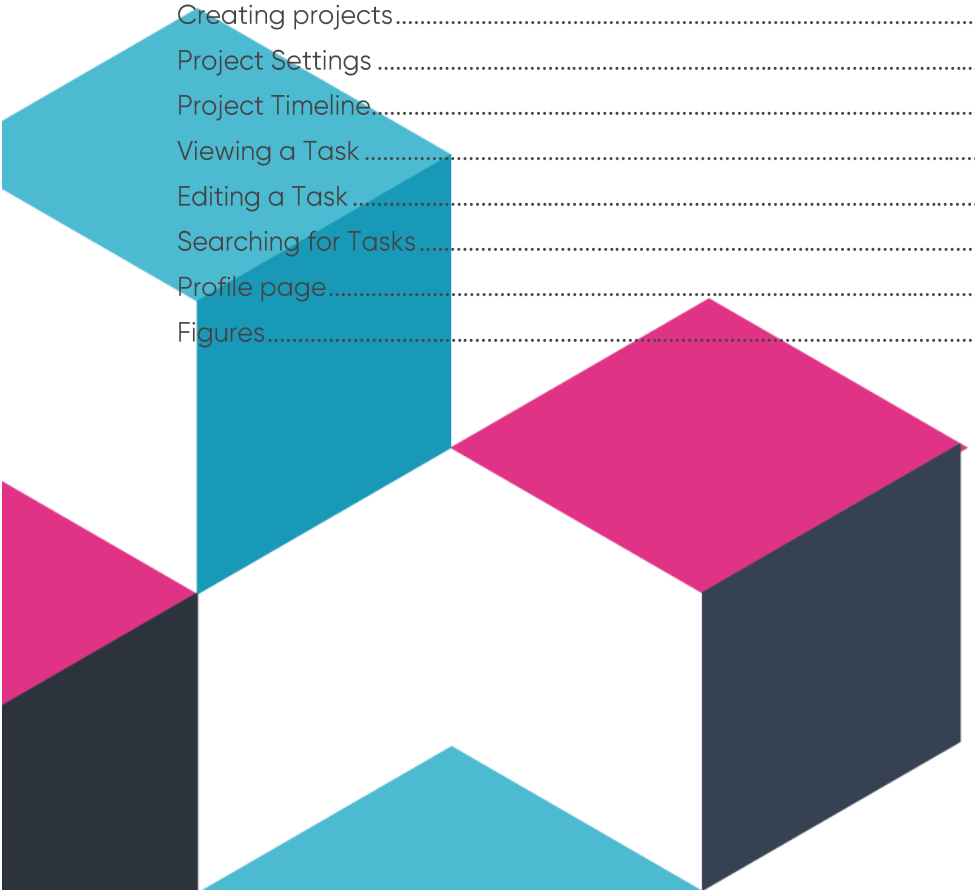
Software Analysis & Design

By Ben Belcher

Word count: 3236

Contents



The System

This system is a work management system to help TT Corp manage the various projects and tasks within those projects. The system allows new users to register and then subsequently login to access their assigned projects, additionally users have the option to remain authenticated when the app is closed and upon re-opening the app they only need to provide their quick login pin.

Once authenticated users are presented with a dialog with all their projects and some information regarding that project. If the user is a manager then they also have the option to create a project from this page. After selecting a project to open the user is presented with a full screen window showing the project dashboard, this dashboard was inspired by other work management systems such as Jira or Trello. The dashboard comprises of multiple columns which organize tasks and visually show their progress within the project.

Some other features of the application include:

- A profile page, this page displays the authenticated user's details and all the tasks they currently have assigned to them.
- Automatic email notifications for:
    - Assigning users tasks
    - Assigning users to a project
    - Adding extra users to a project
    - Project completion
- Functionality to search both current project and all projects to find a task by name.
- Each project has its own timeline page, displaying messages posted by project members and a list of project members. There is also a project progress bar on this page and if the user is a manager then they can mark the project as complete once every task is complete.
- Managers can edit the project information, such as changing the project name and assigning/ removing users from the project.

Cohesion & Coupling

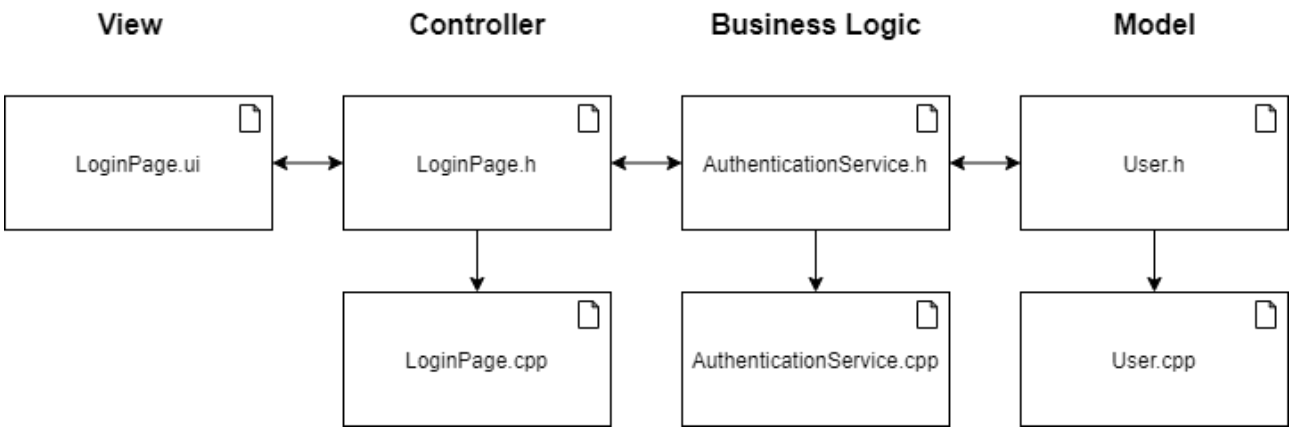
A good software system should have high cohesion and low coupling, this is to ensure minimal dependencies between components and promote reusable code. In my design and implementation I am aiming for a very high level of cohesion, to achieve this I am going to only group methods that are similar and serve the same function. For example I'm planning on having a Project model to create and return values of that object, and then a separate class to handle all the data operations associated with that model.

Then to ensure that I will have loose coupling between modules/ components I be using dependency injection rather than having the dependency tightly coded into the class. Dependency injection works by creating the object (dependent) in separate class to where it is used. Once the object is constructed I can inject it into the data service class.

Application Architecture

To aid in my goal of achieving high cohesion and low coupling I need to structure the application in a specific pattern. Doing this not only achieves high cohesion and low coupling, it also improves code readability and code integrity.

The architecture I am going to use is an MVC based pattern, as this pattern is simple to implement using QT. MVC architectures also improves the software quality as it will work more efficiently resulting in a more responsive experience and consume less of the users resources. The diagram below illustrates my desired implementation of this pattern.



QT by default provides a .ui file which will only hold the declaration of ui elements, then working with that ui file is a class (header and cpp file). Those files work to control what happens to those ui elements. For example button clicks or other events fired when the user interacts with the ui. So to complete my implementation I simply need to create the business logic and models.

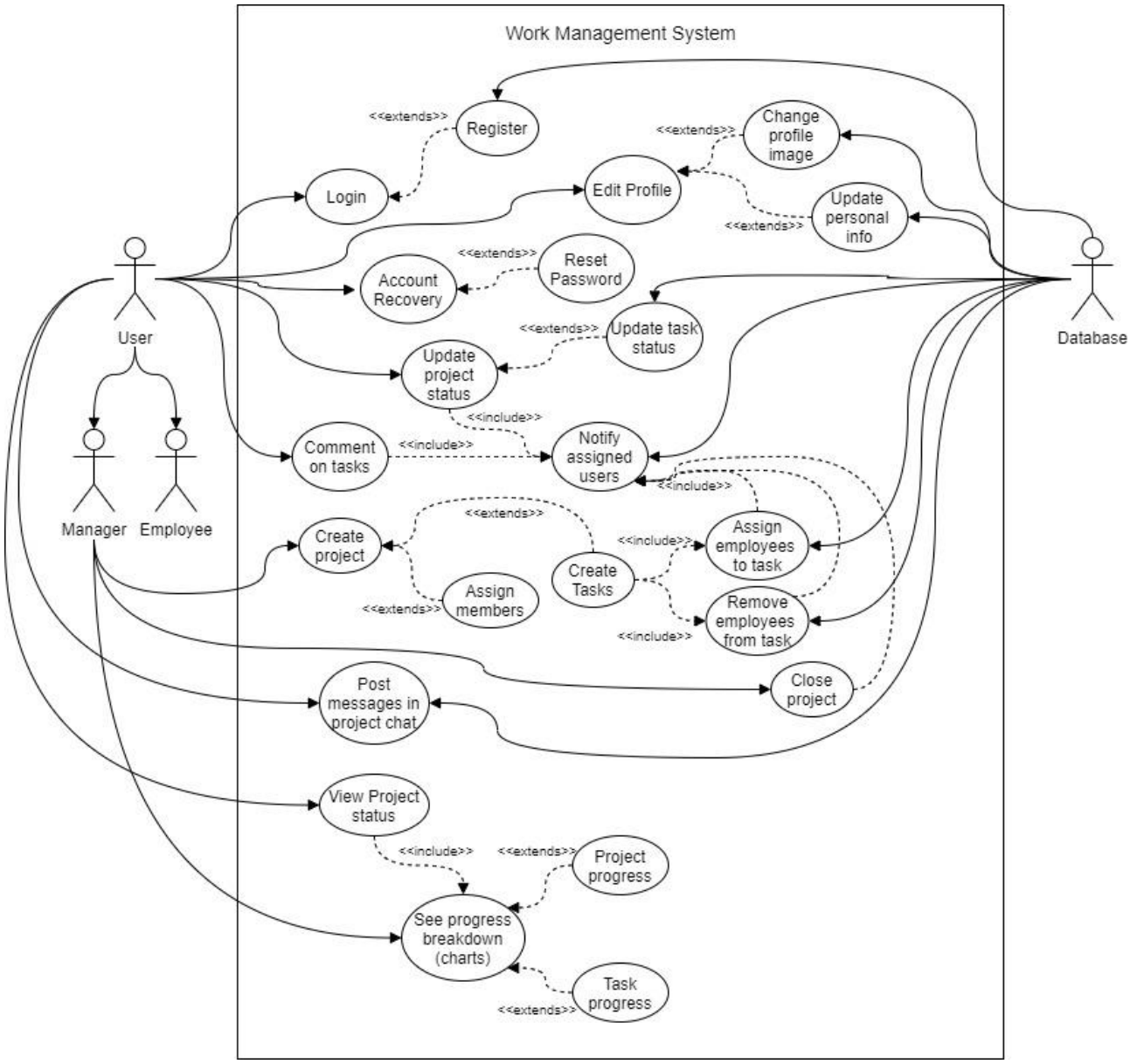
Which does make this is a slight adaptation of the traditional MVC architecture but I feel for my use case it works to abstract the different functionalities between classes that work together to achieve similar functionality.

