

# **Design, build, test, and deploy a piece of software for a specific purpose**

Matt Hyde

registration: [Private]

## 1. Introduction

This project is to design, build, test, and deploy a piece of software for a specific purpose. The purpose chosen is project management as there are many software solutions available with various tools to support people's projects. It can be hard to find the system that best fits what you require and so research will be conducted to determine what features users need and desire. The main aim is to produce an efficient and secure project management system which meets the requirements identified through research and analysis. As research is conducted over the project, new requirements will be discovered. Because of this, an Agile approach will be used to develop updated versions of the system which reflect the changing requirements. The objectives are to:

- Complete the first iteration:
  1. Analyse potential risks and create plans to mitigate them
  2. Create a project plan to help monitor progress
  3. Research the features of existing, similar systems
  4. Research the use of project management software through an online survey
  5. Investigate existing surveys on project management
  6. Use the research to engineer the system requirements
  7. Define test cases for the system requirements
  8. Create models of the system to guide implementation
  9. Implement the system with suitable technologies
  10. Design the system with considerations for mitigating security threats
  11. Create unit and functional tests of the system's reliability and integrity
  12. Create system documentation and user guides to support end-users
  13. Validate the first iteration of the system with the test cases
- Complete the second iteration:
  1. Review the research and update requirements
  2. Update the test cases, models and documentation
  3. Implement the new features along with unit and functional tests
  4. Validate the second iteration of the system with the test cases

## 2. Progress Report

### 2.1. Research

#### 2.1.1. Similar Systems Analysis

The project began with an analysis of existing project management systems. This was done to identify common features that would be required in the project system. Asana, Clickup, Jira, Monday.com and Trello were chosen as they are popular systems with various features. The analysis has been presented as a feature matrix to show a simple comparison of the features available in each system. The results identify that the features available across the free-to-use versions of each system consist of:

- User sign-up and login
- Creation of projects and tasks with deadlines
- Kanban boards
- Collaboration on projects
- Assigning team members to tasks
- Updating the status of a task
- Notifications
- File sharing

The full feature matrix can be seen in table 1 in appendix A.

#### 2.1.2. Online Survey

To identify the main user-base for the finished system and the features they require, an anonymous online survey is being conducted. The survey uses a closed-question asking participants to select the features they use. This collects quantitative data that can be presented as a bar chart to show clear trends in feature popularity. The survey also uses open-questions to collect qualitative data on why features are important to the participant and the features they would like added to the software they use. The results will be regularly analysed and used to update the requirements over the project.

### **2.1.3. Investigation into existing surveys**

Results from existing surveys have been used to support the requirements. According to White and Fortune (2002), Gantt charts are one of the most popular tools used by project managers. This is supported by the fact that Gantt chart creation was common in the feature matrix. Zwikael and Globerson (2006) found communication between the project team and stakeholders important for project success. Group messaging was common in the feature matrix and could be used to facilitate communication with stakeholders within the system. As the importance of these features has been highlighted by the research, they should be considered for additional system requirements.

## **2.2. Design**

### **2.2.1. MoSCoW Analysis**

To identify what features must be prioritised to develop a system that meets the needs of common users, a must-have, should-have, could-have, won't-have (MoSCoW) analysis was performed. The must-haves cover the features available across the free-to-use versions of the systems in the feature matrix. The should-haves and could-haves are the features that were present in most of the systems. The wont-haves are the uncommon features. The results can be read in table 2 in appendix B.

### **2.2.2. Shall Statements**

A set of requirements covering the must-have features was created. Team messaging is included to support team and stakeholders communication. Accessibility settings are also included to make the system usable for a larger group of people. The requirements are written as shall statements to make their meaning clear and unambiguous. FR is used to denote a functional requirement, NFR is used to denote a non-functional requirement. The shall statements must be adhered to during implementation so that the system meets the requirements. The shall statements can be read in table 3 in appendix C.

### **2.2.3. Test Cases**

Test cases were defined for each shall statement, covering their expected inputs and outputs. Once an iteration of the system has been produced, the test cases can be used

to validate the system and ensure that it meets all the requirements. The test cases can be found in table 4 in appendix D.

#### **2.2.4. Use Case Diagrams**

Use case diagrams were created to visualise the tasks users can perform in the system. This was done to explain how the system reacts to user interaction in a way which meets all of the shall statements. They can also be used to validate the system during testing by attempting to perform a task and comparing the result with the relevant diagram. The use case diagrams can be seen in appendix E.

#### **2.2.5. UML Class Diagram**

A unified modelling language (UML) class diagram was created to begin planning a suitable method of implementing the system's classes and methods. This diagram should be used as guidance when developing early versions of the system and must be updated as changes are made and new features are implemented. The class diagram can be seen in figure 12 in appendix F.

### **2.3. Technologies**

#### **2.3.1. Architecture**

The Model-View-Controller architecture has been chosen for this system. This was chosen to simplify development by separating the application into layers so that changes can be made to one layer without having to make large changes to the rest of the application.

#### **2.3.2. Front-end**

React will be used to produce the front-end as it allows users to efficiently interact with the system through fast and responsive views. The Vite development environment will be used to speed up implementation as it can quickly set up React projects and display changes made in the code without restarting the development environment. The Tailwind CSS framework and DaisyUI library have been used to create clear user interface (UI) components to improve the user experience. The React testing framework will be used to create tests to ensure that the front-end is working correctly.

