**SW Engineering CSC 648-848**

**OnTask**

**Section 4 Team 5**

Rhoanna Perez (Team Lead), Christopher Su (Backend Lead), Jia Li (Frontend Lead)

Nelson Pang (Scrum Master), Seongjung Kim (Github Master), Michael Abolencia

**Milestone 2**

**March 18, 2022**

**History Table**

|  |  |
| --- | --- |
| **First Submission** | **March 18, 2022** |
| **Last Revision for M5 Submission** | **May 13, 2022** |
|  |  |

**1. Data Definitions V2**

|  |  |  |
| --- | --- | --- |
| **Name** | **Definition** | **Usage** |
| Task | Individual tasks created by the user in the to do list, containing the date, title, category, and priority | Users create tasks that they can complete and see in a list format |
| To do list | A list of tasks that a user creates | Student users have the ability to use collaborative to-do lists set upon their **teacher**, given a user ID |
| Date | The day that a user assigned a task | Users can see when they created a task |
| Title | The name of a user’s task (ex: Finish assignment 1 for CSC 413) | Users can view a list of tasks that they’ve titled themselves |
| Category | A group that a user creates for each task (ex: categorizing a task under **exams**, **assignments, reminders**) | Allows user to view tasks by category |
| Priority | The importance of a task established by the user (ex: Priority 1 for assignment due tomorrow) | Allows users to view tasks by priority (1st priority first to 3rd priority last) |
| Deadline | The due date that a user defines for each task (ex: Finish assignment due 3/19) | Allows users to view tasks in chronological order of deadline |
| Collaborative to-do list | A to-do list that is shared among other users | Keeps track of tasks between users |
| Creator | The person that creates a to-do list, term commonly used for collaborative to-do lists | Specifies who created the to-do list |
| Viewer | A person who can view a to-do list, often shared from a collaborative to-do list | Shows who can view a creator’s list. Creator can specify who can edit. |

**2. Functional Requirements V2**

**F.1: Sign-in/Login/Logout, Priority 1**

* Users can sign up for accounts and set it up based on what they need it for.

**F.2: Login/Logout, Priority 1**

* Users are able to log in and log out through their accounts.
* All users within our list of personas will be able to create an account as a way to register with the to-do list application, and will be able to login and logout.

**F.3: Categorize Tasks, Priority 1:**

* All users within our list of personas will be able to create a to-do list and categorize each task into different categories. The user can create their own categories and filter them out into each. In the different view layouts, users can sort their tasks by the categories.
* Students like Michelle are able to determine what categories tasks can be sorted into such as school work or feeding their pets which can be categorized into home necessities.

**F.4: Assign Priority, Priority 2:**

* A user is able to assign different importances to their to-do list items.
* The user would be able to drag and drop the important tasks from top to bottom.
* All users within our list of personas can assign priorities to their tasks. The different priorities on tasks can be set with different notifications. If the task has higher priority, it can be set with multiple reminders or with an alarm. If they are missed, there would be a snooze-like button. The application will remind them several minutes later. Less prioritized tasks will still have notifications, but they can be missed.

**F.5: Add a Friend, Priority 3:**

* Once an account is created, the user will be generated with an account ID.
* Users will be able to add a friend through a user’s account ID. A user will invite and request another user to become a friend, and if they would like to be friends, they can accept the invite.
* A friends list will allow users to easily share a to-do list with a friend and request a friend to either view/edit a to-do list.
* Clicking onto a friend’s icon will bring the user to their friend’s tasklist/calendar. Tasks that are created private will not be visible unless changed by the owner. **F.5.1-Priority 3**

**F.6: Creating a Collaborative To-do List, Priority 1:**

* Users will be able to create a collaborative to-do list. As opposed to having your own to-do list (which is private and only viewed by the owner), a collaborative to-do list allows a user to invite other users to a list that they can edit and view.
  + In order for a user to be added to the collaborative to-do list, user A needs to invite user B to the to-do list. User B needs to accept an invitation to edit/view. If they do not want to be added to the list, they can decline the invitation.
* A collaborative to-do list will allow users to assign tasks to other users and view what collaborative tasks need to be completed by all users who can view the list.
  + When adding a task, a user will have the option to name the task and assign the task to another user within the collaborative to-do list.
* Once a user is assigned to a particular task, the user will be able to see the task on their personal dashboard as a list things to complete. **F.6.1-Priority 2**

**F.7: Notification Pop-Up, Priority 3:**

* When the project task is created or deleted. **F.7.1-Priority 1**
* When someone has sent a comment or notes to a user. **F.6.2-Priority 2**
* When someone has assigned a task or project. **F.7.3-Priority 1**
* When a user has received email or comments. **F.7.4-Priority 3**
* When someone accepts or declines others invitation to join a project **F.7.5-Priority 2**
* When a task that a user has either created or assigned is completed. **F.7.6-Priority 1**
* When a task that a user has either created or assigned is uncompleted. **F.7.7-Priority 1**
* If the user sets a due date and time, notification pop up will be shown before the due date. **F.7.8-Priority 1**

**F.8: Add Notes to Tasks, Priority 1:**

* Users can save all the relevant task information right inside user’s task **F.7-Priority 1**
* Users can add notes for collaborative to-do lists for themselves or for others to see **F.7-Priority 1**

**F.9: Mark Task As Complete, Priority 1:**

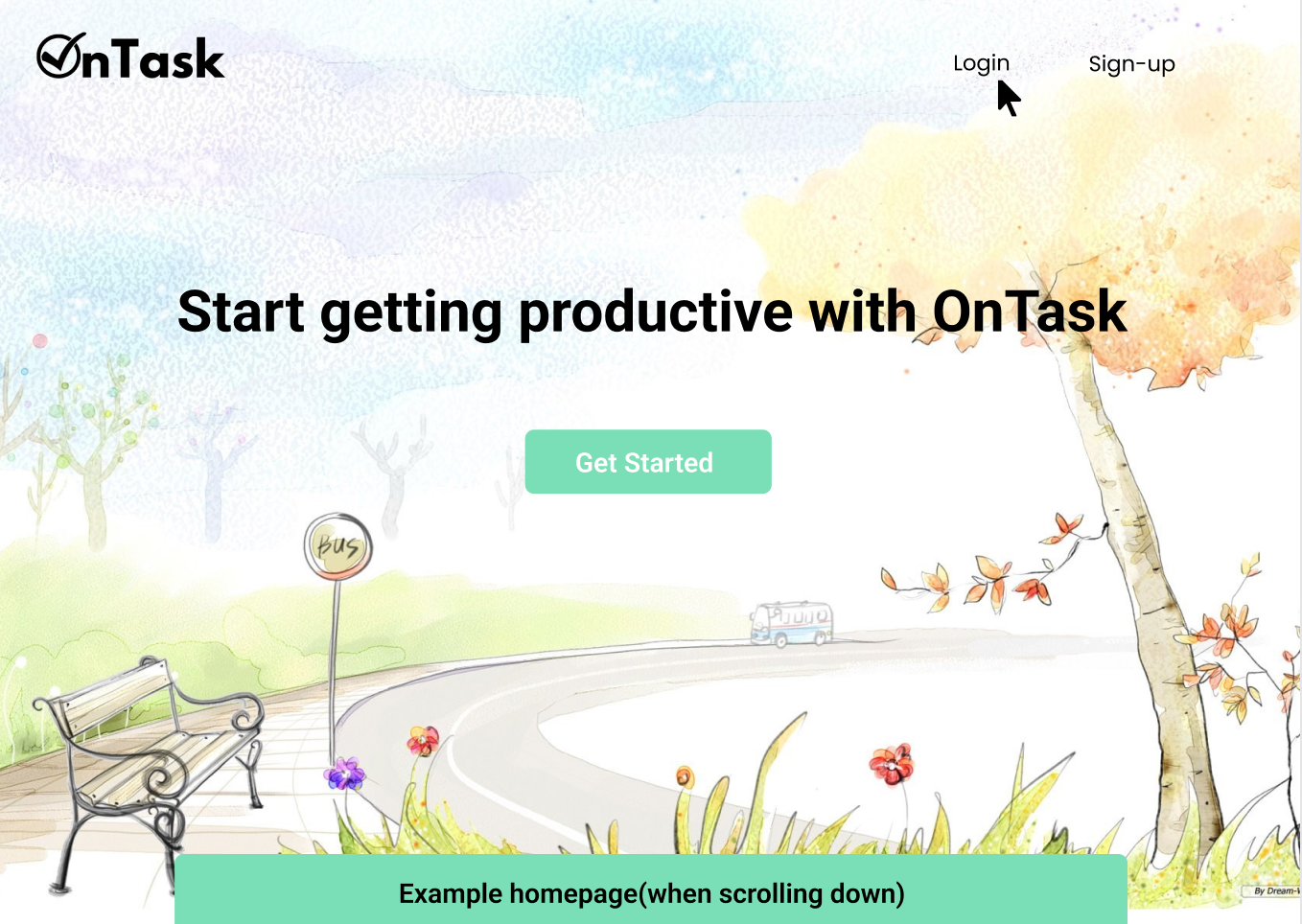
* Users can mark a task as complete by clicking the checklist **F.8-Priority 1**
* After marking it as complete, the task will be crossed off their to-do list and will be removed, showing the rest of the tasks that are yet to be completed **F.8-Priority 1**