Another aspect of the application's architecture is security, as the project and task data could be sensitive I am planning to implement robust authentication and authorization to ensure that only authorized users can see the data.

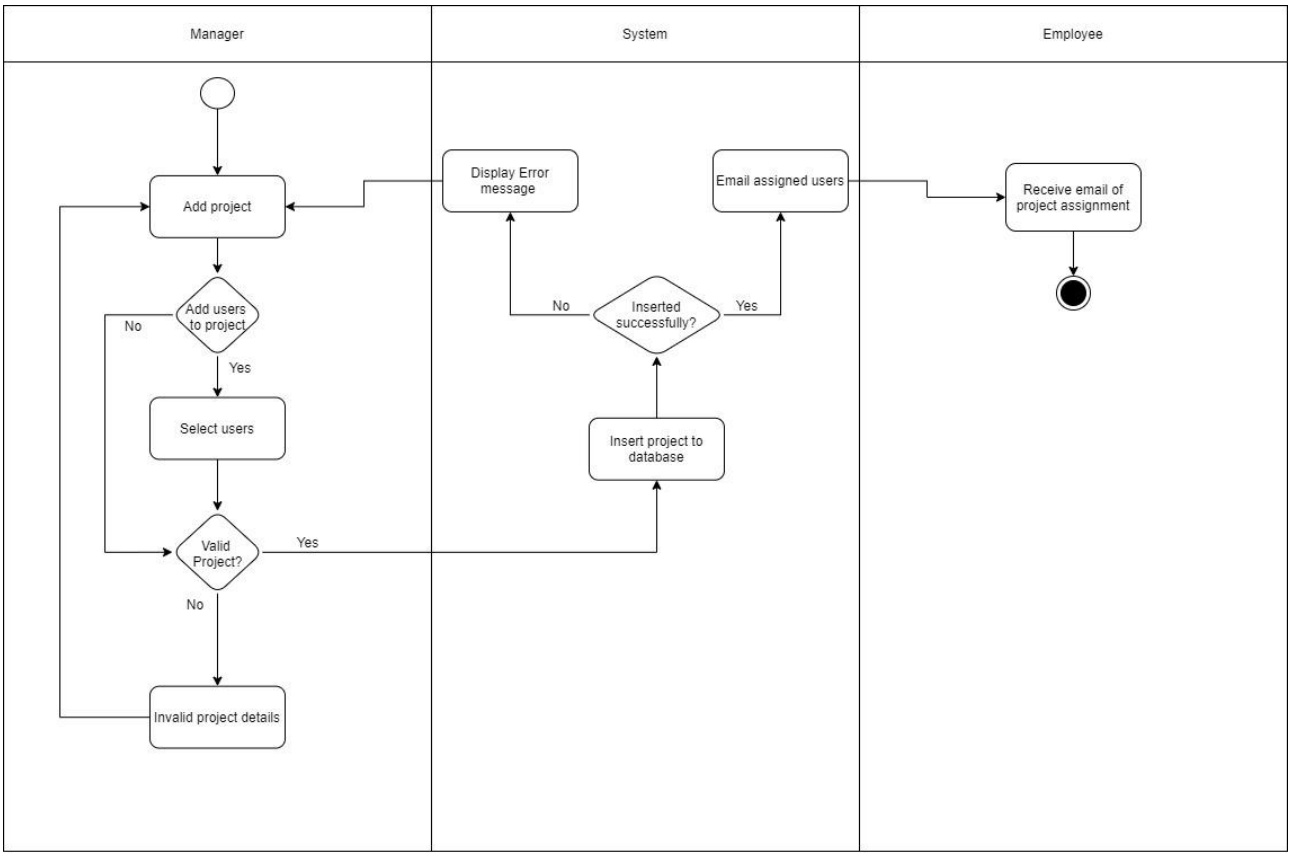
Furthermore I plan on encrypting the local database and provide only the application with the key, this ensures that if someone gains access to the database file, they cannot open it and view the data. Both of these combined with in app validation will provide my application with a good level of security.

Use Case Diagram

This is the proposed use case diagram for the application, there is only two users of the system and they mostly have the case uses. The manager has some extra use cases such as detailed breakdowns of each projects status and the ability to close a project once its complete. Furthermore as this application is a standalone desktop app there is only one external actor which is a database. There is no need for other web services.

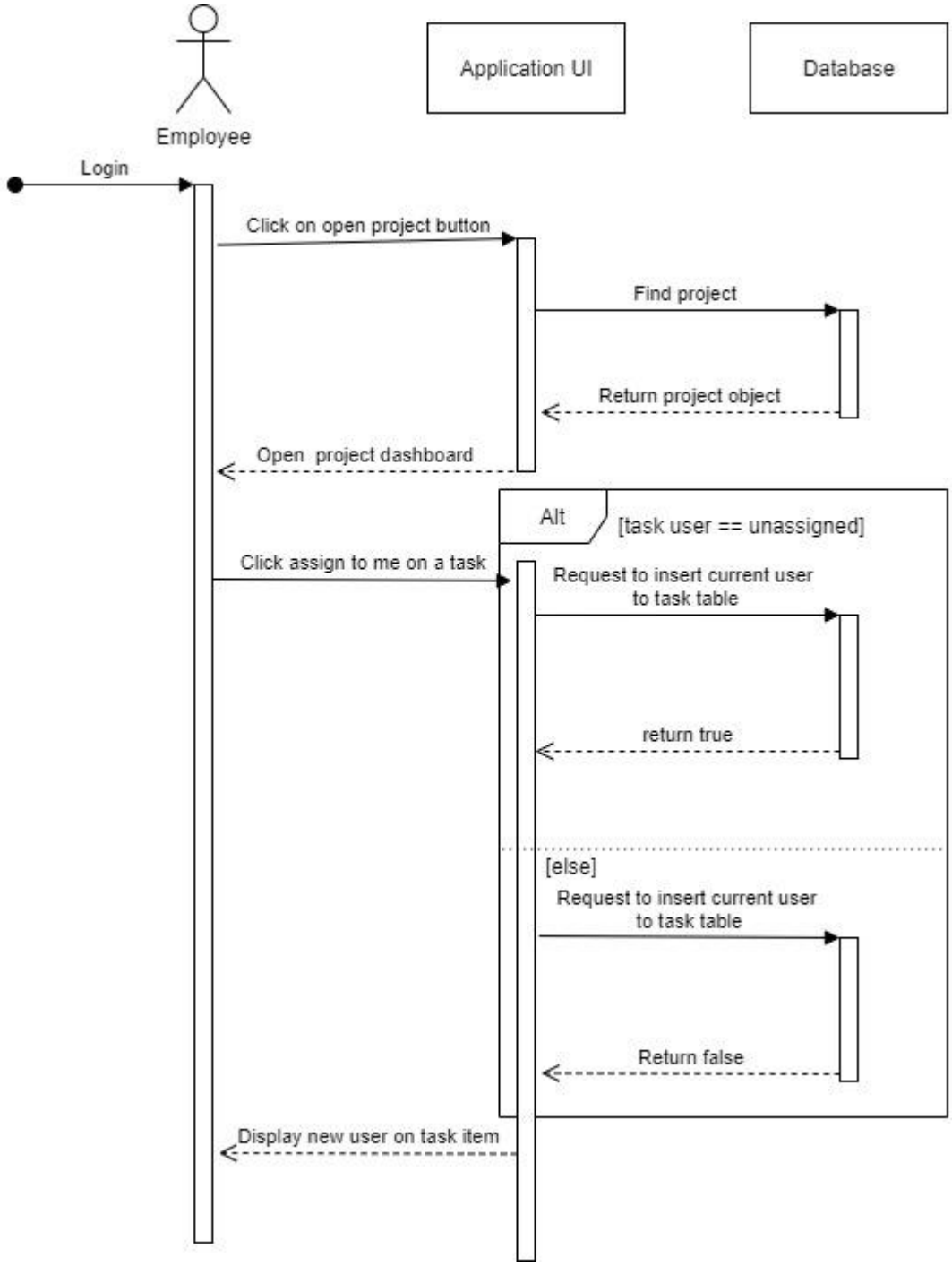

Work Management System
Register
Login
Edit Profile
Change profile image
Update personal info
Reset Password
Account Recovery
Update task status
Update project status
Comment on tasks
Notify assigned users
Create project
Assign members
Create Tasks
Assign employees to task
Remove employees from task
Post messages in project chat
Close project
View Project status
Project progress
See progress breakdown (charts)
Task progress
<<extends>>
<<include>>
User
Manager
Employee
Database

