Peter Rogelj

Ključne besede: programski inženiring, programsko opremo, spletna stran

Izvleček:
Cilj diplomske naloge je implementacija programske opreme, ki bi ljudi poučila, kako njihova tehnologija deluje in kako jo lahko popravijo. Prvič, diplomsko delo poudarja pomembnost problematike, ki jo imamo vsakdanji ljudje, saj je z vedno večjim številom tehnoloških naprav, ki jih uporabljamo, pogostejša potreba po njihovem popravilu. V nalogi preverimo, kako izvedljiva je rešitev tega problema in kakšne lastnosti I naj ima takšen sistem za izmenjavo navodil, da bi čim bolje ustrezal potrebam običajnih uporabnikov sistem deloval v korelaciji z uporabnikom, kaj vsebuje, kako je posamezna komponenta povezana in kaj te komponente počnejo. In končno zaključite z implementirano programsko opremo, ki prikazuje, kako se uporablja, in testira rešitev.

Key document information

Name and SURNAME: Trajche BELEVSKI

Title of the final project paper:
Knowledge sharing system for electronic device repair instructions

Place: Koper

Year: 2022

Number of pages: 54 Number of figures: 29 Number of tables: 7

Number of references: 11

Mentor: Assist. Prof. Peter Rogelj, PhD

Keywords: software engineering, software, website

Abstract:
The main topic of the thesis is implementing software that would educate people on how their technology works and how to repair it. Firstly, the thesis emphasizes the importance of the issue at hand which everyday people have, with the increasing number of technological devices that we use it is more common. Then we move on to explain how feasible this solution and what is the solution to this everyday problem or issue that people have. The main part describes how the system will work in correlation to the user, what it contains how each component is connected and what these components do. And finally, finish with the implemented software displaying how it is used and testing the solution.

ACKNOWLEDGEMENTS

I would like to capture this opportunity to express my gratitude towards the University of Primorska, principally to the faculty of mathematics, natural sciences, and information technology (FAMNIT) for lending me the privilege to be a foreign student in Slovenia and allowing me to pave my path towards my future career and success.

I thank Doctor Peter Rogelj and I wish to express my deepest gratitude to my mentor for his willingness to help me complete my studies. Secondly without his support and guidance during each step of the way this thesis would not be possible, thanks to his courses and ways of teaching I was able to learn significantly more and would like to praise him not just as a mentor but as a great professor with immense positive energy and great composure including his teaching methods and his whole demeanor towards the students.

Finally, I would like to thank my family for all the support and encouragement that they gave me especially since we had a rough experience with the pandemic they came through in the toughest times, and last but not least my friends and all of the people I met through this journey they made these past few years big memories to count and look back on.

Trajche Belevski
July 05, 2022

LIST OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ETC	et cetera
E.G.	for example
SRS	Software Requirements Specification
UML	Unified Modeling Language
SRS	Software Requirements Specification
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
PHP	Hypertext Preprocessor
SQL	Structured Query Language
DB	Database
URL	Uniform Resource Locator
URI	Uniform Resource Identifier

1 INTRODUCTION

In the last decade, all type of technology has advanced exponentially to a degree where any normal human cannot function in today's society without some type of technological device. Apart from individuals many organizations, workspaces, companies, and factories require the functionality of these devices since on a daily basis in order to function more effectively almost everyone is now dependent on these devices which can be crucial if some of them do not work properly or have any kind of small defect. Which can vary of all sorts such as Mobile devices, Computers, Laptops, Displays, Keyboards, etc.

The people have been advancing with the technological use and I do not see it stopping anytime soon, as a matter of a fact I believe it will only exponentially rise, therefore a lot of these people or so-called users of their technology, on the contrary, are not aware of what they actually possess or how those components in their technological devices work. We as people must learn and adapt to the new rise of the technological era in a way where we must understand the component and devices we use to a certain degree, otherwise, we are just using tools that we are not aware of how they actually work.

The need and dependency on these devices should be eye-opening to a lot of people since they are becoming so essential if a small error interrupts our usage and connection with this device it may cause a lot of traffic or inconveniences which becomes a bigger issue in the long run.

These issues and problems are a lot more common than people think, we just do not pay enough attention to it, normally each person can call up a repair man and repair their problem with a given technological device, but this is not always feasible since the costs may be way more than anticipated in comparison to the issue at hand, instead of repairing or fixing the issue by ourselves we must resort and depend on these people which occasionally do not repair to normal standards but the minimum requirement for the device to work.

2 PROBLEM DEFINITION

The problem/need to be able to repair one's own hardware or software is an important part of everyday life since in today's world we are all surrounded by hardware and technology more and more, every person has some kind of technology in order to progress and be able to be a part of today's society as a whole, hench increasing the usage and need for technology in modern life.

Therefore, a lot of people possess technology occasionally that they do not know how it works and most likely do not know how to repair a certain problem or a broken part of their piece of technology. The need to be competent enough to change or repair certain parts of a component is becoming more necessary by the day, as the use of technology rapidly increases, it is a feat that many people like myself want to accomplish which can be easily possible with this proposed system.

The problem here is everyone is lacking the basic fundamentals of how software and hardware components can be repaired or replaced, there is no need to pay for technical support where it is clearly feasible to do on your own, but by lack of knowledge makes it not possible.

Another issue would be, that there are many sources of information on how to proceed with replacing/repairing certain components or solving an issue with one's software, simply there are too many sources of information rather than having them all in one place.

Simply put one Google search will lead to many locations and different information of a simple problem on how to proceed when resolving it, examples such as YouTube, Reddit, PDF documentation, and other sources which can be found through Google or any other search engine.

The main focus is that people instead of learning the basics of their technology are obligated to find a corresponding repair man which can fix the problem at hand, by the lack of knowledge the everyday person will struggle with these problems, instead of this option the proper solution would be improving this knowledge with a learning and teaching system which can educate the average person that does not know much about their technology which is a must in today's society.

On the other hand, another issue that many people might encounter with the lack of knowledge in this field like myself, they might attempt some sort of repair when a problem is diagnosed on their technological component, they might find some information on the internet but this might not be significant enough to fix the problem as a whole, meaning a failed repair causing an even bigger issue or problem than before, making this a real troublesome matter.

3 REVIEWS OF CURRENT SOLUTIONS

Since our problem is a bit obvious to a certain degree, people have already taken matters into their own hands by trying to solve the issue at hand understanding that while the people that are into technology understand this spectrum but to the regular person that is not interested this may not be the case. So, this everyday person will try to find an existing solution to their problem with a lack of knowledge and experience on the problem at hand, meanwhile, some solutions already exist which might not be the most optimal or the best for our scenario.

Here we will do a short review and explanation of a few solutions and why they may not work or are just not optimal enough and can be done much better in a more efficient fashion.

3.1 YouTube

For this existing solution Figure [1], since it is an enormous platform where videos are uploaded, constantly we come to issues with filtering the search content that we get from YouTube.

Because there are way too many videos already existing that are not in relation to repairing technology it occasionally gives us videos not completely related to the topic that we wished to search for, hench making the searching for solutions quite difficult.

A lot of videos that already exist are sometimes too broad on solutions and don’t work for specific laptop models or mobile phones, some of them need a specific tutorial or solution found by the exact name or model name of the component which YouTube generally does not offer good filtering from their search bar.

Another issue is if a video has e.g., 900 views, and it’s closer or exactly to the solution we are looking for it will not be mainly recommended but will recommend a video not completely related solely because it has more views or a better like to dislike ratio.

Therefore, making this solution is not the most ideal way of learning and repairing technology.

Figure 1: YouTube logo

3.2 Reddit

For this existing solution Figure [2] again, the searching might be a problem but it's not the same case as YouTube since Reddit has a better route by possessing subreddits related to a certain topic which helps minimize to a certain field, but this does not diminish the process of searching if a lot of content is available since the titles are not always concrete making it sometimes difficult to use correct keywords for searching.

But we reach the same argument where information is too broad, filtering is not the best when searching for a solution and there can be a more efficient way of doing all of this work, on Reddit one will not have all of the information necessary needed in one place, we might be missing some information from documentation which cannot be cited, either it will be video content with a title and no description or there will be just an image or a link to some documentation which either way we may be lacking what we need to resolve our problem.

Figure 2: Reddit logo

3.3 iFixit

This existing solution Figure [3] is quite similar to what we are doing here, but not quite what we want to have since regularly the solutions are too broad, may be hard to understand, and not always with a visual representation since a lot of them only use images and a small amount of text explaining what is actually going on.

On the other hand, the videos which are uploaded as solutions are simply just YouTube videos that we can also search on YouTube making this functionality not quite useful. We do not have multiple solutions in one place and we might need to search on other platforms as well. Making this also not the most optimal solution to our problem.

Figure 3: iFixit logo

4 THE PROPOSED PROBLEM SOLUTION

From a technological point of view, the best solution would be a website.

For everyone who has a problem the easiest way the find a solution is by using Google, which means they are most likely going to open some website that contains a solution to their problem, also a website concept is easier to implement and more easily accessible to the everyday user.

Speaking about today's world, it is almost inevitable to own some device, where people will definitely need to repair something regarding the device they own, now they can do it themselves.

A website system where we can find solutions by searching for them with exact model names and find solutions where the video contains a title, description, comment section, rating system, linking together other resources from already existing solutions (YouTube, Reddit, PDF files) basically taking everything and putting it in one place for more efficient work.

The system will contain video content which can be uploaded by any person, representing a solution on how to proceed with repairing or replacing some hardware or even how to solve technical errors around software, or any type of technology in general.

The main purpose of this system is to solve the problem as mentioned, through educational videos done by people for the people, one can learn and experience how such technology works or repair it.

Following this, the system will contain a description content of each solution where there will be brief details explaining the solution depending on how much the user would like to type, in addition to this feature the solution will contain references to other sources of similar solutions or other ways of solving the problem, such as YouTube links, Reddit posts, other types of documentation, in essence, it will be all information gathered in one place.

In addition, the system will contain a comment section where other users can ask questions or follow-up questions regarding the solution or resources available, the system will also include a rating system on each individual solution where it offers a selection of an upvote or a downvote expressing one's opinion of how valid and resourceful the solution and resources were.

Also, the system will contain a search engine where it will allow users to search for specific solutions or documentation required for their problem, making the experience easier.

On another note, a user does not need to upload a brand-new solution that he has created, simply he can upload an existing video that he found useful but for example, it’s not exactly all of the information in it, so he can upload additional information that he has found helpful for this case, in conclusion, it would be connecting multiple sources of information towards one solution of his problem which just the video does not solve.

5 FEASIBILITY STUDIES

From a technological point of view, the project requires a dynamic website.

The word dynamic here must mean that the content must be modified in real-time, it needs to be accessible for everyone with internet access, and should be a simple website displaying categories of solutions displaying fixing problems.

It is operationally and economically feasible because it is a solution that already exists, it should be acceptable for everyone who wishes to use it or needs to use it and it is going to be free of charge to use, the platform is not going to request any personal data from the user apart from his general requirements such as email address and a password.

The most complex task about this system apart from all the technical requirements will be the video management and the correlated comment section with each video, in addition to videos, there will be a comment section available for other users to give opinions or other ways possible for fixing the problem or in general just feedback if the solution is good or bad.

The difficult part would be to get more people involved in the website since it works better if there are more people to submit more solutions, reviews, and comments to contribute to better tutorials and more knowledge on how to repair or replace software/hardware. The project is economically viable it is not expensive to create or expensive to sustain, the only economic finances that are needed are for using a server to sustain the website and make it work 24/7.

Currently, we are not focusing on the economic feasibility of the project since it is just a prototype as part of the final thesis, no focus will be put on economic plans.

But never the less there are ways to cover such economic spectrums, such as the idea of importing some type of advertisements on the software where it would be visible partially on some parts of the page where it would not disturb the viewer or make their experience unenjoyable this would cover some expenses to a certain extent, other options are acquiring some sponsor deals, depending on how big the sponsor may be the income and coverage would be greater.

