



Ministry of Higher Education and Scientific Research
and Information and Communication Technologies
General Directorate of Technological Studies



Higher Institute of Technological Studies of Kef
Department of Computer Technology

REPORT OF DEVELOPMENT INTERNSHIP

SUBJECT

**Development of a web application to maintain
and manage day-to-day tasks**

Realized by : Mohamed Rayen Haddaoui

Organization : Foufousa
Adress : Cyberparc, Rue d'Alger, Le Kef 7100
Phone : +216 96 398 237
E-mail : contact@foufousa.com

Academic Year: 2022-2023

Thanks

We reserve these lines with great pleasure as a token of appreciation and respect to everyone who has helped this work come to be, whether directly or indirectly.

First and foremost, we would like to express our gratitude to **Mr.Radhouen Foudhaili**, the **CTO** of **Foufusa**, for his kind welcome, his confidence, and his encouragement, all of which helped us push forward.

Additionally, we would like to extend our sincere appreciation to the professors at ISET Kef who, despite their other obligations, gave us their help with a lot of dedication, tolerance, and kindness.

Summary

General introduction	3
Chapter 1: General context of the project	2
Introduction	2
1.1 Company presentation	2
1.1.1 Organization chart	2
1.2 initial study	3
1.2.1 Study of the existing	3
1.2.2 Criticism of the existing system	4
1.2.3 Proposed Solution	4
Conclusion	4
Chapter 2: Specification of requirements	5
Introduction	5
2.1 Project planning	5
2.1.1 Actors' Presentation	5
2.1.2 Functional requirements	5
2.1.3 Non-functional requirements	6
2.2 General Use Case diagram	7
Conclusion	7
Chapter 3: Conception	8
Introduction	8
3.1 Overall system architecture	8
3.1.1 MVC design pattern	8
3.1.2 General class diagram	9
3.2 sequence diagrams	10
3.2.1 Login sequence diagram	10
3.2.2 Register sequence diagram	11
3.2.3 Password reset sequence diagram	12
3.2.4 Add task sequence diagram	12
3.2.5 Update task sequence diagram	13
3.2.6 Delete task sequence diagram	13
Chapter 4: Realization	14
Introduction	14
4.1 Working environment	14
4.1.1 Hardware environment	14
4.1.2 Software environment	14
4.1.3 Programming languages	15
4.1.4 Frameworks	15
4.1.5 Database Management System	15
4.2 Main graphical user interfaces	16
4.2.1 Login page	16
4.2.2 Signup page	16
4.2.3 Task list page	17
4.2.5 Delete Task confirmation page	18
4.2.7 new password page	19
General conclusion	20

List of figures

Figure 1 : Company logo	2
Figure 2 : Foufousa organization chart	3
Figure 3 : General use case diagram	7
Figure 4 : Model-View-Controller (MVC)	9
Figure 5 : General class diagram	9
Figure 6 : Login sequence diagram	10
Figure 7 : Registration sequence diagram	11
Figure 8 : password reset sequence diagram	12
Figure 9 : Add task sequence diagram	12
Figure 10 : update task sequence diagram	13
Figure 11 : Delete task sequence diagram	13
Figure 12 : Login Page	16
Figure 13 : Signup page	16
Figure 14 : Task list page	17
Figure 15 : Add/Edit Task page	17
Figure 16 : Delete Task confirmation page	18
Figure 17 : Reset password page	18
Figure 18 : new password page	19
Figure 19 : Admin dashboard	19

List of tables

Table 1 : The functionalities of the system by actor	6
Table 2 : Hardware environment	14

General introduction

We will discuss the internship project that we conducted with the **Foufusa** team with you.

Through the use of web development tools and the understanding of the method the company uses to organize its work, this project allowed us to utilize and enhance our skills.

This report summarizes the work we accomplished for this project, which entailed creating a web application that maintains and handles day-to-day tasks.

Therefore, after giving a broad overview of Foufousa , we'll illustrate how we conducted our project's initial research, and then we'll concentrate on the procedures we used to carry out the project.

Chapter 1: General context of the project

Introduction

This introductory chapter is devoted to the presentation of the general context of the project.

I will start by introducing the hosting organization for my project. Next, I will provide a summary of my project by outlining the issues to be resolved and the suggested solutions.

1.1 Company presentation

Foufusa Engineering & Services is a company that specializes in providing top-notch services in the automotive field. Their services include vehicle safety and cyber security, system engineering, IoT, mechanical engineering, and software development.



Figure 1 : Company logo

Their team of highly skilled professionals is dedicated to delivering the best solutions and services to meet their clients' needs. They provide a wide range of services, including vehicle safety and cyber security, system engineering services, software development services, mechanical engineering services, embedded systems, and IoT services.

1.1.1 Organization chart

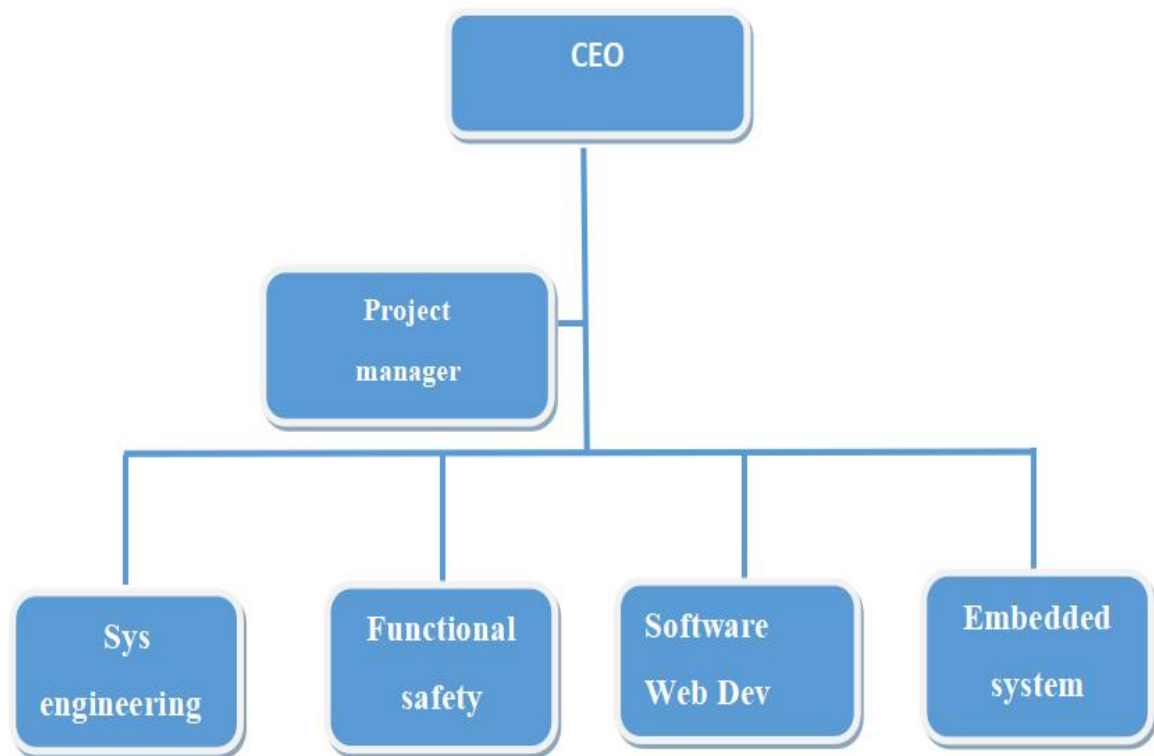


Figure 2 : Foufousa organization chart

1.2 initial study

1.2.1 Study of the existing

Although many people use to-do apps, each user approaches the apps differently depending on their unique needs.

We list the following qualities of to-do apps:

- easy set up (getting the app up and running quickly).
- Users may easily and rapidly add tasks to their to-do lists.
- the flexibility to quickly manage tasks and search to-do lists.

1.2.2 Criticism of the existing system

Advertisements and other easily accessible online distractions make it hard for us to stay focused on our tasks and make it difficult for us to do our work efficiently.

Switching back and forth between tasks frequently may give us the impression that we are being productive when we are not.

users often lose track of their time and delay work.