Activity Diagram



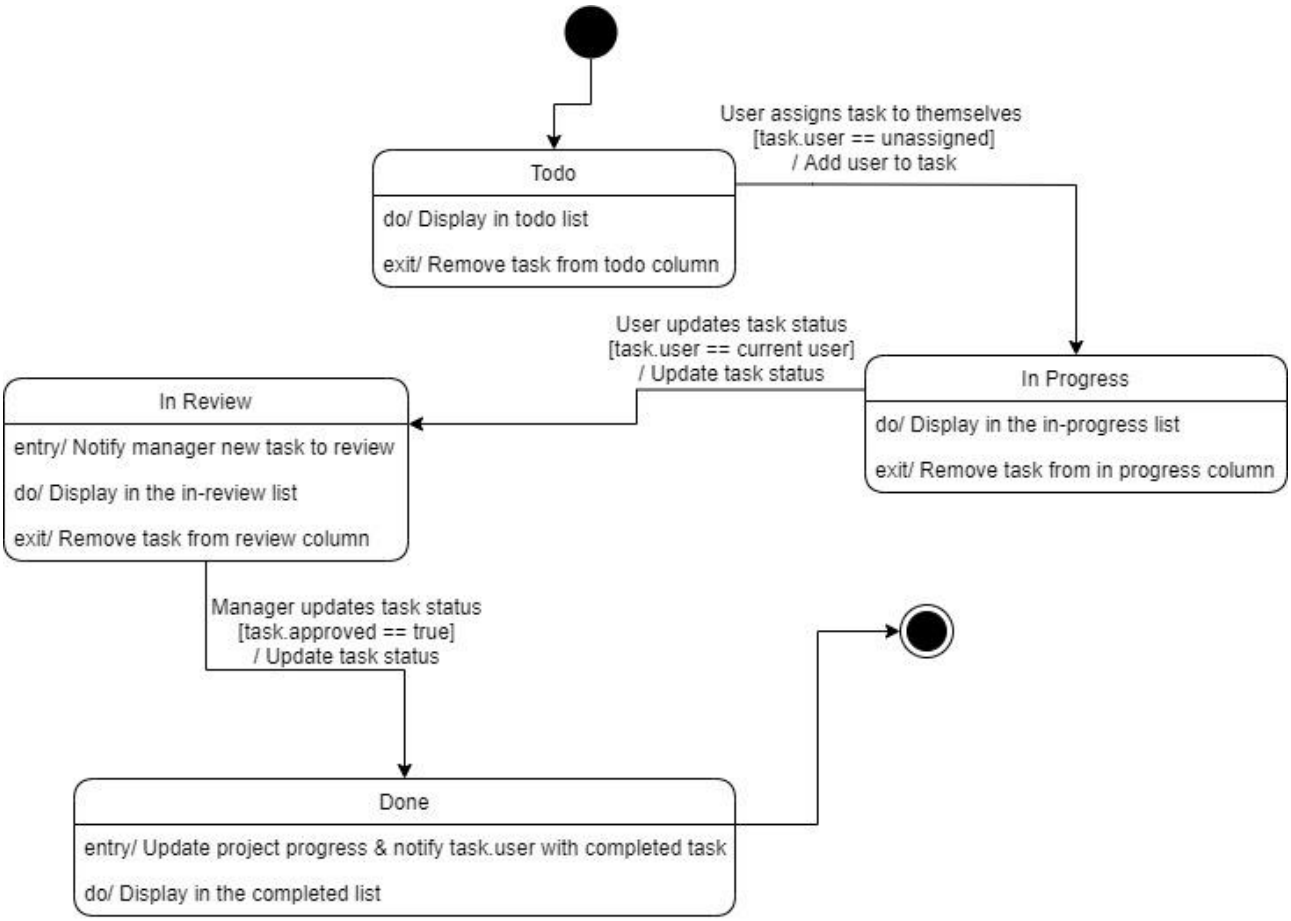
This activity diagram depicts the process of a manager creating a new project and assigning users to it. There are three swim lanes in the diagram to show how actions preformed by each entity relate and coordinate to create a complete process. These entities are **The Manager**, **The Application**, and **The End User**. The flow starts with the manager adding a project using the new project form, if the manager decides to add users to the project upon creation then they can select a set of users from a list. From there the project is checked for its validity. If the project form has errors then an error message will display and put the manager back to a blank form. If the project is valid then the system will insert the project into the database. Once inserted the system will then send email notifications to all the project members, letting them know that they have been added to a project. The last step of this diagram is the employee reading that email.

Sequence Diagram



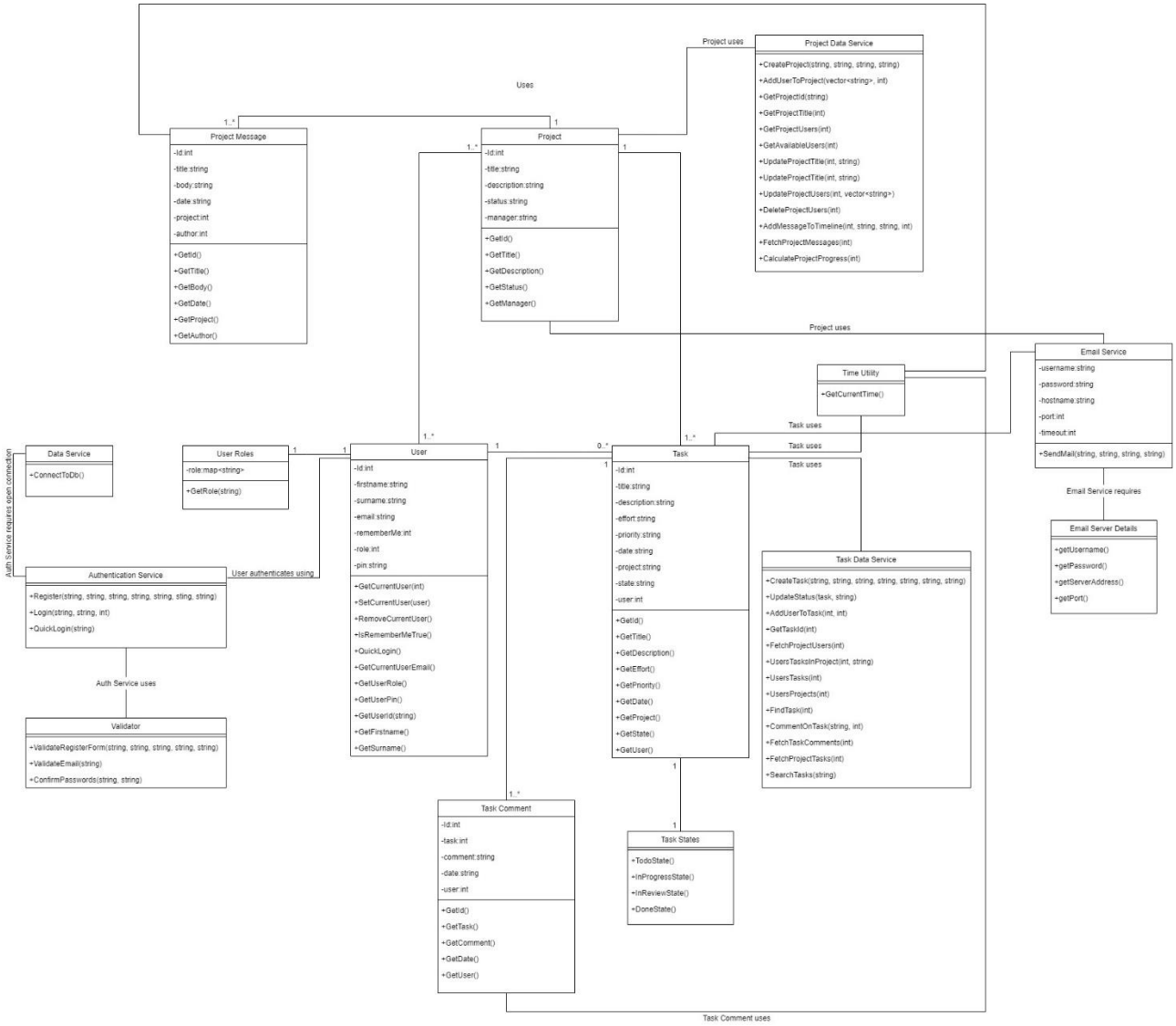
This sequence diagram is showing the sequence and events for the user assigning themselves a new task within a project. The diagram starts with the user opening their desired project and if the project can be opened a dashboard appears for the project, the user then clicks on the assign to me button on a task. A condition starts. If the task already has a user assigned to it then an error will appear informing the user someone else is already assigned, else the user will be assigned to the task.

State Diagram

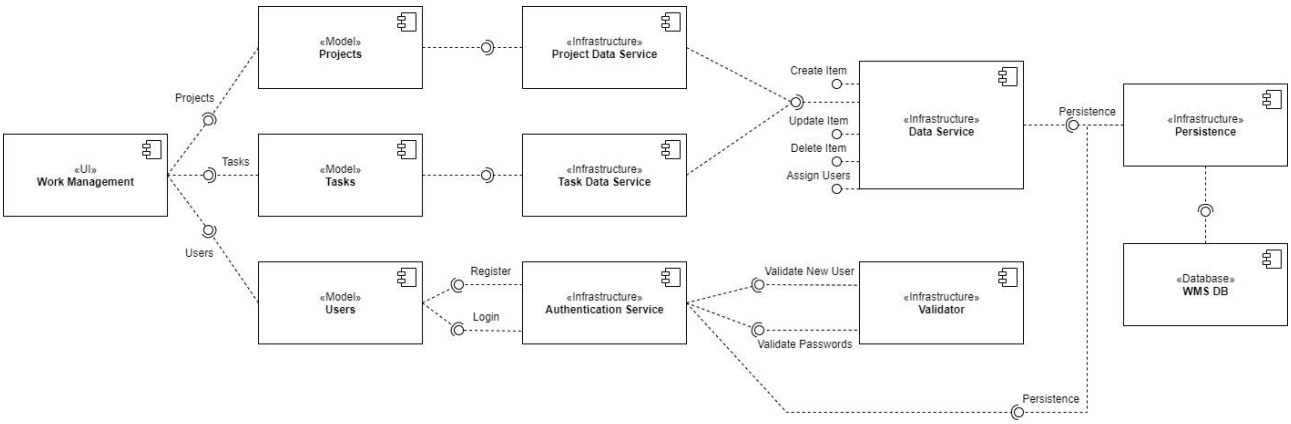


The state diagram above is displaying the different states that a task will go through and also some of the functions that need to happen during the transition of states. The initial state of a task will be todo and once a task has been created it will automatically be placed in todo. The task can only leave the todo state when it has been updated in the database by a user, when the task does leave the todo state a check is preformed to ensure that the task is assigned to the current user. Then the widget will move to the in progress list. Then when the task leaves the in progress state a check is preformed to ensure that the requesting user is the same as the user assigned to the task. When the task enters the review state an email notification will be sent to the project manager, letting them know that there is a task to review. If the manager approves this task then it can be moved to the done state. Upon entry for the done state an email notification will be sent out to the original assignee of the task, letting them know the task is marked as done.