### **2.3.3. Back-end**

Papers investigating back-end frameworks have been researched and used to select Django for the project's back-end. Kaluža et al. (2019) compared several back-end frameworks and found Django the best overall and is highly testable, whilst allowing you to validate incoming data. This makes Django a suitable choice for the objectives of testing the system and mitigating security threats. The unittest framework will be used to create tests to ensure that the back-end is working correctly.

### **2.3.4. Authentication**

Authentication is handled by Auth0. This was chosen as it provides easy-to-use UIs for signing up, logging in and resetting passwords. Another advantage is that Auth0 encrypts user information, helping to meet the objective of mitigating security threats.

## **2.4. Development**

### **2.4.1. Front-end development progress**

A welcome message and login/sign-up button is displayed when first accessing the system. Clicking the button navigates to the Auth0 UI. Once logged in, the landing page displays cards with information on the different pages and buttons to navigate to them. There's a header with a navigation menu button in the top left, a title button in the middle and a welcome message with the user's name in the top right alongside the logout button. The header is fixed throughout the app to simplify navigation. At the bottom of the screen there is a fixed footer with links to user resources and a contact section where users can submit their email. There's a page for error handling, informing the user of the issue that occurred with a link back to the home page. Backups are stored in a GitHub repository. Screenshots of the application can be seen in appendix G.

## **3. Progress Evaluation and Risk Analysis**

### **3.1. Current achievements**

All the work that needed to be done at this stage has been completed. Including, the initial research into the problem area, creating early iterations of the system requirements

and design, and planning the technologies that will be used. This was vital for creating a clear plan for producing a system that will meet the requirements of the end-users. Additional progress has also been made on the development of the UI.

### **3.2. Problems Encountered**

Gaining ethics approval for the survey took longer than planned, delaying the data gathering. Once the study was approved, it was shared on social media however, gaining participants has been difficult. To deal with this, the survey will need to be shared more frequently across a wider range of platforms. If this does not work, further research will be conducted into existing surveys on project management. The results gathered from other studies can be used to support the project requirements if the amount of data gathered by the online survey is insufficient.

### **3.3. Next steps**

Work on the UI will continue, research into libraries and packages that can be used to implement the features may be required. Back-end development will begin with Django but if progress is slow, Express.js will be used as it's a more familiar framework and can be used to get the project back on track. Testing frameworks will be used to create unit and functional tests throughout development. A database management system will be selected when the back-end development begins. The first iteration of the system should be completed by March, research results will then be reviewed and planning will begin for the second iteration. To identify potential risks and mitigation plans, a risk analysis was performed and can be read in table 5 in appendix H. A Gantt chart has been used to organise tasks and guide development and can be seen in appendix I.

## **4. Conclusion**

In conclusion, research has been used to engineer requirements and define test cases. Models of the system have been created to guide the implementation. Suitable technologies have been selected to produce an efficient and secure system. The front-end coding has began, with the system allowing sign-up, login and navigation through a simple UI. The next steps have been outlined and a risk analysis has been performed.

## References

- Kaluža, M., Kalanj, M., and Vukelić, B. (2019). A comparison of back-end frameworks for web application development. *Zbornik veleučilišta u rijeci*, 7(1):317–332.
- White, D. and Fortune, J. (2002). Current practice in project management — an empirical study. *International Journal of Project Management*, 20(1):1–11.
- Zwikael, O. and Globerson, S. (2006). From critical success factors to critical success processes. *International journal of production research*, 44(17):3433–3449.

## A. Feature Matrix

Table 1: Feature matrix of similar systems

| Feature name                | Asana                 | Clickup | Jira | Monday.com            | Trello                |
|-----------------------------|-----------------------|---------|------|-----------------------|-----------------------|
| User sign-up                | Yes                   | Yes     | Yes  | Yes                   | Yes                   |
| User login                  | Yes                   | Yes     | Yes  | Yes                   | Yes                   |
| Create projects             | Yes                   | Yes     | Yes  | Yes                   | Yes                   |
| Kanban boards               | Yes                   | Yes     | Yes  | Yes                   | Yes                   |
| Gantt charts                | Yes (on paid account) | Yes     | Yes  | Yes (on paid account) | Yes                   |
| Calendars                   | Yes                   | Yes     | Yes  | Yes (on paid account) | Yes (on paid account) |
| Collaboration               | Yes                   | Yes     | Yes  |                       | Yes                   |
| Group messaging             | Yes                   | Yes     | Yes  |                       | No                    |
| Direct messaging            | No                    | Yes     | No   |                       | No                    |
| Create tasks                | Yes                   | Yes     | Yes  |                       | Yes                   |
| Task list                   | Yes                   | Yes     | Yes  | No                    | No                    |
| Task dependencies           | No                    | Yes     | Yes  | Yes (on paid account) | No                    |
| Assign person to task       | Yes                   | Yes     | Yes  |                       | Yes                   |
| Task status                 | Yes                   | Yes     | Yes  |                       | Yes                   |
| Task priorities             | Yes                   | Yes     | No   | Yes                   | No                    |
| Start dates                 | Yes (on paid account) | No      | Yes  | No                    | Yes                   |
| End dates                   | Yes                   | Yes     | Yes  |                       | Yes                   |
| Notifications               | Yes                   | Yes     | Yes  |                       | Yes                   |
| Notifications for messages  | Yes                   | Yes     | Yes  |                       | No                    |
| Notifications for tasks due | Yes                   | Yes     | Yes  | Yes (on paid account) | Yes                   |
| Analytics                   | Yes (on paid account) | Yes     | Yes  | Yes (on paid account) | No                    |
| File sharing                | Yes                   | Yes     | Yes  |                       | Yes                   |
| Accessibility settings      | Yes                   | Yes     | No   | No                    | Yes                   |
| Whiteboards                 | No                    | Yes     | No   | No                    | No                    |
| Portfolios                  | Yes (on paid account) | No      | No   | No                    | No                    |
| Goals                       | Yes (on paid account) | Yes     | Yes  | No                    | No                    |
| Workloads                   | Yes (on paid account) | Yes     | No   | Yes (on paid account) | No                    |
| Task table                  | No                    | Yes     | No   | Yes                   | Yes (on paid account) |
| Risk management             | Yes (on paid account) | No      | No   | No                    | No                    |
| Mind maps                   | No                    | Yes     | No   | No                    | No                    |
| Time-sheets                 | No                    | Yes     | No   | No                    | No                    |
| Document collaboration      | No                    | Yes     | No   | No                    | No                    |
| Clips sharing               | No                    | Yes     | No   | No                    | No                    |
| Forms                       | Yes (on paid account) | Yes     | Yes  |                       | No                    |
| Code collaboration          | No                    | No      | Yes  | No                    | No                    |