1.2.3 Proposed Solution

I was able to create a list of possible solutions based on users' needs after studying and testing a number of to-do applications and the themes I derived from them:

- Make it easy for users to manage todo lists, track time and set constraints in one place.
- Add the ability to easily add tasks and to search todo lists.
- A minimalist design without adverts helps to minimize distractions.
- Add the ability to track time and set deadlines.

Conclusion

My goal in this chapter was to outline the overall context of my project. I presented the host organization and the project study. The following chapter will concentrate on analyzing and specifying the requirements of our system.

Chapter 2: Specification of requirements

Introduction

After contextualizing the project, we present the actors as well as the functional and non-functional requirements of our application. Then the final step is to model the problem.

2.1 Project planning

2.1.1 Actors' Presentation

We list the actors who are likely to interact with the system. First, we must define what an actor is. An actor is an entity that exists outside the system. It represents a person or another computer system that anticipates one or more services provided by an access interface. It communicates with the system by sending and receiving messages. The following actors will interact with our system:

- **User** : A user's primary role is to create and organize tasks within the app. He may use the app to keep track of his daily tasks, work assignments, personal goals, and other responsibilities.
- **Admin** :The admin role is focused on ensuring that the app is reliable, secure, up-to-date, and optimized for the needs of users.

2.1.2 Functional requirements

The goal of this app is to help users organize and manage their tasks and responsibilities. with a simple and user-friendly interface for creating, updating, searching, and deleting tasks, as well as setting deadlines to help prioritize tasks. The table below depicts the application's various functionalities by actors.

Actors	Features
User	<ul style="list-style-type: none"> • Create account • Login • Reset password • Create tasks • Edit tasks • Delete tasks
Admin	<ul style="list-style-type: none"> • Login • Reset password • View users • Add users • Delete users • View lists • Manage lists

Table 1: The functionalities of the system by actor

2.1.3 Non-functional requirements

In this section, we will discuss the app's non-functional requirements:

- **Maintainability:** The code must be readable and understandable.
- **Reliability:** The application should operate consistently without errors.
- **Performance:** The proposed solution must be fast, reliable and continuously operational.
- **Security:** The application's data must be secure.

2.2 General Use Case diagram

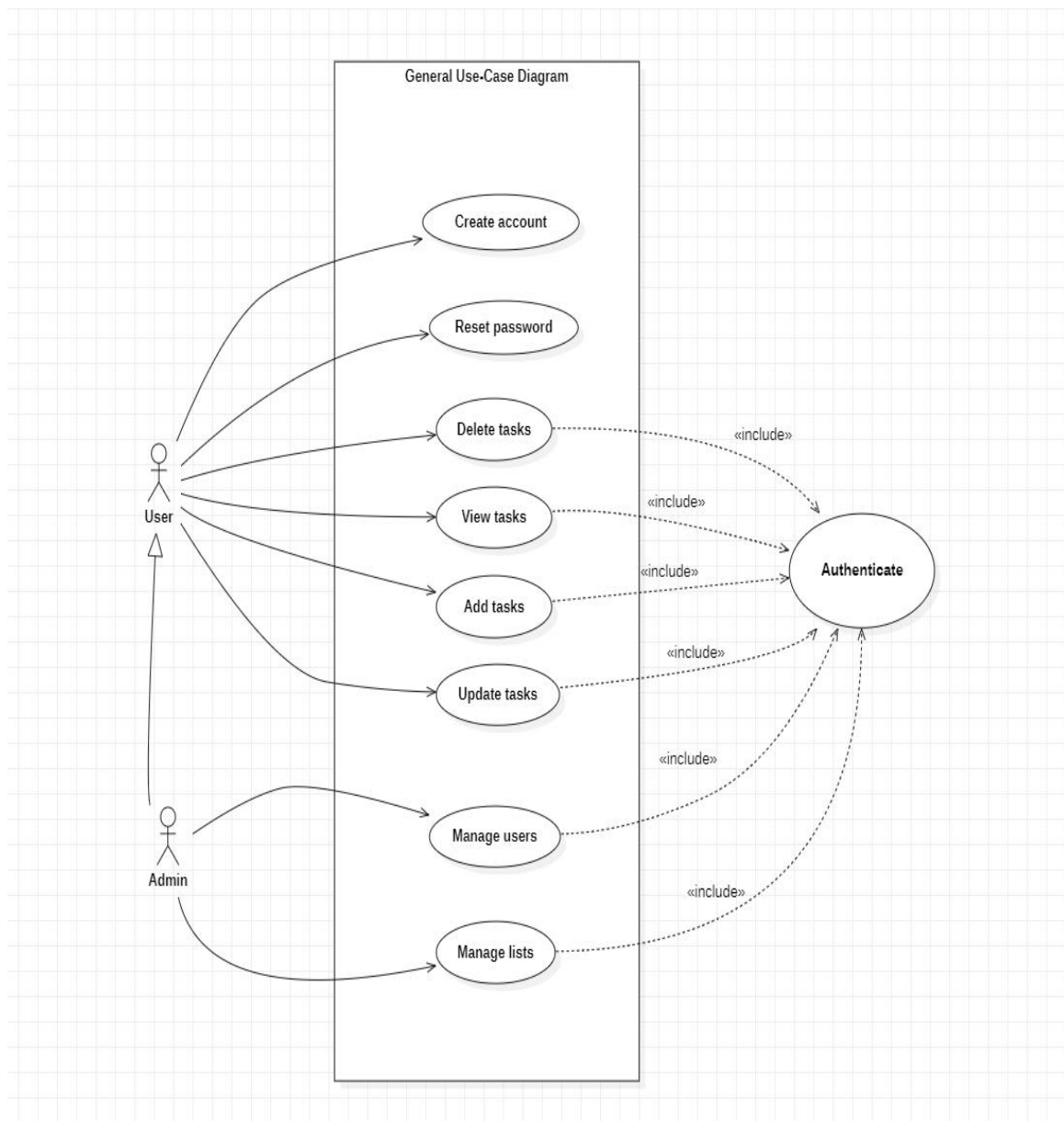


Figure 3: General use case diagram

Conclusion

The specification and analysis of our project's needs provide a clearer vision of the subject addressed as well as a deeper understanding of the tasks to be completed. After going through this section, it's time to assess the tools and technologies available to help us achieve our goals.

Chapter 3: Conception

Introduction

Design is an essential step in the software life cycle. It provides an overview of the solutions used to solve the application's problems. In this chapter, we first present the solution's general architecture, which is based on the MVC architecture, and then we draw the class diagram and the various sequence diagrams.

3.1 Overall system architecture

3.1.1 MVC design pattern

The Model-View-Controller (MVC) is a software design pattern that separates an application's data, user interface, and control logic into three separate components. This separation allows for easier maintenance and scalability of the application.

Here's how the MVC pattern is typically implemented:

- **Model:** The model component is responsible for managing the application's data and business logic. It receives input from the controller, performs the necessary operations on the data, and sends the results back to the controller for display to the user. The model does not have any knowledge of the user interface.
- **View:** The view component is responsible for presenting the data to the user. It receives data from the controller and displays it in the user interface. The view does not have any knowledge of the application's data or business logic.
- **Controller:** The controller component is responsible for managing the flow of data between the model and the view. It receives user input from the view, performs the necessary operations on the data in the model, and updates the view with the results. The controller acts as an intermediary between the model and the view and does not have any knowledge of the application's data or business logic.