Class Diagram



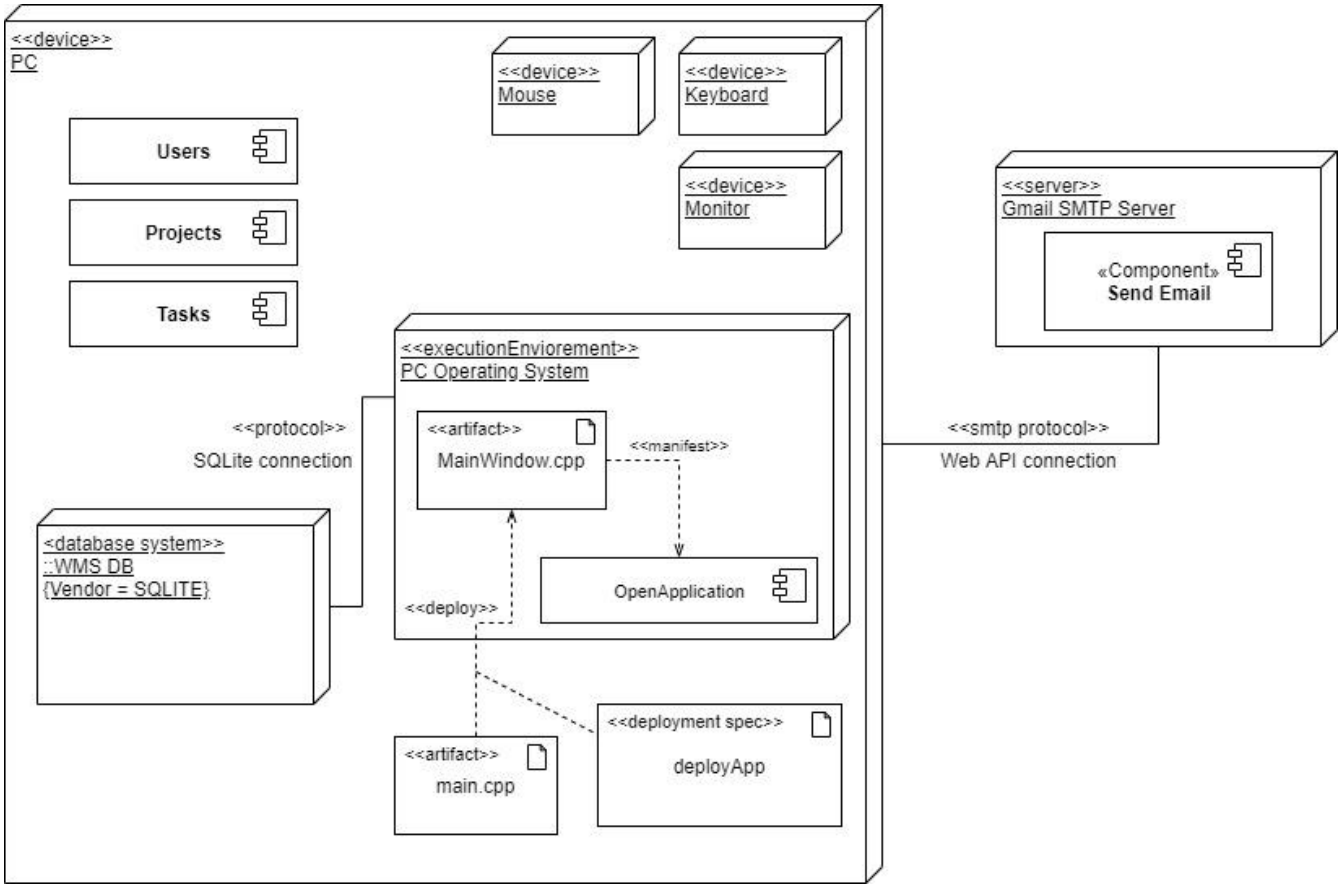
Component Diagram



The Component diagram shows how all of the different components will integrate with each other and which components provide what methods. The diagram starts from the UI components and the drills down

to models which interface with their corresponding data services, these data services provide most of the business logic and data functionality regarding the models. Finally the last layer of components is the persistent storage which is the application database.

Deployment Diagram



My deployment diagram only consists of two nodes which is the users pc and the Gmail smtp server, the entire application is localized to their pc and the only external service is the smtp server to send automated emails to users. The database engine is going to be an SQLite database which is file based, thus remaining on the users device. Furthermore, the execution environment holds the source files (only main window is depicted here but there will be more source files). Then to execute these files the deployment specification will invoke main.cpp which is the entry point to the application.

Data Storage

For this application I am planning on storing all permanent data in a local database using SQLite, my reasoning behind this decision is that unlike file storage a database is structured and allows for querying. This will allow me to easily retrieve data without using complex algorithms or regular expressions, ultimately making the application lighter on resources and improving the data integrity and scalability. Its very difficult to scale file storage compared to a database, no matter what searching approach you use with files the entire file would have to be opened and read to retrieve data. When a database only has to be opened once and then only the specified table/s has to be queried.

Email Servicing

My application supports automatic emailing to project members and to implement this I wrote my own service which was inspired by the third party library SmtpClient. The reason I decided to make my own implementation was because I didn't require all the functionality that library offered and for this use case of sending simple text emails it works well. The service interfaces with googles SMTP server under the account of WMSsystem@gmail.com.

User Guide

This section of the report outlines how to use the application with screenshots for visual reference.

Creating an account

To create an account, simply open up the app and click on the *Register* button (figure 1). Once clicked a form will appear (Figure 2), then fill in all the fields correctly to register. If you make a mistake, like entering not matching passwords, an error will appear that is relevant to the error you made (Figure 3). If registration is successful then a success message will appear, (figure 4).

To continue click on the login button, this will open a new dialog where you can enter your email and password to authenticate (Figure 5). If you decide to remain authenticated across sessions then check the remember me box. If you enter the wrong details on this form then you will receive an error message (Figure 6) and once logged in you will receive another success message (Figure 7).

Opening a project

Now you have authenticated you can select a project that you have been assigned to. Figure 8 shows what this screen would look like from a manager account, the only difference between a manager and user for this page is users don't have the option to create a new project. This page also shows the state of a project, figure 8 shows an example of what each state would look like.

Once you open your desired project you will be redirected to a full screen window (figure 9) and this is the main dashboard for the project. The centre of attention are the task columns, these display lists of tasks within the project. Each task card has the title, description and the assigned user, if a task is unassigned then it will say *unassigned*.

Assigning tasks

If you are a user you can assign any task in the todo column to yourself. To do this right click on a task card and a menu will appear (figure 10) then click the *assign to me* link. Other ways to assign tasks is when the task is created, only managers can create new tasks. To create a new task managers will have a button in the bottom left of the page called *Create Task*. Click on that button the a form will popup (Figure 11).

To assign a user to the task, select a user from the drop down (figure 12), when you create the task the user assigned will receive an email notification that a task has been assigned to them. To create the task ensure all fields are complete and then click the *create task* button. If successful you will see a success message and the new task appear in the todo list.

Creating projects

Another manager only feature is creating projects. A project can be created either from this dashboard or from the project selection screen, in either scenario once you click on the respective *create project* button a form will appear (figure 13). To create the project complete all the fields and assign the relevant users. To assign a user, navigate to the available users list and right click on an email to see a link to assign that user (Figure 14). Once assigned the user will move to the list on the left and this indicates they are going to be in that project. If the list of available users is very long then you can use the search bar to filter that list (Figure 15).

Once all fields are complete, click the create project button. If the project was successfully made a dialog will appear (Figure 16) and once you close this dialog the create project dialog is also closed and your back to the original page you where on when clicking the create project button.

Project Settings

Only project managers can see the project settings page (Figure 17), to open this dialog a manager needs to click on the *project settings* button from the project dashboard. This page allows the project manager to add or remove users from the project and edit the project name. To edit the project name just change the value in the *Project Name* textbox and click the save button. If you want to update the project users then right click on an email address in either column and a menu will appear to either add or remove that user. Figure 18 shows both of these menus.

Project Timeline

Any project member can view the project timeline (Figure 19), to get to the project timeline click on the *project timeline* button from the main dashboard. Once on the page project members can see a list of other project members and a chatroom specific to that project. If you are a project manager you also have the option to mark the project as complete from this page if all the tasks within that project are complete.

To post in the project chat simply fill in the input fields with a message title and content, then click the send button. All project messages are private to every project and are saved in a persistent database.

Viewing a Task

To get more details about a specific task within a project you can double click on the task card in any of the lists on the main dashboard (Figure 9). This will open a new dialog with the details about that task (Figure 20). If you are the project manager or the current assigned user to that task you have the option to edit the task details, this is indicated with the red *edit* button seen on figure 20. If you want to add a comment to the task then you can do so by adding your comment to the *Write a comment* textbox and click the comment button. Any project member can add comments to a task.

Editing a Task

Upon viewing a task, if you have the permission to edit you'll see a red button *Edit Task* which can be seen in figure 20. When you click this button the ui on the dialog will change (Figure 21), to allow for user input. A dropdown for Priority and effort will appear and so will a text edit for the task description. Once you are happy with the changes you can click the *Save* button to commit these changes.

Once saved the dialog ui will change again to viewing mode and you will see the new changes here and on the main dashboard which remains open behind this dialog.

Searching for Tasks

If you need to find a specific task then you can search for it by name. There are two ways to find tasks, the first is to only find tasks within the project that you currently have open. To search the current project simple start typing the task title in the searchbar on the left (Figure 22) and see the tasks on the dashboard change to match your search. The search happens automatically every time you press a key and to reset the board to see all tasks just remove all text from the search bar.

The other way to find tasks is to use the search bar at the top of the page called *Find a task in any project*. If you enter either a full task title or part title and click the search button a new dialog will open showing you the search results (Figure 23). From this dialog you can see the basic information about each task and have the option to open it. If you open the task it displays the dialog to view a task. From there the functionality is the same. To close the dialog and resume to the project dashboard simply click the close button.

Profile page

Every user can also view their profile page (Figure 24), this page can be accessed from the project dashboard via the *Profile* button located in the top right corner of the page. This page displays the users basic information such as their name, email address and role. The page also displays current tasks that the user has assigned to them, the user can open these tasks and this will open the view task page and from there the functionality is the same.