## B. MoSCoW Analysis

Table 2: MoSCoW Analysis

| Must Haves                                    | Should Haves           | Could Haves       | Wont Haves             |
|---|------------------------|-------------------|------------------------|
| User sign-up/login                            | Accessibility settings | Start dates       | Code collaboration     |
| Creation of projects and tasks with deadlines | Team messaging         | Task tables       | Document collaboration |
| Kanban boards                                 | Gantt charts           | Project analytics | Whiteboards            |
| Invite users to projects                      | Calendars              | Workloads         | Risk management        |
| Remove user from projects                     | Forms                  | Task dependencies | Mind maps              |
| Accept/decline invite                         |                        | Goals             | Clips sharing          |
| Assign team members to tasks                  |                        | Time sheets       | Portfolios             |
| Update the status of a task                   |                        | Task priorities   | Direct messaging       |
| Notifications                                 |                        |                   |                        |
| File sharing                                  |                        |                   |                        |

## C. Shall Statements

Table 3: Shall statements

| Key   | Requirement   | Description  | Type |
|-------|---|--|------|
| S1.1  | A user shall be able to sign-up to the system with a username, email and password.          | Users need to be able to create accounts so they can access the system and its features.   | FR   |
| S1.2  | A user shall only be able to sign-up with an email that does not already have an account    | User email addresses must be unique and so the system will not allow one email address to have multiple accounts.  | FR   |
| S1.3  | A password policy shall be used to make user's passwords more secure.                       | Users will be made to create strong passwords to help protect their accounts from malicious users.   | FR   |
| S1.4  | User login information shall be stored securely.  | The usernames, emails and passwords must be encrypted and protected from malicious users.  | FR   |
| S2.1  | Users shall be able to login to the system.   | Users must be able to enter their details in the login form and then gain access to the rest of the system.  | FR   |
| S2.2  | Users shall be immediately informed as to why they've failed to login to the system.        | If a user fails to login, they will be prompted by the system explaining why.  | NFR  |
| S3.1  | Users shall be able to logout of the system.  | Users must be able to see and click a button that allows them to logout of their account when they are finished using the system.  | FR   |
| S4.1  | Users shall be able to create projects.   | Users must be able to access a projects page where they can create projects with a title, description, tasks with deadlines and end date.  | FR   |
| S4.2  | The validity of project information shall be checked when a user tries to create a project. | If a user tries to add invalid information such as project end date that is in the past, the system must notice and prompt the user to change it before they can create the project. | FR   |
| S4.3  | Projects shall be presented as a Kanban board.  | A project and its tasks will be presented as a Kanban board with cards representing tasks and columns for tasks that need to be done, are in progress and are done.                  | FR   |
| S5.1  | A user that creates a project, shall be assigned as its team leader.                        | A user that creates a project will have a more permissions than other users who join the project, including updating and deleting the project and inviting or removing users.        | FR   |
| S6.1  | A team leader shall be able to update their project.  | Team leaders must be able to make changes to their project information, such as the title, tasks or end date.  | FR   |
| S7.1  | A team leader shall be able to delete their project.  | If a team leader needs to cancel a project, they must be able to delete it in the app.   | FR   |
| S8.1  | A team leader shall be able to invite other users to the project.                           | A team leader must be able to type in a user's email address and send them an invitation to work on the project.   | FR   |
| S9.1  | A team leader shall be able to remove users from their project.                             | A team leader must be able to access a list of users on their project and have the option to remove them from the project.   | FR   |
| S10.1 | Users shall be able to view invitations to projects.  | Users can access the projects page and see the invitations they have received.   | FR   |
| S10.2 | Users shall be notified when they've been invited to a project.                             | Users who are invited to a project will be sent a notification that can be read by clicking on the notification icon in the nav bar.   | FR   |
| S10.3 | Notifications shall be indicated to the user with an icon.                                  | When the user has unread notifications, the notification icon in the nav bar will have a blue dot above it, to inform the user that they have a new notification.                    | NFR  |
| S11.1 | Users shall be able to accept or decline an invite.   | Users must have the option to click a button and join the project they've been invited to, or to decline and dismiss the notification.   | FR   |
| S12.1 | Users shall be able to view the information of a project they are a member of.              | Through the projects page, a user can click on a project they are a member of and view its information, including the description, tasks, team members and deadlines.                | FR   |
| S13.1 | Users shall be able to assign themselves to tasks.  | Users can click on a project task and assign themselves as the team member who will work on it.  | FR   |
| S14.1 | Users shall be able to update the status of a task.   | Users can update the status of their assigned tasks to "done".   | FR   |
| S15.1 | Users shall be notified one week before a task or project is due.                           | Users need to be notified that a project they are a team member of or a task they are assigned to is close to being due.   | FR   |
| S16.1 | Project members shall be able to view the progress of the project.                          | Through the project's page, team members must be able to see how many tasks have been completed and how many still need to be done.  | FR   |
| S16.2 | Project members shall be able to view progress as a percentage bar.                         | Team members should be able to see a simple percentage bar which represents how close they are to completing the project.  | NFR  |
| S17.1 | Project members shall be able to message each other in a group message board.               | Team members can access a group message board where they can communicate with each other.  | FR   |
| S17.2 | Project members shall be notified with an icon when the message board has new messages.     | The button to access the message board will have a blue icon above it when there are unread messages.  | NFR  |
| S18.1 | Users shall be able to leave a project.   | Users can choose to press a button to leave a project and will no longer be able to view the project or receive notifications for it.  | FR   |
| S19.1 | Users shall be able to change their user settings.  | Users can access a settings page and select their preferred font-size and colour theme, with options available for colour blind users.   | NFR  |
| S20.1 | Users shall be able to access a user guide that helps them navigate the system.             | In the footer of every page, the user can be directed to a user guide page with instructions for using the system.   | FR   |
| S21.1 | Users shall be able to get in contact with support staff if they need help.                 | In the footer of every page, the user can enter and send their email address, if they require support.   | FR   |
| S22.1 | Users shall be able to share files with their team members.                                 | Within a project, there must be a files section where team members share files with the project's team members.  | FR   |
| S23.1 | Users shall be able to download files from their project.                                   | Within a project's file section, users must be able to download a file that was uploaded by a team member.   | FR   |