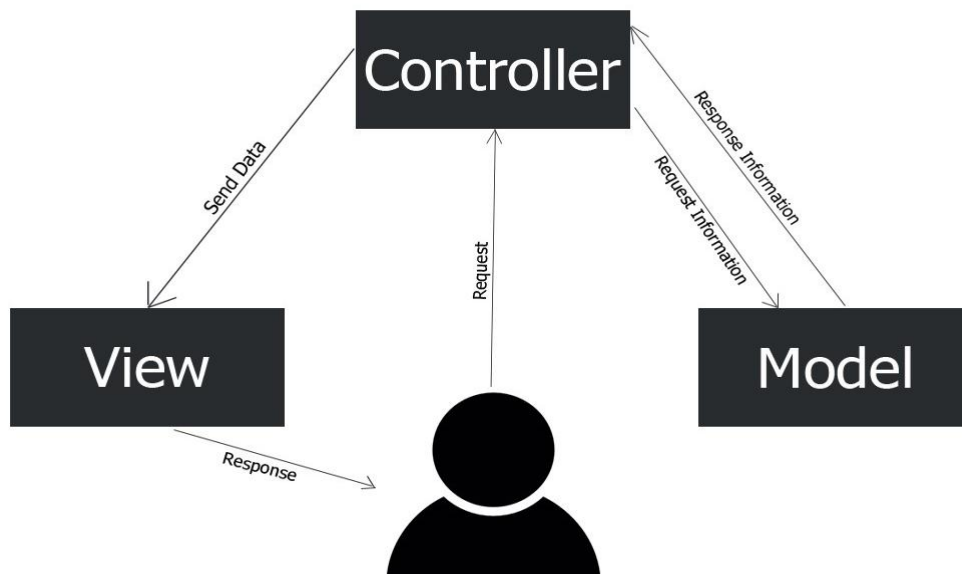


Figure 4: Model-View-Controller (MVC)