6 REQUIREMENTS ANALYSIS

In software engineering and systems engineering, requirements analysis focuses on the tasks that determine the needs or conditions to meet the new or altered product or project. Requirement analysis is critical to the success or failure of a systems or software project. The requirements should be documented, actionable, measurable, testable, traceable, related to identified business needs or opportunities, and defined to a level of detail sufficient for system design.

With this type of analysis, we try to understand the concepts of user requirements and system requirements and why these requirements should be written in different ways, we must understand the difference between functional and non-functional software requirements.

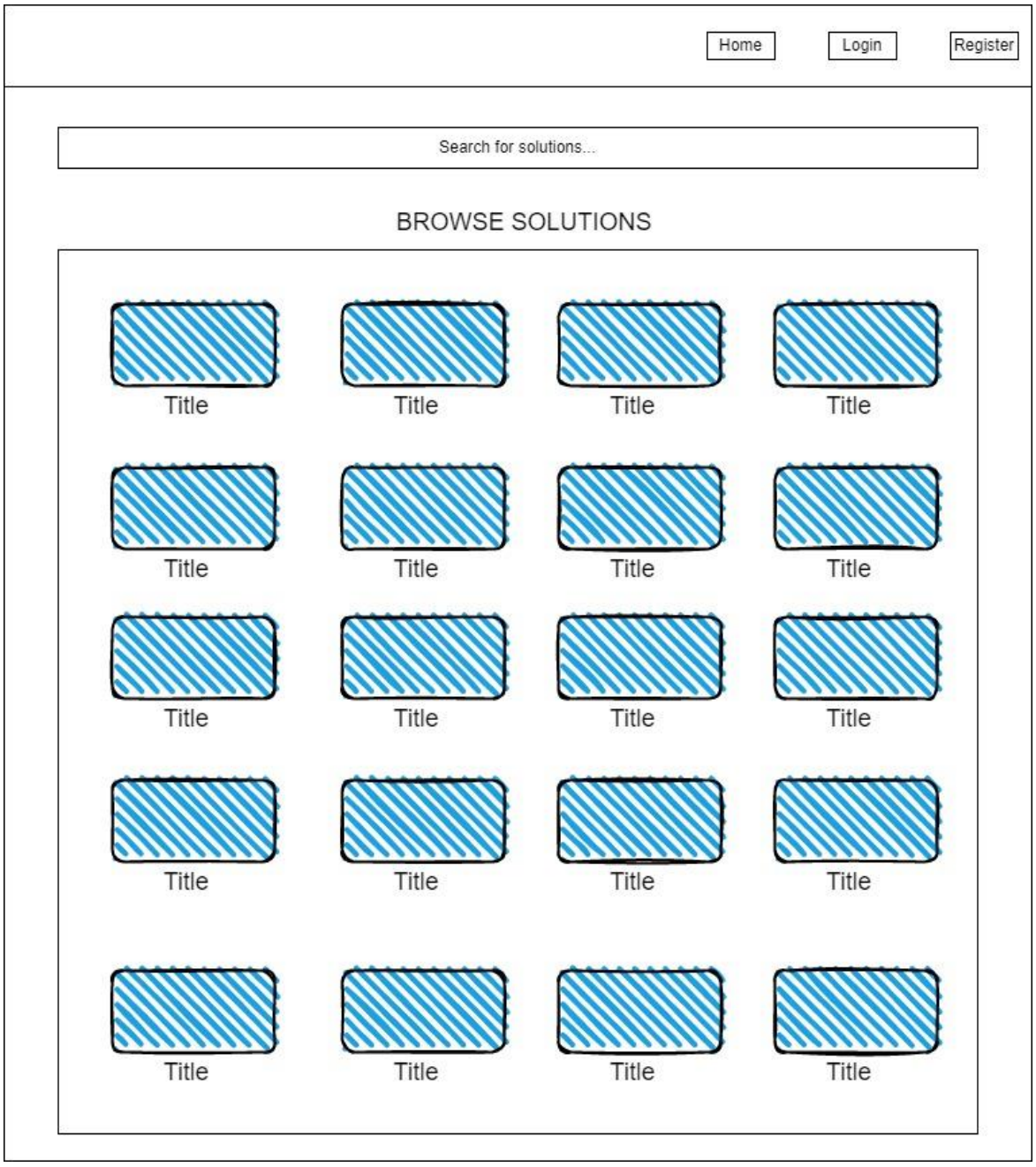

Figure 4: Website home page mockup

Before we move forward with the rest of the analysis, we would like to present a mockup of how we thought of the shape and looks of our website, Figure [4]. Starting with the home page of the software, it is represented with standard login and register configuration, where we can become a member of the community of our website to operate on it fully.

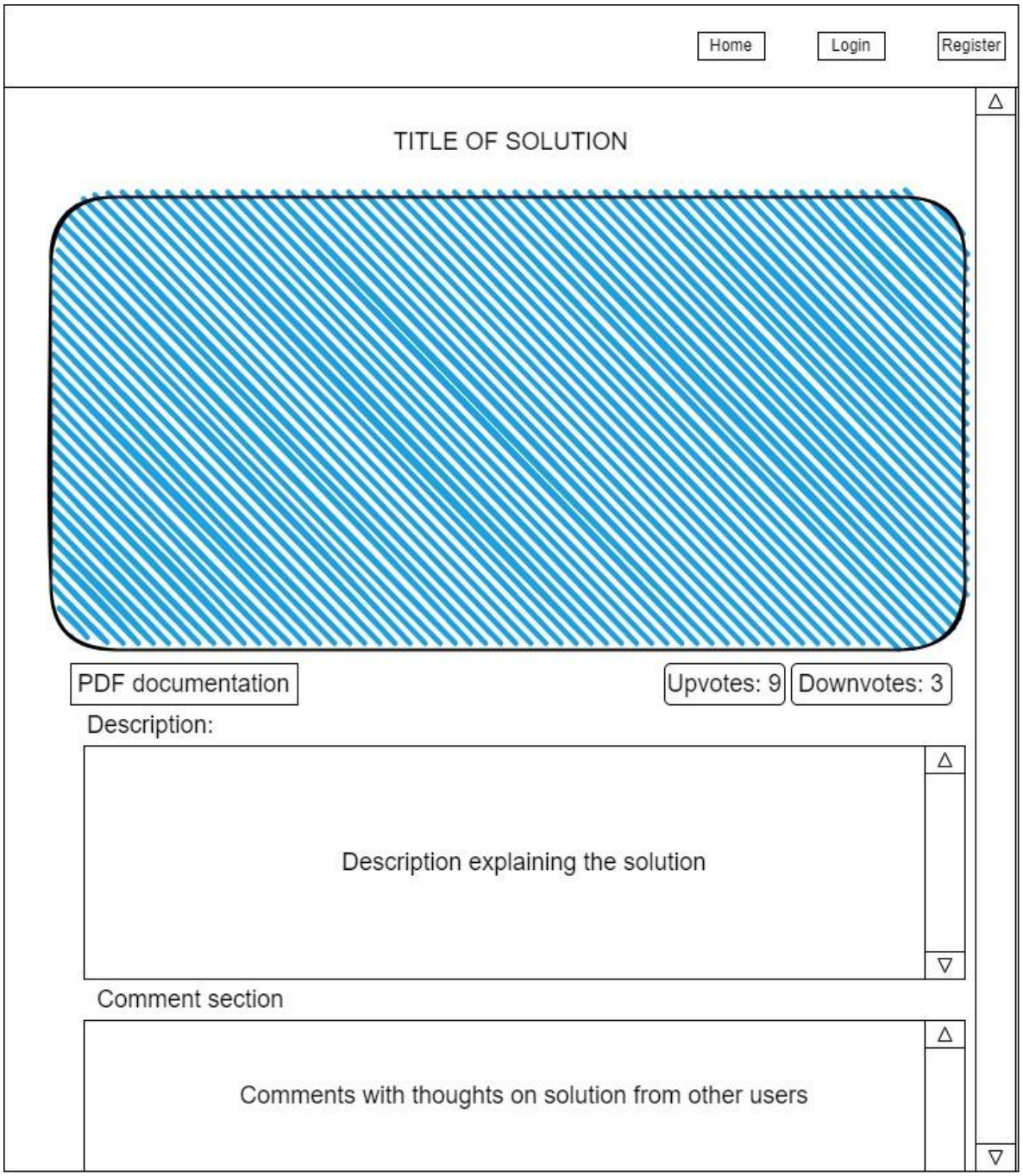

Figure 5: Website solution page mockup

Moving on to the second half we can notice Figure [5], a window where one of the solutions is opened and has many description components such as description, pdf documentation, comments, ratings, and so on. We will discuss and explain all of the properties shown in the mock-up further in the thesis.

6.1 Functional Requirements

These are statements of services the system should provide, and how the system should react to particular inputs and how the system should behave in particular situations. In some cases, the functional requirements may also explicitly state what the system should not do. When expressed as user requirements, the requirements are usually described in a fairly abstract way. However, functional system requirements describe the system function in detail, its inputs, outputs, exceptions, and so on. Functional requirements for a software system may be expressed in several ways, in our case, we will display them with a Use case diagram and its own description representing and explaining how it works.

6.1.1 Use Case Diagram

Use-cases Figure [6] identify the individual interactions of the user with the system, they can be documented with text or linked to UML models that develop the scenario in more detail. In its simplest form, a use-case identifies the type of interaction and the actors involved.
Here we can vividly see our representation of how our use cases of the website are linked and used in correlation with the user and administration, regarding each use case furthermore will be presented with use case descriptions.

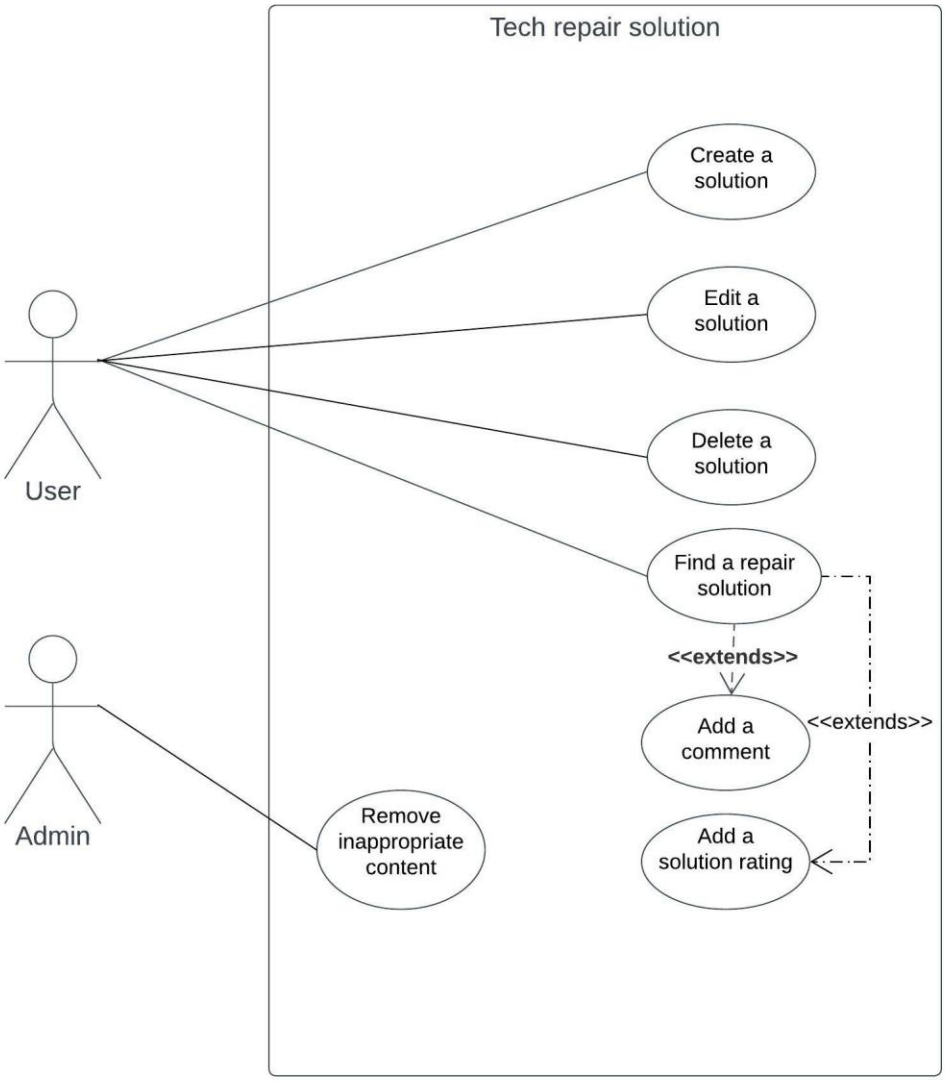

Figure 6: Use Case Diagram

6.1.2 Use case descriptions

A use case is a written description of how users will perform tasks on our software. It outlines, from a user's point of view, a system's behavior as it responds to a request. Each use case is represented as a sequence of simple steps, beginning with a user's goal and ending when that goal is fulfilled. Use cases add value because they help explain how the system should behave and, in the process, they also help brainstorm what could go wrong. They provide a list of goals and this list can be used to establish the cost and complexity of the system [3].

