

Use Case Descriptions

Use Case: View homework schedule

Goal: The user wants to view their scheduled homework by day of the week

Preconditions: n/a

Trigger: User opens the app or navigates to main page

Scenario:

1. User opens the app or navigates to the main page
2. User views the tasks of the current day
3. User may tap on a task for its full info
4. User navigates to view other days of the week using the top toolbar

Exceptions:

1. No tasks have been added by the user: schedule is empty of any homework assignments

Open Issues:

1. Should this display both homework assignments and events from their calendar?

Use Case: Add task

Goal: The user wants to add a task

Preconditions: n/a

Trigger: The user selects the add button

Scenario:

1. The user decides to add a task and selects the add button
2. The user fills in the name of the task
3. The user fills in the due date of the task
4. The user fills in how long they think the task will take
5. The user adds any subtasks
6. The user taps the create task button

Exceptions:

1. The user does not fill in all required info(name, due date, predicted time): the create button is not responsive until required fields are entered. This is made visually clear to the user.
2. The user changes their mind: There is a cancel button.

Open Issues:

1. After a task is added should the user have the option to add another(delaying running the scheduling algorithm until they are done adding)?

Use Case: Add subtask

Goal: The user has a task that can be broken into multiple subtasks

Preconditions: There is a task to add the subtask to

Trigger: The user is on the initial add task screen or is looking at the task and selects the add subtask option

Scenario:

1. User selects add subtask
2. User inputs name of the subtask
3. User inputs how long they think the subtask will take
4. User taps the add subtask button

Exceptions:

none

Open Issues:

If the time of all subtasks combined are smaller than the time of the overall task should the overall time not be updated or is this a conflict?

Use Case: Update task info

Goal: The user wants to change the information of a task or delete it

Preconditions: The task exists to be updated

Trigger: The user is looking at a task and selects edit

Scenario:

1. The user selects edit
2. The user changes any fields they want or deletes the task
3. The user edits any subtask or deletes it
4. The user selects the update button

Exceptions:

1. The user changes their mind: there is a cancel button

Open Issues: none

Use Case: Dismiss task

Goal: The user has completed a task or part of a task and wants to mark it

Preconditions: There is an existing task

Trigger: User swipes left on a task or subtask on the calendar view or selects dismiss when viewing the task

Scenario:

1. The user is looking at their schedule and swipes left on a task/subtask scheduled

Or

1. The user has viewing a task and selects the completed button for the whole task or subtask

Exceptions:

1. The user accidentally swiped and wants to undo it: swiping right undo it. a task is not deleted until the day they were scheduled for is over

Open Issues:

1. Should there be a way to mark a task partially finished if no subtasks exist to be marked off?

Use Case: View homework list

Goal: The user wants to view all homework

Preconditions: n/a

Trigger: User navigates to task list

Scenario:

1. User navigates to task list
2. User views the tasks and reads name, due date and percent completed
3. User may tap on a task for its full info

Exceptions:

1. No tasks have been added by the user: list is empty of any homework assignments

Open Issues: none

Use Case: Add/remove calendar

Goal: The user wants to add or remove a calendar

Preconditions: n/a

Trigger: user navigates to add calendar page

Scenario:

1. User navigates to add calendar page
2. User reads list of calendars already used
3. User Selects another calendar

Or

4. User removes a calendar

Exceptions:

1. No calendars are available: the user reads a message saying this

Open Issues: none