University of the Peloponnese
School of Economics, Management and Informatics
Department of Informatics and Telecommunications

Undergraduate Thesis

An online communication system for the citizens and the municipality of a city.

Andreas Maragkakis
AM: 2022201500085

Supervisor:

Emmanuel Wallace
Assistant Professor

Tripolis, 12,2020

Andreas Maragkakis
Department of Informatics and Telecommunications
University of Peloponnese

Acknowledgements

Throughout the writing of this thesis, I have received a lot of support and assistance.

Firstly, I would like to thank my supervisor, Assistant Professor Emmanuel Wallace, whose knowledge and experience help me develop my thesis. Your feedback was always on point and helped me learn new things.

Additionally, I would like to thank my parents and my brother for their unlimited support and consultation. Your help was significant. Also, I would like to thank my friends in Greece and abroad, who believed in me during good and bad times.

Abstract

Nowadays cities have a lot of problems (e.g broken traffic lights). Citizens are facing those problems every day and they want to make them known to the borough to address them accordingly. Unfortunately, the communication between the citizens and the municipality of a city is usually not existing or it exists to a small degree. This thesis aims to provide an online communication system, in which those two parties can use it to communicate with each other and solve the city's problems together. Firstly, the citizens use an app to report a problem (e.g. a damaged recycle bin) to the municipality. Then the borough uses a Website to see those reports and it can send the appropriate task force to solve them. The app was developed for Android devices using the Android Studio IDE (Integrated development environment). The main programming language used for the app was Java. The Website was developed using internet technologies (HTML, PHP, CSS) and it communicates with the app through a common database. Cities around the world can use this online system to create a communication channel between the citizens and the municipality. This way those two parties can work together to eliminate the various problems that exist inside their cities.

Contents

Chapter 1

Introduction

The municipality of a city must be able to solve the city's problems as quickly as possible. The citizens of the city can help them toward that goal, by reporting the various problems that the city has to them. This way the borough can detect and solve problems faster. However, the communication between the municipality and the citizens in most cases does not exist. This thesis aims to create an online communication platform for the citizens and the municipality of a city.

Most of us had lived in a city for a period. City life can be quite problematic. You have to deal with different kinds of problems (e.g broken stop sign) that will make your everyday life difficult. Moreover, the municipality can't keep track of all the problems happening around the city, as a result, a lot of problems are remaining unsolved for a couple of years. These things can make a city quite dysfunctional. We can see how important is to create a connection between everyday citizens and the municipality of the city. This thesis aims to create that bond between those two parties and thus making cities a better place to live in.

This tool will help the a) citizens and b) the municipality of a city. Firstly, the citizens will have a tool more precisely an app in which they will report the unsolved problems of their city. Secondly, the municipality will be able to detect and keep track of the problems happening around the city, and thus be able to solve them quicker.

1.1 Objectives

This project aims to complete two main objectives.

1.The creation of an android app which the citizens will use to make reports about various incidents around their city.

2.The creation of a website for the municipality of a city, in order to keep track of the citizen's reports.

1.2 Dissertation outline

- Chapter 1 Introduction

This chapter provides an introduction to the thesis by explaining how this thesis will affect the real world, what are the objectives of the thesis, and how the rest of the dissertation is structured. .

- Chapter 2 Mobile apps

In this chapter, the reader will first learn general information about mobile apps. Then the chapter continues with the categorization of the apps into 3 different groups a)Native apps, b)Web-based apps, c)Hybrid apps.

- Chapter 3 Android operating system and Introduction to Android Studio

This chapter provides a lot of information about the Android operating system and Android Studio. It begins with the architecture of the Android operating system, and finally, it reviews Android Studio (IDE).

- Chapter 4 Development of the thesis

This chapter outlines the process that took place to complete the thesis. Furthermore, in this chapter, the reviewer can understand the tools which were used to create this project.

- Chapter 5 Conclusions

This chapter summarizes the objectives of the thesis and suggests paths for further development.

Chapter 2

Mobile apps

A mobile application also referred to as a mobile app or simply an app is a computer program or software application designed to run on a mobile device such as (phone, tablet). Apps are generally downloaded from application distribution platforms or (App stores) which are operated by the owner of the mobile operating system such as the Google Play Store. Some apps are free and others have a price. Usage of mobile apps has become increasingly prevalent across mobile phone users [1].

Apps usually are divided in 3 categories:

1) Native app

These apps are programmed to function only on a specific mobile platform. For example, an app that was developed to target the Windows operating system does not run on the Android operating system.

2) Web-based app

These apps have been developed using Web type programming languages and technologies. They are executed on a web server and not on a specific operating system and they are required to have internet access for proper behavior.