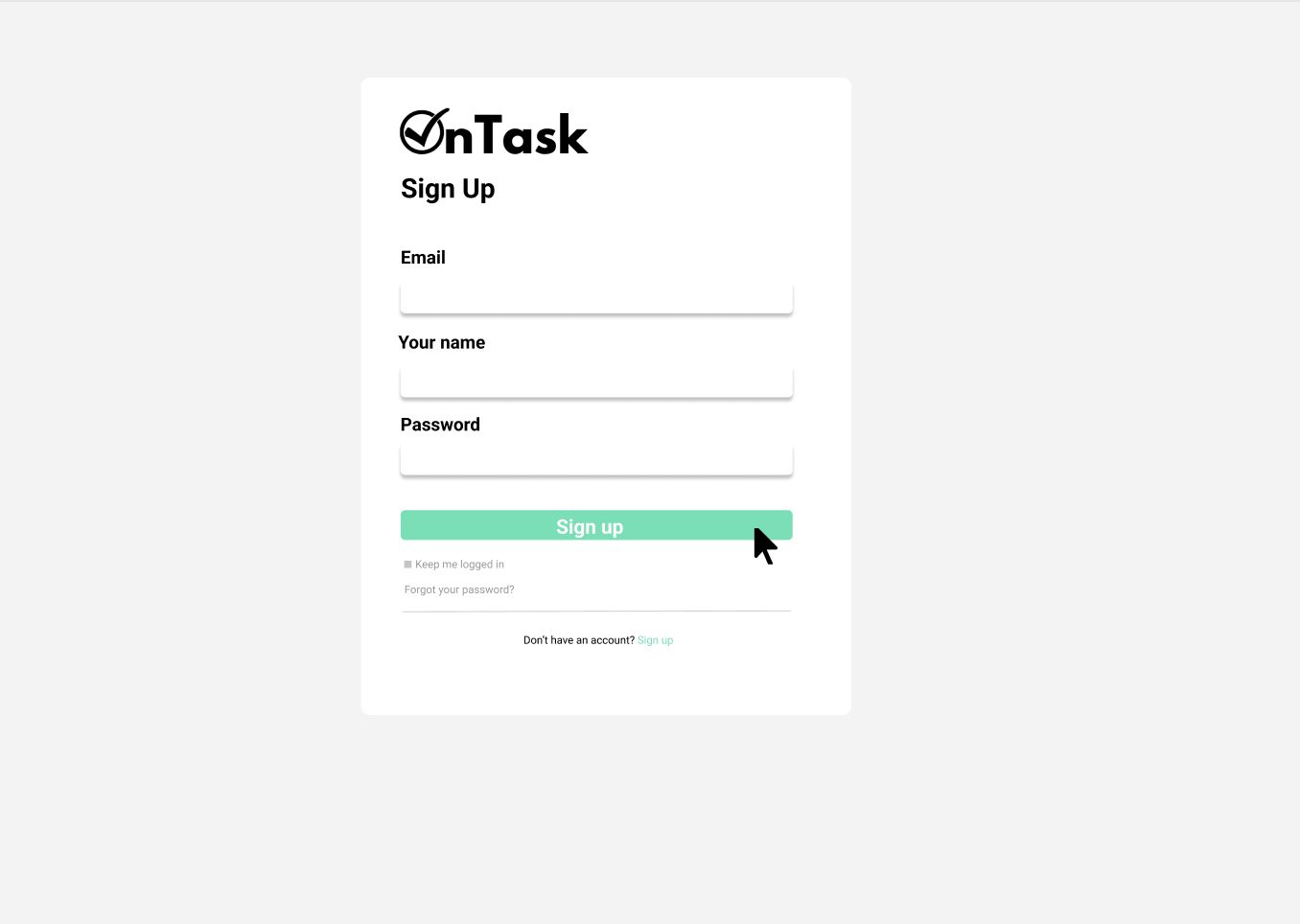
**F.10: Work-Study Intervals, Priority 1:**

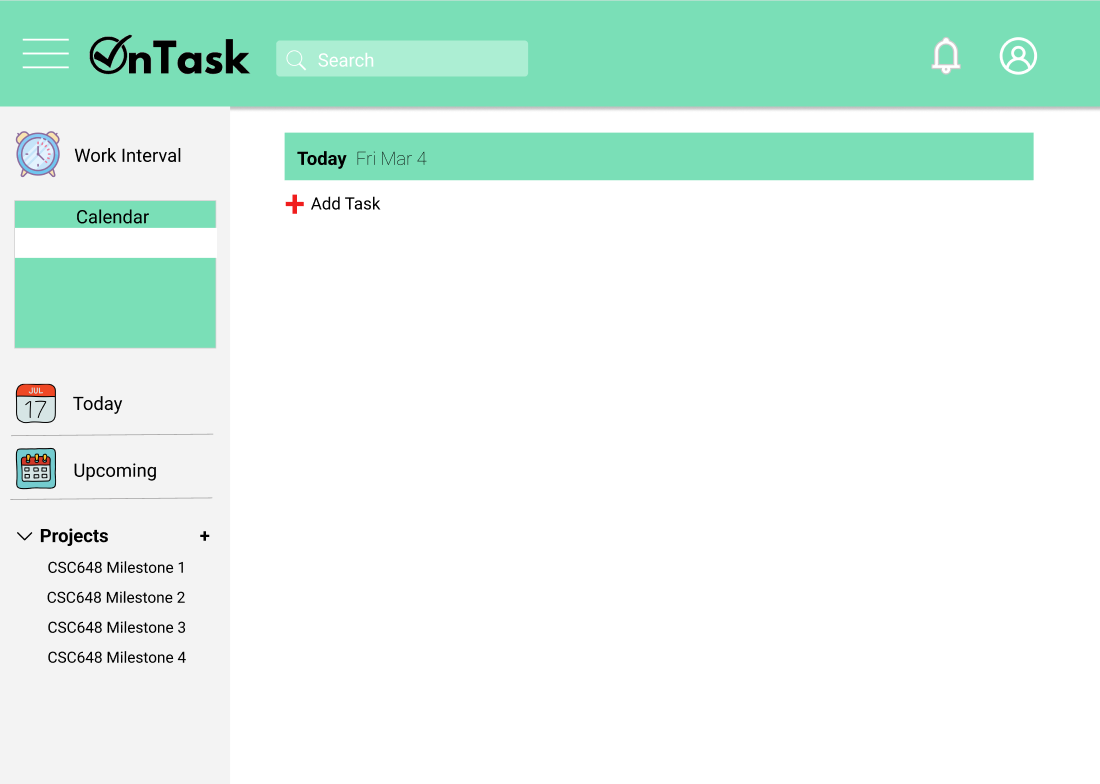
* User sets the amount of time they want to study and decide how long they’d like to study, and how long they’d like to have a break.
* With the information given by the user, the app will set a timer for the user to study.
* Once the timer goes off, it’ll allow the user to take a break for the designated amount they initially requested.
* The intervals will continue until the set amount of time the user wants to study will end. This will allow students like Michelle, teachers like Henry, project managers like Lisa, to work and study efficiently and be able to take adequate breaks throughout the day.