Table 1: Use case description of "Create solution" use case

Name	Create solution
Actor	User
Requirements	User – System should allow user to propose a solution
Trigger	Clicking upload button
Basic flow	1. New window opens where user will do upload and creation of a solution. 2. The user uploads their video solution. 3. Thumbnail must be inserted by the user. 4. Title is inserted by the user since it is mandatory for uploading the solution (which contains name of device and what is being fixed). 5. Description must be entered by the user since it is also mandatory, containing images, links to other solutions, documentation and text explanation 6. Clicking finish button checks if mandatory fields are fulfilled. -
Postconditions	-Solution is available to other users. -Solution is stored in a database.
Alternative flows	1a: The system recognizes an invalid video format being uploaded. 1a1. The system logs the error and the use case ends. 2a: The system recognizes an invalid thumbnail file format being uploaded. 2a1. The system logs the error and the use case ends. 3a: The description is left empty which is required to be fulfilled. 3a1. The system logs the error and the use case ends. 4a: The system recognizes an invalid or empty Title. 4a1. The system logs the error and the use case ends.

Table 2: Use case description of "Edit solution" use case

Name	Edit solution
Actor	User
Requirements	User – System should allow user to edit their created solution
Trigger	Clicking edit button on solution
Basic flow	1. The user opens their uploaded content. 2. Select the solution of their choice. 3. Clicks edit button. 4. New window opens. 5. Begin with editing of already existing sections. 6. Enter a new title. 7. Upload a different thumbnail. 8. Edit description. 9. Clicking finish button checks if mandatory fields are fulfilled.
Postconditions	-Edited solution is available to other users. -Changes are updated in a database.
Alternative flows	1a: The system recognizes an invalid thumbnail file format being uploaded. 1a1. The system logs the error and the use case ends. 2a: The description is left empty which is required to be fulfilled. 2a1. The system logs the error and the use case ends. 3a: The system recognizes an invalid or empty Title. 3a1. The system logs the error and the use case ends.

Table 3: Use case description of "Delete solution" use case

Name	Delete solution
Actor	User
Requirements	User – System should allow user to delete their existing solution
Trigger	Clicking delete button on solution
Basic flow	1. The user opens their uploaded content. 2. Select the solution of their choice. 3. New window opens. 4. Clicks delete button. 5. Confirmation of type "Are you sure?" appears where user must click confirm to continue. 6. Solution is removed.

Postconditions	- Solution is no longer available to other users. - Database is updated and sets the status of the solution to “deleted” and after a short period it is removed.
Alternative flows	1a: The system displays confirmation of type “Are you sure?” in order to confirm the user actually wants to remove their solution. 1a1. The user declines the deletion.

Table 4: Use case description of “Find a repair solution” use case

Name	Find a repair solution
Actor	User
Requirements	User – System should allow user to find any solution they require if its available
Trigger	Browsing and opening a solution or searching and opening a solution
Basic flow	1. The user browses through the home page, or goes through the search bar. 2. User find a solution on home page, or what they searched for from the search bar. 3. User views the solution either video or documentation format, or both if documentation is provided in the description by the creator of solution.
Postconditions	-Solution is available as long as user which uploaded it does not remove it. -Content is available as long as Admin deems them not inappropriate.
Alternative flows	1a: The system search does not find searched solution. 1a1. The system displays message informing no search matches the find. 1b: The system search does match an empty search. 1b1. The system displays message informing the search box cannot be empty. 2a: The user browses through the home page finds and selects a solution. 3a: The user searches through search bar finds and selects a solution.

Table 5: Use case description of "Add a comment" use case

Name	Add a comment
Actor	User
Requirements	User – System should allow user to add a comment on an existing solution
Trigger	User starts writing comment in comment box
Basic flow	1. The user locates the comment section under the solution. 2. User selects the new comment window. 3. User types a comment in the window. 4. User clicks submit button on comment window. 5. New comment is displayed under solution.
Postconditions	-Comment made by user stays as long as user does not change it or remove it. -Comments are available as long as Admin deems them not inappropriate.
Alternative flows	1a: The user selects the edit button of their comment: 1a1. User edits the content of their comment as long as the comment window it not empty. 1a2. If user leaves empty window while editing, will display error. 2a: The user leaves an empty comment: 2a1. The system logs that it's an invalid input. 3a: The user selects delete button of their comment and removes it.

Table 6: Use case description of "Add a solution rating" use case

Name	Add a solution rating
Actor	User
Requirements	User - System should allow user to add a rating on an existing solution
Trigger	Clicking the rating button
Basic flow	1. The user locates the rating system under the solution. 2. The user sees the rating ratio numbers, which represent how good of a rating the solution has. 3. The user selects an upvote or a downvote. 4. The selection is visible on the rating numbers.

Postconditions	-Rating stays the same as long as user does not change it or remove it.
Alternative flows	1a: The user clicks on their selected option of the rating to be able to remove it.

Table 7: Use case description of "Remove inappropriate content" use case

Name	Remove inappropriate content
Actor	Admin
Requirements	Admin – System should allow admin to remove inappropriate content when he finds and deems them going against guidelines
Trigger	Admin selects and deletes inappropriate content
Basic flow	1. The admin browses through solutions regularly. 2. Admin observes other solutions. 3. Admin checks reports of inappropriate content from other users. 4. Found inappropriate solutions. 5. Solution is removed. 6. Found inappropriate comments. 7. Comment is removed.
Postconditions	-Solution is removed by admin for breaking guidelines. -Comment is removed by admin for breaking guidelines.
Alternative flows	1a: The admin updates the community guidelines. 1a1. The admin adds new guidelines. 1a2. The admin removes guidelines.

6.2 Non-Functional Requirements

These are constraints on the services or functions offered by the system. They include timing constraints, constraints on the development process, and standards. Non-functional requirements often apply to the system as a whole. They do not usually just apply to individual system features or services. Non-functional requirements, as the name suggests, are requirements that are not directly concerned with the specific functions delivered by the system, they may relate to emergent system properties such as reliability, response time, and store occupancy. Alternatively, they may defll1e constraints on the system such as the capabilities of devices and the data representations used in system interfaces.

Non-functional requirements are rarely associated with individual system features therefore, they may specify system performance security, availability, and other emergent properties. This means that they are often more critical than individual functional requirements. System users can usually find ways to work around a system function that doesn't really meet their needs.

6.2.1 Operational requirements

- The system shall be able to contain 100-200 project uploads per minute.
- The system shall store the date when a project was uploaded, also a date on each comment to have information of how current the comment or project is.
- The system shall ensure 24/7 operation with an uptime of at least 98%.
- The system shall ensure the speed of the search engine to respond immediately after text is entered.
- The system shall operate using an English language template
- Access to the system will be available to anyone and everyone since it is a free website any person can create an account.
- The privacy requirements for creating an account in the system are only: Name and E-mail address.
- The system requires ordinary users, who either search for help or give help to people in need, the other users are the administrators.
- The system shall provide security of their data where it will only be stored in a database.
- The system shall include a tutorial on how to operate the website.
- The system shall provide guidelines available to be read.
- The system shall provide extensibility allowing the maintainer to add additional features and customizations in the future.

7 SYSTEM DESIGN

System design is the process of designing the elements of a system such as the architecture, modules and components, the different interfaces of those components, and the data that goes through that system, the activity of translating an SRS into a system design model, the system design model divides the system into subsystems (components), describe their behavior, their relationship, and interfaces. Subsystems provide services to other subsystems the service is a group of related operations. When designing a system (architecture) it is necessary to define subsystems in terms of the service that subsystems provide.

7.1 Component Diagram

In UML, a component diagram Figure [7] depicts how components are wired together to form larger components or software systems, they are used to illustrate the structure of arbitrarily complex systems. A component diagram allows verification that a system’s required functionality is acceptable.

One way of illustrating the provided and required interfaces by the specified component is in the form of a rectangular compartment attached to the component element. Another accepted way of presenting the interfaces is to use the ball and socket graphic convention.

A provided dependency from a component to an interface is illustrated with a solid line to the component using the interface from a "lollipop", or ball, labeled with the name of the interface. A required usage dependency from a component to an interface is illustrated by a half-circle, or socket, labeled with the name of the interface, attached by a solid line to the component that requires this interface.

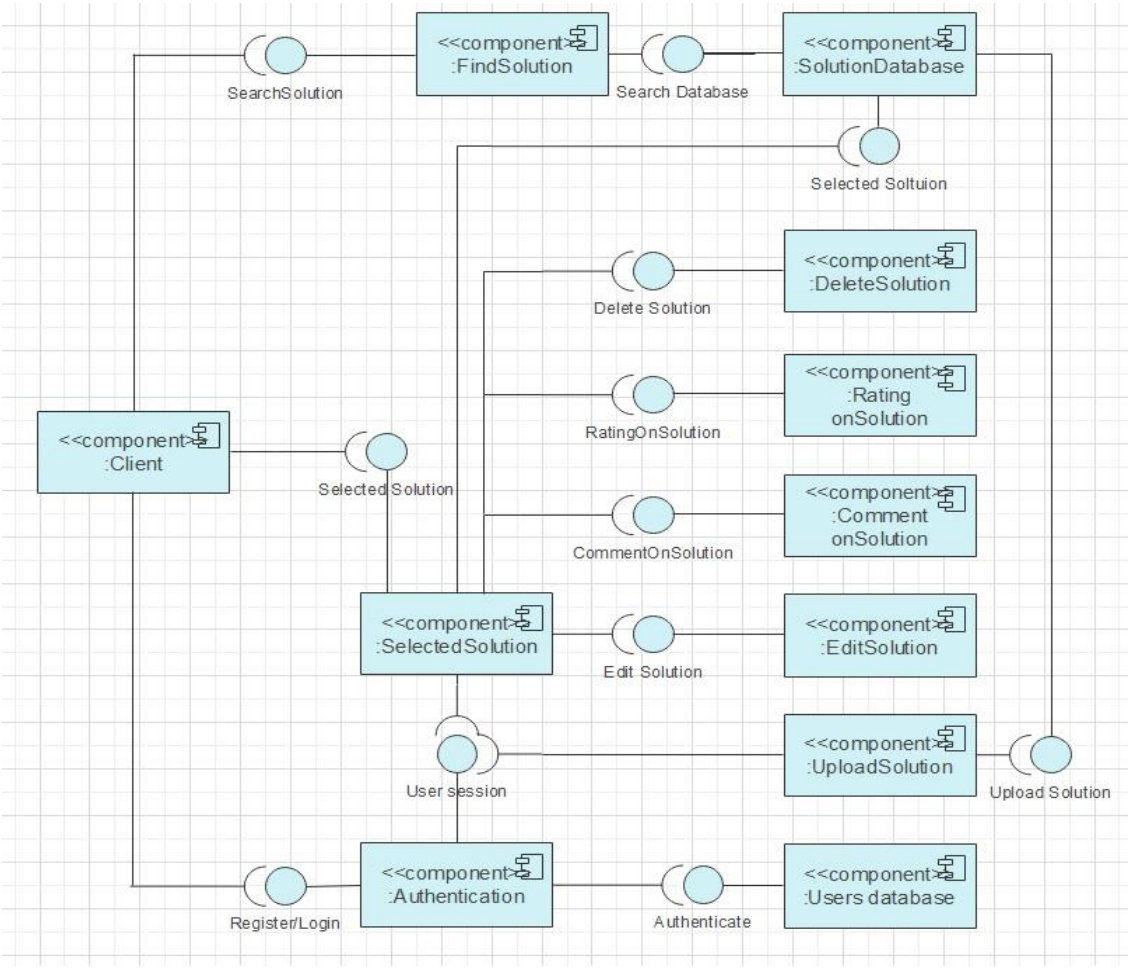

Figure 7: Component Diagram

7.2 Component Design

With component designing we get to provide further information on each component which is being displayed in Figure [5], allowing us a better understanding of how each component is connected with each other informing what this component does and with all of the combined what the system design beholds.

