CVWO Project : Task Management Application

Application Name

StudenTask

Value Proposition

StudenTask is a **task management application** to provide NUS students a fast and convenient way to manage their tasks.

Tools

Backend will be written in *Rails* and Frontend will be in *React*.

User Stories

As a	I want to	so that	Priority
user	record my task	I can view them.	***
user	delete my task	I can reduce clutter.	***
user	login to my account	access my personalised task.	**
student	arrange my tasks	I can attend to more important/urgent/relevant tasks.	**
student	mark my task as completed	I have a record that the task is done.	***
impatient student	have shortcuts to direct me to commonly used NUS websites	I can be easily directed to accomplish my task.	*

Use Cases

For the use cases below, unless otherwise mentioned, the system is *StudenTask* and the actor is the user of the *StudenTask*.

Use case: UC01 : Add Task

MSS

1. User chooses to add task.
2. User enters the task details.
3. User confirms.
4. StudenTask displays the newly added task.

Use case ends.

Extensions

3a. StudenTask detects an error in the entered data.
3a1. StudenTask requests for the correct data.
3a2. User enters new data.
Steps 3a1-3a2 are repeated until the newly entered data is correct.
Use case resumes from step 4.

Use case: UC02 : Delete Task

MSS

1. User chooses a task to delete.
2. StudenTask requests for confirmation.
3. User confirms.
4. StudenTask removes the deleted task from the display. Use case ends.

Extensions

2a. User does not confirm the deletion.
2a1. StudenTask goes back to the task display.

Use case ends.

Use case: UC03 : Sort Task

MSS

1. User chooses a criteria to sort the task by
2. StudenTask displays the tasks in the sorted order

Use case ends.

Use case: UC04 : Login Account

MSS

1. User enters username and password
2. User confirms details.
3. StudenTask authenticates and allows user to access account.

Use case ends.

Extensions

3a. StudenTask detects an error in the entered data.

3a1. StudenTask requests for the correct data.

3a2. User enters new data.

Steps 3a1-3a2 are repeated until the newly entered data is correct.

Use case resumes from step 3.

Use case: UC05 : Mark task as completed

MSS

1. User chooses a task to mark as completed.
2. StudenTask archives the task and marks it as completed.

Use case ends.