**3. UI Mockups and Storyboards**

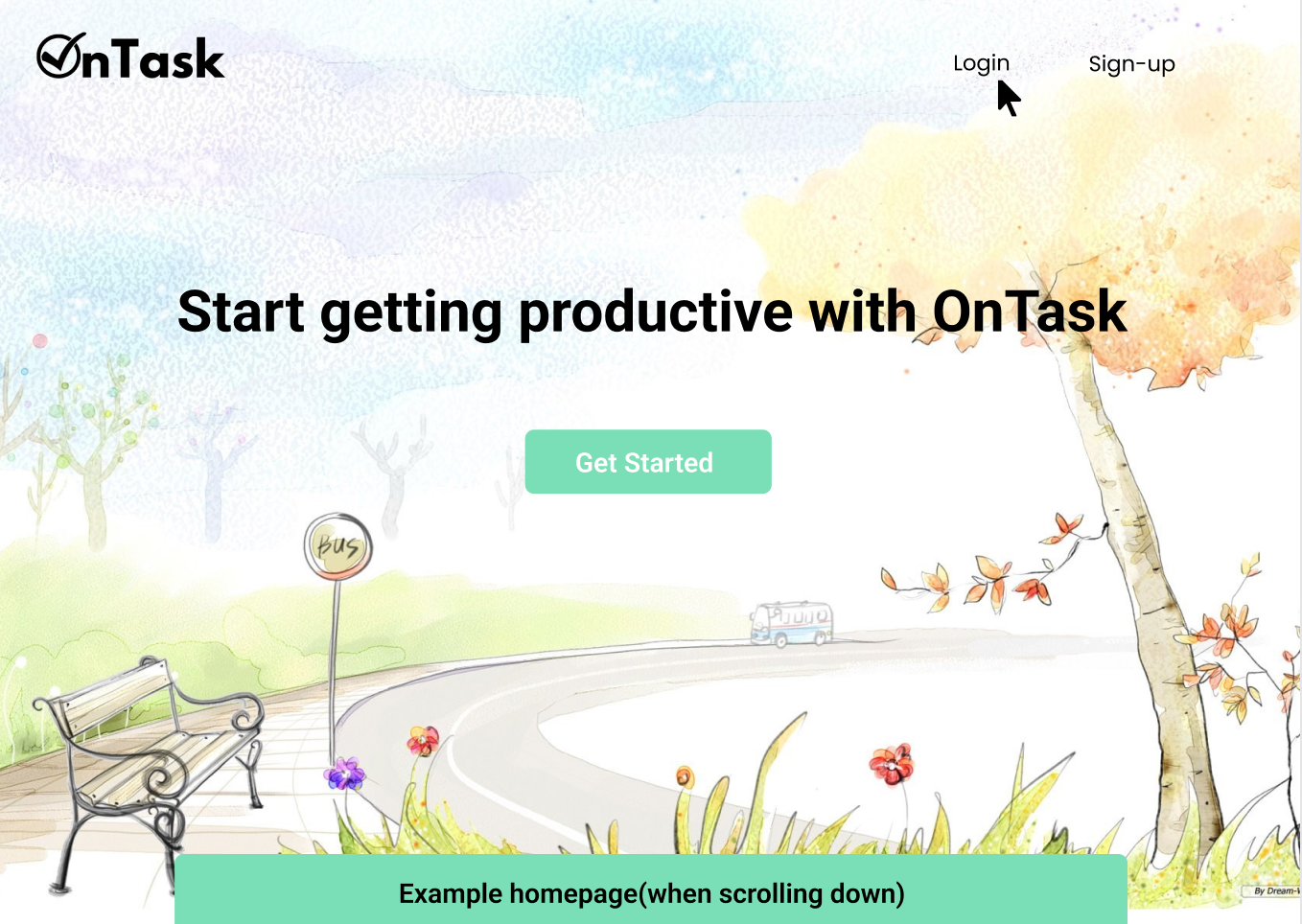
**3.1: Logging In Function**

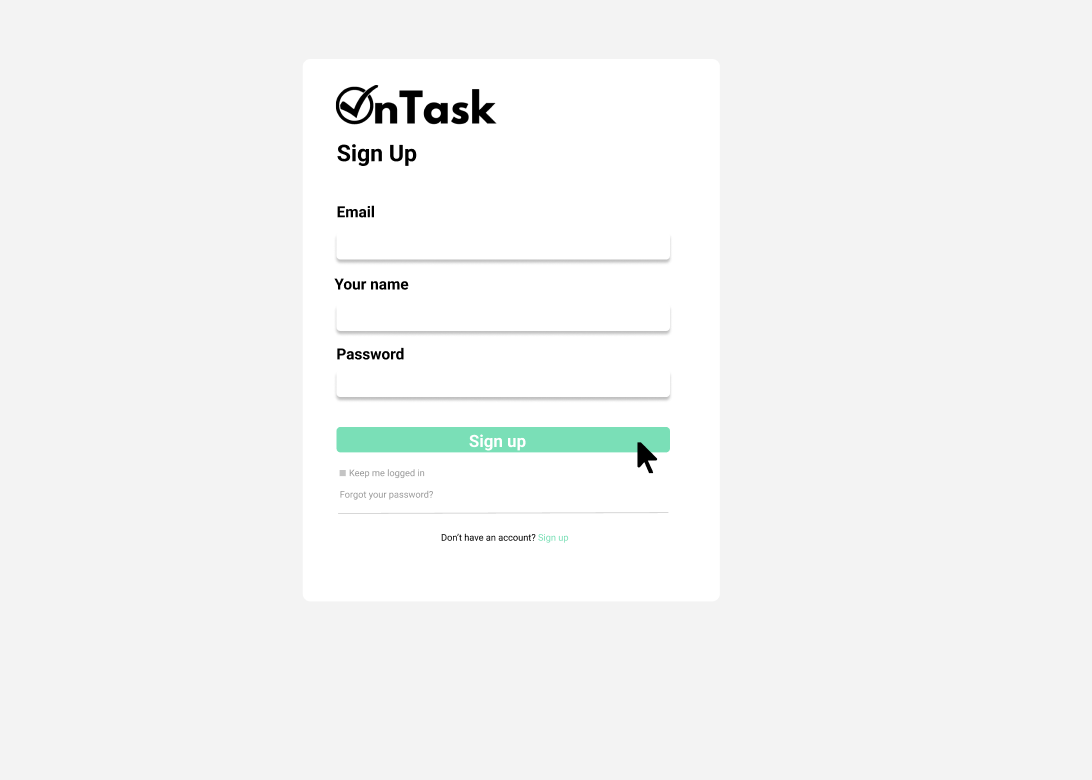