- Client:

- The person using the software, allows him to start searching for a solution or the other option of registering/logging in to the system and allowing the user more functionalities available on the web services.

- Find Solution:

- Can be accessed either if the client is logged in or not, allowing the user to either locate or search through a search bar for a given solution.

- Database (SQL):

- The database component is where all of the video solutions are stored, but firstly there have to be some input solutions through “Upload Solution” which will update the database with available solutions to choose from.

- Selected Solution:

- Implying the selected solution which was previously found by the user, allowing other dependencies for other components.

- Delete Solution:

- For this component to function a solution has to be selected, afterwards the option to delete the solution is available to the user who uploaded it.

- Rating on Solution:

- For this component to function a solution has to be selected, meanwhile, any user including the owner of the solution can give their opinion and rate how useful the solution was.

- Comment on Solution:

- For this component to function a solution has to be selected, while the user has to be logged in, they can leave their opinion and thoughts on the solution via a comment section where each user can leave as many comments as they desire.

- Edit Solution:

- For this component to function a solution has to be selected, and the user has to be the owner of the solution in order for an editing option to be available, allowing the user to change the description, title, or thumbnail.

- Upload Solution:

- Allowing the user to select a solution upload it, insert a title according to the solution, select a thumbnail appropriate for the solution, and insert a description describing the solution and the option to link other solutions such as YouTube links, Reddit posts, documentation and other.

- Authentication:

- This component authenticates if the user credentials are correct for logging in, or while registering to confirm if the credentials are already used or not.

- User database:

- This component stores the credentials of the users who are members of the software.

8 SYSTEM IMPLEMENTATION

With the implementation as envisioned in the proposed problem solution, it will be a website. This system will contain several functionalities such as Register, Login, search for a solution, upload solution, view other solutions, comment on a solution, rating a solution etc.
By using the following technologies: HTML, CSS, JavaScript, PHP, SQL, and GIT the software was made possible.

8.1 Technologies Used

HTML: Is a markup language that defines the structure of your content, it consists of a series of elements that you use to enclose or wrap different parts of the content to make it appear a certain way, or act a certain way. It is used for creating code to structure a web page and its content. Using HTML in our PHP file we were able to create the home page of the website [4].

CSS: Is a simple design language intended to simplify the process of making web pages presentable, CSS handles the look and feel part of a web page. Using CSS we can control the color of text, style of fonts, spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs, variations in display for different devices, and screen sizes as well as a variety of other effects. Using CSS, we were able to create a much more presentable website by doing some simple styling in order for better looks [5].

JavaScript: Is a scripting or programming language that allows us to implement complex features on our web pages, every time a web page does more than just sit there and display static information for us to look at, displaying timely content updates, animations, graphics, scrolling jukeboxes, etc. We can surely assume that JavaScript was probably involved. Using JavaScript, we were able to implement all of the logic and functionalities of the website easily, allowing us to create any feature that we desire [6].

PHP: Is a widely used open-source general-purpose scripting language that is especially suited for web development and can be embedded into HTML. Instead of lots of commands to output HTML, PHP pages contain HTML with embedded code, the PHP code is enclosed in special start and end processing instructions that allow you to jump into and out of “PHP mode”. By mainly using PHP we were able to create our implementation [7].

SQL: Is used to communicate with a database where all of our data is stored, it is the standard language for relational database management systems. SQL statements are used to perform tasks such as updating data on a database or retrieving data from a database. We created tables and mainly used SQL databases to store our data and access it [8].

GIT: Is a free and open-source distributed version control system designed to handle everything from small to very large projects with speed and efficiency. Git was used for creating a project which has high performance [9].

Libraries: In our implementation, the following libraries were a part of it: jQuery which is a fast, small and feature-rich JavaScript library, Bootstrap which is a free and open-source front-end development framework which is built on HTML, CSS, and JavaScript.

8.2 Website Functionalities

As a starting point, I would like to first present what the implementation of our software looks like so that we can get a better understanding and the ability to dissect each functionality and for better visualization of what is going on. Here we can find the home page of our software where we can clearly see in the top right corner that each person using the software has the ability to either Register or Login depending if they are a returning user or if they wish to have access to additional functionalities. Depending on that if a person is not registered, they will be able to view the software as any registered person, therefore we can vividly see our home page has some solutions already available from us to choose a solution and view it but that person will not be able to use some functionalities.

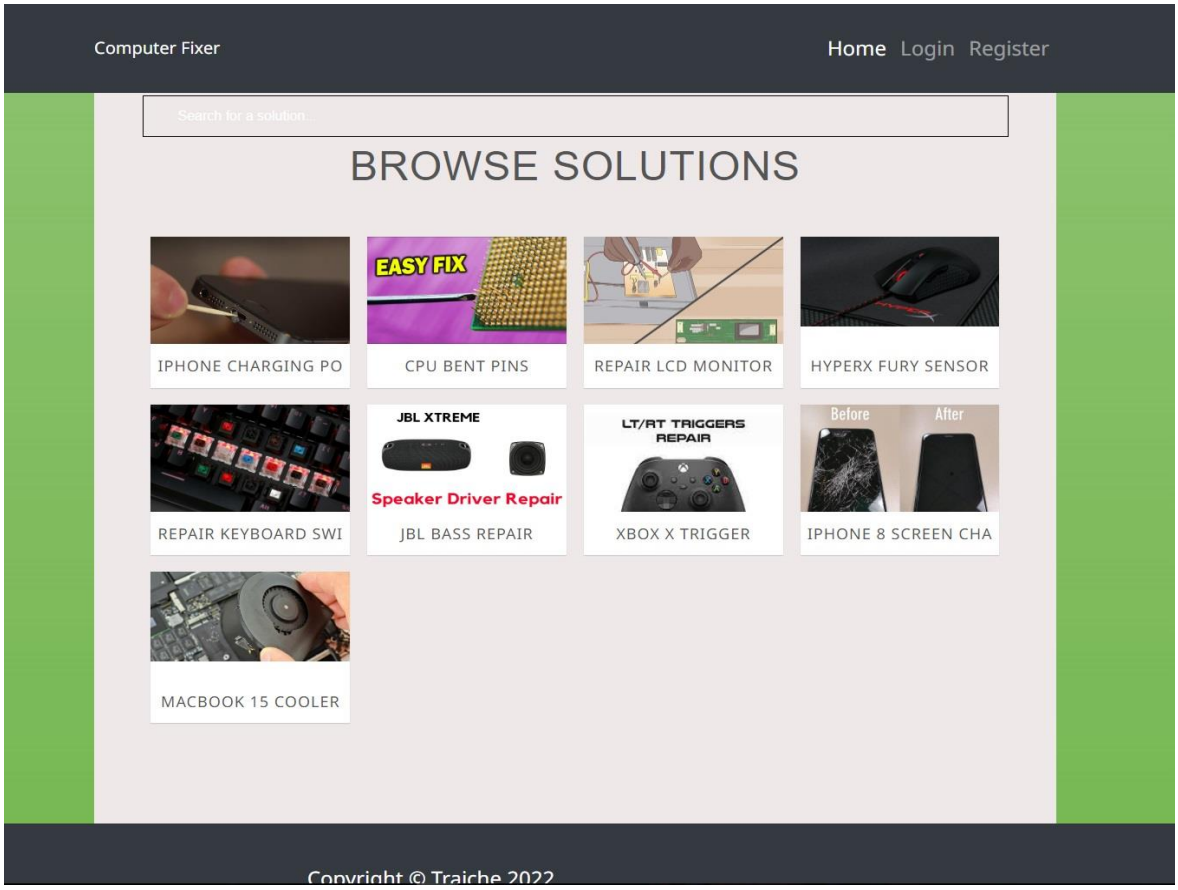

Figure 8: Home page

Simply by browsing through such solutions we can select one free of choice if the desired solution is visible, if for example there are too many uploads done by other users, simply

we can use our search engine which allows the person using the software to search by title name giving the ability of free choice on any solution as we can see on Figure [8].

Figure 9: Search engine

Once the desired solution is selected Figure [9], there are limitations to the functionalities available to the person using the software as mentioned before because they are not a registered user and not logged in therefore, not having all functionalities available such as the ability to leave comments or rating on the selected solution, these functionalities will be discussed separately in better details when we move on.

After our selection we can view the solution which was published by a user, below the solution it contains a description box containing details regarding the solution, upvote and downvote boxes containing ratings of other users, the additional pdf documentation regarding the solution (if it is necessary and uploaded by the user), and finally a comment section where each user can give their feedback and thoughts on the selected solution.

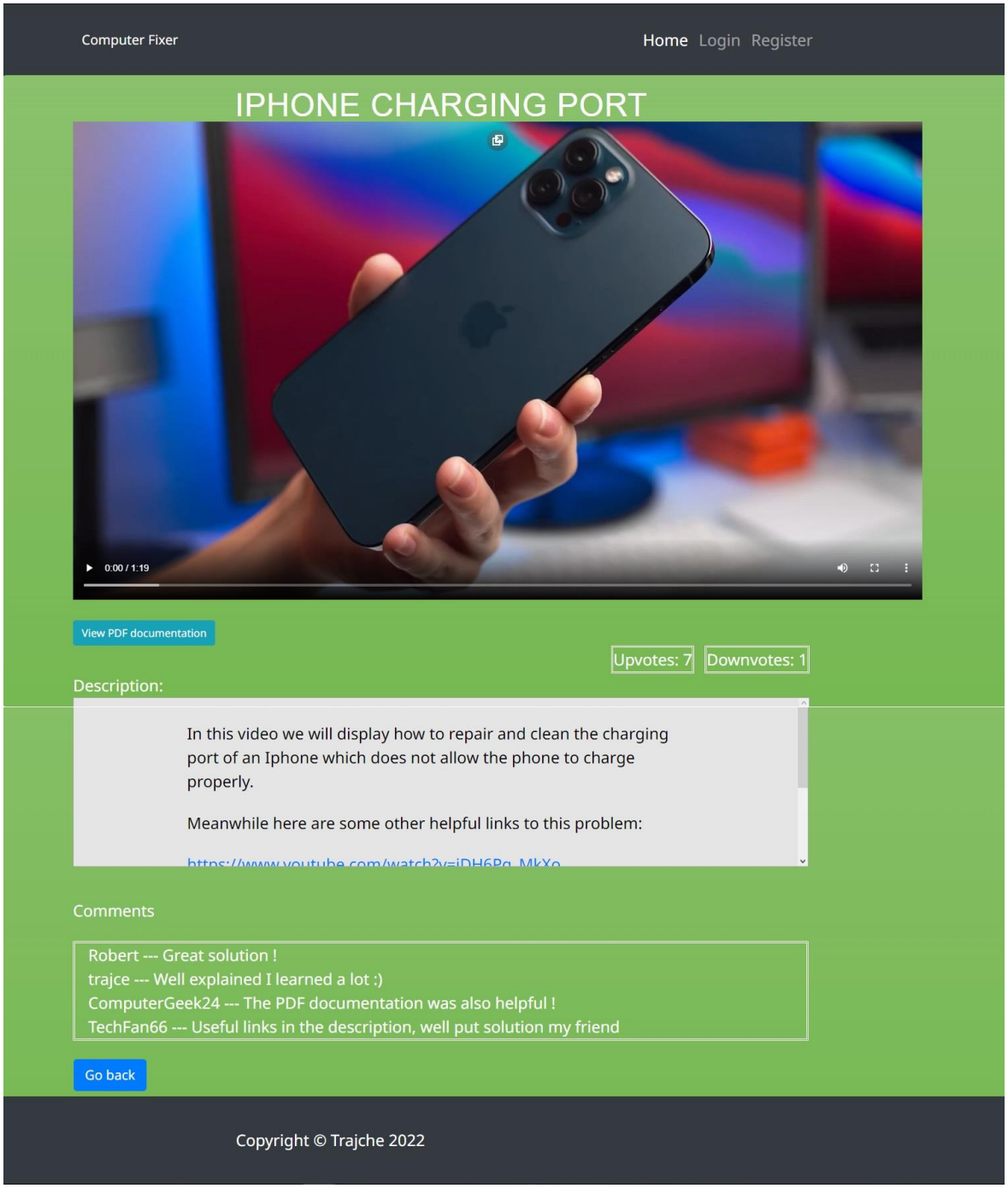

Figure 10: Solution preview

But in another case Figure [10], if the person using the software is registered and signed in, they can use the other functionalities such as leaving comments and rating the solution we can see an image example in Figure [11].