Figures

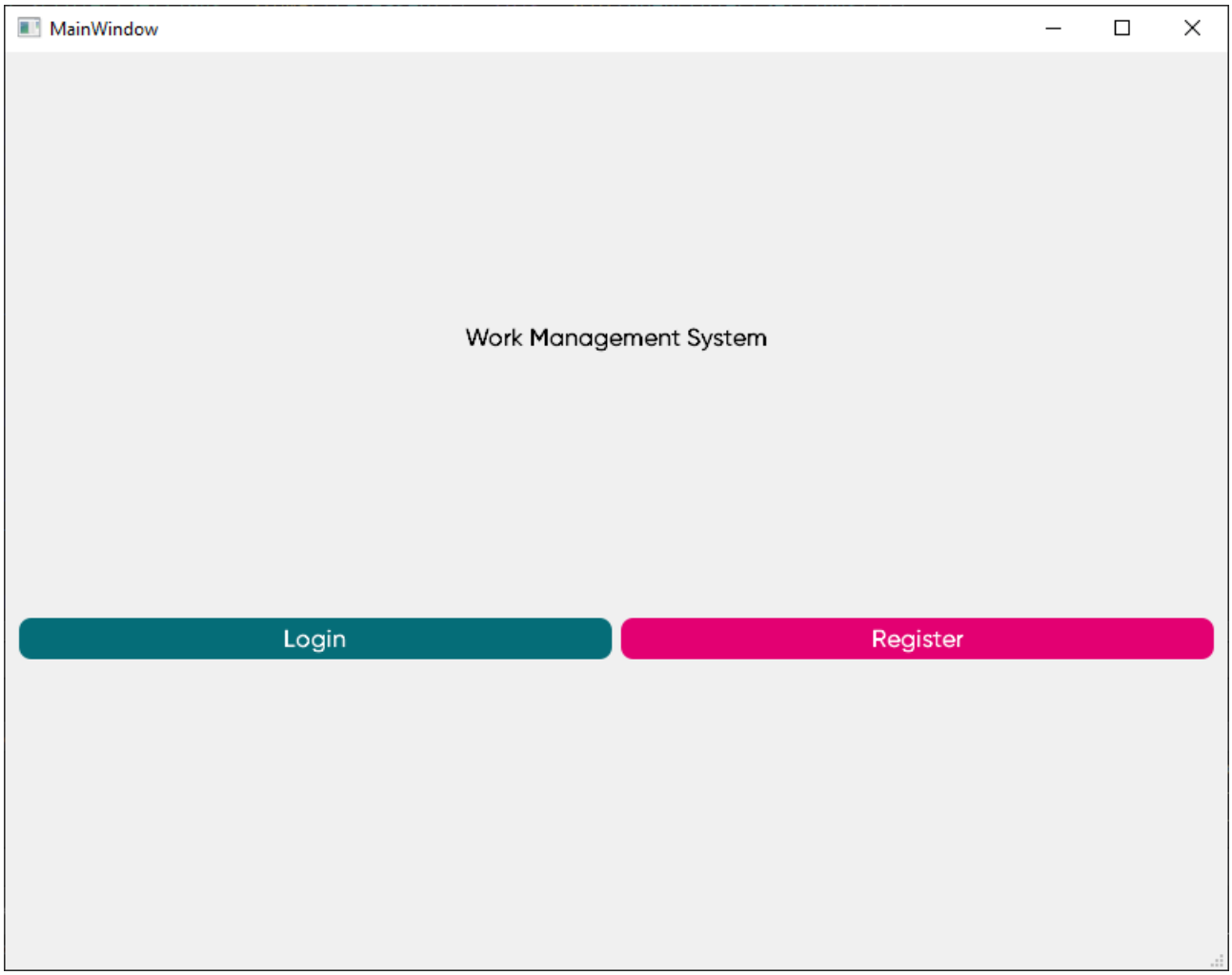


Figure 1 Startup screen

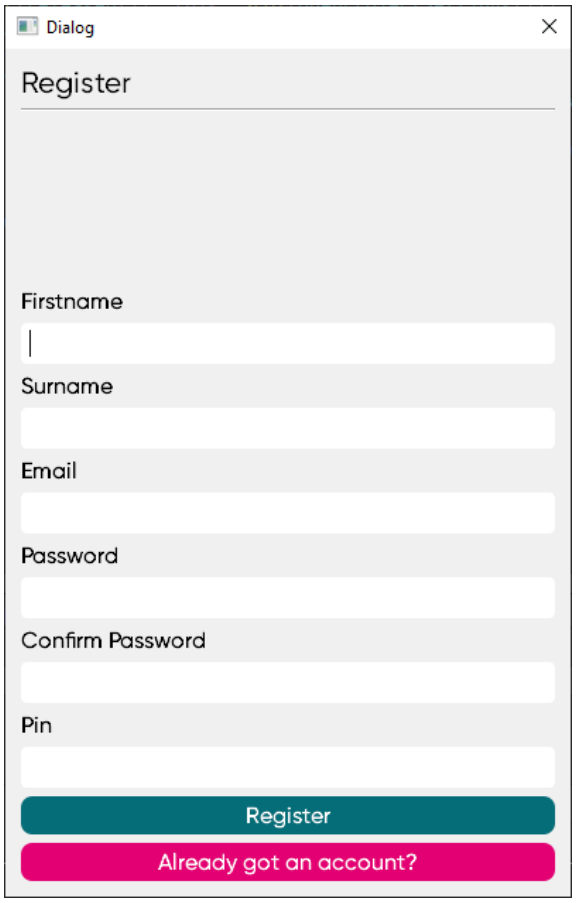


Figure 2 Register form

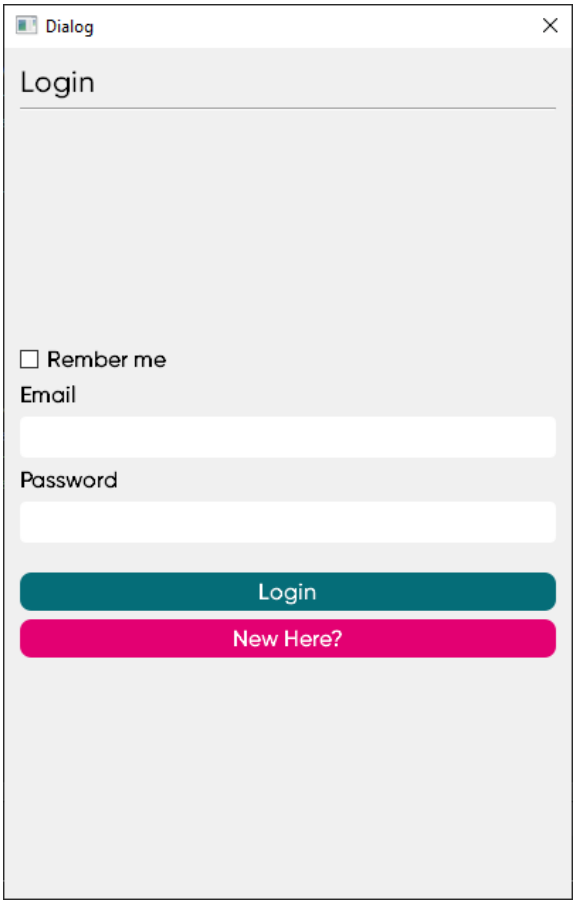


Figure 5 Login form

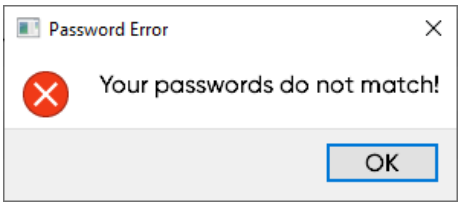


Figure 3 Password error

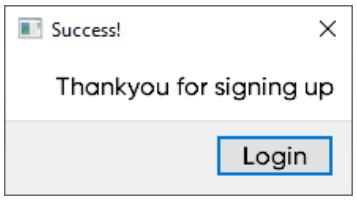


Figure 4 Registration success

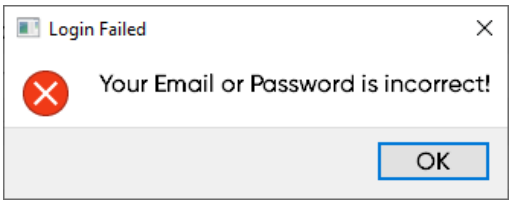


Figure 6 Login failure