3) Hybrid app

The hybrid app is a combination of native and web-based technologies. These apps are developed to function on different mobile platforms. Often hybrid apps are exhibit lower performances in comparison with the rest categories.

Mobile application development requires the use of specialized integrated development environments (IDES). Mobile apps usually are first tested within the IDE and then they are tested on real mobile devices.

Chapter 3

Android operating system and Introduction to Android Studio

3.1 Architecture of Android

Android is a mobile Linux-based operating system, created mostly for mobile devices. Figure 3.1 shows the major components of the Android operating system.

3.1.1 The Linux Kernel

Android is based on the Linux kernel. Linux kernel is a free, monolithic, operating system kernel. It provides a variety of functionalities such as threading, memory management, and many more.

3.1.2 Hardware Abstraction Layer (HAL)

The hardware abstraction layer (HAL) provides standard interfaces that expose device hardware capabilities to the higher-level Java API framework. The HAL consists of multiple library modules, each of which implements an interface for a specific type of hardware components, such as the camera or Bluetooth module. When a framework API makes a call to access device hardware, the Android system loads the library module for that hardware component [2].

3.1.3 Android Runtime

For devices running Android version 5.0 (API level 21) or higher, each app runs in its own process and with its own instance of the Android Runtime (ART). ART is written to run multiple virtual machines on low-memory devices by executing DEX files, a bytecode format designed especially for Android that's optimized for minimal memory footprint. Build toolchains, such as Jack, compile Java sources into DEX bytecode, which can run on the Android platform [2].

3.1.4 Native C/C++ Libraries

Many core Android system components and services, such as ART and HAL, are built from native code that requires native libraries written in C and C++ [2].

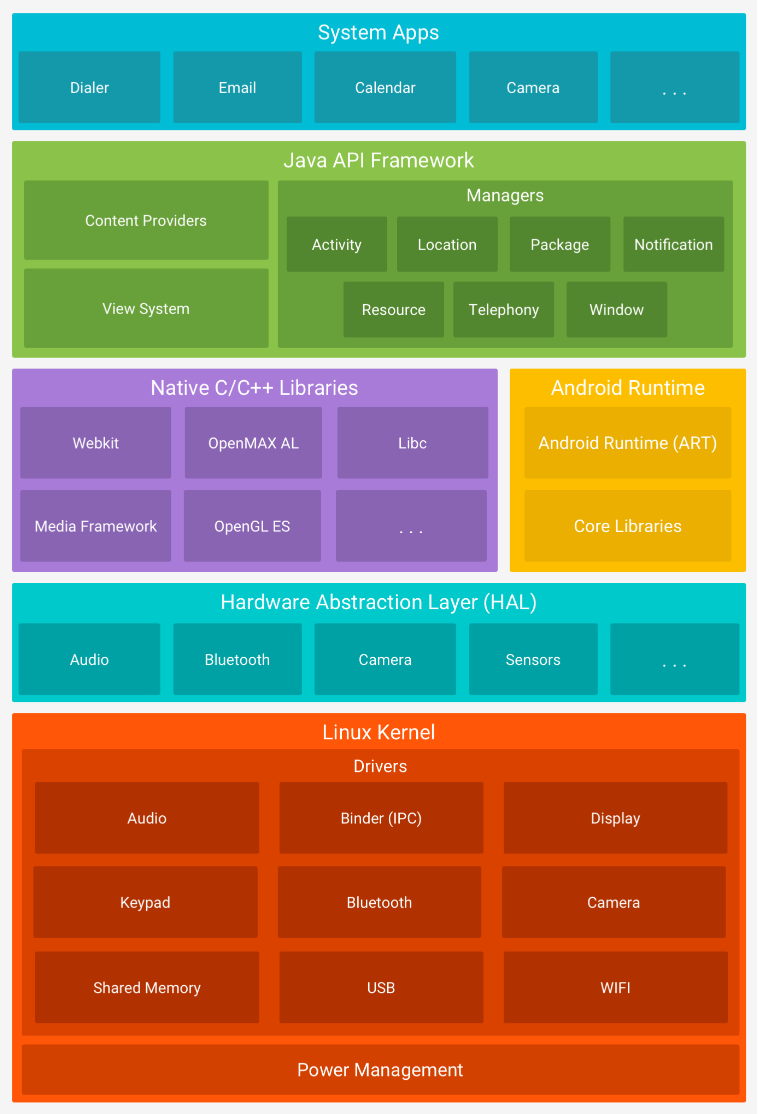


Figure 3.1. The Android software stack. From Platform Architecture,by Google Developers,2020, https://developer.android.com/guide/platform. Copyright APACHE LICENSE, VERSION 2.0

3.1.5 Java API Framework

Programmers in order to create Android apps need to access some of the Android OS functionalities. For that reason, Android provides APIs (application programming interface) written in the Java language. Programmers use these APIs to access core services that the operating system provides.