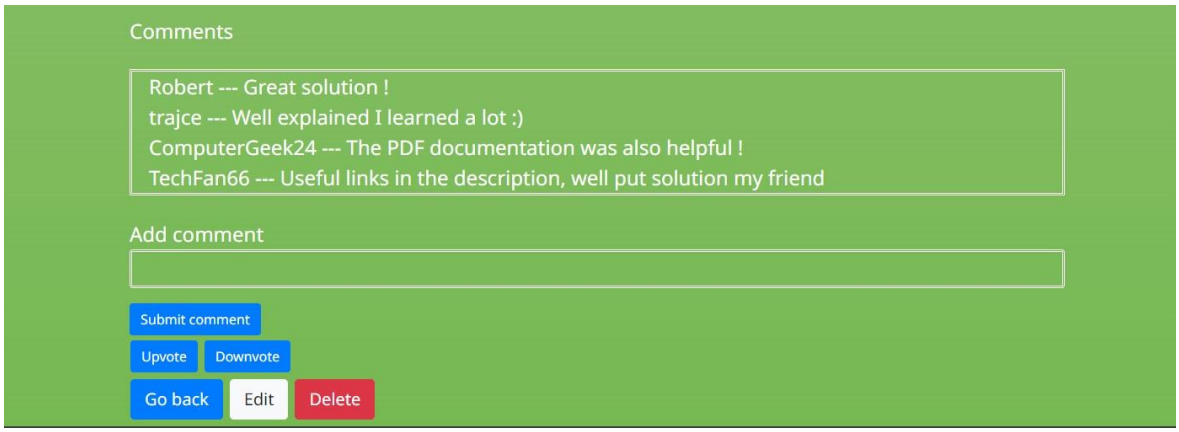

Figure 11: Logged in user

```
<div class="container" style="margin-top: 130px;">

    <div class="page-header">
        <?php
        if(isset($_GET['title'])) {
            $title = urldecode($_GET['title']);
            echo "<h1 id='video-title'>$title</h1>";
        } else {
            echo '<h1>Video player</h1>';
        }
        ?>
    </div>

<?php
if(isset($_GET['url'])) {
    $url = urldecode($_GET['url']);

    echo '<video src="' . $url .'" type="video/mp4" controls></video>';

}
else{
    header('location: index.php');
}
?>
```

Figure 12: Video code example

For our video presentation Figure [12], no special library was used and was formatted fairly simple, when the video is uploaded it is uploaded within a destination directory where it is stored, having a unique ID and URI which would be the destination of this video.

When the corresponding title is presented in connection to the same ID of the video by matching the URI of the video it will simply be displayed on the page.

As we are familiar with how the system looks and works, we would like to present each functionality in bigger detail to explain how the complete software works.

- Login and Register

As with any commonly used website, the person using it most likely would need an account to become a user of the website Figure [13]. With this functionality we allow any person using our website to register with a valid name, email address, and unique password, making them a unique user and getting the ability to access additional functionalities of the website.

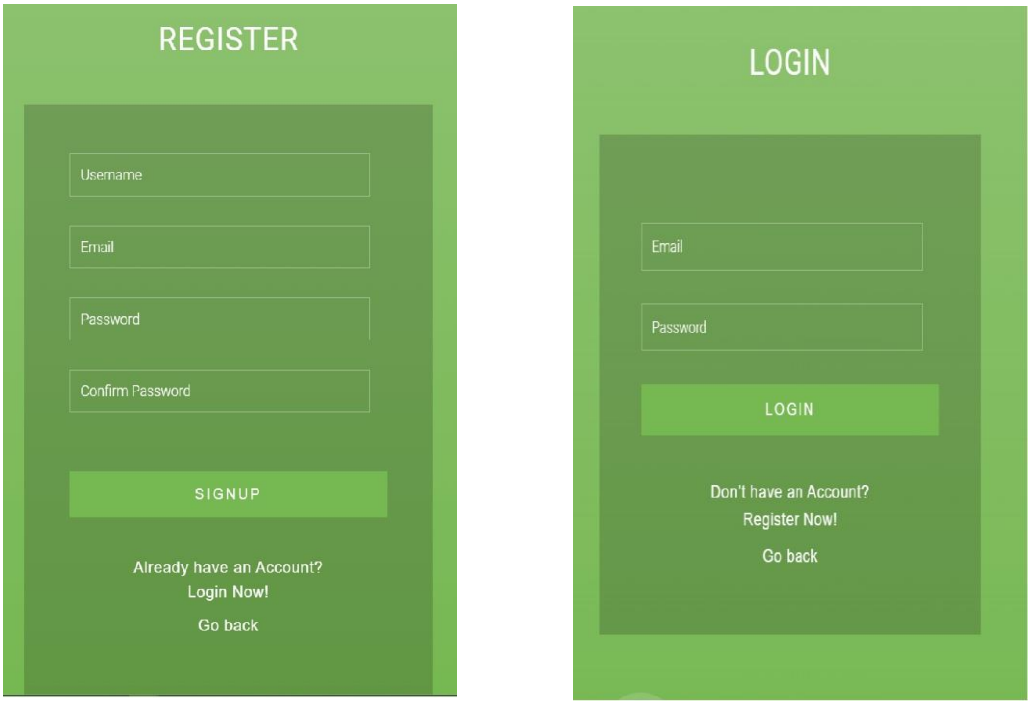

Figure 13: Login and Register window

```php
// REGISTER USER
if (isset($_REQUEST['form_action'])) {
    $form_action = $_REQUEST['form_action'];
    if($form_action == "register") {
        // receive all input values from the form
        $user = mysqli_real_escape_string($db, $_POST['user']);
        $email = mysqli_real_escape_string($db, $_POST['email']);
        $password1 = mysqli_real_escape_string($db, $_POST['password1']);
        $password2 = mysqli_real_escape_string($db, $_POST['password2']);

        // form validation: ensure that the form is correctly filled
        if (empty($user)) { array_push($errors, "Username is required"); }
        if (empty($email)) { array_push($errors, "Email is required"); }
        if (empty($password1)) { array_push($errors, "Password is required"); }

        if ($password1 != $password2) {
            array_push($errors, "Passwords must match");
        }

        // register user if there are no errors in the form
        if (count($errors) == 0) {
            $query = "SELECT * FROM users WHERE email='$email'";
            $results = mysqli_query($db, $query);

            if (mysqli_num_rows($results) == 1) {
                array_push($errors, "There is already an account with the given email.
                                     Please login or use a different email!");
                header('location: ../index.php');
            }

            $password = md5($password1);//encrypt the password before saving in the database
            $query = "INSERT INTO users (username, email, password)
                      VALUES('$user', '$email', '$password')";
            mysqli_query($db, $query);

            $query = "SELECT id FROM users WHERE email='$email'";
            $results = mysqli_query($db, $query);

            if (mysqli_num_rows($results) == 1) {
                $id = mysqli_fetch_assoc($results);
                $_SESSION['user'] = $email;
                $_SESSION['userid'] = $id['id'];
                $_SESSION['success'] = "You are now logged in";
            }
            else {
                array_push($errors, "Failed to register user in database,
                                     please try again or contact system admin!");
            }
            header('location: ../index.php');
        }
        else{
            header('location: users.php?errors=' . urlencode(serialize($errors)));
```

Figure 14: Register code example

Presenting the registering process through our code, we can notice we have a variable for each field: user, email, and two for the password to be able to have verification if it is matching, we must have each field filled in otherwise we cannot proceed.

After the verification process and everything matches, we check through the database to see if the current email is already in use otherwise, we proceed by encrypting the password and saving it in the database, therefore providing better and more optimal security and safety for the user, this way even the maintainer will not be able to view any personal data of the user, Figure [14].

```php
    else if($form_action == "login"){
        $email = mysqli_real_escape_string($db, $_POST['email']);
        $password = mysqli_real_escape_string($db, $_POST['password1']);

        if (empty($email)) {
            array_push($errors, "Username is required");
        }
        if (empty($password)) {
            array_push($errors, "Password is required");
        }

        if (count($errors) == 0) {
            $password = md5($password);
            $query = "SELECT * FROM users WHERE email='$email' AND password='$password'";
            $results = mysqli_query($db, $query);

            if (mysqli_num_rows($results) == 1) {
                $id = mysqli_fetch_assoc($results);
                $_SESSION['user'] = $email;
                $_SESSION['userid'] = $id['id'];
                $_SESSION['success'] = "You are now logged in";
                header('location: ../index.php');
            }else {
                array_push($errors, "Wrong username and/or password!");
            }
        }
        else{
            header('location: users.php?errors=' . urlencode(serialize($errors)));
        }
    }
    else{
        array_push($errors, "Invalid action!");
    }
```

Figure 15: Login code example

By the login code example Figure [15], we notice the similarities with the register code, since we are going through a similar process of checking and verifying if the existing data of a user matches the inputted data by the user in order to have a valid login.

Otherwise, we display a login error.

- Search Solution

  As mentioned before about browsing through the home page this time we will take the case if the user is registered and logged in. By browsing the home page each user has the ability to select a solution free of choice, whereas if they are unable to find the desired solution to their problem, they can simply use the functionality of our search engine and have the ability to search by title name of the given solution. After the solution is found the user possesses the choice of selecting it and opening this solution, afterwards as seen in the previous figures the user can watch the video solution, read the description inputted by the user, read the PDF file documentation inputted by the user, and read other comments.

- Upload Solution

  By pressing the “Upload Solution” button Figure [16], the user has the ability to upload a solution of their choice, which will be in a video format, after the solution selection the user is obligated to insert a proper title corresponding to the solution they are making, input a valid thumbnail in relation to their solution, input a valid PDF documentation in correlation with the solution if that is necessary and finally leave a valid description explaining the solution they have created Figure [18], on the other hand, the user has the ability to link other solution which may be relevant to the solution they are creating.

  This is one of the main points of our solution allowing the ability to link multiple sources and links where it will be merged into one place to enable easier solution searching.

