**CVWO Mid-Assignment Submission** Emily Ong Hui Qi (A0240634M)

**1. User Stories & User Requirements**

**1.1 Target User Groups**

|  |  |  |
| --- | --- | --- |
| **As a…** | **I want to…** | **So that…** |
| Student | Keep track of assignment deadlines | I can plan and complete assignments on time |
| Student | Know what my friends are doing | I can be motivated to complete my tasks as well |
| Student | Categorize my tasks based on different academic subjects | I know in which subject I am pressed for submissions |
| User | Use applications that are aesthetically pleasing | I feel motivated to use the application |
| User | Use applications that runs on multiple platforms | It is more convenient |
| User | Use applications with a secure authentication mechanism | I can use the application without risking my privacy |
| Working Adult | Use applications that are frictionless and fuss-free | I can reduce learning costs and focus on what is needed |
| Working Adult | Be reminded of my calendar events and emails | I do not miss out on an important topic |
| Project Team Player | Collaborate with other people on the application | We can work together on projects |
| Project Team Player | Organize multiple projects in one place | I do not lose track of any projects |

**2. Execution Plan**

**2. 1 Implementation Details**

|  |  |
| --- | --- |
| Technology Stack | * Frontend: React.js with Redux (for state management) and Bulma UI (for lightweight CSS styling) * Backend: Go with Gin (for HTTP web-framework) and Gorm (as an Object-relational mapping to interact with the database) * Database: PostgreSQL |
| Infrastructure | * Setup a Github repository as a monorepo for the frontend and backend code, since the application is small, and it makes for consistent deployment and easy visibility of the entire application. * Setup code quality tools using Github actions and create test cases and mocks for both frontend and backend to improve code coverage. |

**2.2 Database diagram**

Diagram

Description automatically generated

**2.3 Page views**

Authentication page:

Graphical user interface, application, Teams

Description automatically generated

Home page:

Graphical user interface, application, Teams

Description automatically generated

Dashboard page:

A screenshot of a computer

Description automatically generated with medium confidence

**3. Future work**

|  |  |
| --- | --- |
| Database | * Currently, file-based SQLite is used, which is inconsistent in deployments. It is recommended to use PostgreSQL. |
| Integrations | * Enable users to integrate tools such as their calendar or emails into the application, e.g. through external public APIs such as the Google Calendar API. |
| Tags Management | * Currently, users can only create tags. Users should be able to edit and delete tags. |
| Collaboration | * Set up boards that share many-to-many relationships with users i.e enabling different users to collaborate on the same board |
| User Experience | * Use Cron jobs to send reminders about tasks deadlines to task owners |