Mid Assignment Submission

Cynthia Lee A0190363H

Use cases

System: App (To-do manager)
Actor: User

Use case: UC01 – Create task
MSS:
1. User chooses to create task.
2. App requests user to enter the task details.
3. User enters the requested details and submits.
4. App creates the task and stores it in the database.
5. App redirects User to view task and details on main page.

Use case ends.

Extensions:
3a. User enters invalid or insufficient details.
3a1. App informs User to enter the necessary information.
3a2. User enters new data.
Steps 3a1-3a2 are repeated until the data entered are correct.
Use case resumes from step 4.

Use case: UC02 – View tasks
MSS:
1. User chooses to view tasks.
2. App directs User to main page to view all tasks.
3. User selects an individual task to view more details (if any).
4. User chooses to return to main page.

Use case ends.

Use case: UC03 – Update tasks
MSS:
1. User selects an individual task to view (UC02).
2. User chooses to edit the particular task.
3. App loads the task and its existing details into a form.
4. User edits the form to make changes to the data and submits it.
5. App updates the task information in the database.
6. App reflects the changes to the user.

Use case: UC04 – Delete tasks
MSS:
1. User selects an individual task to view (UC02).
2. User chooses to delete the task.
3. App asks the user to confirm the action.
4. User confirms.
5. App deletes the task.

Use case: UC05 – Tagging tasks
MSS:
1. User selects an individual task to view (UC02).
2. User chooses to add tags to a task.
3. User selects the tags for the task and confirms.
4. App saves the tags into the database.

Use case: UC06 – Sorting tasks 1. User chooses to sort the tasks. 2. User selects criteria for the tasks to be sorted by (for example, tags, state of completion, date created, keyword etc.). 3. App sorts the tasks and presents it to the viewer.

Execution plan

Task	**Expected completion date**
Readings and tutorials Setting up basic application	14 Dec
Follow online tutorials to create a basic React-On-Rails App. • Familiarise with syntax and structure • Basic task structure • Basic CRUD functionalities ○ Currently completed: C, R, D	30 Dec

• Implement ability to update task
• Implement more task features
  ○ Add more relevant fields which might be useful
  ○ Eg. Date/time component to schedule tasks
• Begin implementation of tagging system
	3 Jan

Overseas	6 Jan
• Continue implementation of tagging system	10 Jan

• Improve user interface
  ○ Try to implement tree/calendar view to be able to view tasks by schedule
  ○ Improve visibility of tagging system
	14 Jan

Buffer	17 Jan

• Implementation of optional tasks
  ○ Cron
  ○ Typescript
• Deployment
	22 Jan

• Finalising and refining user interface • Implementation of additional features if necessary • Submission of deliverables	25 Jan

Reference websites used so far for basic app implementation:

1) https://www.pluralsight.com/guides/building-a-crud-interface-with-react-and-ruby-on-rails
2) https://www.digitalocean.com/community/tutorials/how-to-set-up-a-ruby-on-rails-project-with-a-react-frontend

Difficulties faced

1. Lengthy readings for Rails which took a lot of time and delved a lot into specific details and functionalities. I am still struggling to understand how to properly integrate React and Rails (eg. how to ensure a change in a React component results

in a change in the database, for example), but this is perhaps due to my currently insufficient knowledge and experience in both.

2. Time management. I found it difficult to stick to my schedule as unexpected difficulties caused me to be stuck along the way, taking more time than expected to complete the tasks. Since then, I have tried to include buffers in my schedule to better deal with them. I am also trying to devote more time to this assignment.