3.1.6 System Apps

Android has already Pre-installed a set of important apps such as SMS messaging app, internet browsing app, and many more. Furthermore, the user can also choose to install third-party apps and thus making his device more diverse.

3.2 Introduction to Android Studio

To develop my app, I use Android Studio. Android Studio is an IDE (Integrated development environment) developed and managed by Google, which is being used to develop android applications. Android studio provides an auto-generated code editor which helps you to write your code faster. Moreover, It's gradle feature helps you in compiling and building the application. Furthermore, Android studio has a rich layout editor which allows you to drag-and-drop UI components, preview layouts on multiple screen configurations, and many more.

Android Studio also provides an Android emulator. Within the Android emulator, the programmer can test his application on different virtual devices. Also, he can choose which Android API level wants for his device. Android emulator can simulate most of the functions of a real android device. It can simulate incoming phone calls, text messaging, and many more. My project targets the 24th Android API level.

Chapter 4

Development of the thesis

My goal in this thesis is to create an app in which the user can detect a problem in his city and sent a message to the appropriate authorities to fix it. Moreover, I also create a website for the municipality to keep track of the user's problems and solve them. For example, a citizen may found a broken traffic light and he opened the app, and inform the authorities about it. The authorities, using the website, can view all the reports of the citizens. At the same time, they can also see the exact location where those reports came from. This way the state can detect and solve the problems of the communities easier and more efficient.

To develop my app I decide to use Android Studio (IDE). As I mentioned already this IDE is the best option if you want to develop Android apps. I use Java as my main programming language for developing this app. Java is a class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible [3]. As for the website I use 3 main internet technologies HTML, CSS, PHP.

Moreover, I use MariaDB, a relational database management system (RDBMS), to create a database to keep track of my data. Finally, to bring all of my services online, I use the XAMPP package. XAMPP is a completely free, easy to install Apache distribution containing MariaDB, PHP, and Perl. The XAMPP open source package has been set up to be incredibly easy to install and to use [4]. By using these services, I manage to make a connection between my app, my website, and my database. That way I reach the main goal of this thesis.

4.1 App development

I create my app using Android Studio, which is an integrated development environment. Android Studio helps you write, maintain, and execute your code efficiently. It is designed specifically for Android app development. Android Studio lets you develop your app using a variety of languages e.g. JAVA, KOTLIN, and many more. My language of choice was JAVA because, I was more familiar with it, and it's one of the two languages, which google suggests for android app development. In parallel with JAVA, I also use android's XML language. With this language, I was able to create the User-Interface of my app.

Firstly I create a login page for my app (Figure 4.1). An already register user can type the username and the password and press login to enter the app. Moreover, if the user types the password or the username wrong, then an error message will appear, and the user will not be able to login into the app.

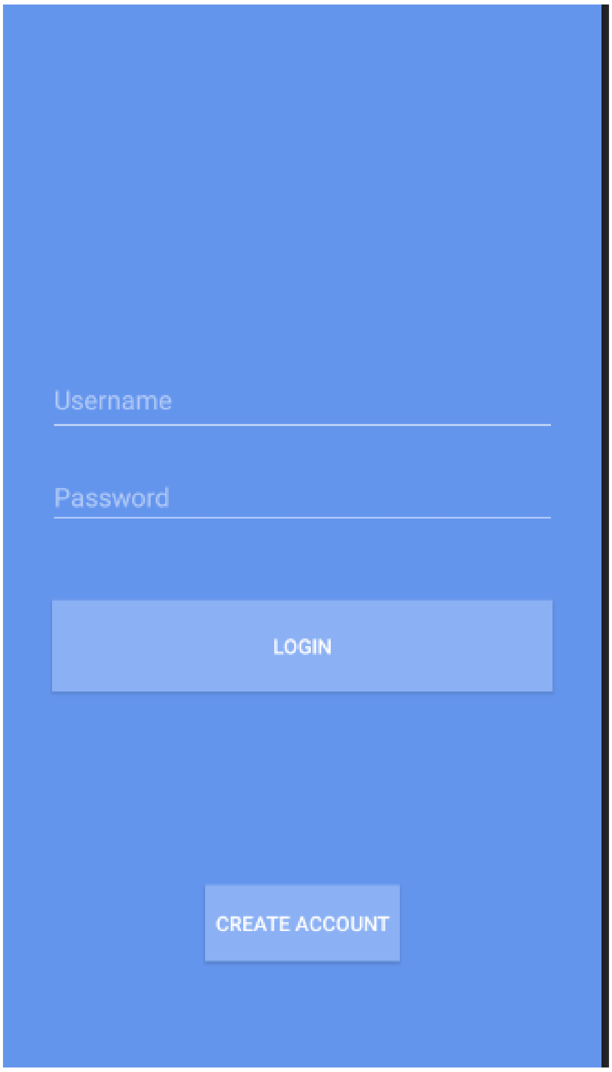


Figure 4.1. Login screen

The users who have not yet registered can press the CREATE ACCOUNT button on their screens. After pressing this button, a new screen will appear. On this screen, the user can type the username, password, and then Confirm Password. If the password and the Confirm Password do not match, an error message will appear, and the user has to retype them. Furthermore, if the typed username already exists in the database, an error message will appear, and then the user has to choose a different username.