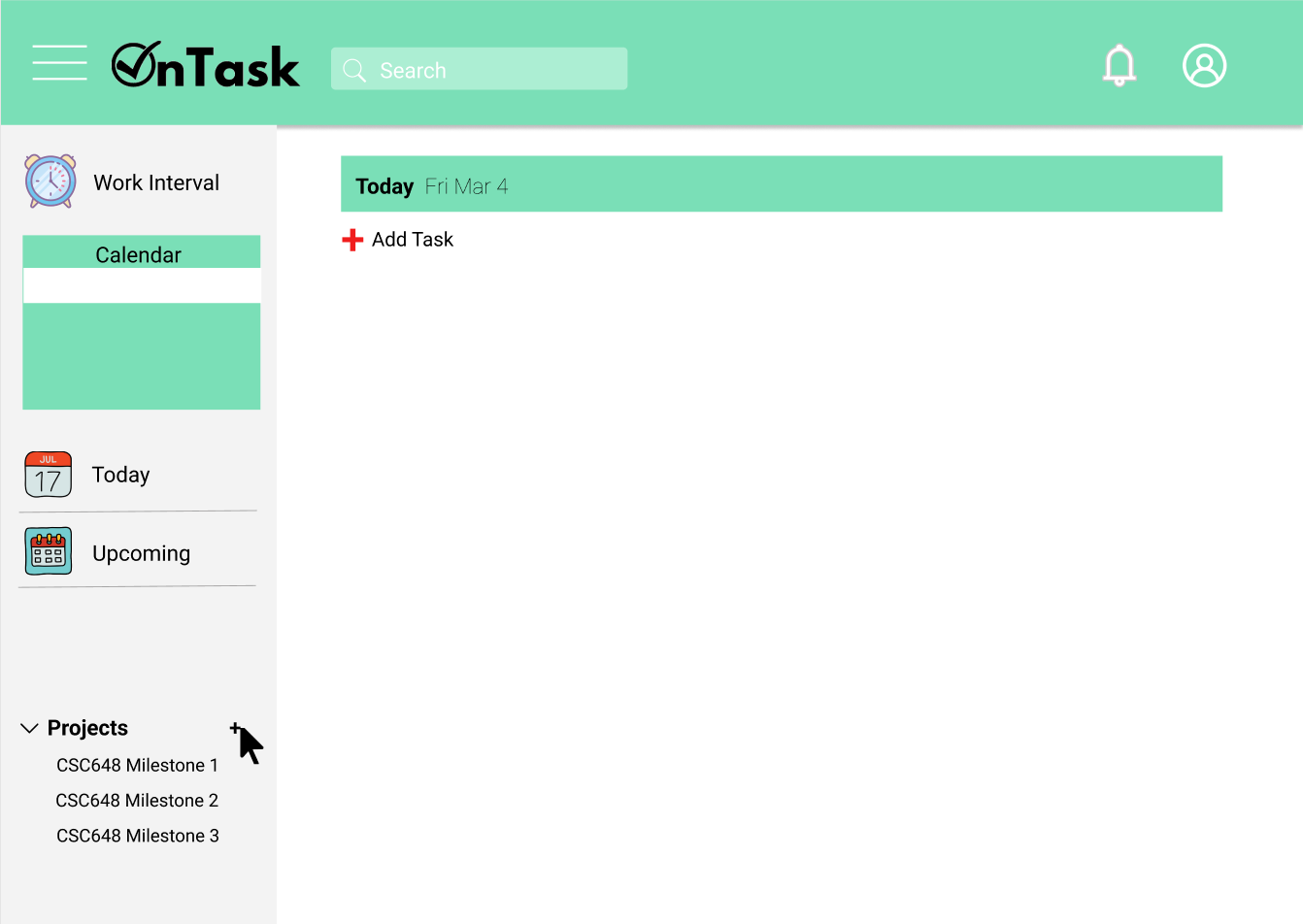


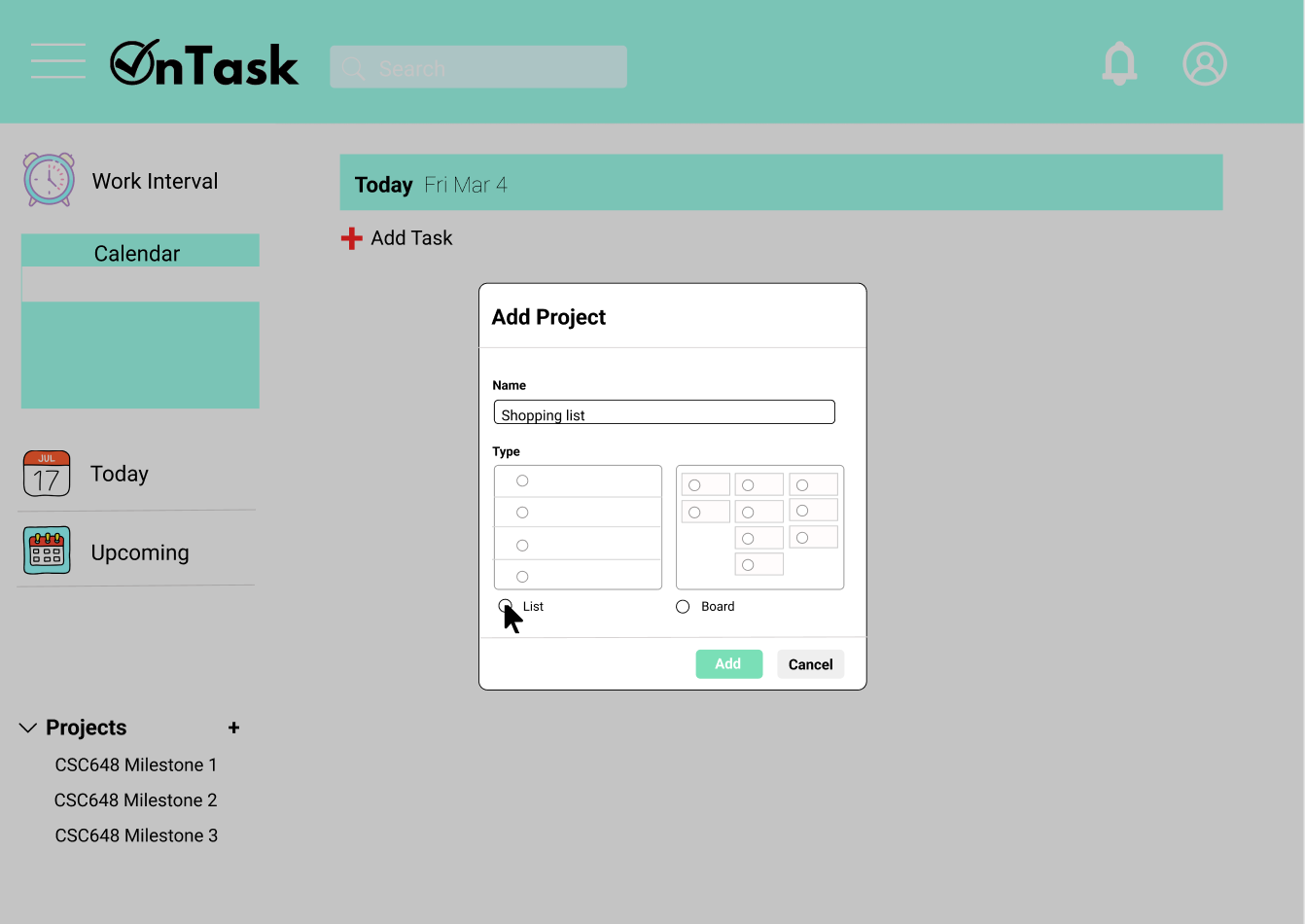
**3.2: Sign Up Function**

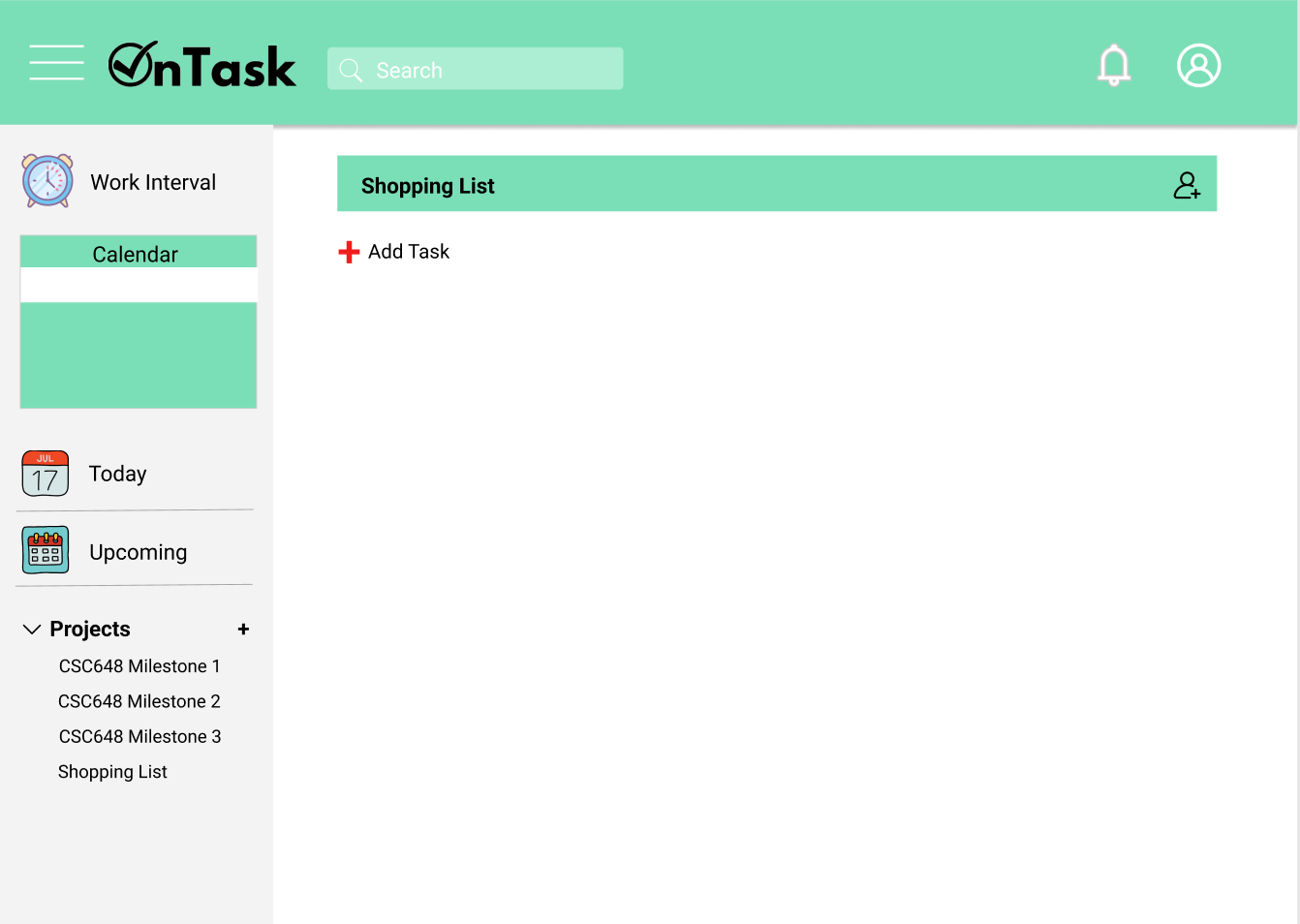