## D. Test Cases

Table 4: Test cases

| Test ID | Feature ID and Description  | Input Operations  | Expected Output  |
|---------|---|---|--|
| T1.1    | S1.1. A new user completes the sign-up form with valid information.                   | 1. Click the sign-up button<br>2. Enter valid username, email and password<br>3. Click "Continue"   | Account is created, user is navigated to the home page with a welcome message in the top right of the screen                             |
| T1.2    | S1.2 A new user completes the sign-up form with an email that already has an account. | 1. Click the sign-up button<br>2. Enter username, password and an email that has already been used<br>3. Click "Continue"   | Account is not created, the user is prompted that the email is already in use and that they'll need to use a different email to continue |
| T1.3    | S1.3 A new user completes the sign-up form with an invalid password.                  | 1. Click the sign-up button<br>2. Enter username, email and invalid password  | User is prompted that the password is not valid and informed as to what the requirements are.  |
| T2.1    | S2.1. A user completes the login form with valid information.                         | 1. Click the login button<br>2. Enter username/email and password<br>3. Click "Continue"  | User is navigated to the home page with a welcome message in the top right of the screen.  |
| T2.2    | S2.2. A user completes the login form with invalid information.                       | 1. Click the login button<br>2. Enter invalid username/email and password<br>3. Click "Continue"  | User is not logged in and is prompted that their login information is incorrect.   |
| T3.1    | S3.1. A user clicks the logout button when they've finished with the system.          | 1. Click the logout button  | User is logged out and redirected to the login page.   |
| T4.1    | S4.1. A user creates a project with information.                                      | 1. Navigate to the projects page<br>2. Click the "Create new project" button<br>3. Complete the project information form<br>4. Click the "Confirm" button   | A project is created and the user is assigned as its team leader.  |
| T4.2    | S4.2. A user creates a project with invalid information.                              | 1. Navigate to the projects page<br>2. Click the "Create new project" button<br>3. Complete the project information form with invalid date<br>4. Click the "Confirm" button                       | The project is not created and the user is prompted with what information is invalid.  |
| T5.1    | S6.1. A team leader updates their project's information.                              | 1. Navigate to the projects page<br>2. Click the "View Project" button<br>3. Click the "Update Project" button<br>4. Complete the project information form<br>5. Click the "Confirm" button       | The project is updated with the new information.   |
| T6.1    | S7.1. A team leader deletes their project.  | 1. Navigate to the projects page<br>2. Click the "View Project" button<br>3. Click the "Delete Project" button<br>4. Click the "Confirm" button   | The project is deleted from the system.  |
| T7.1    | S8.1. A team leader invites a user to the project.                                    | 1. Navigate to the projects page<br>2. Click the "View Project" button<br>3. Click the "Invite to Project" button<br>4. Enter the user's email into the form<br>5. Click the "Send Invite" button | The user is invited to the project.  |
| T8.1    | S9.1. A team leader removes a user from the project.                                  | 1. Navigate to the projects page<br>2. Click the "View Project" button<br>3. Click the "Team Members" button<br>4. Click the "Remove" button next to the name of the user you want to remove      | The user is removed from the project, unassigned from their tasks and can no longer access the project.                                  |
| T9.1    | S10.1. A user views a project invitation.   | 1. Navigate to the projects page<br>2. Click the "View Invitations" button  | The user can see the invitation along with the project information.  |
| T9.2    | S10.2. A user views a project invite notification.                                    | 1. User sees a blue dot above the notification icon in the nav bar<br>2. Click the notification icon  | The user can read the project invite notification.   |