After the completion of all fields, the user can press the CREATE ACCOUNT button, the user‘s account will be created and saved in my database, and then he will be redirected to the login screen of my app. The user now can log in and use the app.

After the login page, the user now is directed to the map screen (Figure 4.2). The main thing you see on this screen is a Google map. To create this map I use Google Maps API (application programming interface). To use this API, firstly you need to have an API key, which you can use to get access to Google Maps servers.

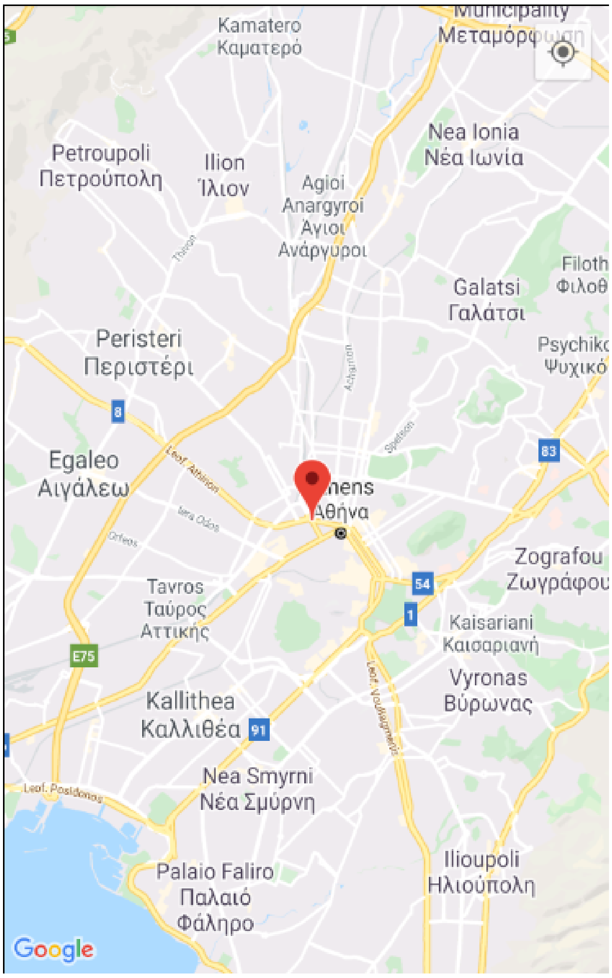


Figure 4.2. Map screen

After having the key you need to make it available to your app. Google has provided specific instructions on how to use the key in your app. This API lets you have access to the Google Maps server and display their map to your app. The Google API provides additional information about the user's location, and also allow user interaction with the map. The map shows by default the location of the user. The user then can move the map with his hand and locate the area, in which a problem exists. After locating the area of interest, the user can then double-tap on this area of the map. This action will direct the user to a new screen.

On this screen (Figure 4.3), the user can explain the difficulty he is facing, by typing it inside the "Enter your problem here box". Then the user can press the send button. The send button redirects the user to the previous screen, the map screen, at the same time the location, which you pin on the map, and the text which you wrote inside the problem box, are sent to my database and saved there. Now the user can do the same steps and report another problem, or he can close the app.