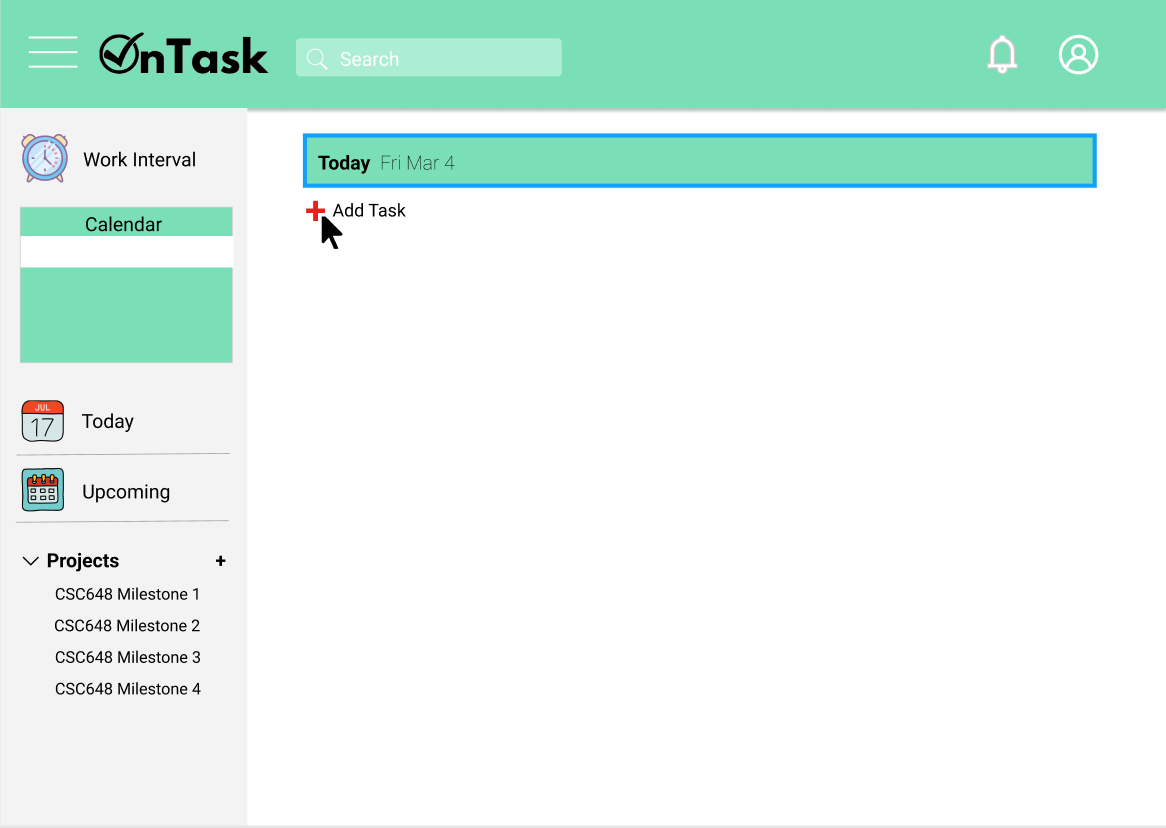
**3.3: Create a To-Do List**

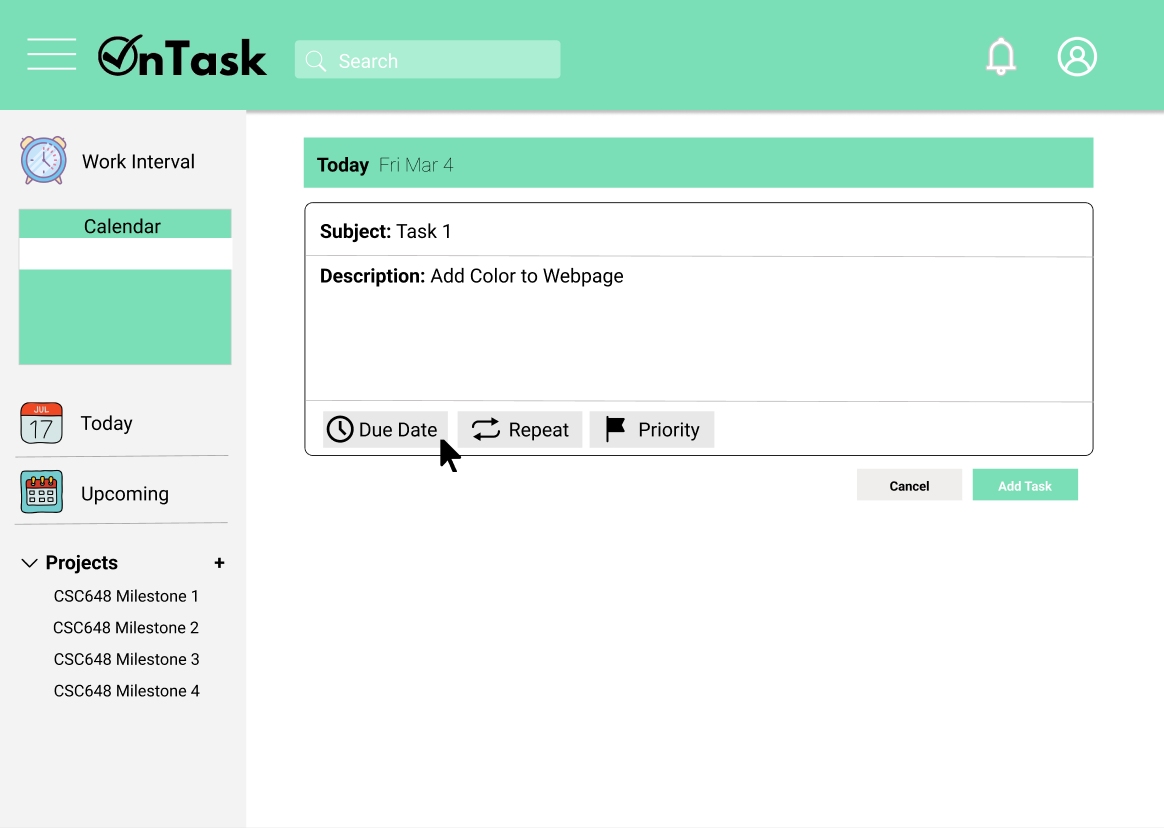
****

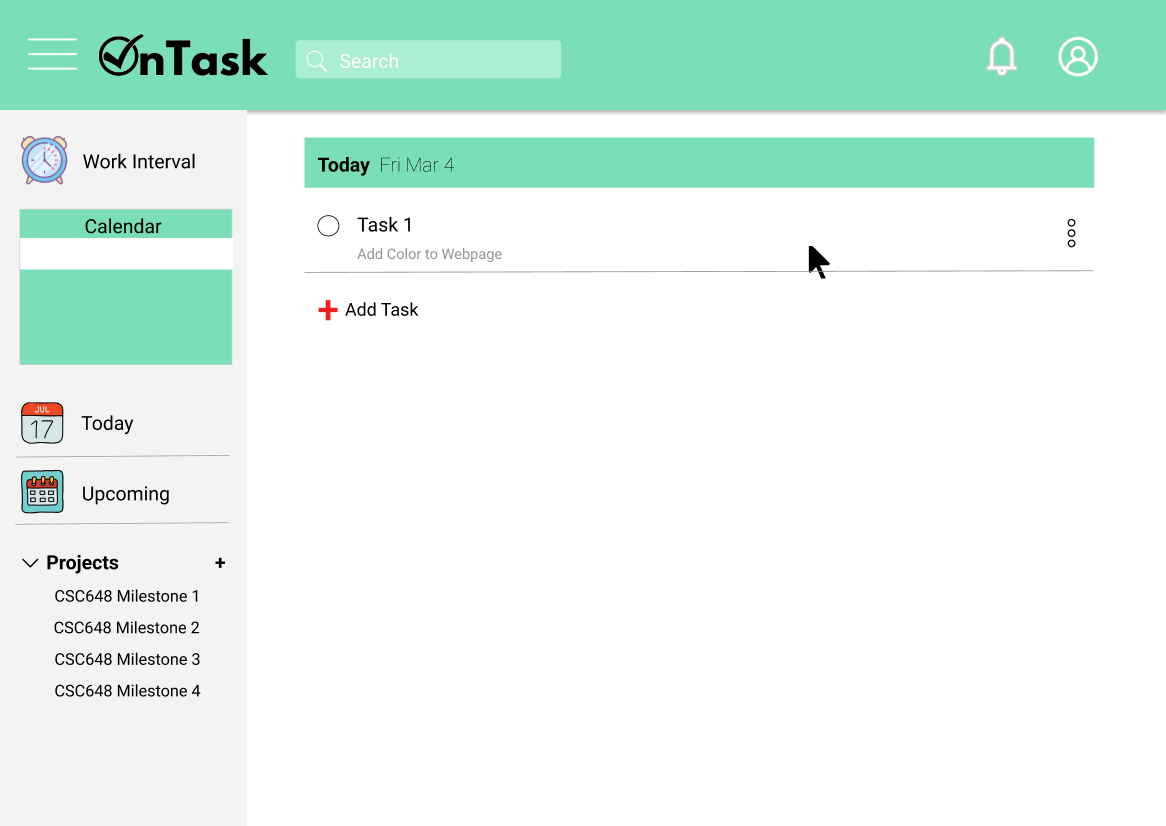
****

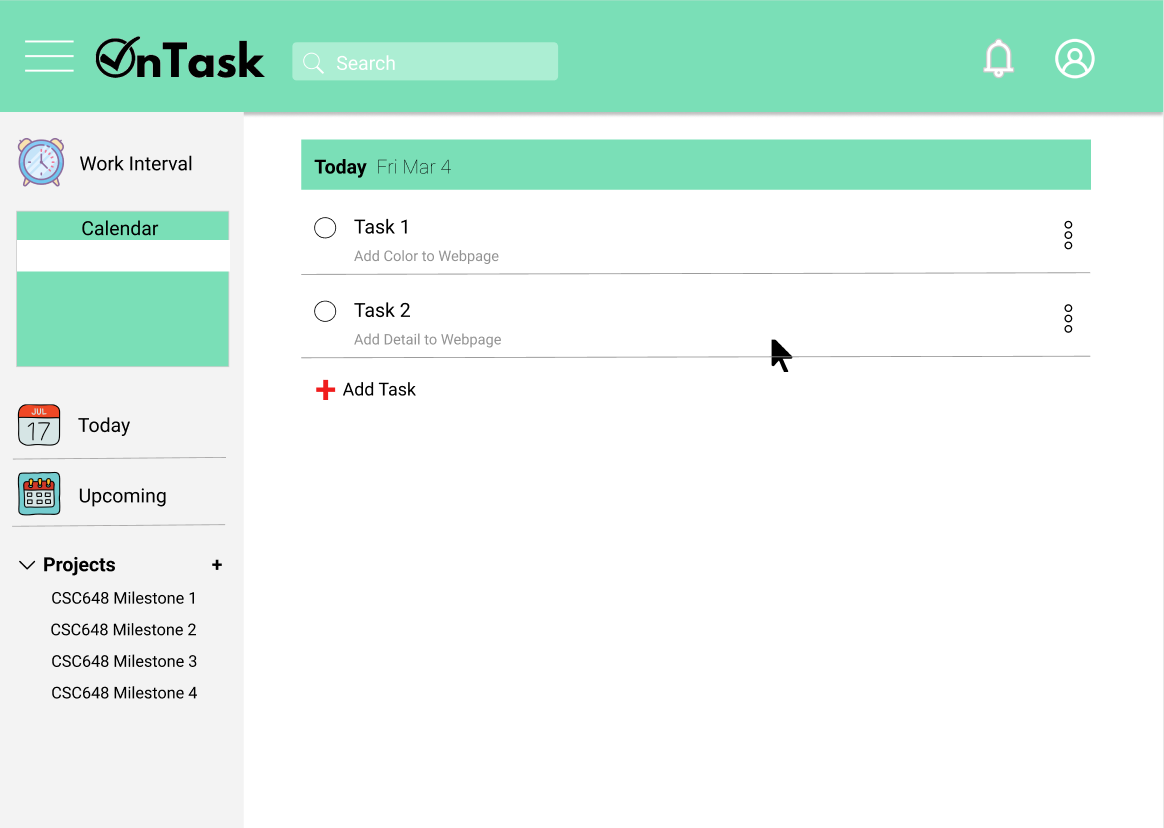