## [Private]

---

| Test ID | Feature ID and Description  | Input Operations   | Expected Output  |
|---------|---|--|--|
| T10.1   | S11.1. A user accepts or declines a project invitation.                           | 1. Navigate to the projects page<br>2. Click the "View Invitations" button<br>3. Click the "Accept" or "Decline" button  | If the user accepts, they are now a team-member on the project, if they decline, the invitation disappears.                                    |
| T11.1   | S12.1. A user views a project's information.                                      | 1. Navigate to the projects page<br>2. Click a project's "View Project" button   | The user can see all of the project information.   |
| T12.1   | S13.1. A user assigns themselves to a project task.                               | 1. Navigate to the projects page<br>2. Click a project's "View Project" button<br>3. Click on an uncompleted project task that is unassigned<br>4. Click the "Assign To Me" button                   | The task will be updated with the user assigned to it.   |
| T13.1   | S14.1. A user marks a task they are assigned to as done.                          | 1. Navigate to the projects page<br>2. Click a project's "View Project" button<br>3. Click on a task they are assigned to<br>4. Click on the "Mark As Done" button                                   | The task will be updated with its status changed to "Done".  |
| T14.1   | S15.1. A user is notified that a task or project is due in a week.                | 1. User sees a blue dot above the notification icon in the nav bar<br>2. Click the notification icon   | The user can see a notification informing them that the task or project is due in one week.  |
| T15.1   | S16.1. A user views the progress of a project.                                    | 1. Navigate to the projects page<br>2. Click a project's "View Project" button   | The user can see the number of tasks completed compared to the total number of tasks   |
| T15.2   | S16.2. A user views a percentage bar of the progress of a project.                | 1. Navigate to the projects page<br>2. Click a project's "View Project" button   | The user can see a percentage bar representing the project's progress  |
| T16.1   | S17.1. A user interacts with a project's group message board.                     | 1. Navigate to the projects page<br>2. Click a project's "View Project" button<br>3. Click on the "Message Board" button<br>4. Type a message in the text box<br>5. Click "Send Message"             | The message is sent to the group and the user can see all the group messages   |
| T16.2   | S17.2. A user is notified when there is a new message in the group message board. | 1. Navigate to the projects page<br>2. Click a project's "View Project" button   | The user can see an icon above the "Message Board" button informing them that there are new messages   |
| T17.1   | S18.1. A user leaves a project.   | 1. Navigate to the projects page<br>2. Click a project's "View Project" button<br>3. Click the "Leave Project" button  | The user is removed from the project team member list, is unassigned from any active tasks and no longer receives notification for the project |
| T18.1   | S19.1. A user updates their settings.   | 1. Navigate to the settings page<br>2. Click on the "Large Font" button<br>3. Select a new colour theme<br>4. Click the "Save Changes" button  | The font size is increased and the colour scheme has changed   |
| T19.1   | S20.1. A user access the user guide.  | 1. Click the "User Guide" button in the footer   | The user is navigated to the user guide  |
| T20.1   | S21.1. A user contacts staff for support.   | 1. Enter an email address in the form in the footer<br>2. Click the "Contact Us" button  | The user's email is stored in the back-end so that staff can get in touch with them  |
| T21.1   | S22.1. A user uploads a file to their project's files section.                    | 1. Navigate to the projects page<br>2. Click a project's "View Project" button<br>3. Click the "Project Files" button<br>4. Click the "Upload File" button<br>5. Select the file they wish to upload | The user's file is uploaded and stored in the project's file section   |
| T22.1   | S23.1. A user downloads a file from their project's files section.                | 1. Navigate to the projects page<br>2. Click a project's "View Project" button<br>3. Click the "Project Files" button<br>4. Click on the file they wish to download<br>5. Click the "Confirm" button | The file is downloaded and stored on the user's machine  |