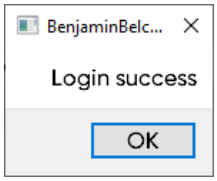


Figure 7 Login success

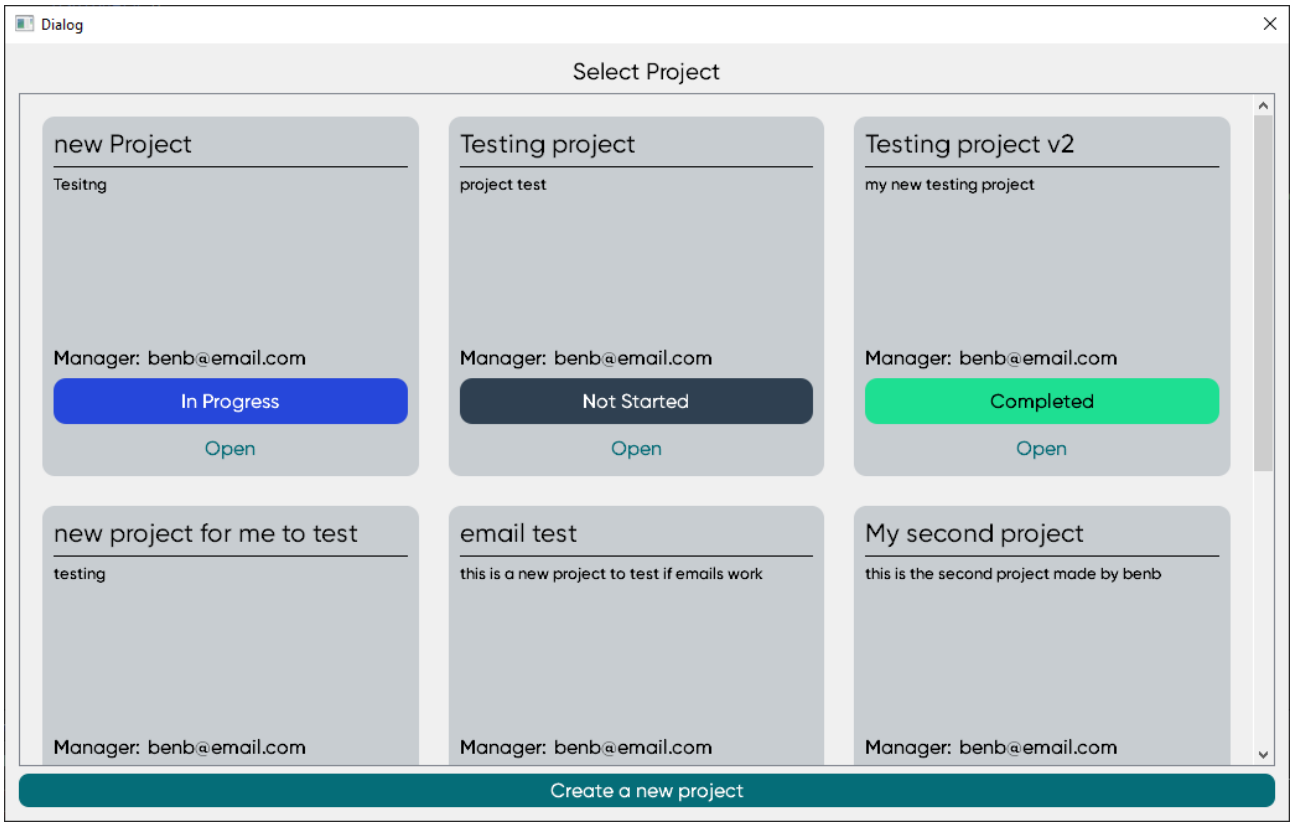


Figure 8 Project select screen

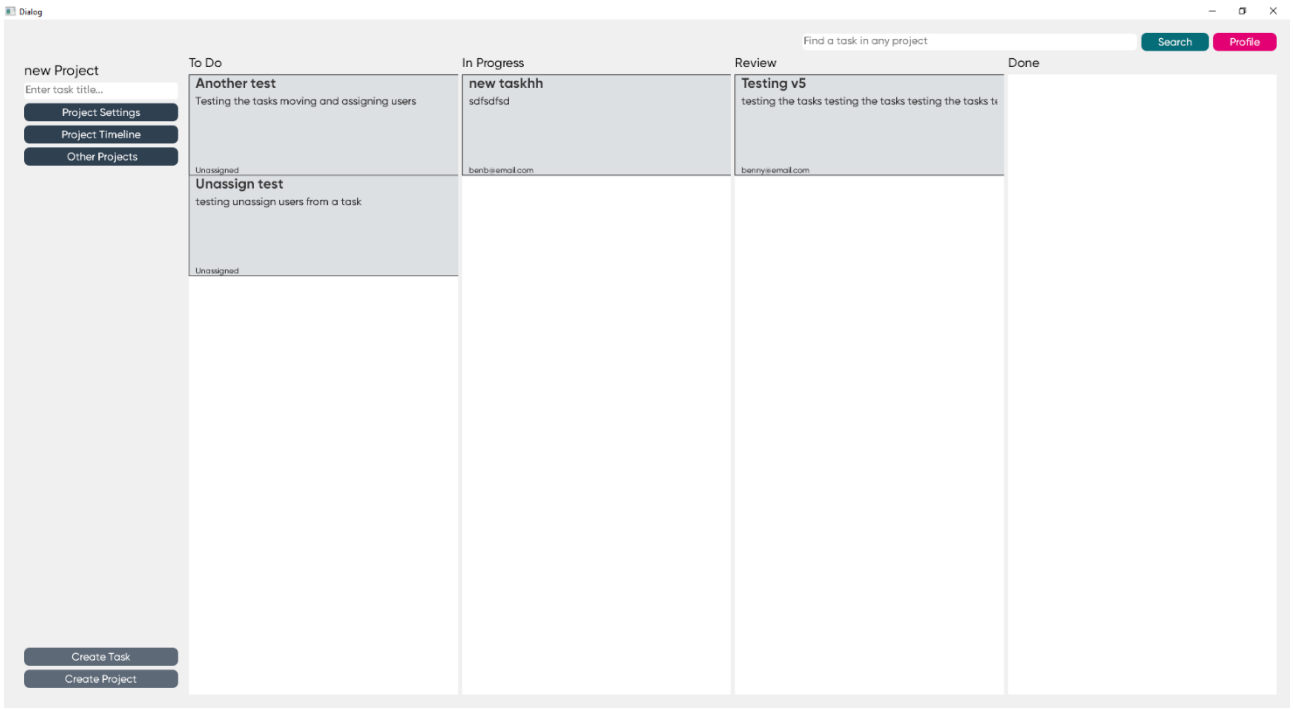


Figure 9 Project dashboard

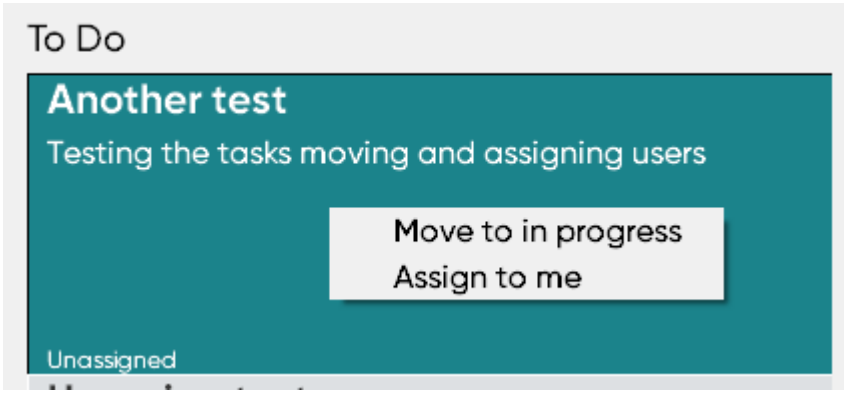


Figure 10 Task menu

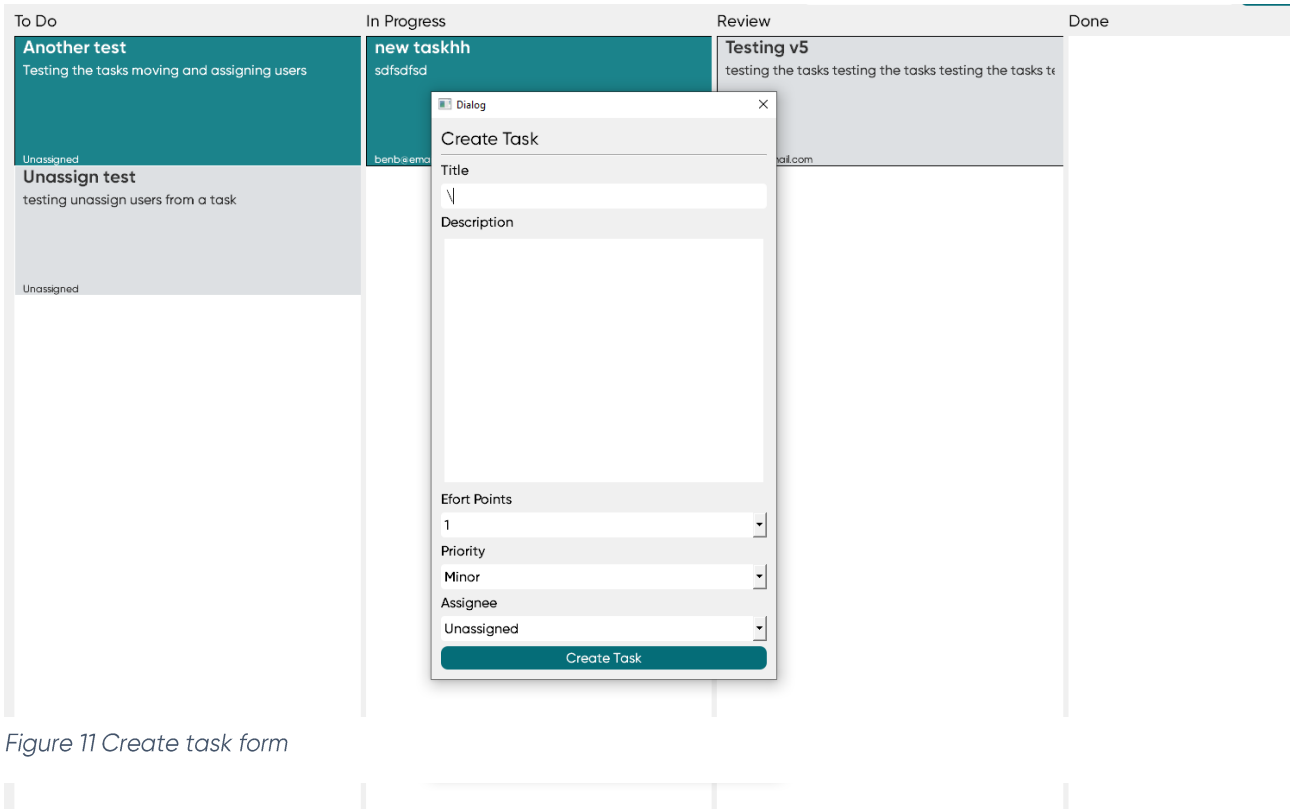


Figure 11 Create task form

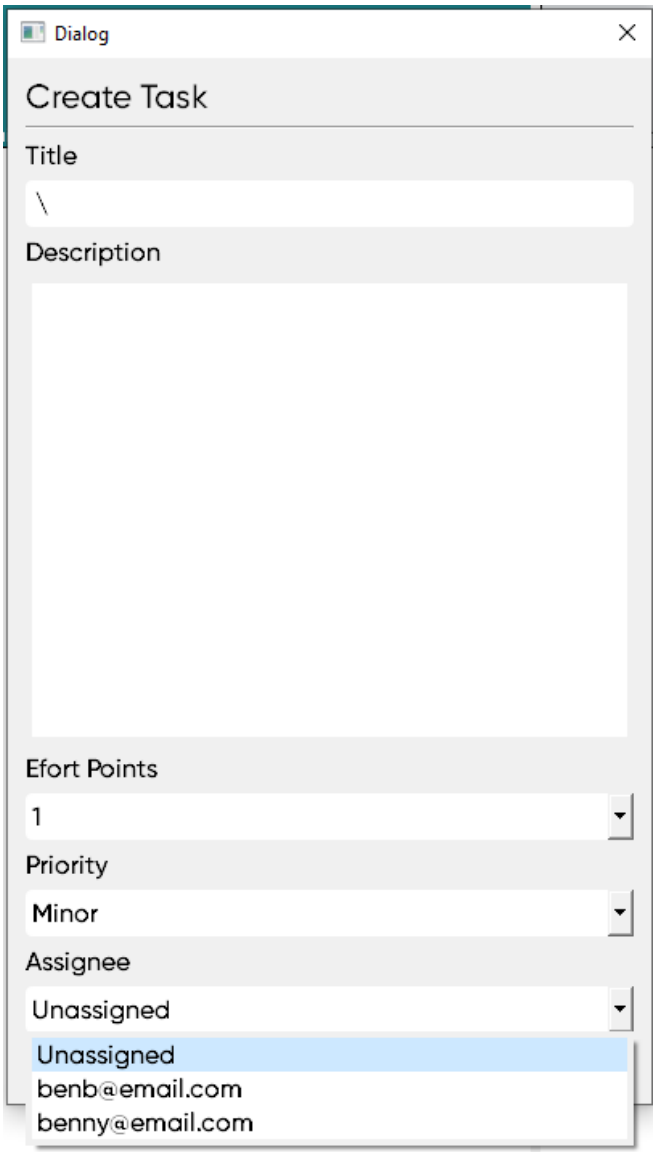


Figure 12 Create task form with dropdown