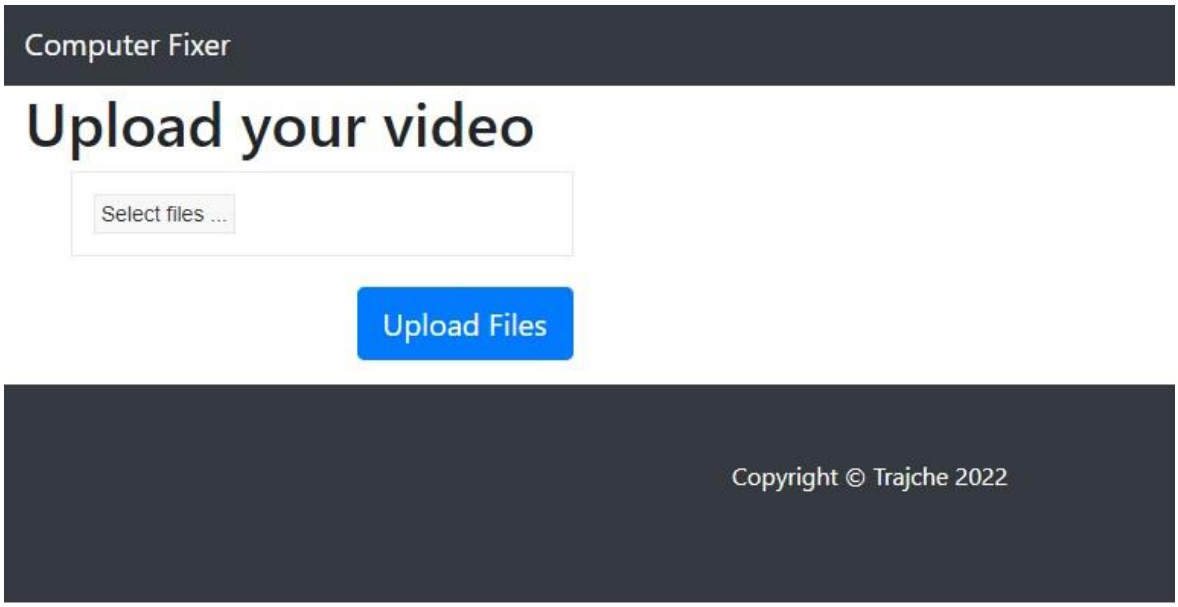

Figure 16: Upload solution part 1

```php
<?php
// Loop through each file
for( $i=0 ; $i < $total ; $i++ ) {
    if($_FILES['file']['size']<=0){
        continue;
    }
    //Get the temp file path
    $tmpFilePath = $_FILES['file']['tmp_name'][$i];

    //Make sure we have a file path
    if ($tmpFilePath != ""){
        //Setup our new file path
        $fname_base = substr($_FILES['file']['name'][$i], 0, strrpos($_FILES['file']['name'][$i], "."));
        $fname_ext = substr($_FILES['file']['name'][$i], strrpos($_FILES['file']['name'][$i], "."));
        //$newFilePath = $target_dir . $_FILES['file']['name'][$i] . strtotime("now");
        $newFilePath = $target_dir . $fname_base . "_" . strtotime("now") . $fname_ext;
        //var_dump($newFilePath);

        //Upload the file into the temp dir
        if(move_uploaded_file($tmpFilePath, $newFilePath)) {
            $query = "INSERT INTO videos (userid, filename)
                      VALUES('$userid', '$newFilePath')";
            mysqli_query($db, $query);

            $query = "SELECT id FROM videos
                      WHERE filename='$newFilePath'";
            $results = mysqli_query($db, $query);

            if (mysqli_num_rows($results) == 1) {
                $id = mysqli_fetch_assoc($results);
                $id = $id['id'];
                echo '<input id="input-videoid-' . $i . '
                " type="hidden" name="input-videoid-' . $i . '" value="' . $id . '">';
            }
        }
        ?>
```

Figure 17: Upload solution code example

Firstly, we have to check if the user is logged in Figure [17], after that verification, we loop through each file and get our file path where the uploaded content is being stored.

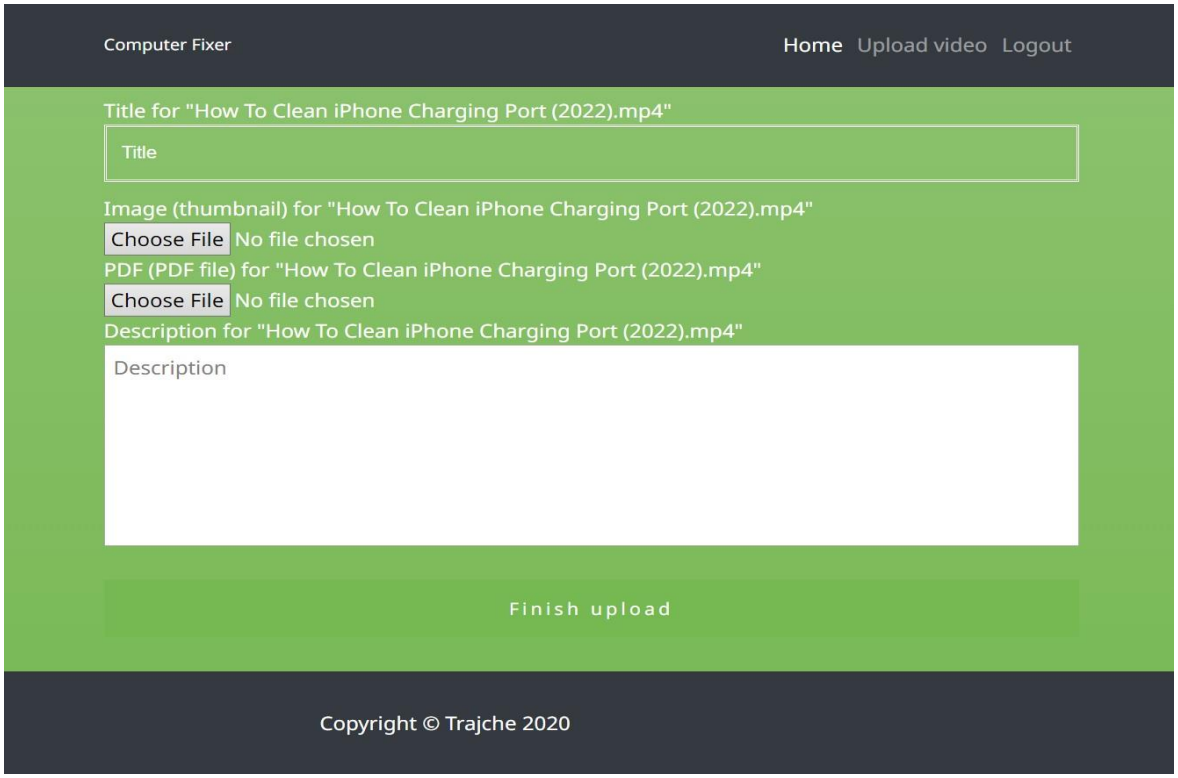

Figure 18: Upload solution part 2

```
?>
<label for="<?= "input-title-$i" ?>">Title for "<?= $_FILES['file']['name'][$i] ?>"</label>
<?php
echo '<input class="form-group" id="input-title-' . $i . '" type="text" name="input-title-' . $i . '" placeholder="Title"
    required="" style="width: 100%; border: double; background-position: -5000px;">';
?>
<label for="<?= "input-thumbnail-$i" ?>">Image (thumbnail) for "<?= $_FILES['file']['name'][$i] ?>"</label>
<?php
echo '<input type="file" id="input-thumbnail-' . $i . '
" placeholder="Select file" name="input-thumbnail-' . $i . '" required="">';
?>
<br>
<label for="<?= "input-pdf-$i" ?>">PDF (PDF file) for "<?= $_FILES['file']['name'][$i] ?>"</label>
<?php
echo '<input type="file" id="input-pdf-' . $i . '
" placeholder="Select file" name="input-pdf-' . $i . '" required="">';
?>
<br>
<label for="<?= "input-title-$i" ?>">Description for "<?= $_FILES['file']['name'][$i] ?>"</label>
<?php
echo '<textarea id="input-description-' . $i . '" type="text" name="input-description-' . $i . '
" placeholder="Description" class="form-control textarea-description summernote" required=""></textarea>';
?>
<textarea type="text" name="hiddenText" class="hiddenText" hidden>
                </textarea>
<div class="clear"> </div>
<div class="clear"> </div>
<?php
```

Figure 19: Upload solution code example part 2

After the solution is uploaded Figure [19], we create new fields that are required and can not be left empty, including title, thumbnail, pdf documentation, and description. All fields are important for a great solution.

- Comment on Solution

The comment functionalities give the user the possibility to share their own thoughts on the solution they are viewing Figure [20], never the less if its negative or positive feedback the comment section is a place for better understanding and improvement, other possibilities are constructive criticism if the solution can be improved in any sort, or perhaps another user may have found a better link or resource of information and they would desire to share it with the rest of the viewers of the same solution.

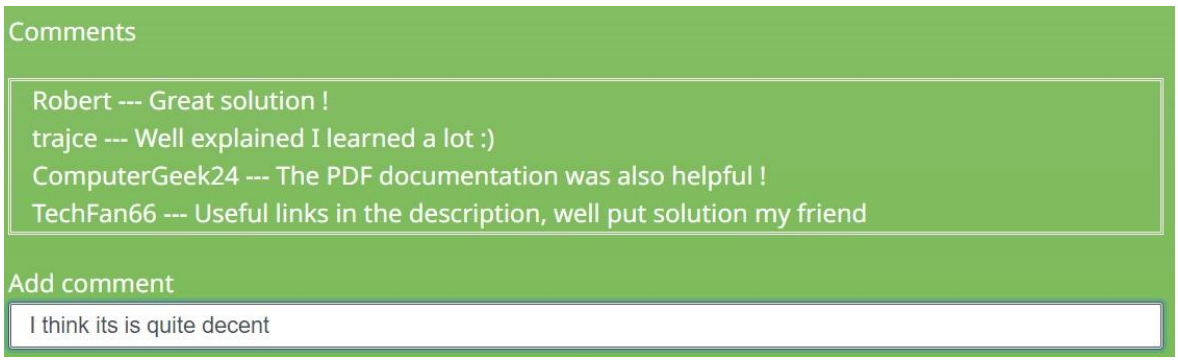

Figure 20: Comment section

- Rating on Solution

The rating system enables the user to express how satisfied they are with the solution Figure [21], depending if they like it, it is expected for the user to select an upvote to signal that the solution is informative and quite useful, on the other hand, it is quite the opposite if the solution is unsatisfactory and not well prepared the user can express a downvote to lower the rating of the solution. Therefore, ratings are important for other users which have not visited the solution signaling to the users if it is actually worth investing their time into this solution.

Figure 21: Rating system

- Edit solution

  The user that created the solution additionally has the ability to edit the content they have created Figure [22], they will be able to press the “Edit” button, and in a new pop-up by their choice they can change the title, thumbnail, pdf document, and description of the solution they created Figure [23].