## E. Use Case Diagrams

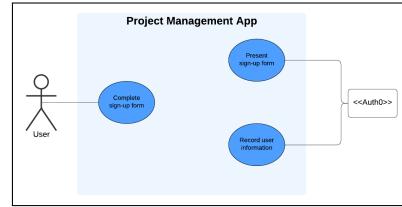


Figure 1: Use case diagram of a user signing-up to the system

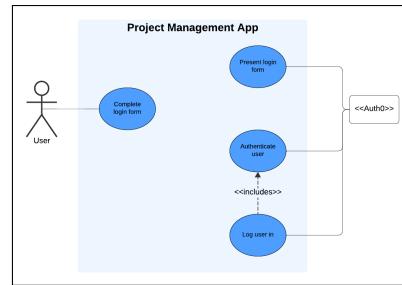


Figure 2: Use case diagram of a user logging into the system

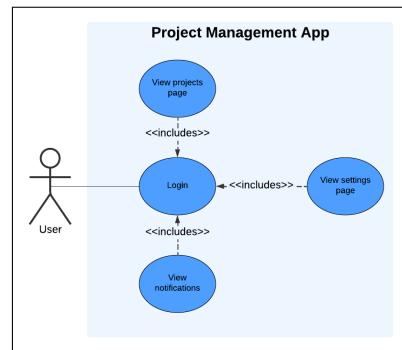


Figure 3: Use case diagram of a user navigating the app

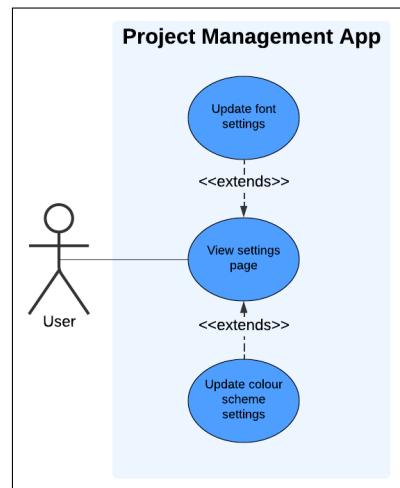


Figure 4: Use case diagram of a user updating their settings

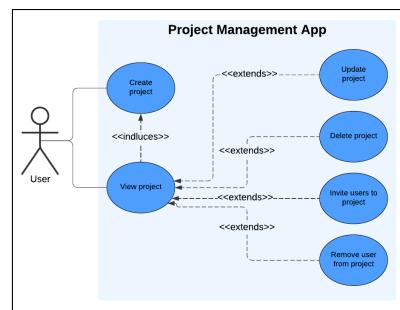


Figure 5: Use case diagram of a user creating and managing a project

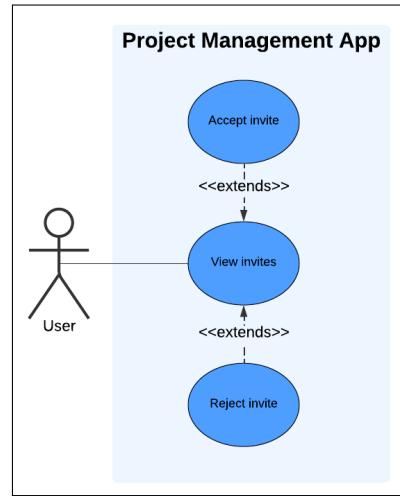


Figure 6: Use case diagram of a user responding to project invites

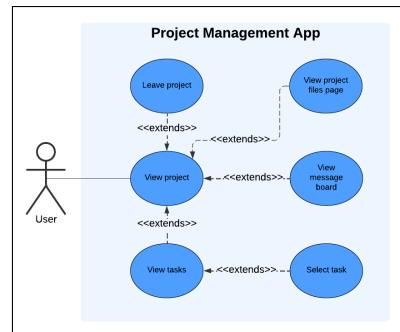


Figure 7: Use case diagram of a team member's view of a project

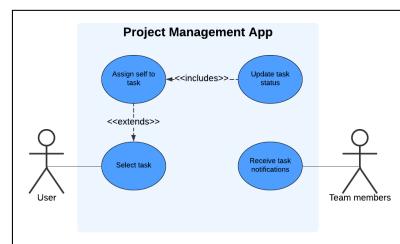


Figure 8: Use case diagram of a team member's view of project tasks

[Private]

---

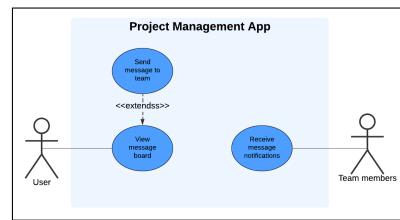


Figure 9: Use case diagram of a team member's view of a project's group message board