3.1.2 General class diagram

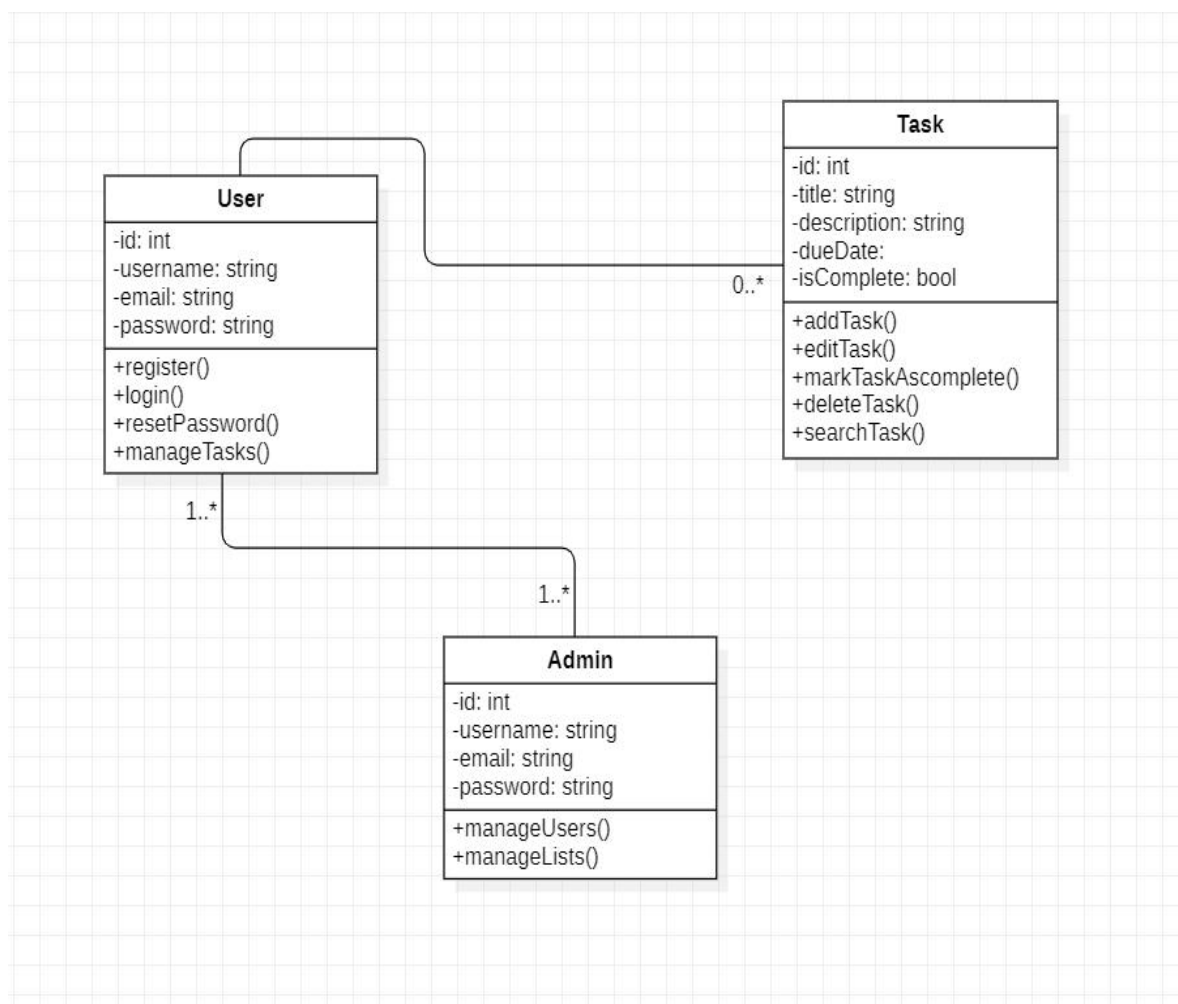


Figure 5: General class diagram

3.2 sequence diagrams

3.2.1 Login sequence diagram

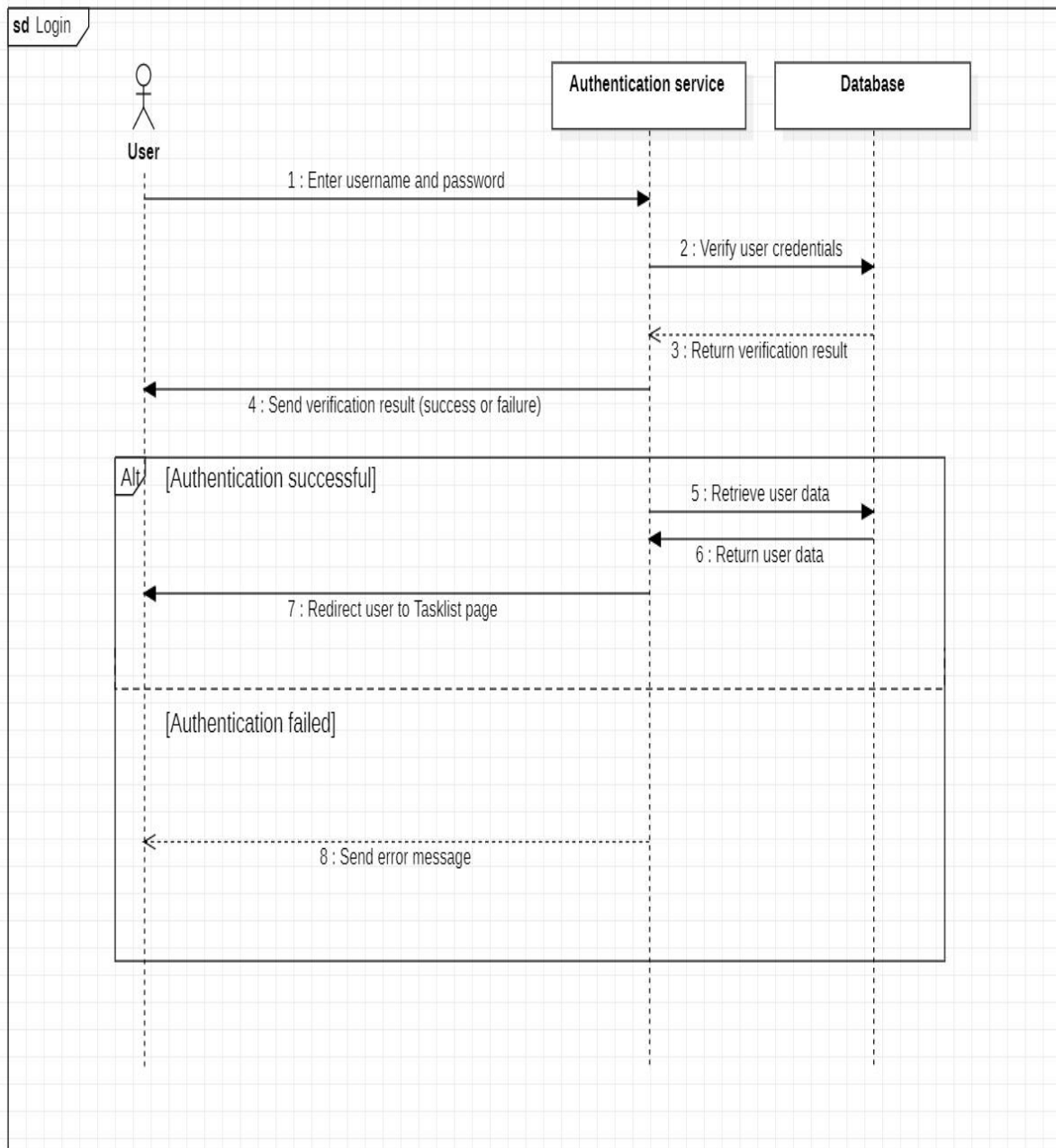


Figure 6: Login sequence diagram

3.2.2 Register sequence diagram

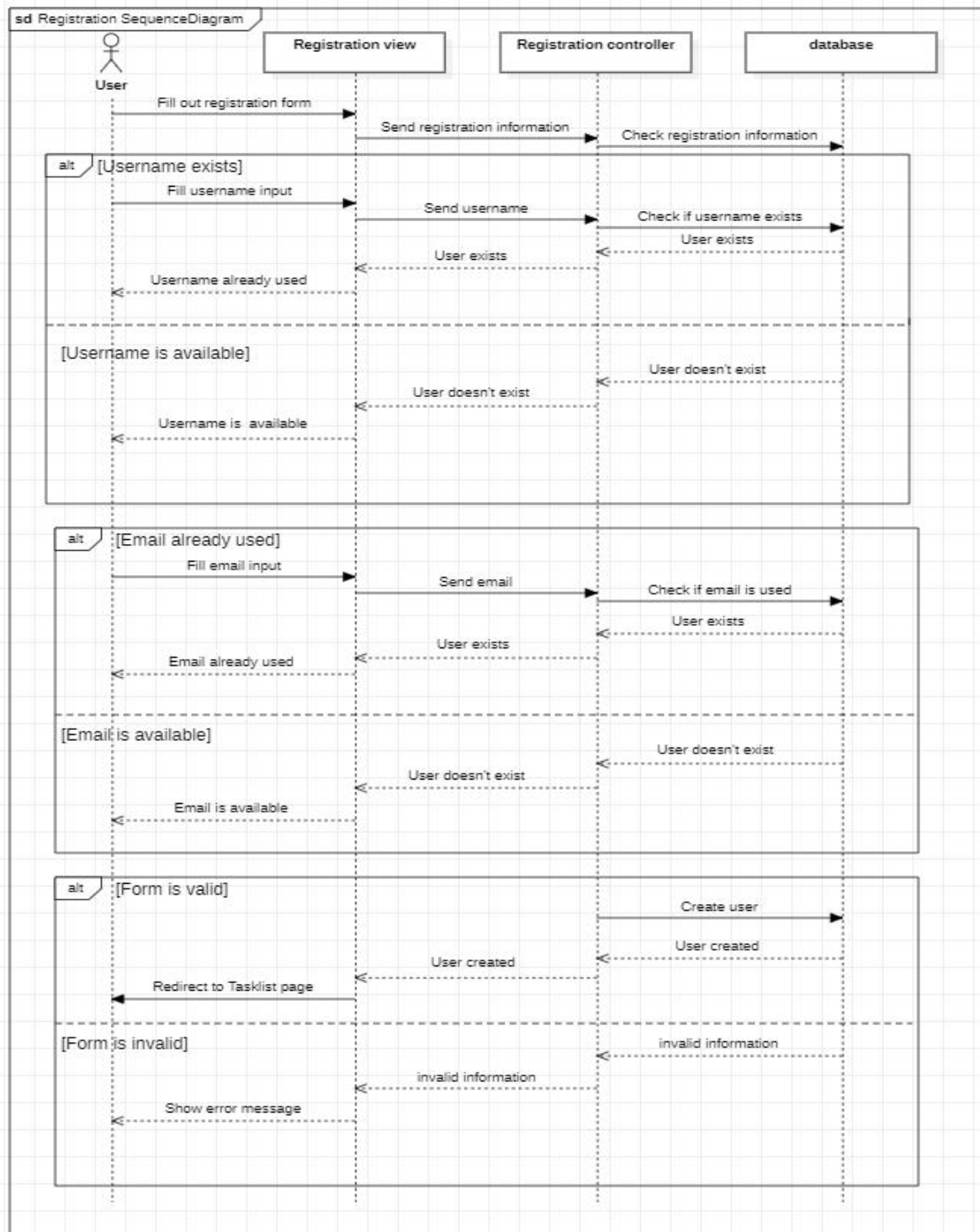


Figure 7: Registration sequence diagram

3.2.3 Password reset sequence diagram

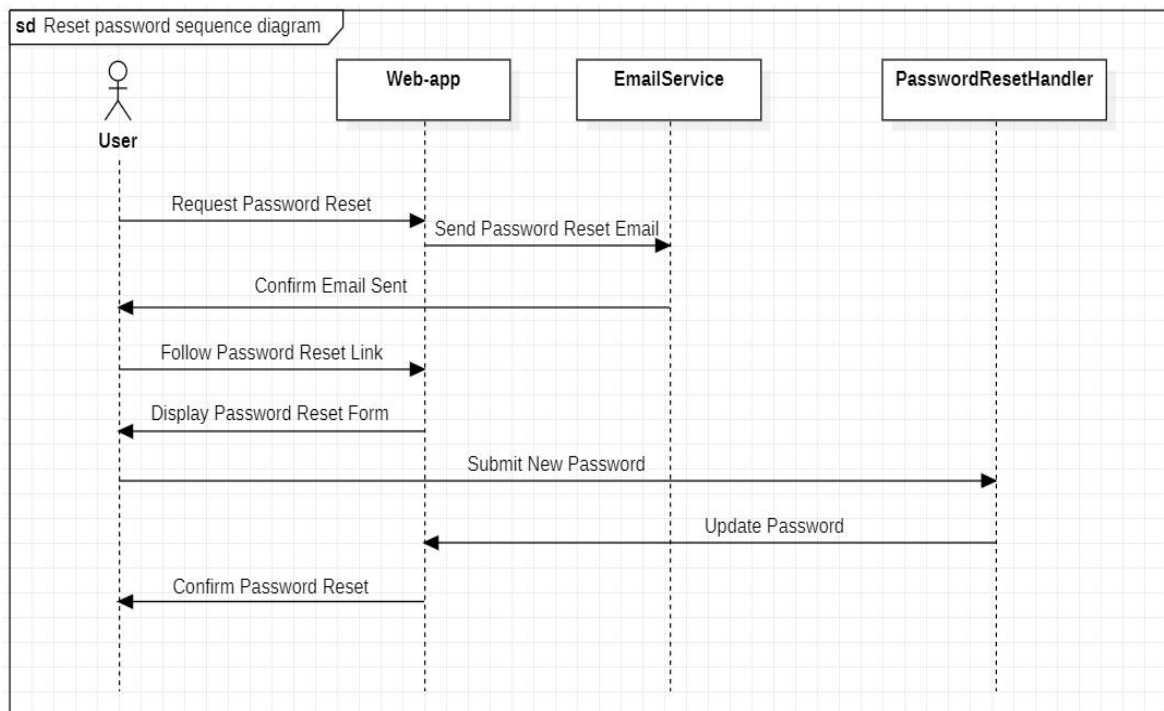


Figure 8: password reset sequence diagram

3.2.4 Add task sequence diagram

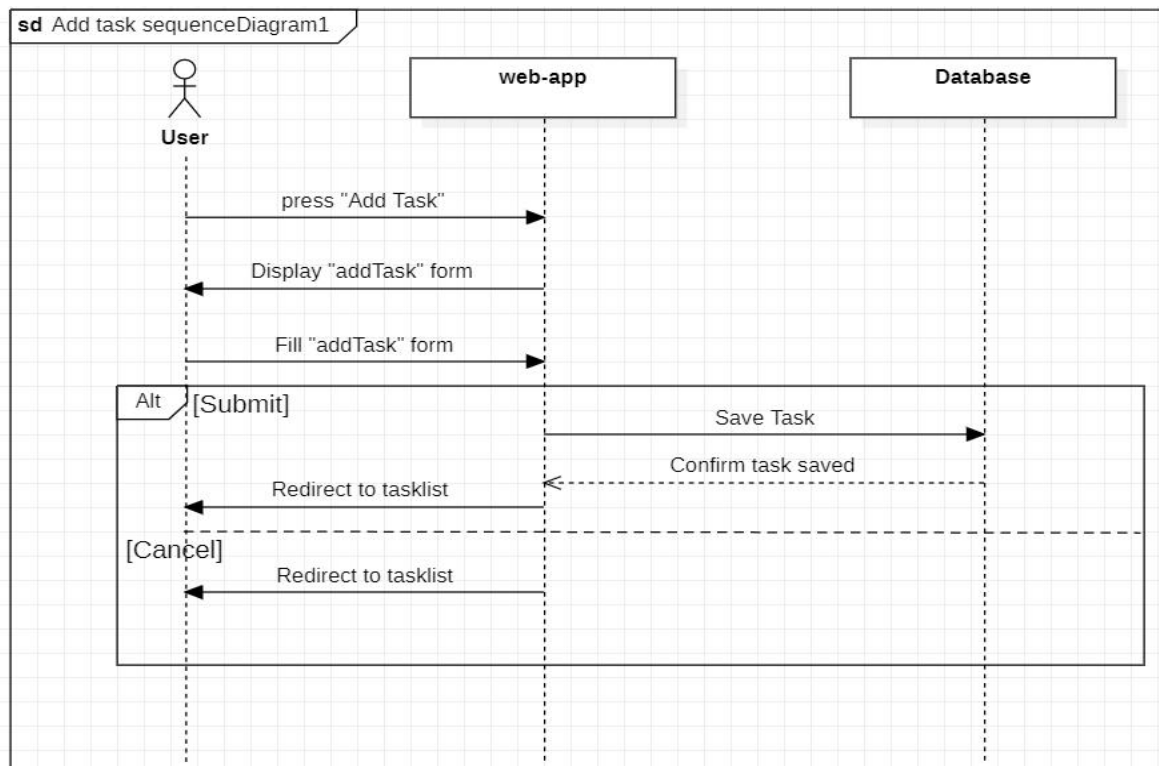


Figure 9: Add task sequence diagram

3.2.5 Update task sequence diagram

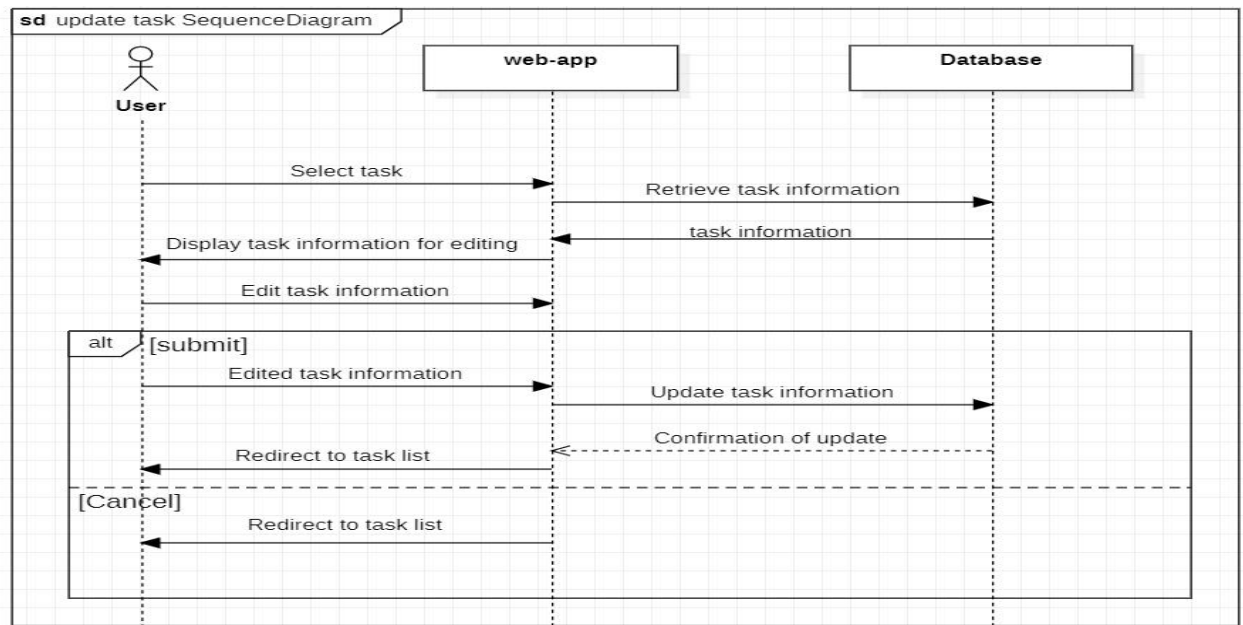


Figure 10: update task sequence diagram

