

CVWO AY2020/21 Mid Assignment Writeup

Basic Use Cases: Actor-User

Use case: U01 – Add Task

Description: User can add task to the “to-do” list

Action Flow:

- 1) User chooses to add task (clicking on the Add Task button)
- 2) User fills in the details of the task and add the task
- 3) The system saves the task in the database (and redirects the user to homepage)

Use case: U02 – View Tasks sorted by Deadline Fields

Description: User can view the tasks in the “to-do” list based on tasks deadline.

Action Flow:

- 1) User chooses to view tasks
- 2) User clicks on the View Task / homepage button in Navigation Bar.
- 3) System presents the tasks to the user which are sorted by date.

Use case: U03 – View Tasks Sorted by Categories

Description: User can filter the tasks based on tasks categories.

Action Flow:

- 1) User chooses to view tasks sorted by the tasks' tag.
- 2) System presents all the categories with its' tasks accordingly.

Use case: U04 – View Individual Task

Description: User can view the details of a task in the “to-do” list

Action Flow:

- 1) User chooses to view a task
- 2) User can click on any task from UCo2, UCo3.
- 3) System presents the details of the task to the user and the ability to edit or delete task.

Use case: U05 – Update/Edit Task

Description: User can update/edit a task in the “to-do” list

Action Flow:

- 1) User chooses to edit a specific task.
- 2) User clicks on the task's edit button.
- 3) User edits the fields of the task and update the task.
- 4) The system saves the task in the database and presents the updated task to the user.

Use case: U06 – Destroy/Delete Task

Description: User can delete a task in the “to-do” list

Action Flow:

- 1) User chooses to delete a specific task.
- 2) User clicks on the task’s delete button.
- 3) The system deletes the task in the database (and redirects the user to homepage).

Use case: U07 – Complete Task

Description: User can complete a task in the “to-do” list

Action Flow:

- 1) User chooses to complete a specific task.
- 2) User clicks on the task’s complete button.
- 3) The system updates the state of the task in the database.

Additional Features:

This is the additional features list that can be added (aside from Optional tasks from 4.0 section in CVWO assignment pdf). These features come from the idea that a to-do list app must have the ability to be flexible to serve its users’ needs.

Additional Task: User Authentication System

User Authentication System in a to-do list application is important to make sure that a user can retrieve his/her own tasks for future visits. Without the addition of User Authentication System, all of the tasks from multiple users will be clumped together as the system does not have the ability to differ between users. Furthermore, it is generally desirable to have a personalized to-do list application because every individual has different tasks.

This feature may include:

- i. The ability to sign up, login, and logout.
- ii. The ability to have personalized to-do list.

Additional Task: To-do List for Teams

Managing a team projects has always been crucial to many individuals, especially those who are working on a big project. Therefore, it might be useful to have an additional feature which enables users to create teams and share the to-do list’s tasks among their peers in the team.

This feature may include:

- i. The ability to create teams and invite other users.
- ii. The ability to create and share tasks among users in a particular team.

Execution Plan:

I have completed most of the main components of the backend side in this to-do list project. Currently, the project uses `sqlite3` for the database to keep the project simple. I have also implemented the `fast_jsonapi` to ease the process of implementing RESTful API in my project. Overall, the CRUD functionality of the project has been implemented.

Personally, I think that the overall design of the web application itself is very important in this particular project. This is based on the fact that a to-do list's main selling point lies in its efficiency and user-friendliness. I am planning to make use of Bootstrap to help beautify the web application in order to enhance user experiences. This also helps to make the web application more user-friendly.

The most important part is to finish the main components of the project. If there is time left for me to implement the rest of the additional feature, I am planning to make use of the *devise gem* for the User Authentication System. Then, I will try to implement the team system of the to-do app. Therefore, I have to further learn about the *devise gem* and implement the database system for both features.

Problems and Challenges

This project is highly challenging for me because I have never touched Ruby on Rails and React, the two main frameworks used. Furthermore, I have never learnt the programming language Ruby and I only have minimum knowledge about Javascript which are the backbone of this project. This is because I am an information security student who learnt C in CS1010 rather than Javascript in CS1101S. Therefore, I have spent most of my time to learn the frameworks and their best practices. I have started on some basic online courses to jumpstart my understanding of the aforementioned frameworks.

Furthermore, I also have never learnt Cron, Typescript, RESTful APIs, Hosting, Redux, or Docker. However, because I feel that understanding the concept of APIs is crucial, I have started my study on it. By now, I have managed to learn briefly about RESTful APIs and implement it in my project.

I am also struggling with the design as I am not proficient in CSS. This is challenging for me because to create a good web application, understanding CSS and familiarizing myself with it is important. Therefore, I have been practicing with it to increase my overall skill. Currently, my next plan is to proceed making the front-end side of the project.

Fortunately, I have some previous knowledge about web development in Django. This really helps me with understanding the new frameworks as I only have to adapt and familiarize myself with them. I have also some brief knowledge about the MVC concept which helps me to understand the mechanism of Ruby on Rails better. This is really beneficial as it enables me to learn other concepts of web development.