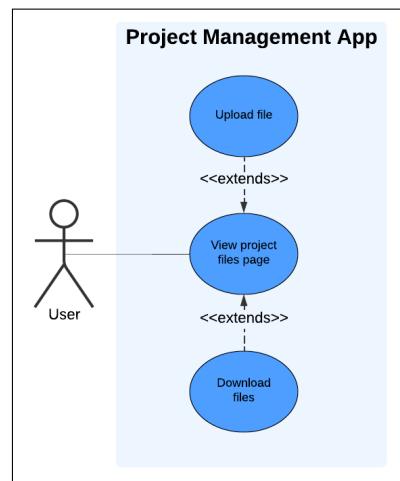


Figure 10: Use case diagram of a team member's view of a project's files page

## F. UML Class Diagram

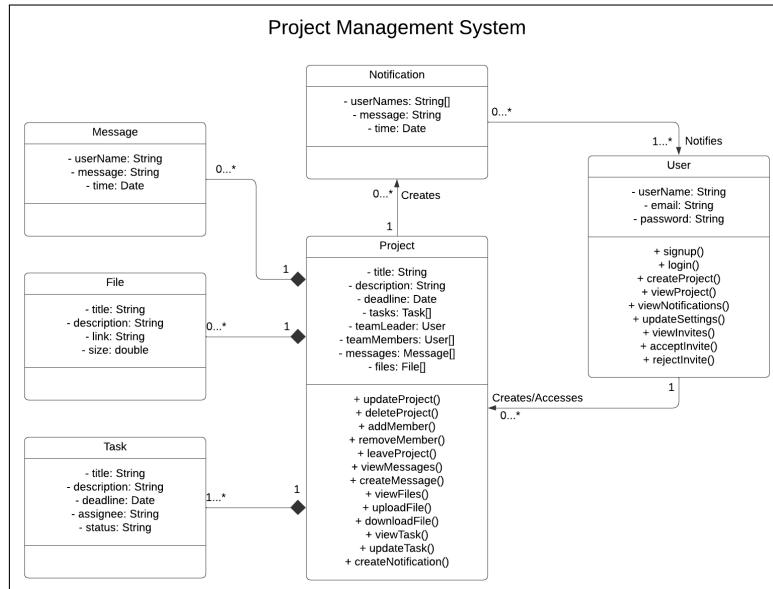


Figure 11: UML Class diagram of the project management system

## G. Front-end development progress



Figure 12: Screenshot of the landing page before a user has signed in

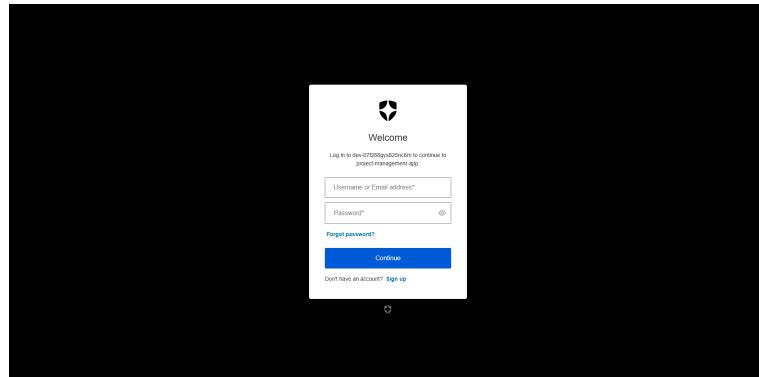


Figure 13: Screenshot of the Auth0 login user interface

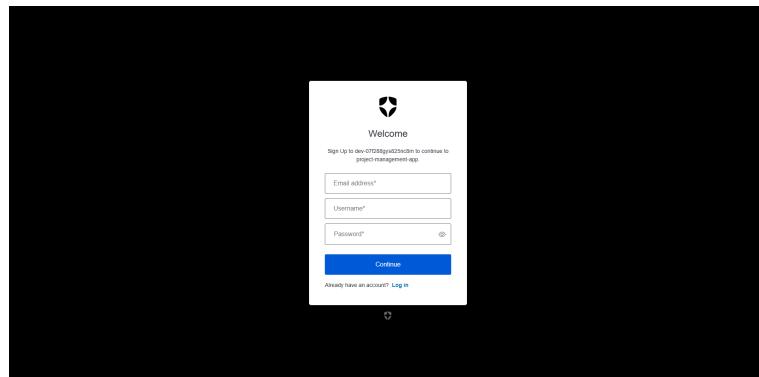


Figure 14: Screenshot of the Auth0 sign-up user interface

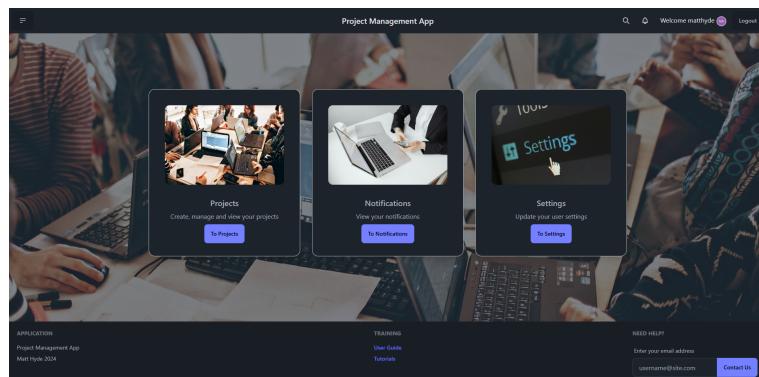


Figure 15: Screenshot of the landing page after a user has signed in

---

[Private]

---

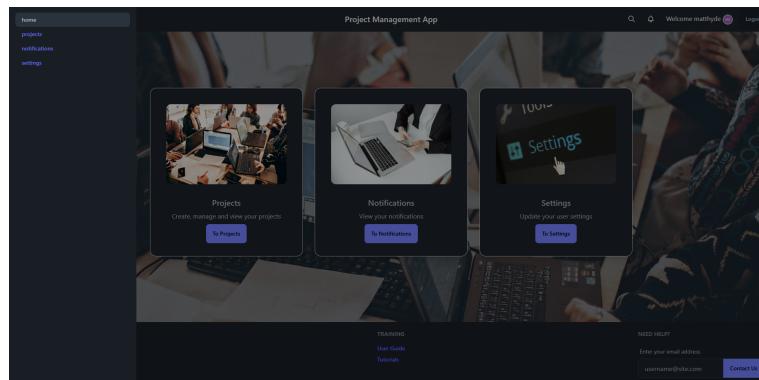


Figure 16: Screenshot of the landing page after the nav menu button has been clicked