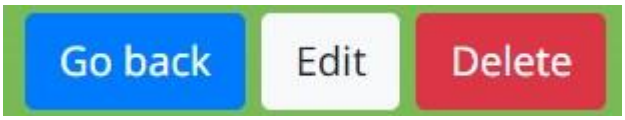

Figure 22: Edit button

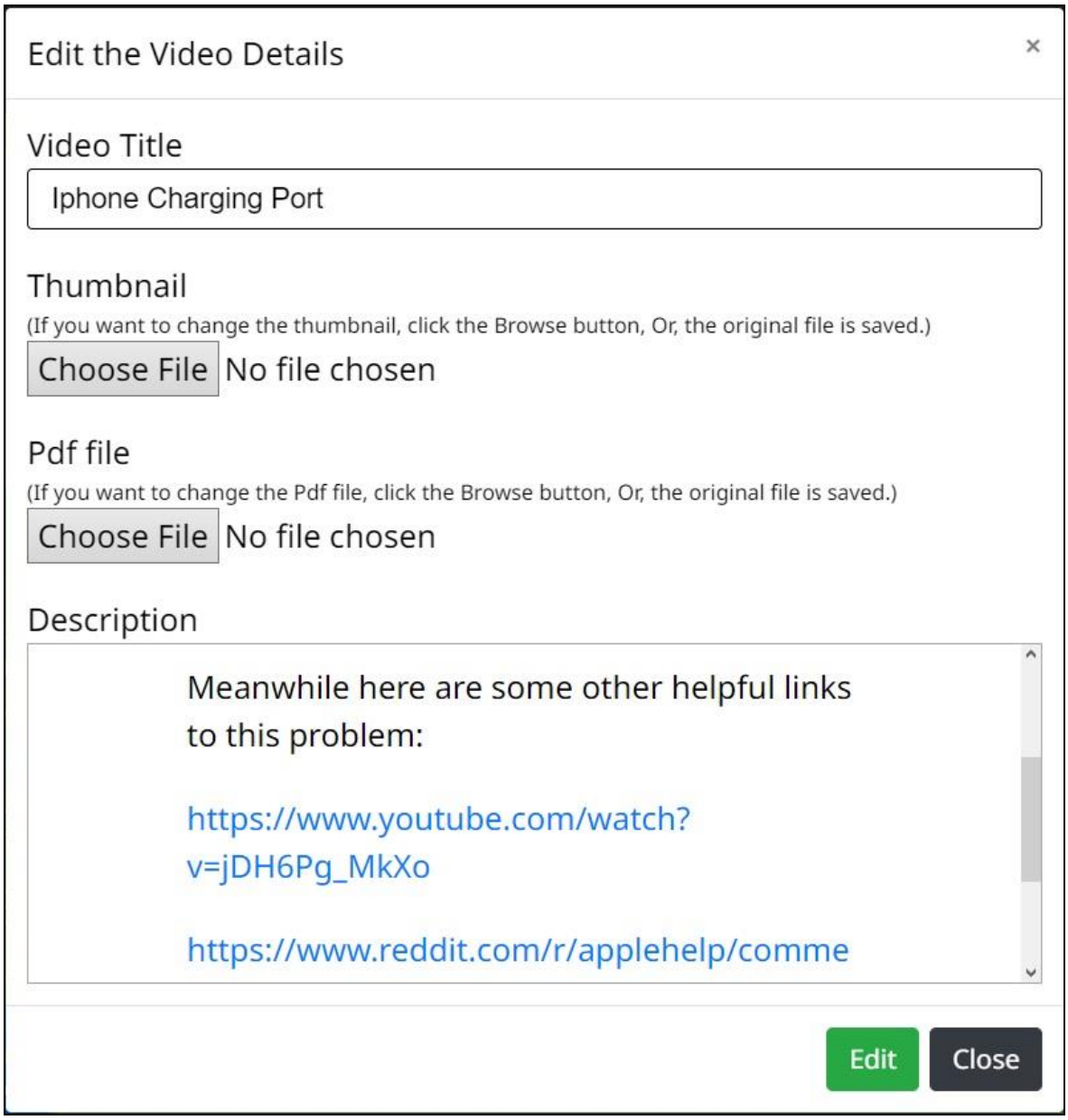

Figure 23: Content editor

```php
// Loop through each file
$id = $_POST['id'];
$title = $_POST['title'];
$descript = $_POST['hiddenText'];

$tmpFilePath = $_FILES['thumbnail']['tmp_name'];

if ($tmpFilePath != "") {
    $fname_base = substr($_FILES['thumbnail']['name'], 0, strrpos($_FILES['thumbnail']['name'], "."));
    $fname_ext = substr($_FILES['thumbnail']['name'], strrpos($_FILES['thumbnail']['name'], "."));
    $newFilePath = $target_dir . $fname_base . "_" . strtotime("now") . $fname_ext;

    if (move_uploaded_file($tmpFilePath, $newFilePath)) {
        $newFilePath = '../' . $newFilePath;
        $query = "UPDATE videodetails SET title='$title', thumbnail='$newFilePath', description='$descript'
              WHERE videoid='$id'";
        mysqli_query($db, $query);
    }
}
else {
    $query = "UPDATE videodetails SET title='$title', description='$descript'
              WHERE videoid='$id'";
    mysqli_query($db, $query);
}
$tmpPdfFilePath = $_FILES['pdf']['tmp_name'];

if ($tmpPdfFilePath != "") {
    $fname_base = substr($_FILES['pdf']['name'], 0, strrpos($_FILES['pdf']['name'], "."));
    $fname_ext = substr($_FILES['thumbnail']['name'], strrpos($_FILES['thumbnail']['name'], "."));
    $newFilePath = $target_dir . $fname_base . "_" . strtotime("now") . $fname_ext;

    if (move_uploaded_file($tmpFilePath, $newFilePath)) {
        $newFilePath = '../' . $newFilePath;
        $query = "UPDATE videodetails SET title='$title', pdf='$newFilePath', description='$descript'
              WHERE videoid='$id'";
        mysqli_query($db, $query);
    }
}
else {
    $query = "UPDATE videodetails SET title='$title', description='$descript'
              WHERE videoid='$id'";
    mysqli_query($db, $query);
}

header('location: index.php');
?>
```

Figure 24: Edit solution code example

We loop through each file and find the existing properties of the solution in our database and we update them accordingly to the desire of our newly input title, thumbnail, pdf documentation, and description their properties are updated and stored again in our database, Figure [24].

- Delete Solution

  Since any user can create and upload their solution it is only right for each user to have the ability to delete their own solution if they deem it is not sufficient enough or it may be has received negative feedback and is not as useful as they may have thought. Therefore, the user will have the ability to press the “Delete” button and an emergency message will appear where it will ask for confirmation if the user is sure that they actually want to delete what they have created otherwise it is a scary scenario to immediately delete without any safety precautions, Figure [25].

Figure 25: Delete solution

```php
$user_logged_in = false;
if(isset($_SESSION['user'])) {
    $user_logged_in = true;
    $user = $_SESSION['user'];
    $userid = $_SESSION['userid'];
}

if(isset($_POST['videoid'])) {
    $videoid = $_POST['videoid'];

    $query = "delete FROM videos WHERE id='$videoid'";
    $results = mysqli_query($db, $query);
    $res = mysqli_fetch_assoc($results);

    $query = "delete FROM videoratings WHERE videoid='$videoid'";
    $results = mysqli_query($db, $query);
    $res = mysqli_fetch_assoc($results);

    $query = "delete FROM videodetails WHERE videoid='$videoid'";
    $results = mysqli_query($db, $query);
    $res = mysqli_fetch_assoc($results);

    $query = "delete FROM videocomments WHERE videoid='$videoid'";
    $results = mysqli_query($db, $query);
    $res = mysqli_fetch_assoc($results);
    $id = $res['id'];

    echo true;
}
?>
```

Figure 26: Delete solution code example

For our delete functionality we have a fairly simple process where we match the IDs of all the properties and remove them from each database accordingly, Figure [26].

8.3 Setting up the environment and configuration

Before the whole code implementation, we had to prepare the environment for the system. The software has been running only locally with a local database, for this to be possible it was inevitable to use external programs that we require. By downloading and installing the required software, for this website XAMPP as a server and Navicat as a DB management was used in order to host and run our software.

8.3.1 XAMPP

XAMPP is an abbreviation for cross-platform, Apache MySQL, PHP, Perl and it allows us to build a website offline Figure [27], on a local web server on our computer. This simple and lightweight solution works on Windows, Linux, and Mac hench the “cross-platform” part.

By running the XAMPP control panel we can start our Apache and MySQL server in order to connect it with our Navicat database [10].

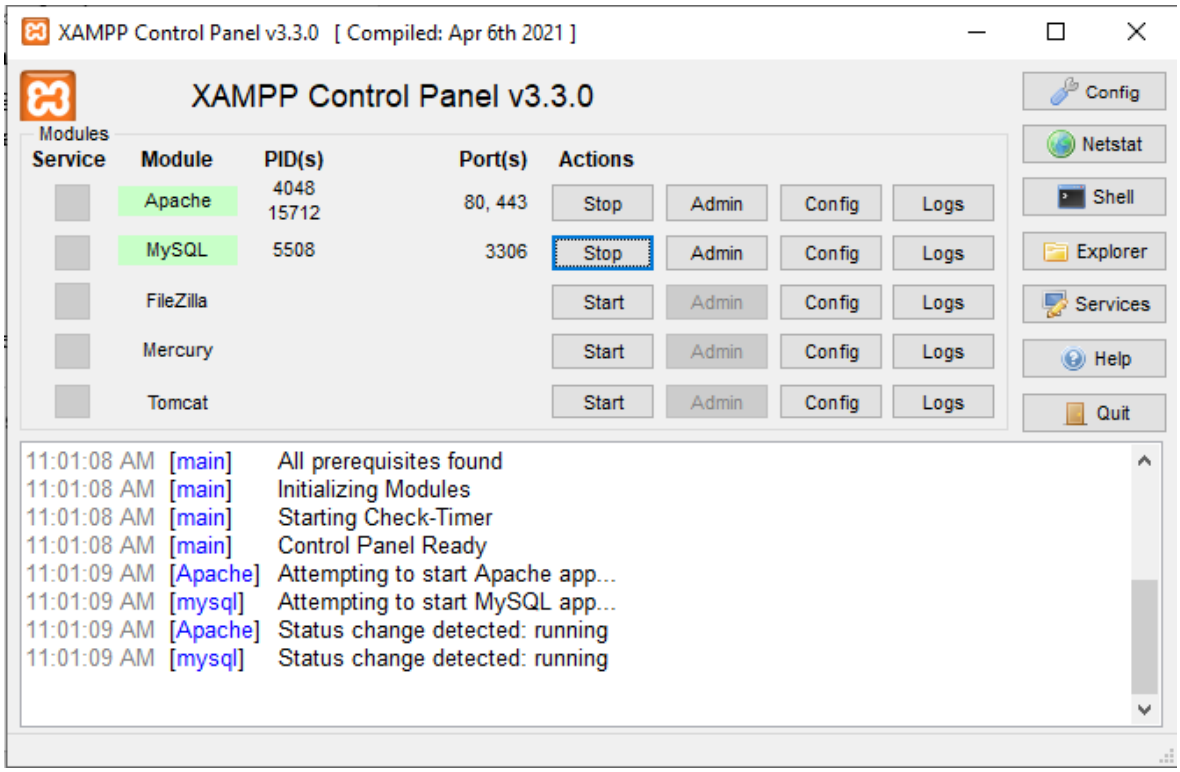

Figure 27: XAMPP control panel

8.3.2 Navicat

Navicat is a database development tool Figure [28], which helps organizations of all sizes build, manage and maintain databases, users can connect with various on-premise and cloud databases such as MySQL, MariaDB, MongoDB, etc. Features include data visualization, code snippet and completion, backup scheduling, virtual grouping, and schema analysis.

The application enables developers to migrate and synchronize data across various databases by using guidelines.

By running Navicat we were able to import the database tables which were created beforehand and link the database with the server allowing our website to function and store data in those tables [11].