3.2.6 Delete task sequence diagram

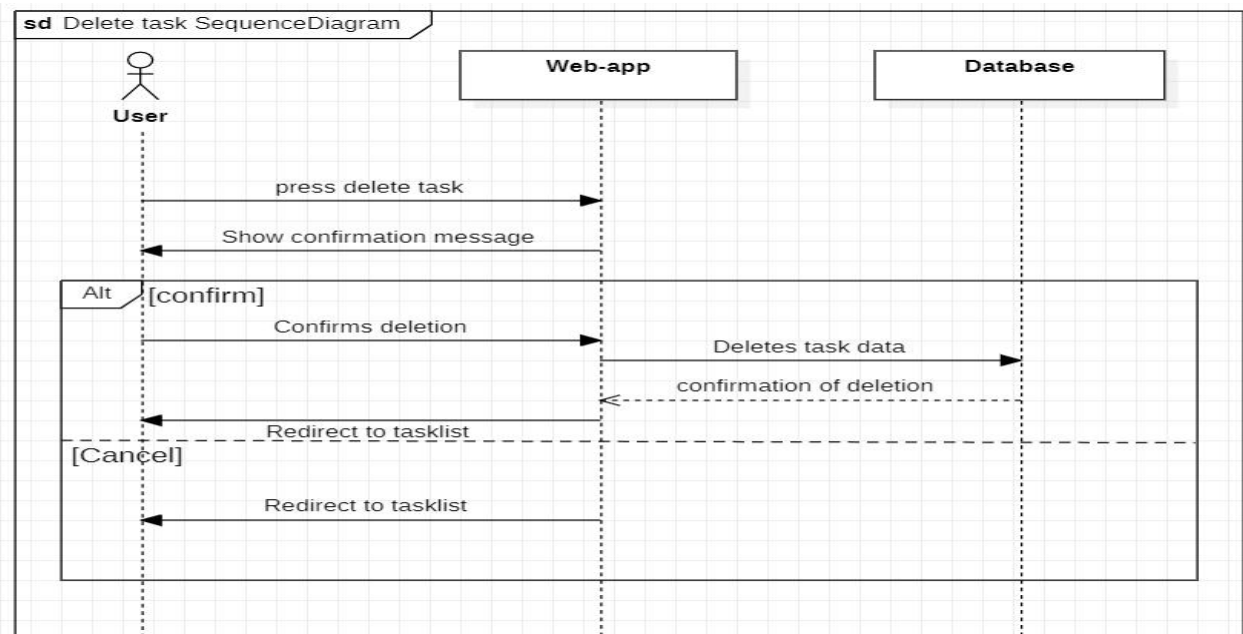


Figure 11: Delete task sequence diagram

Conclusion

This chapter gave a general overview of the architecture and design of my project. Detailed diagrams were included to show the high-level structure of the system, including the modules, components, and their relationships. The final chapter will concentrate on the project's realization.

Chapter 4: Realization

Introduction

The implementation phase marks the end of the software development process. The prior phases' planning and design have now been converted into functional code. Using several screenshots, we'll give a thorough explanation of the work that was done.

