

## **CVWO Mid-Assignment Write-Up Submission**

Lim Su Woon, Sarah (A0190365A)

### **Use Cases**

Name: Adding and editing tasks in to-do list

Actors: User

Normal flow: The user will type in the name of the task to be carried out in the relevant input field and add it to the list. The main page displays all the tasks that have been added as a single continuous list, along with a checkbox next to each one. To mark a task as done, the user simply clicks on the checkbox, where a green tick will appear and the task will be striked out.

Tasks are by default untagged. The task label and its tags may be edited or deleted by hovering and clicking on either the green edit button or red delete button.

When the edit button is clicked, the editing field for the task label and tags will appear, allowing the user to edit the task label and/or edit tags associated with the task. Currently, only tags which already exist may be added. When a task with tags is deleted, only its association to the tag is deleted; the tag instance remains intact even if there are no posts associated with it.

Name: Finding task by tags

Actors: User

Normal flow: The user clicks on the 'Manage Tags' link from the main page, and will be redirected to another page where the tasks are sorted by tags. On this page, there is an input field to allow users to create new tags. Existing tags may also be edited or deleted by hovering and clicking on either the green edit button or red delete button which appears next to the tags.

Each task has a checkbox next to it. To mark a task as done, the user simply clicks on the checkbox, where a green tick will appear and the task will be striked out. The updated status of the task will be reflected in the main page as well.

### **Execution Plan and Suggestions**

Although it may be difficult to implement all intended functionalities within the provided deadlines given my limited prior experience with React on Rails, I hope to be able to accomplish as much as I can.

By mid-assignment submission:

Accomplish basic fundamentals of the web application.

- Basic **CRUD** functions of a to-do list
  - **Create:** Create a new task
  - **Read:** Reads the database and displays the tasks that are in the database
  - **Update:** Edit task labels and keeps track of whether a task has been crossed off
  - **Destroy:** Allow deletion of tasks
- **Tagging System**
  - **Display** associated tags of each task next to it
  - Search the tasks by tags → Search bar (see below)
  - Allow only unique tags to be created
  - **CRUD** for tags
    - Implement Rails 'has\_many through' Active Record association between tags and todos

After mid-assignment submission:

Add other functionalities which would enhance the experience of the user and allow the todo-list app to encourage users achieve their goals and maximize productivity.

- (Tentative plans) Have a **sidebar** in the main page which lists all existing tags and number of tasks associated with the particular tag, allowing users to access a page with all the associated tasks for that tag
- **Search bar** to find tasks by name and/or tags
  - Allow fuzzy searching
- Include **due date** parameter to each task
  - Allow sorting by due date
  - Possibly use Cron to set a countdown timer
- Include **importance** parameter to each task
  - Allow sorting by importance
- Include **colour coding** for each task so users can sort tasks by colour
- **Customise name of user** in the title (eg. "Adam's To-Do List") → allows user to personalise their to-do list (**note: currently only partially implemented**)
- **Drag and drop** (tentative) to allow manual re-arranging of order of tasks, custom to the user
- Formatting and styling

- Cron job
- Deploying to Heroku

### **Current Obstacles**

I had very limited prior experience using React before this assignment, and no exposure to Rails at all, hence this task was certainly challenging to me as I had to pore over many documentations and tutorials before I could even properly set up Rails. I haven't taken the software engineering modules either (CS2103 / CS2113), so Orbital - where my teammate and I developed a web application to manage passwords - was one of my only forays into web development. Thus, it took me some time to learn how to navigate new fundamental concepts such as MVC and React components.

Admittedly with excessive amounts of Googling, I finally managed to implement CRUD functionality for my to-do list. It took much longer than I expected as there were many parts of code that I didn't understand when I referred to tutorials online, and I had to constantly look things up and spent hours just staring at the code trying to understand how it worked. Understanding, implementing and debugging Active Rails Association between the tasks and tags was very challenging as well, and it highlighted the importance of learning about how the database is structured, and how to manipulate the data accordingly.

However, it was extremely gratifying when my application finally worked. Though the code had seemed foreign before, I am starting to become more familiar with using React and Rails. Currently, I am in the midst of implementing the tagging system and testing it to identify any bugs. A major obstacle I am trying to overcome is creating and updating a tag when it is added to the tags of a task, as asynchronous programming means that the tags associated with a post may be updated *before* the new tag is created, causing the new tag to not be added. I am looking into how I may tackle this issue with Promises or the like.

After the mid-assignment submission, I also intend to focus more on styling and work on some of the optional tasks such as integrating TypeScript and deploying the application to Heroku. I also hope to reorganise some components and clean up my code which is currently in progress and hence fraught with unnecessary comments.