**3.4: Add a Task to the To-Do List**

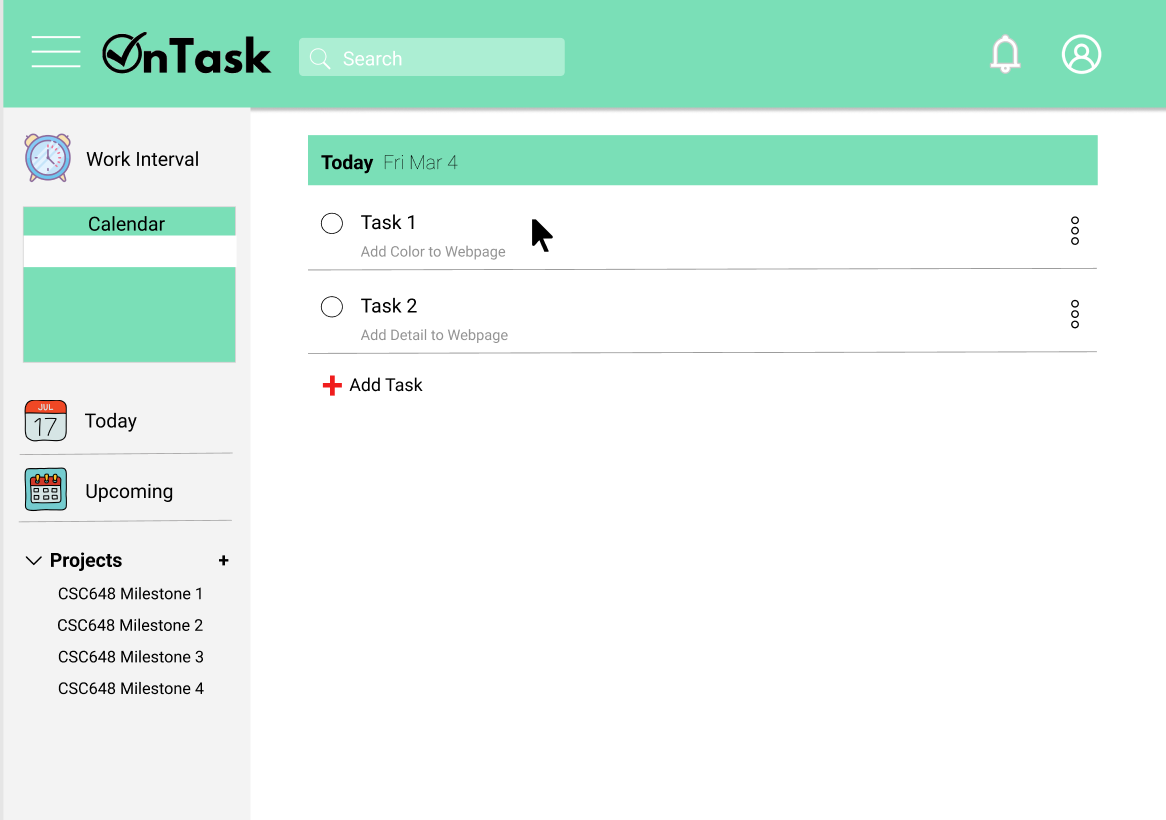


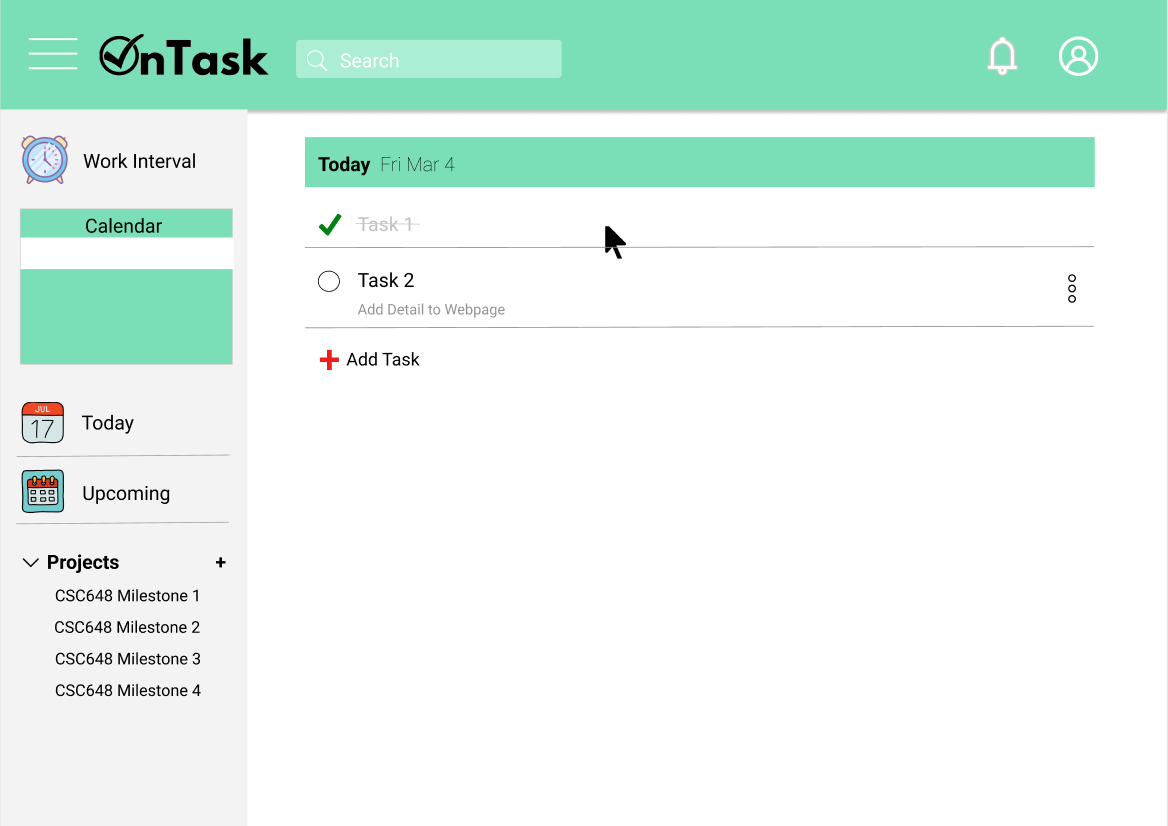