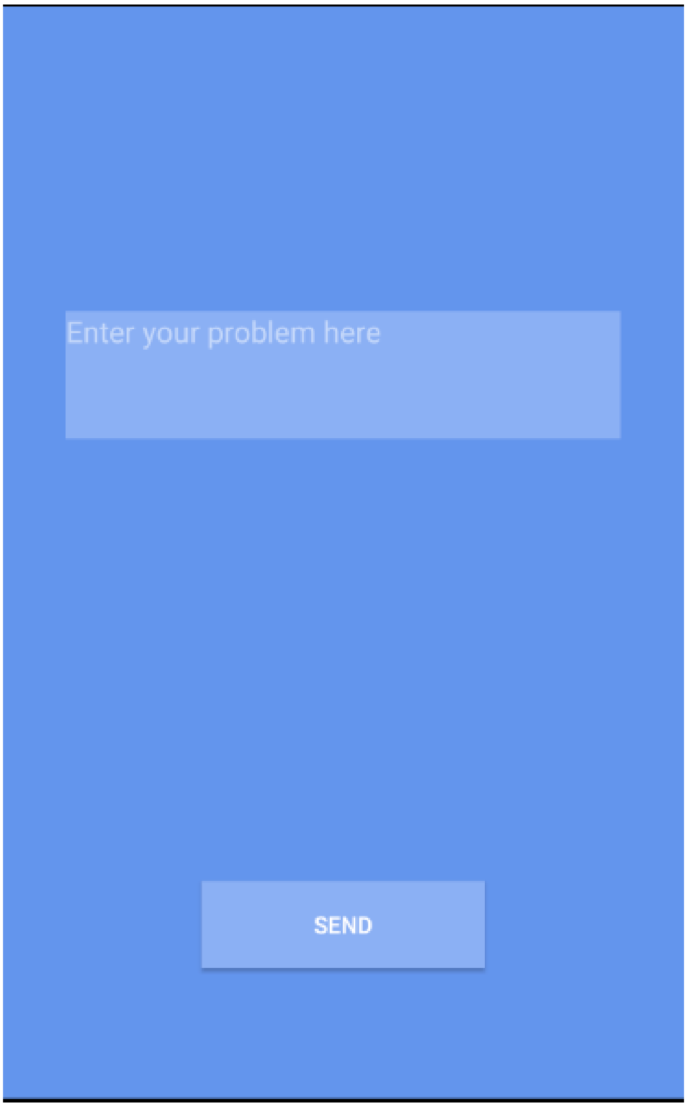
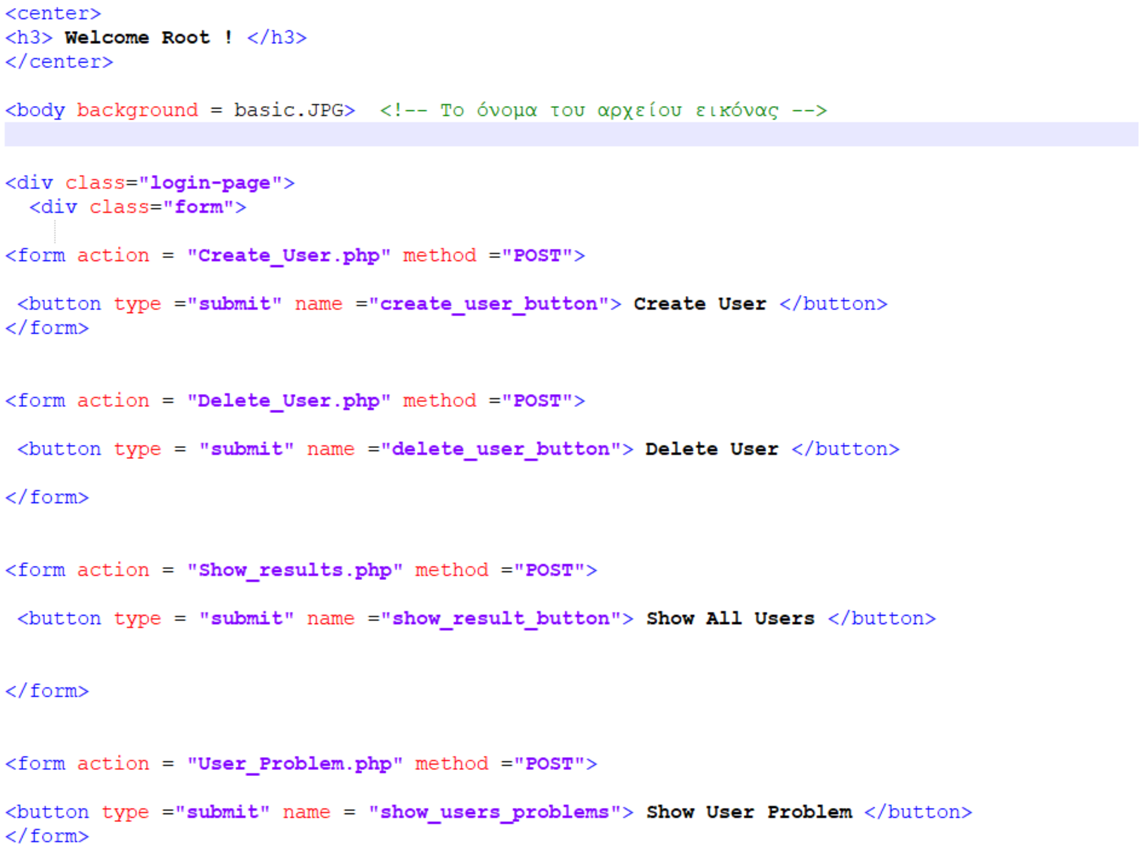


Figure 4.3. Problem screen

4.2 HTML

To create my website, I firstly use the HTML language. Hypertext Markup Language (HTML) is the standard markup language for documents designed to be displayed in a web browser. Web browsers receive HTML documents from a web server or local storage and render the documents into web pages. With HTML you can describe the structure of a web page [5].