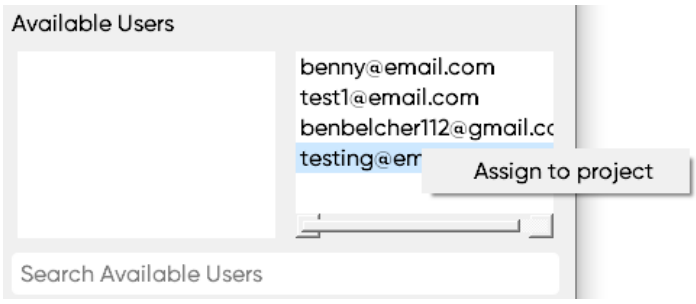


Figure 14 Assign user to project

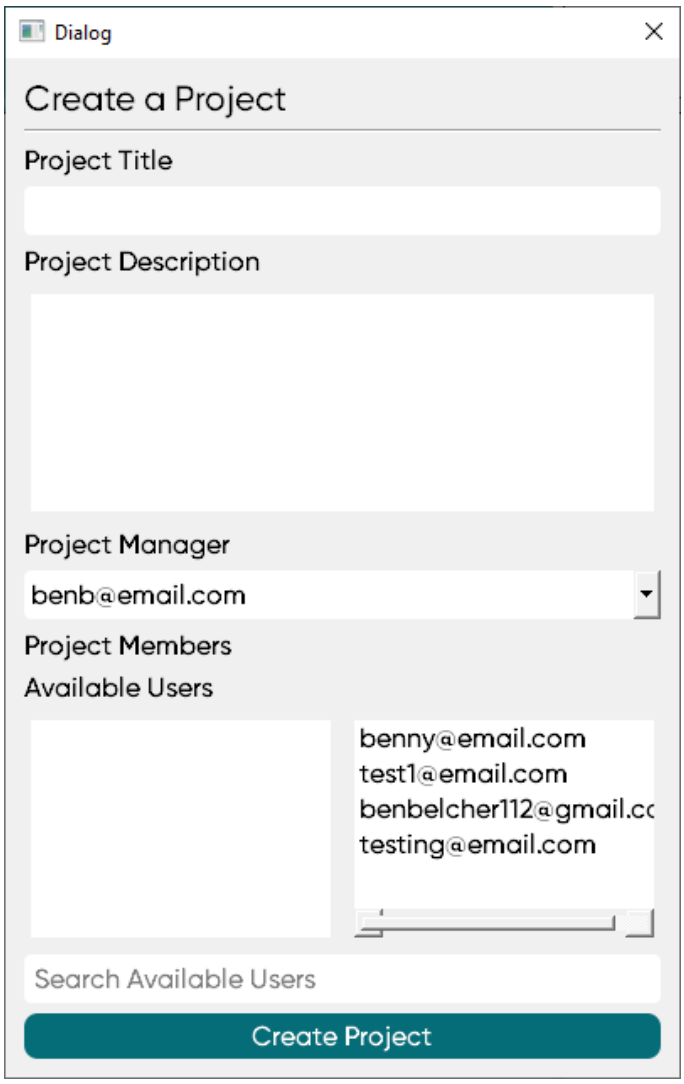


Figure 14 Create a project form

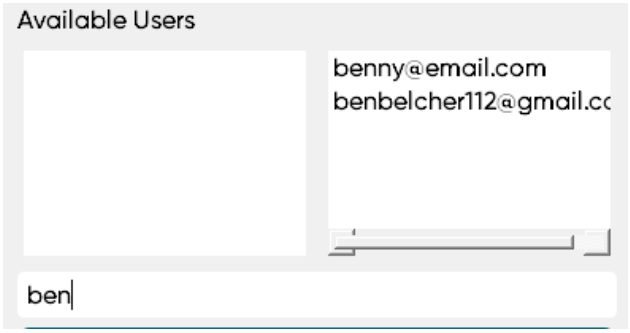


Figure 15 Available users filter

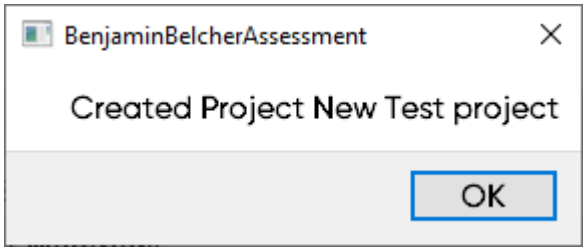


Figure 16 Creating new project

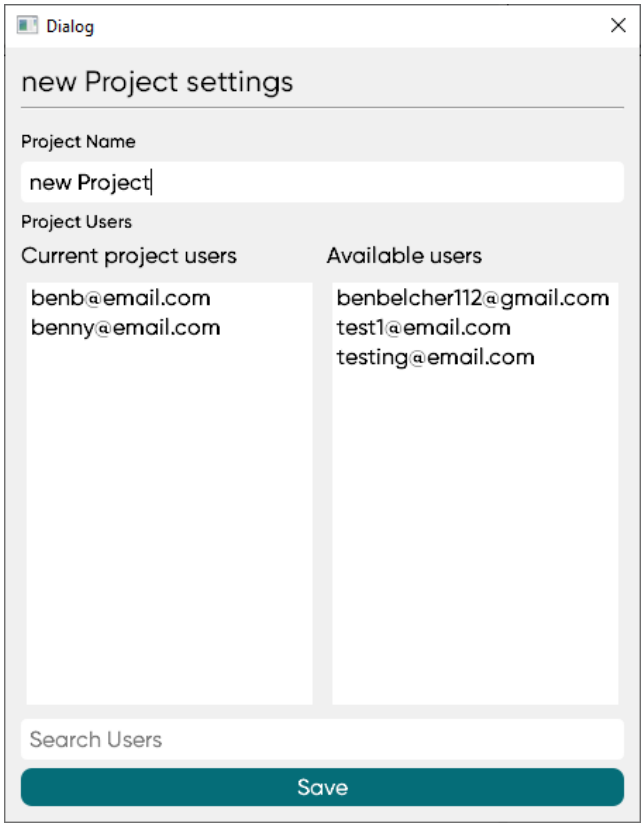


Figure 17 Project settings page

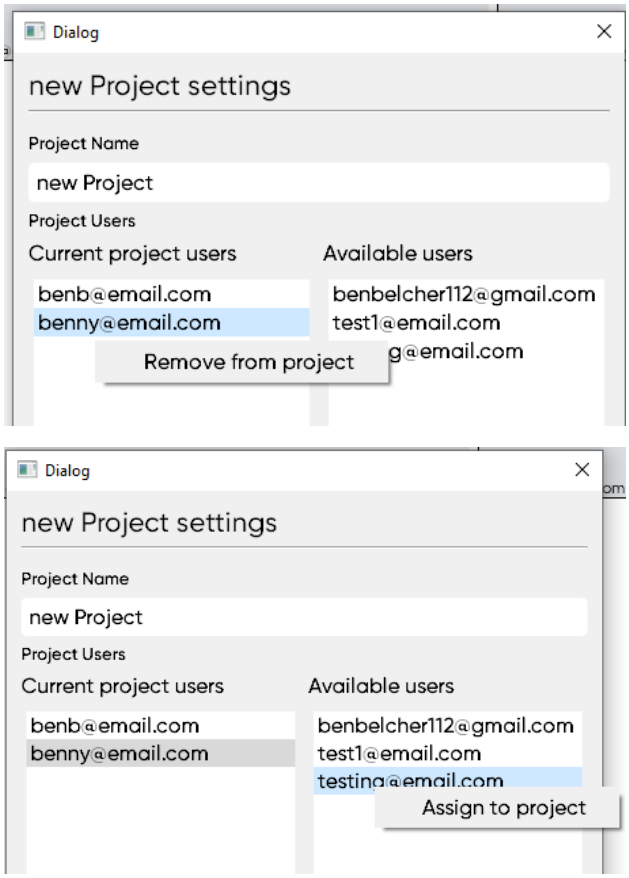


Figure 18 Add or removing users