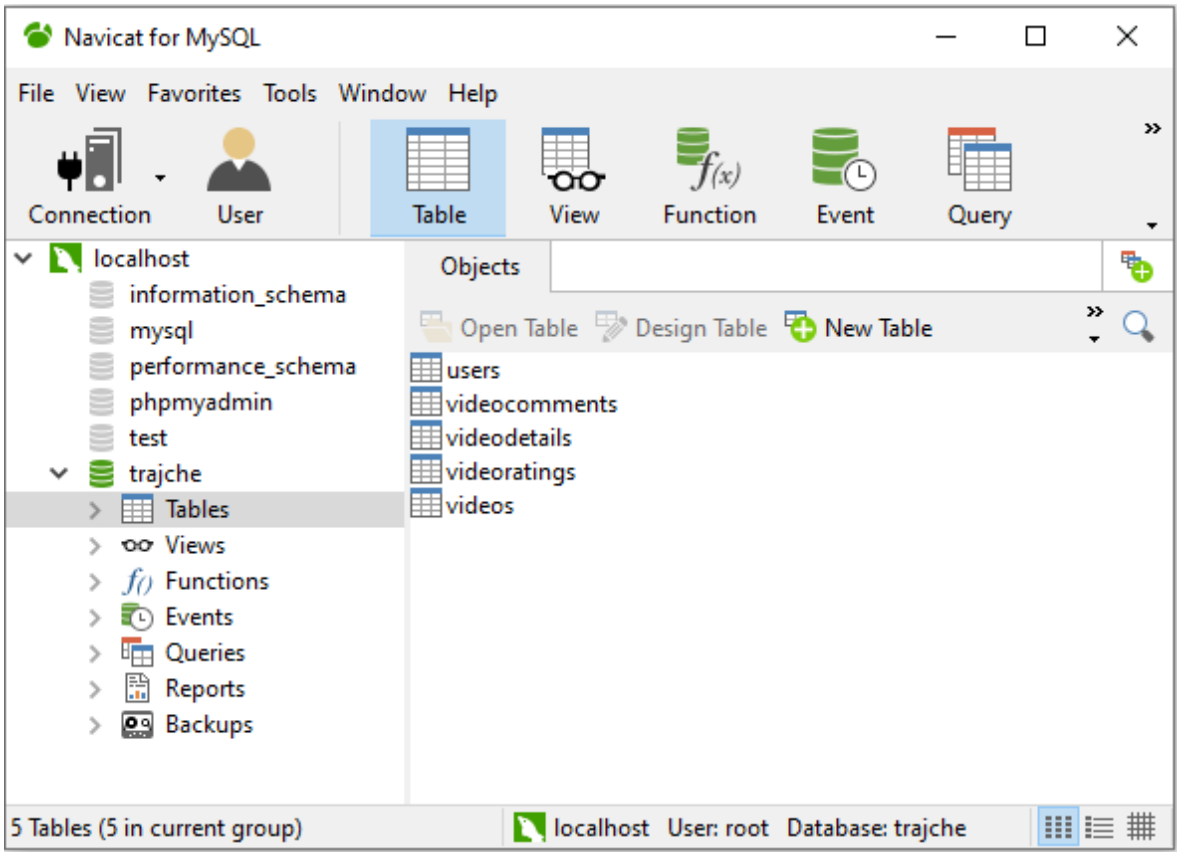

Figure 28: Navicat MySQL database with tables

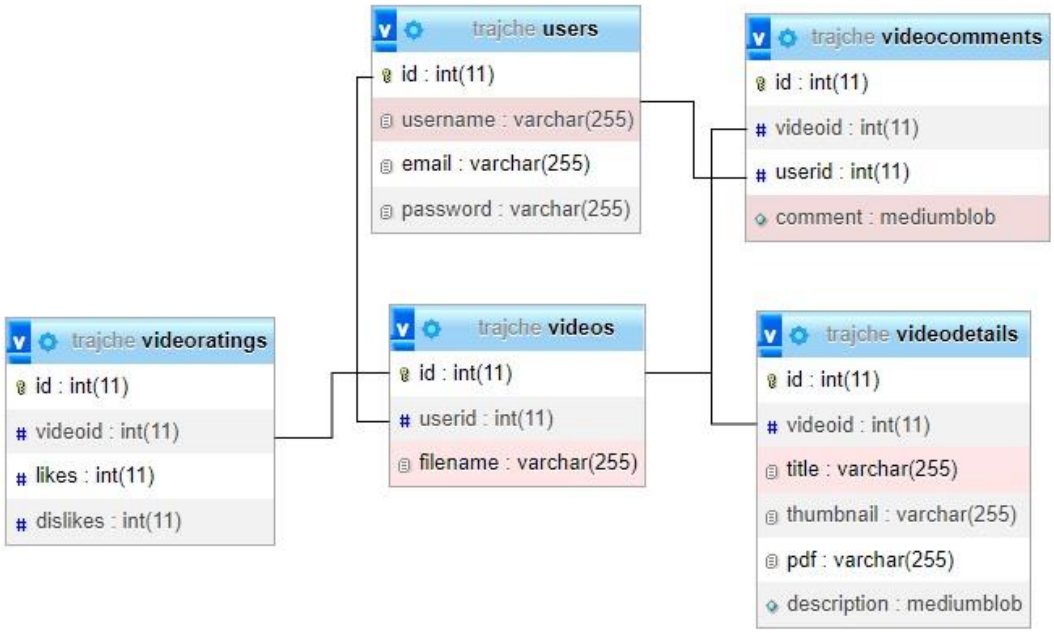

Figure 29: Database relations between tables

In the projected design, tables were created and separated, and the relationships between them were built, Figure [29]. These are the most important tables that will contain all the relevant data input by the users. In the table users, we define their personal data and login information, which is essential for user authentication. The video comments table contains the comment content which is inserted by the user depending on which solution and which user inserts it, hench the relationship with those other two tables. The video ratings table contains the number of upvotes and downvotes of each user corresponding to the selected solution, the video details table contains all of the details of the selected solution and the videos table contains the name and which user has selected that solution.

9 SYSTEM TESTING

Now one of the main questions to ask is whether the project is showing good performance? While this is not a final version, we did not have enough time to complete full testing of the software, therefore we briefly tested the whole system with the requirements for it to function properly for the everyday user.

We have found out it offers all of the capabilities that were stated in the project, since it is a prototype, it is not tested to the full extent perhaps some parts can be improved but we need to do more intense testing for this to be possible and before offering the system to users publicly.

All of the testing's have been running and done locally therefore not tested on a public server meanwhile, the uploading process has not yet been tested with a heavy file such as a maybe one-hour long video which can be a lot of storage to withstand on a server, on the other hand, smaller video which is less storage dense can be uploaded fairly easily and requires not much time.

The important requirements were tested:

- When a person opens any page, they are able to view it, but with limited access to some functionalities
- When a person tries to register, after registration they are redirected to the home page, same as when a person logs in he is also redirected to the home page.
- When a user wants to create a solution, the upload solution button is pressed and the upload process begins, inserting the mp4 file, title, pdf document, and description to complete the upload, afterward solution is available on the home page.
- When the user uses the search engine, the solution in need is searched upon by inserting the title in the search engine, the desired solution is displayed.
- When the user opens a solution, the video format is displayed, the rating is displayed, the comment section is displayed, pdf document is accessible through the press of a button, buttons for rating are displayed, the button for an edit is displayed, the button for deleting is displayed.
- While the solution is opened, users can input a comment, open a pdf documentation, and leave a rating, if they uploaded the solution buttons for editing and deleting are available to be accessed.

10 CONCLUSION

At the beginning of the project, we identified the goal of the diploma thesis as creating a prototype solution to a certain problem which we noticed is occurring more rapidly by each day. In the present work, we first had to define what our problem is, moving on we figured out some existing solutions where some people tried to help solve the problem, and we figured out our own solution and how we can solve this issue. We identified that this software is feasible to create, by doing some requirements analysis and having the functional and non-functional requirements of the system, visualizing the interactions of users with the system using the UML Use Case diagram.

Afterward we created a system design using a Component diagram where we could visualize each component, to be prepared and understand the system implementation. By explaining which technologies were used, and visually representing how the system was created and what it looked like eventually we got a better understanding of the whole software. For the storage and hosting, we reviewed that it was done locally using XAMPP and Navicat where the database tables were presented and displayed how they look like and in the end showed some libraries which helped us achieve some parts of the implementation.

The built prototype can actually fulfill the tasks assigned to the project. While all of the functionalities are working and tested, it is good to note that it is still a prototype and is a work in progress if by means it is decided to be released to the full public by regular users.

Summarizing all of the above, we can say that all of our goals were achieved: defining our problem, presenting the idea of our solution, understanding the feasibility of the project, explaining our requirements analysis with functional and non-functional requirements, creating a system design, finally our implementation and finishing off with the system testing, we demonstrated the principles of software engineering all the way to a fully implemented software. Using modern development methods, we have created a working prototype, that can become the basis of the project. With this project we have learned the various ups and downs of creating software and the difficulties that come along the way, there is a lot of attention to detail and precision involved when creating such a project.

11 DALJŠI POVZETEK V SLOVENSKEM JEZIKU

Na začetku projekta smo si za cilj diplomske naloge zadali izdelavo prototipa rešitve določenega problema, za katerega smo opazili, da se pojavlja vsak dan hitreje. Pri tem delu smo morali najprej opredeliti, kaj je naš problem, nato pa smo ugotovili nekaj obstoječih rešitev, pri katerih so nekateri ljudje poskušali pomagati rešiti problem, in ugotovili smo svojo rešitev in kako lahko rešimo to težavo. Ugotovili smo, da je to programsko opremo mogoče ustvariti z analizo zahtev in funkcionalnimi in nefunkcionalnimi zahtevami sistema, vizualizacijo interakcij uporabnikov s sistemom z uporabo diagrama primerov uporabe UML.

Nato smo ustvarili načrt sistema z uporabo diagrama komponent, kjer smo lahko vizualizirali vsako komponento, da smo pripravljeni in razumeli implementacijo sistema. Z razlago, katere tehnologije so bile uporabljene, in vizualno predstavitvijo, kako je bil sistem ustvarjen in kako je izgledal, smo sčasoma bolje razumeli celotno programsko opremo. Za shranjevanje in gostovanje smo pregledali, da je bilo izvedeno lokalno z uporabo XAMPP in Navicat, kjer so bile tabele baze podatkov predstavljene in prikazane, kako izgledajo, na koncu pa so bile prikazane nekatere knjižnice, ki so nam pomagale doseči nekatere dele implementacije.

Zgrajeni prototip lahko dejansko izpolni naloge, dodeljene projektu. Medtem ko vse funkcionalnosti delujejo in so preizkušene, je dobro omeniti, da je še vedno prototip in je delo v teku, če se na ta način odločijo, da ga redni uporabniki objavijo za celotno javnost.

Če povzamemo vse zgoraj navedeno, lahko rečemo, da so bili vsi naši cilji doseženi: opredelitev našega problema, predstavitev ideje naše rešitve, razumevanje izvedljivosti projekta, razlaga naše analize zahtev s funkcionalnimi in nefunkcionalnimi zahtevami, izdelava zasnova sistema, končno naša implementacija in zaključek s testiranjem sistema smo prikazali principe programskega inženiringa vse do popolnoma implementirane programske opreme. S sodobnimi razvojnimi metodami smo ustvarili delujoč prototip, ki lahko postane osnova projekta. S tem projektom smo se naučili različnih vzponov in padcev ustvarjanja programske opreme in težav, ki se pojavljajo ob tem, pri ustvarjanju takšnega projekta je treba posvetiti veliko pozornosti podrobnostim in natančnosti.

12 REFERENCES

[1] Ian Sommerville, “Software Engineering, Eighth Edition”, 2006.

[2] Roger S. Pressman; Bruce R. Maxim, “Software Engineering: A Practitioner's Approach”, 2019

[3] “Use Cases”. [Online]. Available: https://www.usability.gov/how-to-and-tools/methods/use-cases.html. [Accessed Jul. 2022]

[4] MDN contributors “HTML Basics”. [Online]. Available: https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/HTML_basics. [Accessed Jul. 2022]

[5] Tutorials Point simply easy learning “What is CSS”. [Online]. Available: https://www.tutorialspoint.com/css/what_is_css.htm. [Accessed Jul. 2022]

[6] MDN contributors “What is JavaScript”. [Online]. Available: https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript. [Accessed Jul. 2022]

[7] “What is PHP”. [Online]. Available: https://www.php.net/manual/en/intro-whatis.php. [Accessed Jul. 2022]

[8] SQL course “What is SQL”. [Online]. Available: https://www.sqlcourse.com/beginner-course/what-is-sql/. [Accessed Jul. 2022]

[9] Software Freedom Conservancy “GIT”. [Online]. Available: https://git-scm.com. [Accessed Jul. 2022]

[10] Undsgn “XAMPP Tutorial”. [Online]. Available: https://undsgn.com/xampp-tutorial/. [Accessed Jul. 2022]

[11] Navicat “Navicat for MySQL”. [Online]. Available: https://www.navicat.com/en/products/navicat-for-mysql. [Accessed Jul. 2022]