HTML is an easy language to understand. HTML use elements which are the building blocks of HTML documents. The elements are delineated by tags, written using angle brackets. So, an element is everything from the start tag to the end tag.



```
<center>
<h3> Welcome Root ! </h3>
</center>

<body background = basic.JPG>  <!-- Το όνομα του αρχείου εικόνας -->


<div class="login-page">
  <div class="form">

<form action = "Create_User.php" method ="POST">

 <button type ="submit" name ="create_user_button"> Create User </button>
</form>


<form action = "Delete_User.php" method ="POST">

 <button type = "submit" name ="delete_user_button"> Delete User </button>

</form>


<form action = "Show_results.php" method ="POST">

 <button type = "submit" name ="show_result_button"> Show All Users </button>


</form>


<form action = "User_Problem.php" method ="POST">

<button type ="submit" name = "show_users_problems"> Show User Problem </button>
</form>
```

Figure 4.4. html code

As you can see from Figure 4.4, HMTL uses a variety of elements and tags. By combining different elements, you can create bigger and more complex documents, which later the web browsers will translate into Webpages.

4.3 CSS

As I mention above with HTML, I create the skeleton of my website. Now I need to make it more presentable to people. For that purpose, I use the CSS language. Cascading Style Sheets (CSS) is a style sheet language. CSS is used for describing the presentation of an HTML document. Moreover, when you combine HMTL and CSS you can produce beautiful Websites.

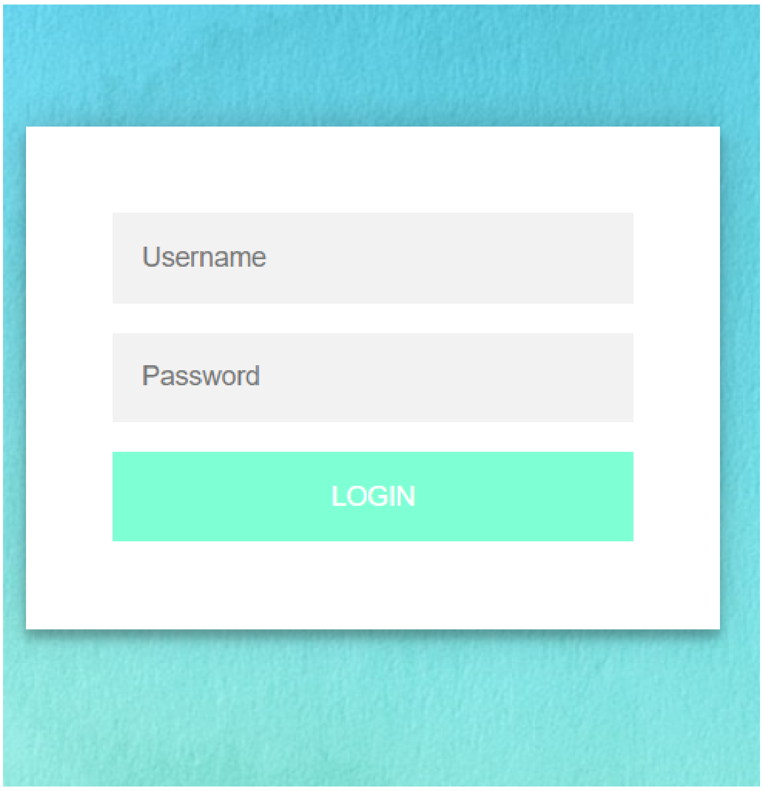


Figure 4.5. index1.php

CSS is independent of HTML, which means that you can have the HTML code the CSS code in different files. This separation of HTML from CSS makes the website more flexible.

For my project, I create one. CSS file called ”index1_styles.css”. I reuse my CSS code inside different locations of my project. That way I manage to reduce the number of code lines that I need to complete my website.

4.4 PHP

The last step in my website creation is to use the PHP language together with HTML and CSS. PHP stands for the recursive initialism PHP: Hypertext Preprocessor. It is a general-purpose scripting language especially suited to web development [6]. Furthermore, PHP works well with HTML, CSS, and databases.

For my website, I use several. PHP files. Firstly I have created a login page called “index1.php”(Figure 4.5). To enter the website you must know the specific password and username, which only the municipality admin must know. If the credential is wrong, then an error message appeared.

After the successful login, the admin is directed to the “Home.php” page (Figure 4.6). Here the admin has several options to choose from.