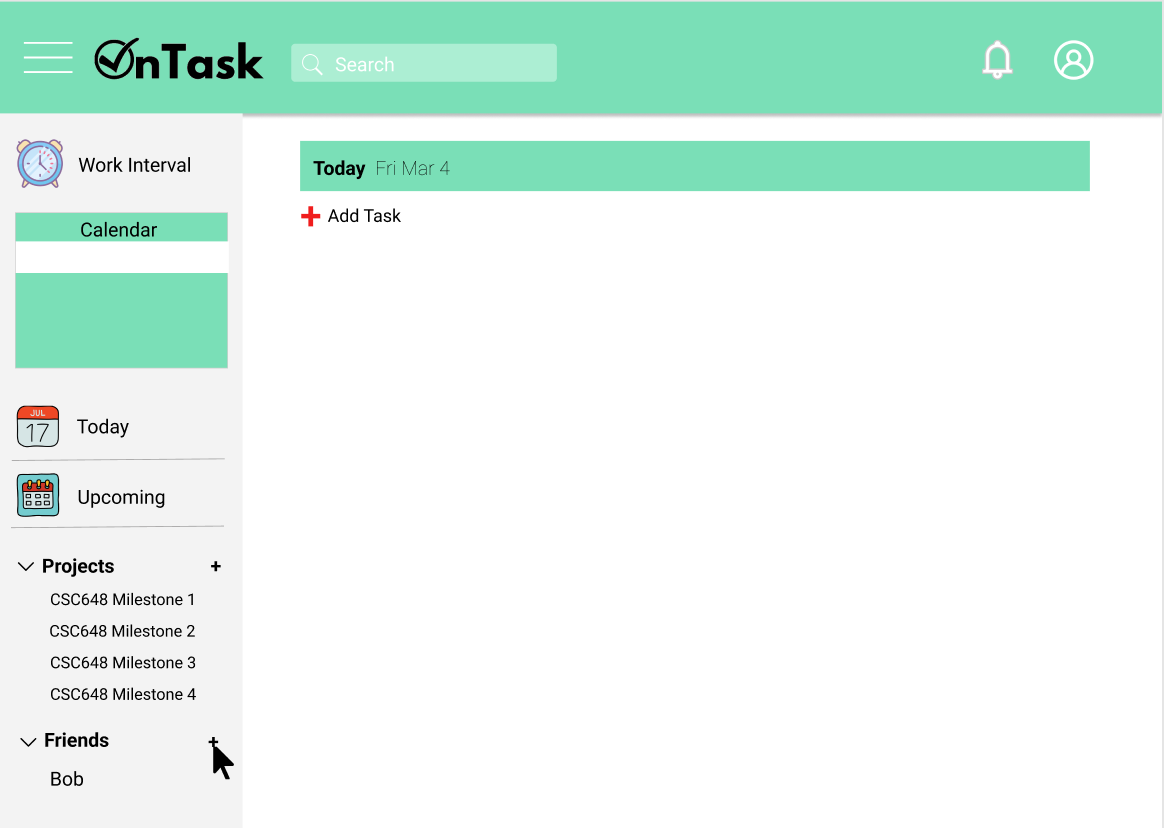


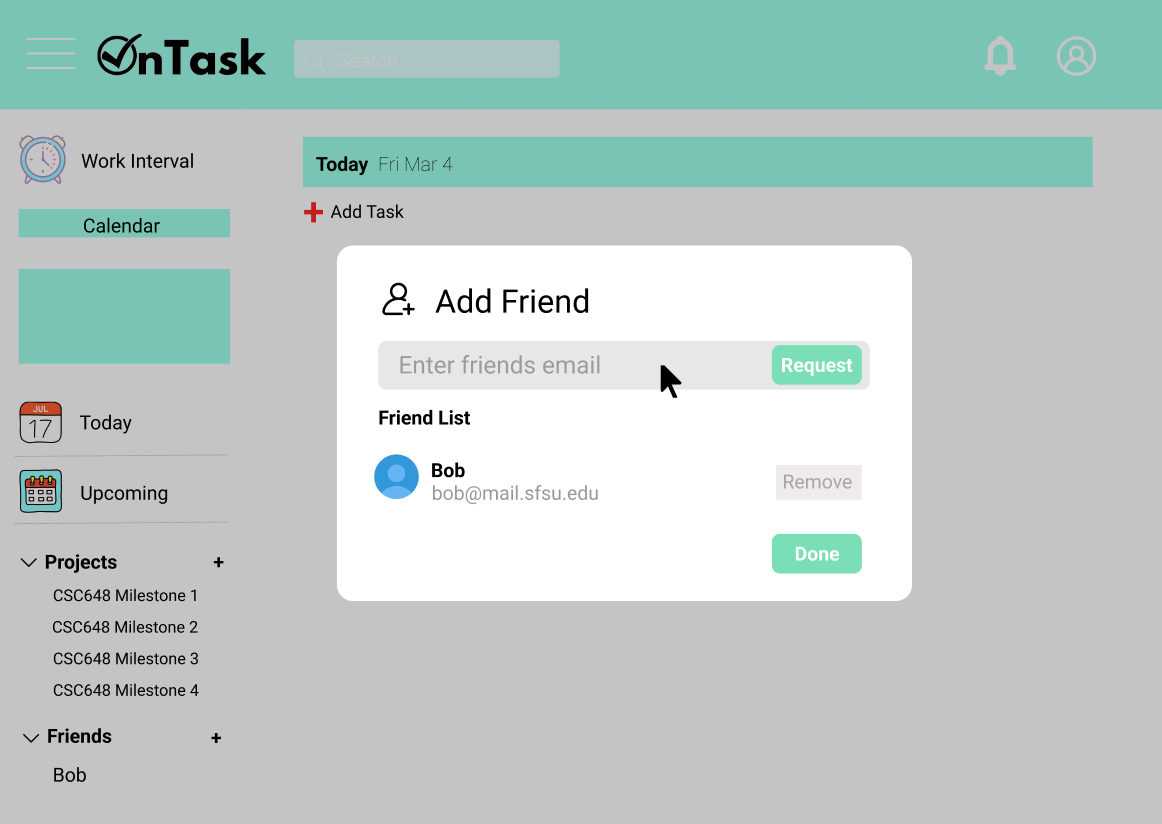
**3.5: Mark a Task as Completed**

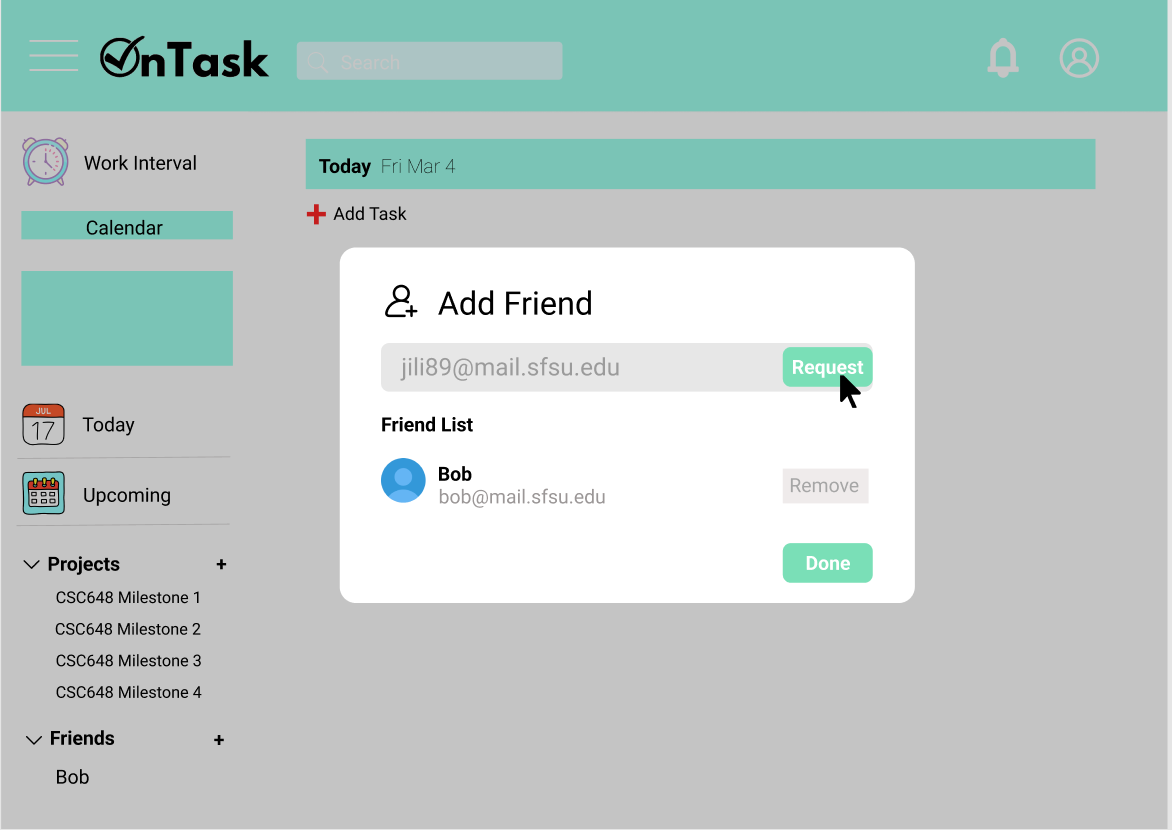