4.1 Working environment

In computer science, the term "environment" refers to all the components and system software, including the operating system, that are used to run an application's programs. In this area, we usually list the different kinds of environments we have used along with their features.

4.1.1 Hardware environment

The machine I used to develop my platform and application had the following technical specifications:

- **Brand: Asus**
- **CPU: Intel Core i7**
- **RAM: 16GO**
- **Operating system: Windows 10 x64**

Table 2: Hardware environment

4.1.2 Software environment

I've chosen software, programming languages, libraries, and frameworks for the development of my web application.



Visual Studio Code, or **VS Code**, is an extensible code editor developed by Microsoft for Windows, Linux, and macOS.



GitHub is a web-based software development management and hosting service that also provides access control and collaboration features for each project such as bug tracking, feature requests, task management, and a wiki.



StarUML is a software engineering tool for system modeling that supports the Unified Modeling Language, Systems Modeling Language, and traditional modeling notations. It is published by MKLabs and is available on Windows, Linux and MacOS.

4.1.3 Programming languages



HTML5, or **HyperText Markup Language 5**, is a version of the popular HTML format used to create web pages. This one is summarized in a markup language, which is used to write the hypertext essential for Web page formatting.



CSS is a style sheet language used to describe the presentation of a document written in a markup language such as HTML or XML (including XML dialects such as SVG, MathML, or XHTML). CSS, like HTML and JavaScript, is a foundational technology of the World Wide Web.

4.1.4 Frameworks



Django is a high-level Python web framework that promotes rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source.

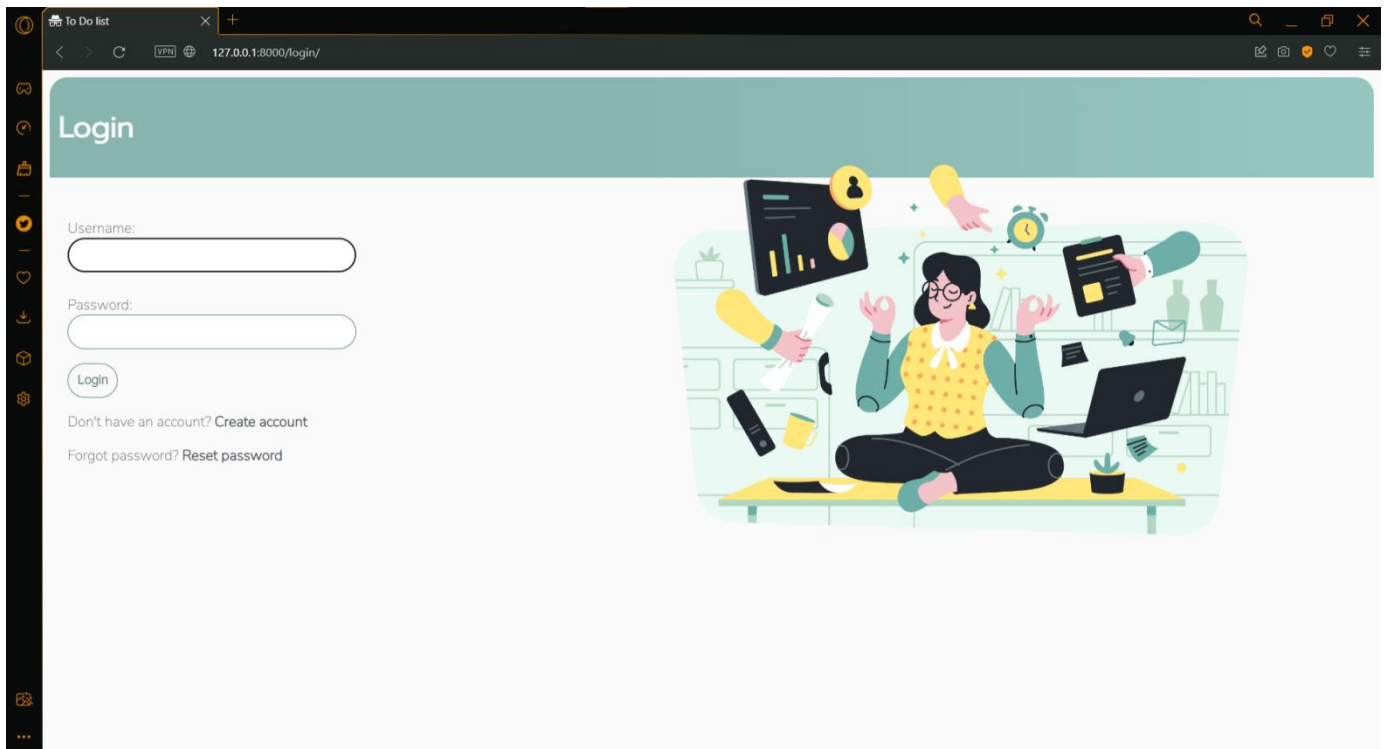
4.1.5 Database Management System



PostgreSQL is a powerful, open source object-relational database system with over 35 years of active development that has earned it a strong reputation for reliability, feature robustness, and performance.

4.2 Main graphical user interfaces

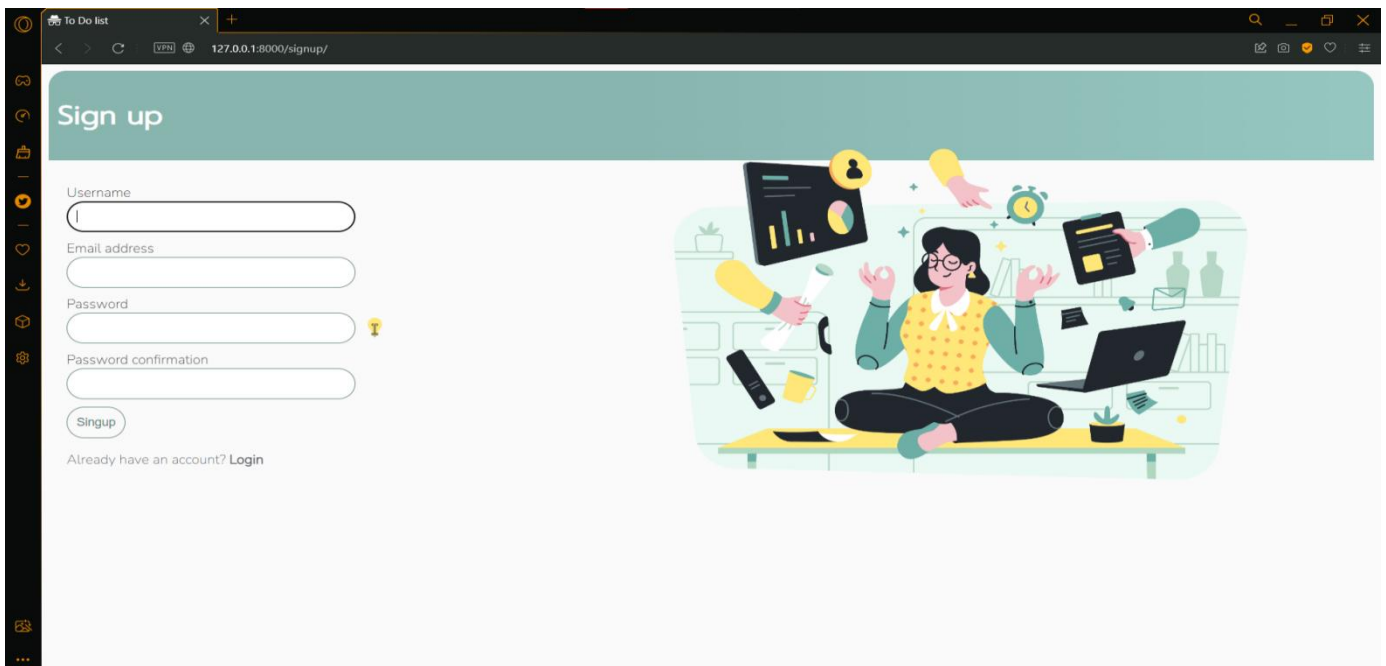
4.2.1 Login page



The screenshot shows a web browser window with the address bar displaying "127.0.0.1:8000/login/". The page has a teal header with the word "Login" in white. On the left, there is a form with two input fields labeled "Username:" and "Password:", a "Login" button, and two links: "Don't have an account? Create account" and "Forgot password? Reset password". On the right, there is a large, colorful illustration of a woman with glasses and a yellow polka-dot vest, sitting cross-legged on a yellow desk. She is surrounded by various office items: a laptop, a smartphone, a clock, a calendar, a coffee cup, and some papers. The background is a light blue with some abstract shapes.