The first option is to create a user. When it’s pressed the admin is directed to a new screen. The admin can now type the username and the password of the user. If the username already exists the admin must choose another. The password and the Confirm password fields must much otherwise

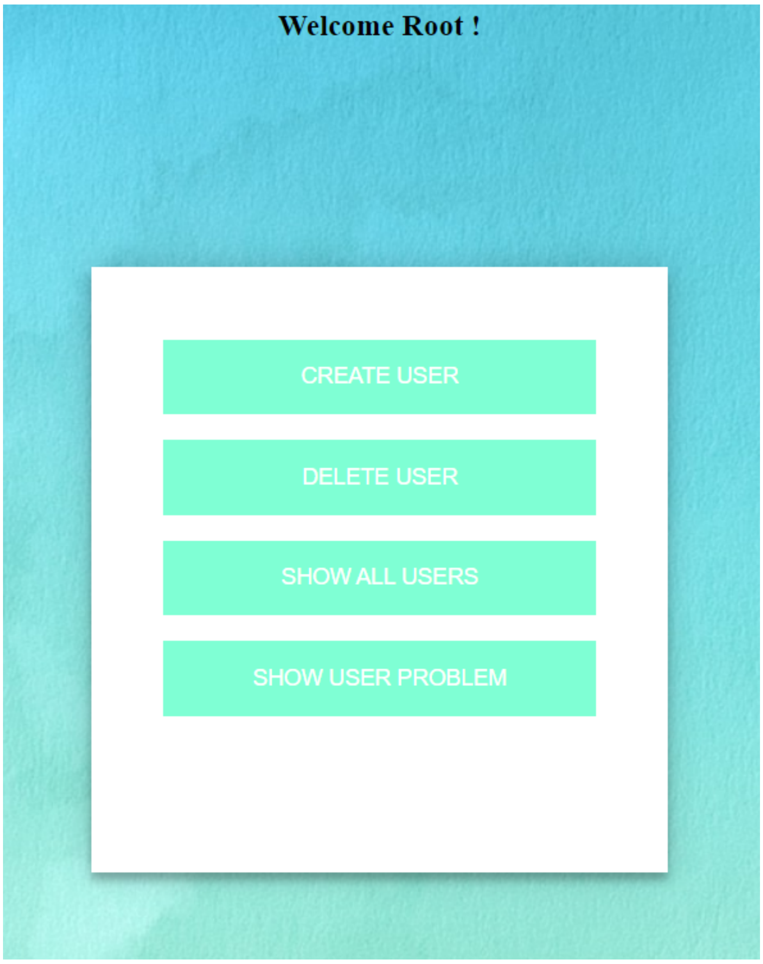


Figure 4.6. Home.php

an error message will appear. After the successful completion of all fields, now the admin can press the CREATE_USER button. Then, the admin is redirected to Home.php and the user has been added to my database.

The second option is to delete a user. The admin is now at the Delete_User.php page. Here the admin can write the Username of the user who wants to delete. If the username does not exist, an error message appeared. After the pressing of the button DELETE_USER, the user with the specific Username is deleted and the admin is redirected to the home page.

The third option is to show all the already existing users. After pressing this button the admin is seeing a list with all the users in the database. This page is called “Show_results.php”.

The last option is to list the user report and the location of this issue. This page is called “User_Problem.php”. The admin can see the issues and the location in ascending order.

4.5 Database

Databases have been important in the last years. Applications and Databases need to work together to save, retrieve, and process critical information about users. For my thesis, I use MariaDB, a relational database management system (RDBMS). Basically, MariaDB is a software, which is helping me create and maintain a database on my computer. As I mention, MariaDB is integrated into the XAMPP software package. So the creation of the database was done with the help of the XAMPP ‘s phpMyAdmin tool (Figure 4.7). This software tool provides you an easy and understandable UI(User Interface) which, lets you administrate the MariaDB software and thus, creating your database.