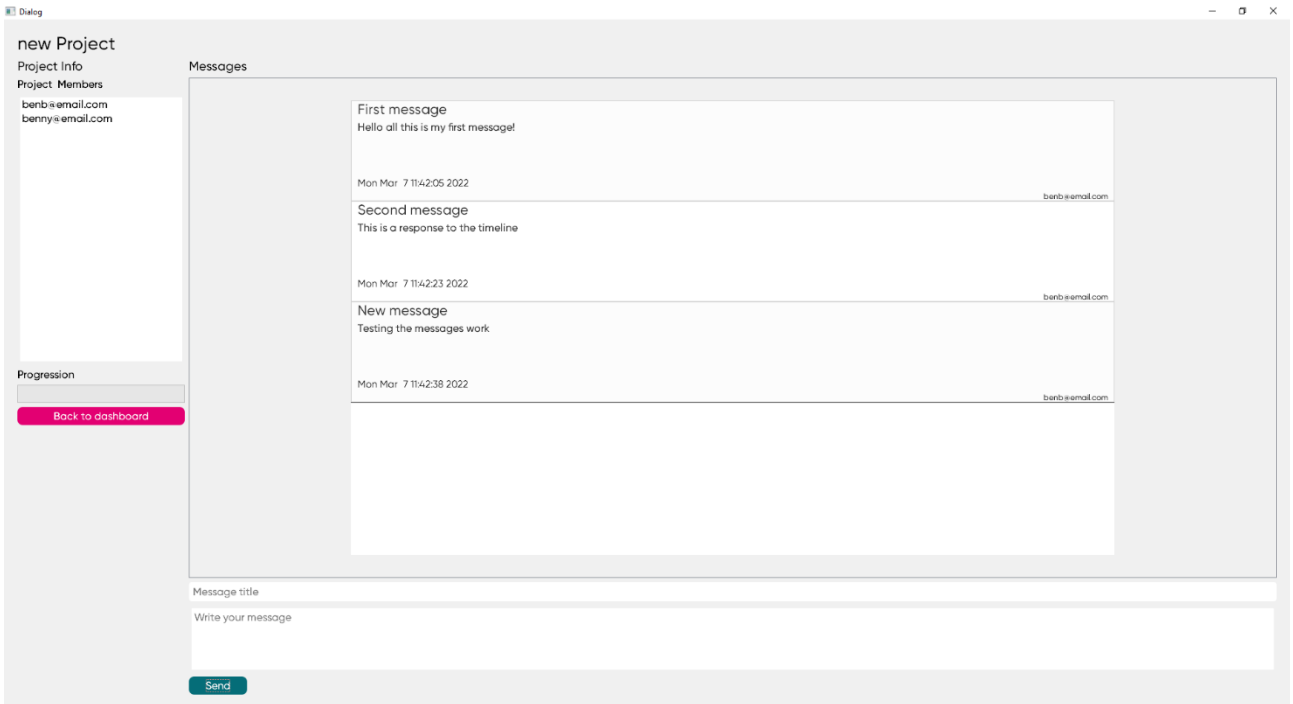


Figure 19 Project chat

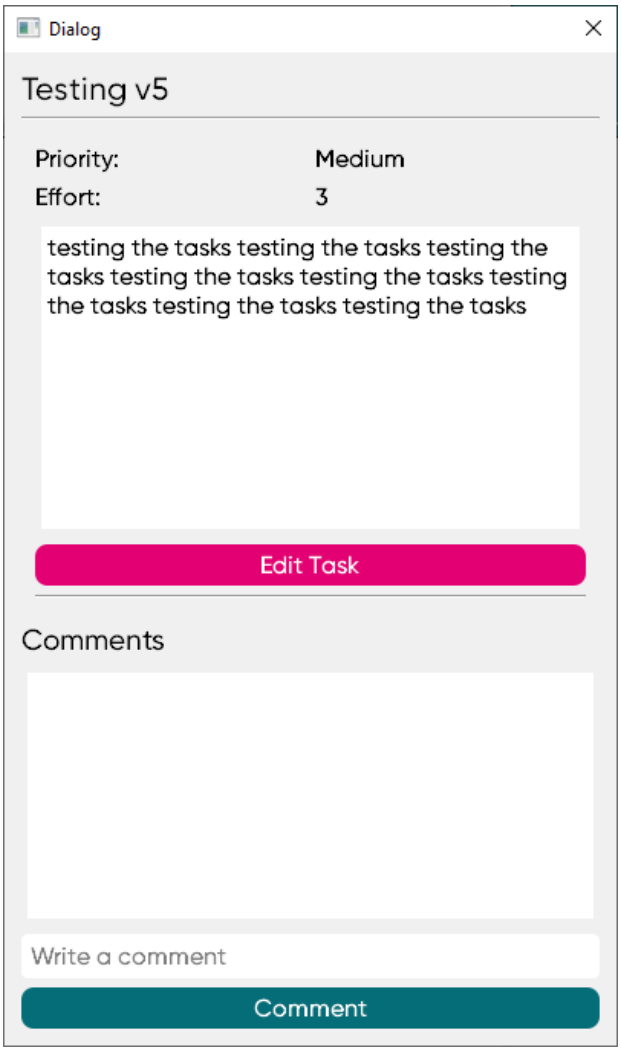


Figure 20 Task detail page

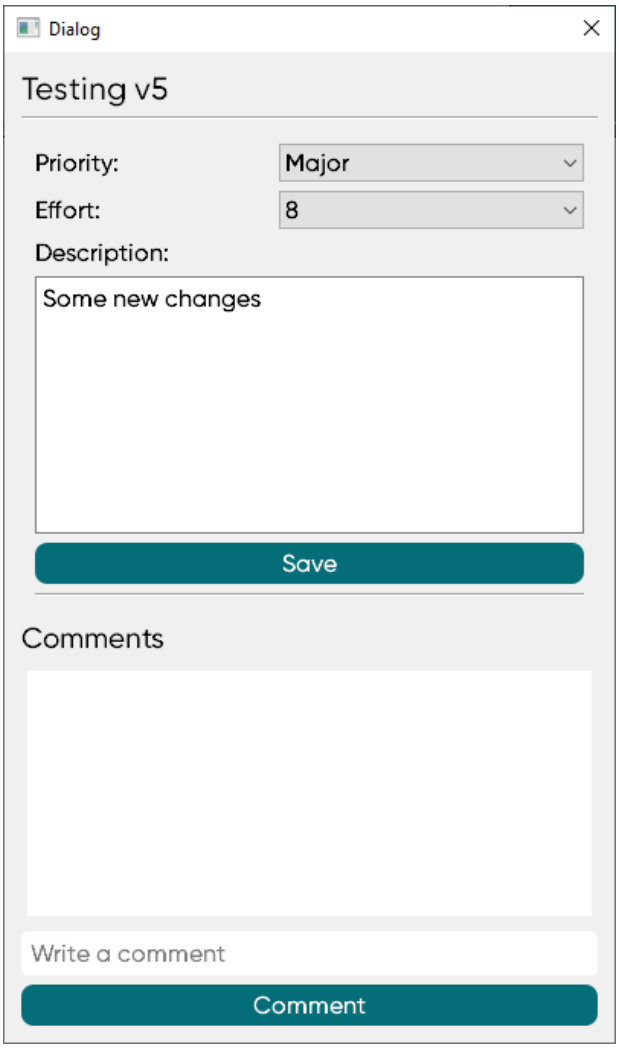


Figure 21 Edit a task

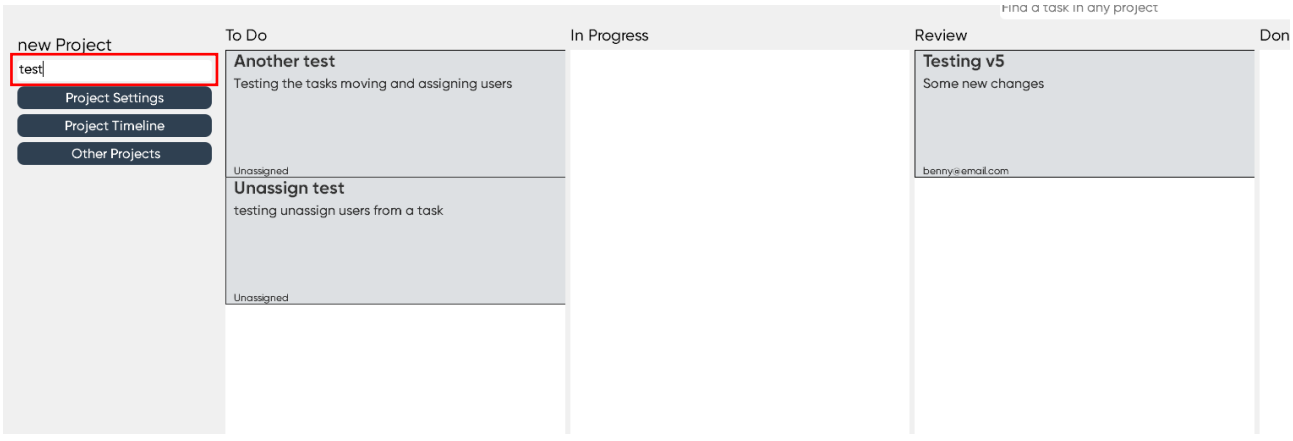


Figure 22 Search current board

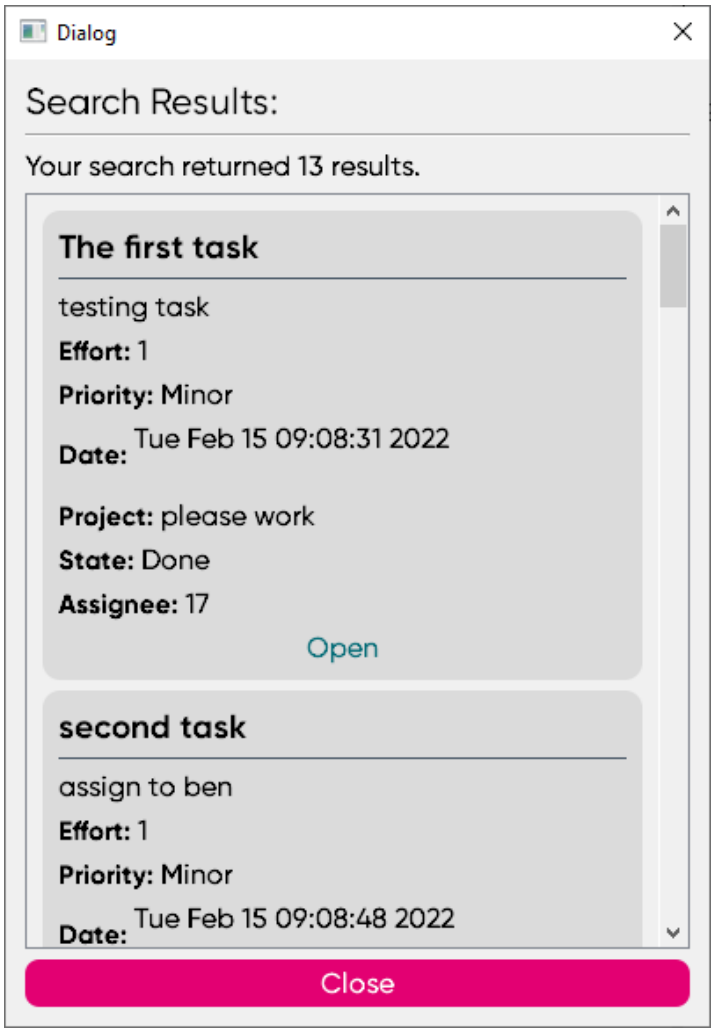


Figure 23 Search results dialog

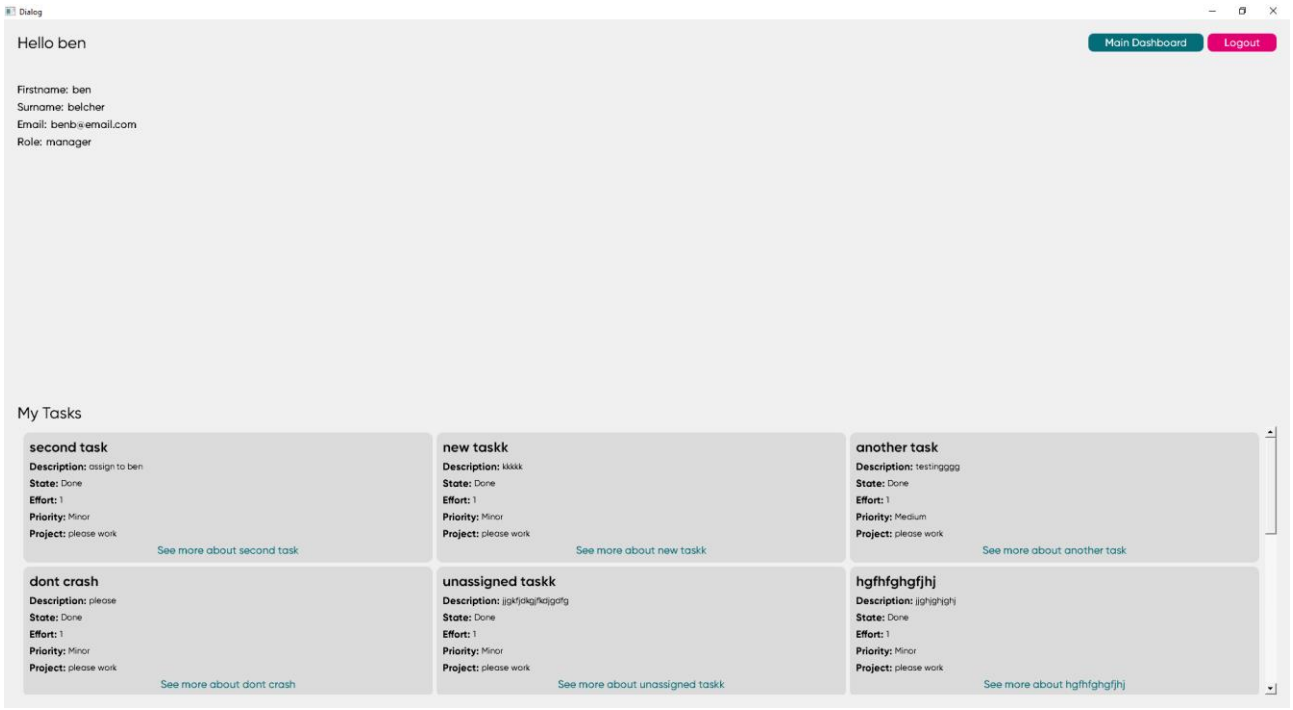


Figure 24 Profile page