Figure 12: Login Page

4.2.2 Signup page



The screenshot shows a web browser window with the address bar displaying "127.0.0.1:8000/signup/". The page has a teal header with the words "Sign up" in white. On the left, there is a form with four input fields labeled "Username", "Email address", "Password", and "Password confirmation", followed by a "Signup" button. Below the button is a link: "Already have an account? Login". On the right, there is a large, colorful illustration of a woman with glasses and a yellow polka-dot vest, sitting cross-legged on a yellow desk. She is surrounded by various office items: a laptop, a smartphone, a clock, a calendar, a coffee cup, and some papers. The background is a light blue with some abstract shapes.

Figure 13: Signup page

4.2.3 Task list page

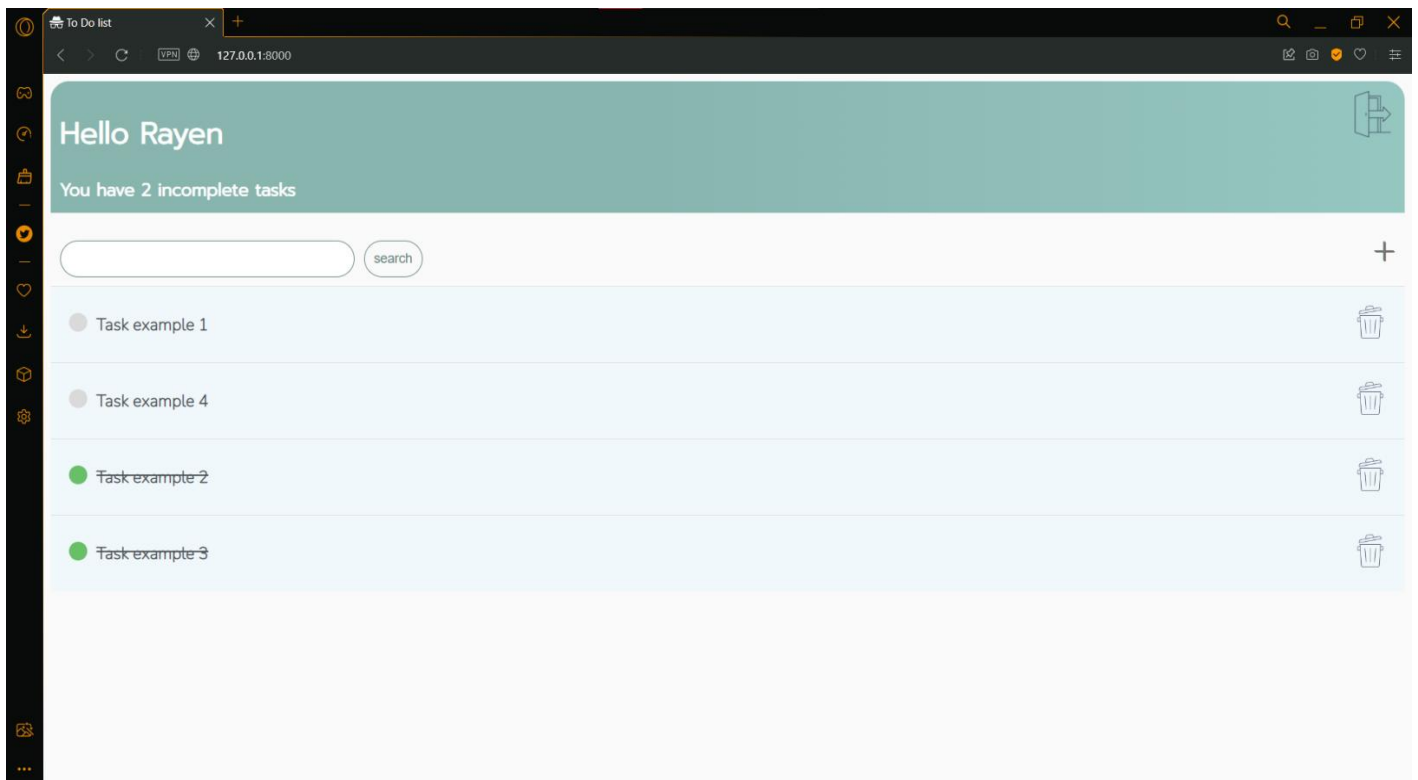


Figure 14: Task list page

4.2.4 Add/Edit Task page

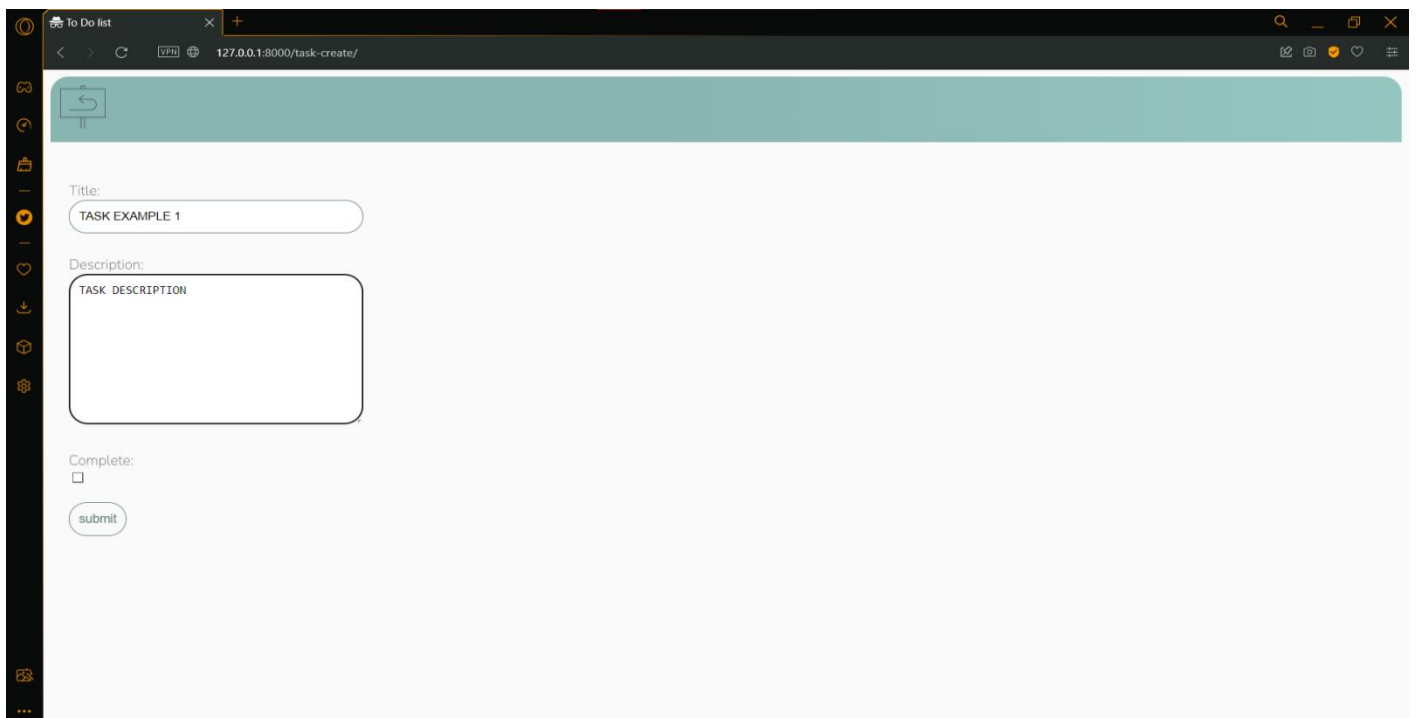


Figure 15: Add/Edit Task page

4.2.5 Delete Task confirmation page

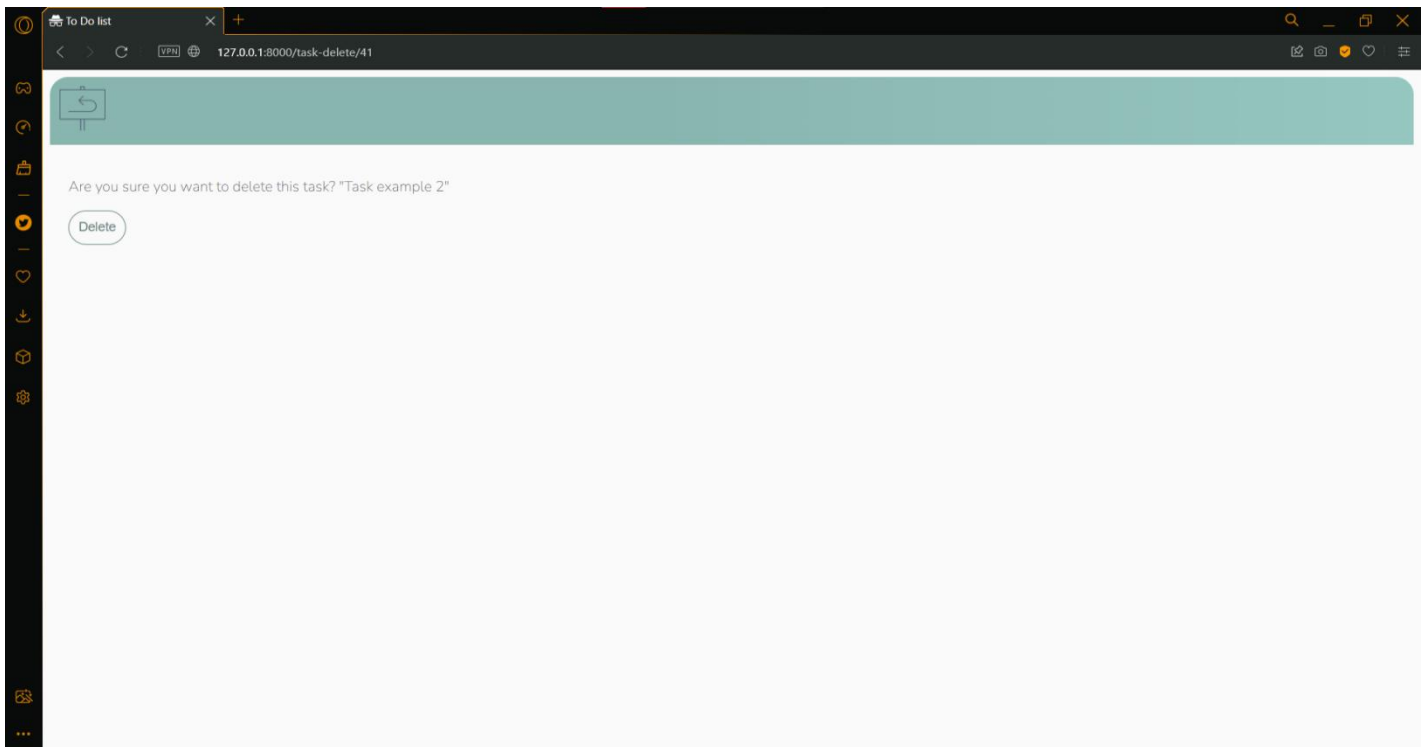


Figure 16: Delete Task confirmation page

4.2.6 Reset password page

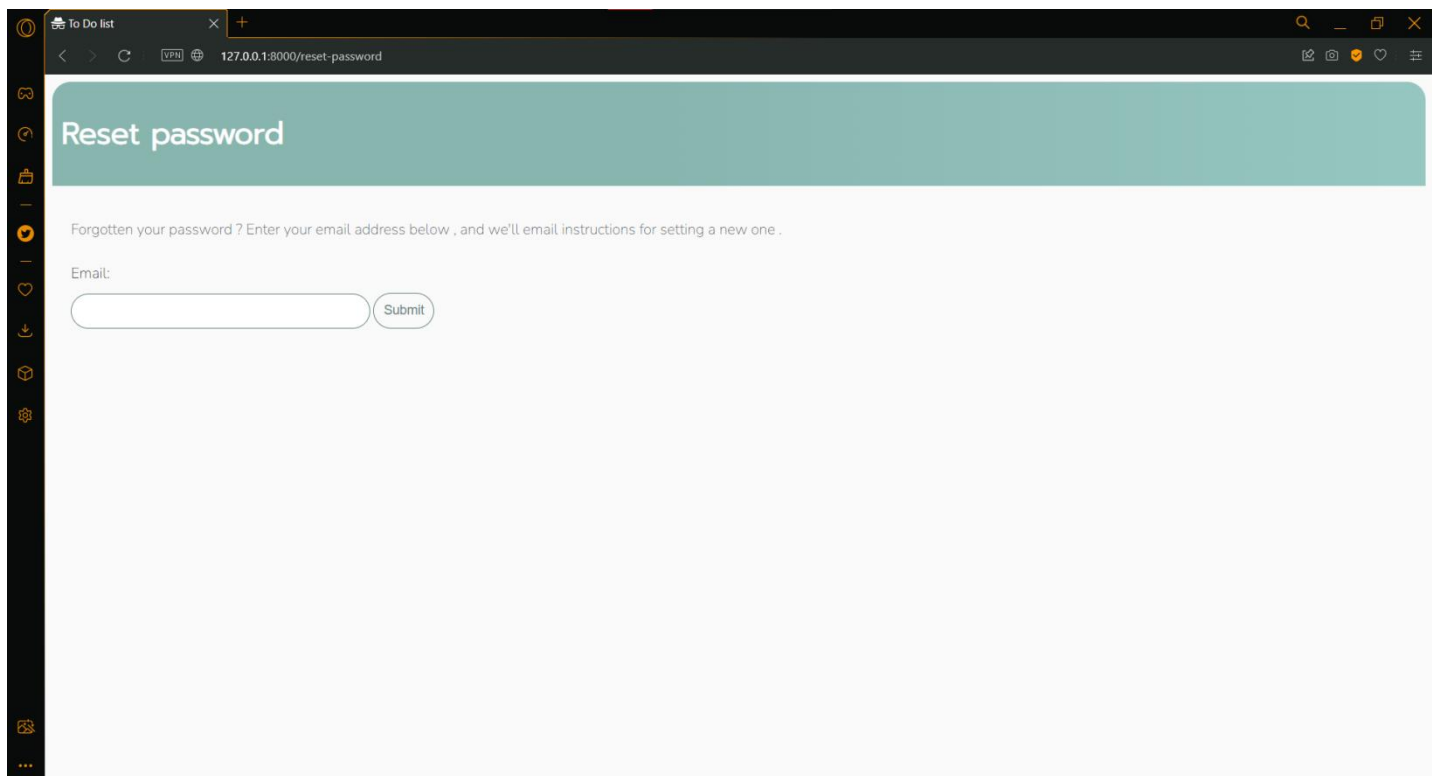


Figure 17: Reset password page

4.2.7 new password page

New password

Please enter your new password.

New password:

- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

New password confirmation:

Figure 18: new password page

4.2.8 Admin dashboard

Django administration

WELCOME, RAYEN. VIEW SITE / CHANGE PASSWORD / LOG OUT

Site administration

AUTHENTICATION AND AUTHORIZATION

Groups [+ Add](#) [Change](#)

Users [+ Add](#) [Change](#)

BASE

Tasks [+ Add](#) [Change](#)

Recent actions

My actions

None available

Figure 19: Admin dashboard

General conclusion

To sum it up, developing a to-do app was both an exciting and challenging project. The app was designed to help users manage their tasks more effectively. Throughout the internship, I gained valuable experience and knowledge in a wide range of technologies, including software development, database management, and user interface design, as well as the importance of proper planning and design prior to project implementation.

I also came to the realization that there is no correct or incorrect method to design a solution, and there is also no ideal approach to do it. Iteration and the inclusion of customer feedback are the only ways to achieve a design goal and create the ideal product.