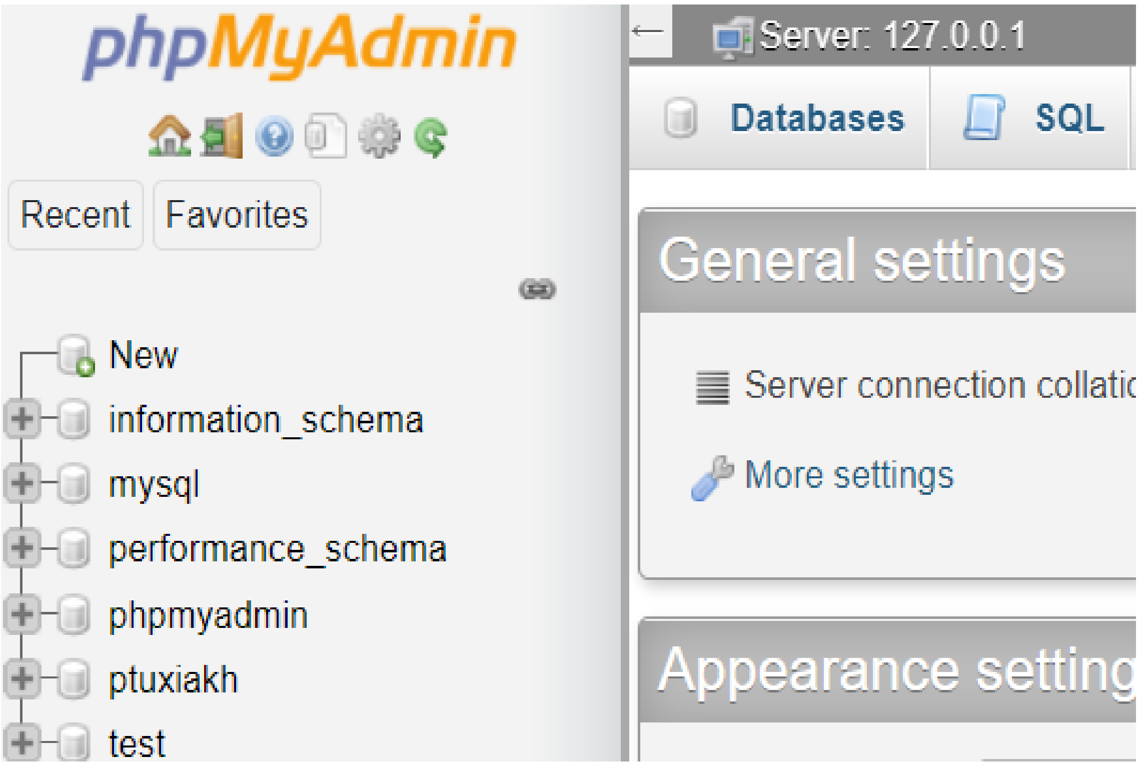


Figure 4.7. phpMyAdmin tool

For my project, I create one database, called “test”, and two tables inside it. The first table I name it “main_table”. This table has two columns: Username and password, which are both text types. This table is used to save the user’s Usernames and passwords. The second table is called “user_problem_text” and it also has two columns: problem_text and Location. This table is used to save the user’s text and the location of the issue. Both are type of text.

After creating my database and my tables, I also need to connect my Website with them. To connect to my database, I create a new .php file called “Connection.php” and inside it, I wrote the code I need to access my database. Now, I have a fully functional website. Additionally, so that my app can access my database, I need to add some extra lines of JAVA code inside my app.

Now I have a fully functional website and app which can communicate with each other through the database. All my objectives have been reached and thus the online communication system has now been completed. The code of my project along with other files can be found in my GitHub repository [7].

Chapter 5

Conclusions

As a citizen of a city, I can feel the lack of communication with the municipality.Moreover, during my research, I did not find good solutions for this communication problem.This thesis is a new initiative that aims to create an online communication tool between the citizens and the municipality of a city. With this tool cities in the future will be more functional.

This online system consists of the following steps:

1. The creation of an android app. Citizens of a city can use this app to report dysfunctions that happening around their city.

2. The development of a website. The municipality uses this website to create, delete, and list the users. Moreover, the website can list the various problems that the users have reported and the municipality can take the appropriate actions to solve them.

Those two objectives have been completely reached as I mentioned in Chapter 4.

5.1 Further Expansion

Further expansion is required to make this system more concrete. A list of the main foreseen tasks follows:

1. The app must reprogram to be able to run in different operating systems (e.g iOS). This way more people will be able to help their city.

2. The website should have a new option which will delete the old user's reports and thus making the database more manageable.

References

[1] Wikipedia contributors. *Mobile app*. URL: https://en.wikipedia.org/w/index.php?title=Mobile_app&oldid=983293824.

[2] Google Developers. *Platform Architecture*. URL: https://developer.android.com/guide/platform.

[3] Wikipedia contributors. *Java (programming language)*. URL: https://en.wikipedia.org/w/index.php?title=Java_(programming_language)&oldid=991892947.

[4] Apache Friends. *XAMPP Apache + MariaDB + PHP + Perl*. URL: https://www.apachefriends.org/index.html.

[5] Wikipedia contributors. *HTML*. URL: https://en.wikipedia.org/w/index.php?title=HTML&oldid=991040534.

[6] PHP Group. *PHP*. URL: https://www.php.net/.

[7] G-S-B-97. *Thesis*. URL: https://github.com/G-S-B-97/Thesis.

[8] Google Developers. *The Android software stack*. URL: https://developer.android.com/guide/platform.