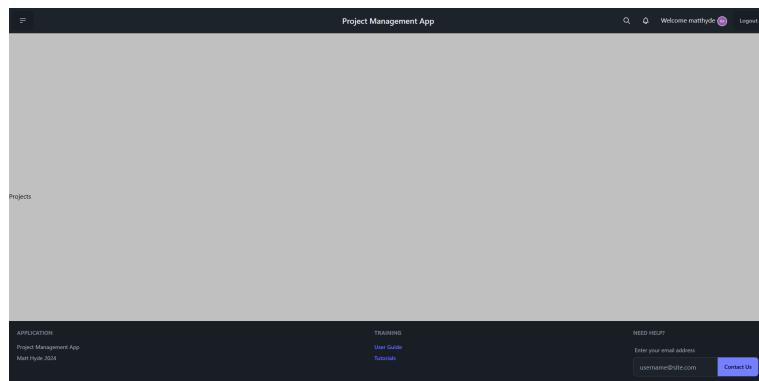


Figure 17: Screenshot of the projects page

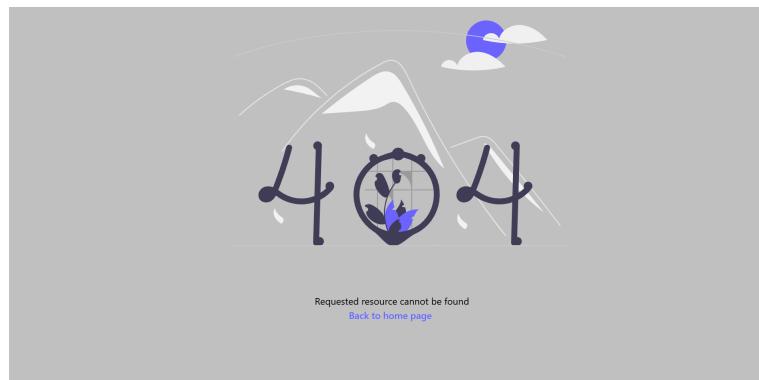


Figure 18: Screenshot of the error page after a HTTP 404 error has occurred

## H. Risk Analysis

Table 5: Risk analysis

| Risk                                | Impact               | Impact level | Probability level | Mitigation  | Control  |
|-------------------------------------|----------------------|--------------|-------------------|---|--|
| Struggle to learn Django            | Project delays       | High         | High              | Switch to Express.js                                  | Research online materials and tutorials before development     |
| Local code files corrupted or lost  | Project delays       | High         | Low               | Use backups to restore system                         | Store online backups of the system in a GitHub repository      |
| Survey doesn't get enough responses | Missing requirements | Medium       | High              | Research similar surveys for more secondary data      | Share the survey frequently across multiple social media sites |
| Scope creep                         | Missing deadlines    | High         | Medium            | Avoid features that are not must-haves                | Create and manage design and implementation plans              |
| Low quality code                    | Poor system          | High         | Medium            | Dedicate more time towards improving and testing code | Implement tests to check the code is working correctly         |
| Fail to implement features          | Project failure      | High         | Low               | Research techniques for implementing the key features | Create and follow UML models of the system                     |
| Fail to meet deadlines              | Project failure      | High         | Low               | Focus on meeting key requirements                     | Create and follow a Gantt chart of the project timeline        |

## I. Gantt Chart

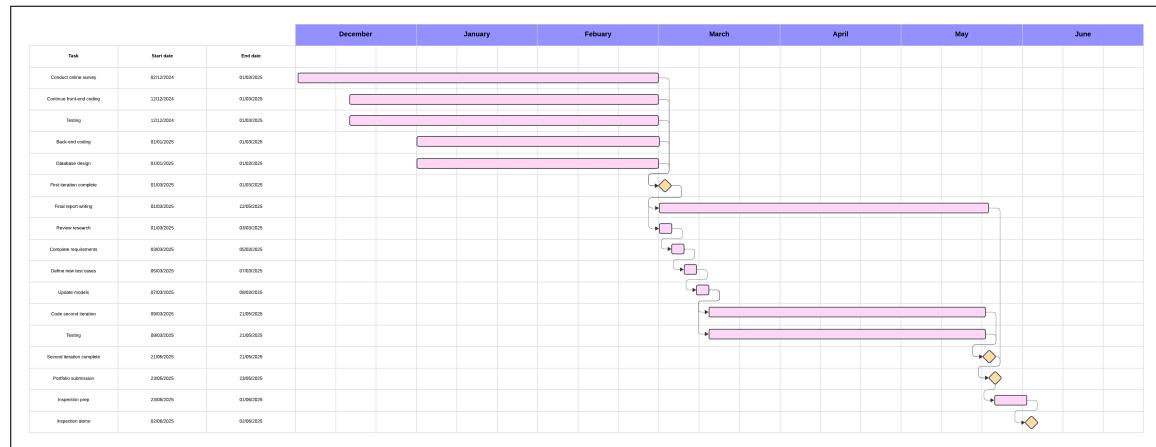


Figure 19: Updated Gantt chart for the project plan