****

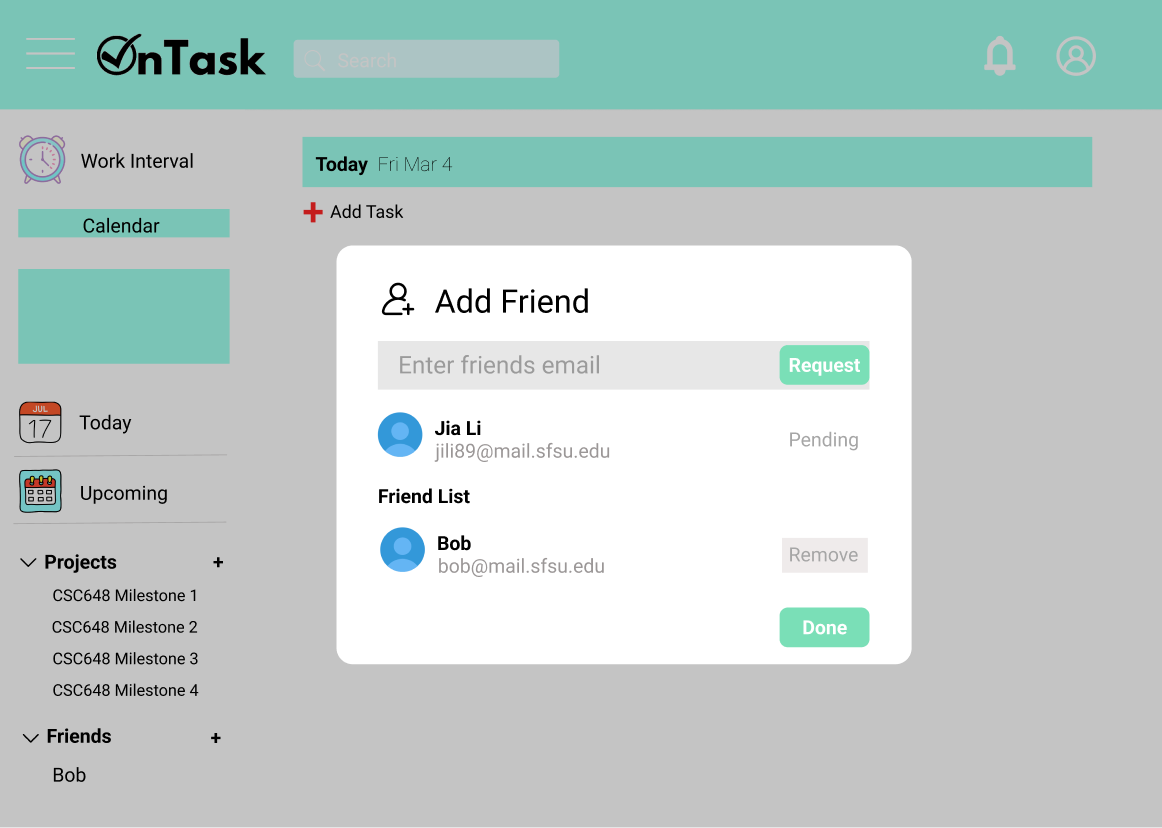
****

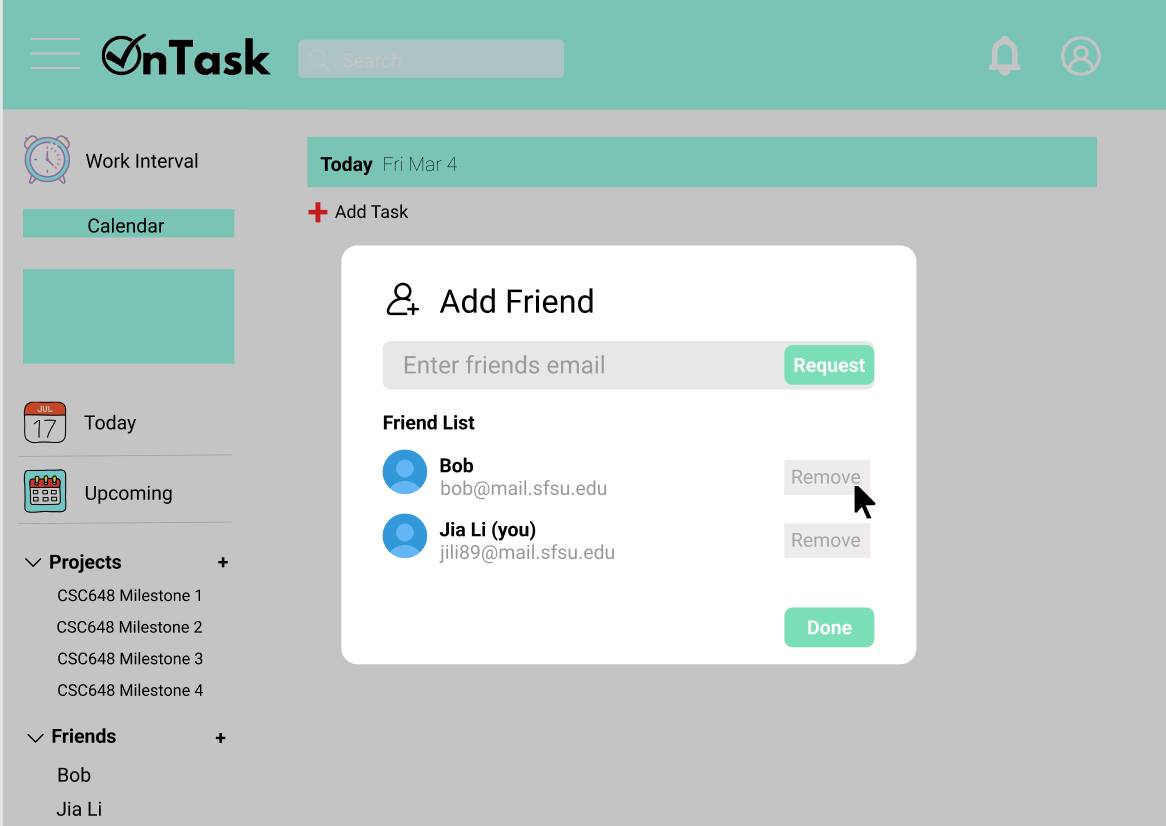
**3.6: Add or Remove a Friend**

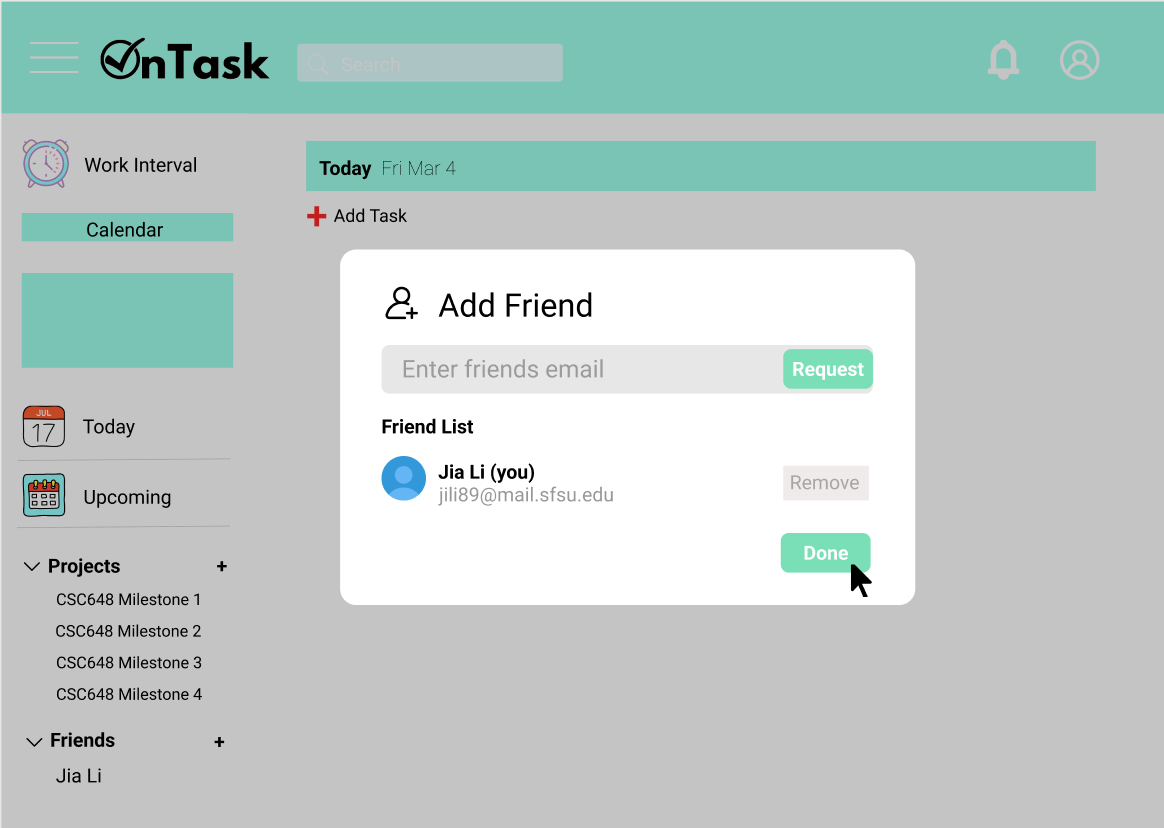
****

****

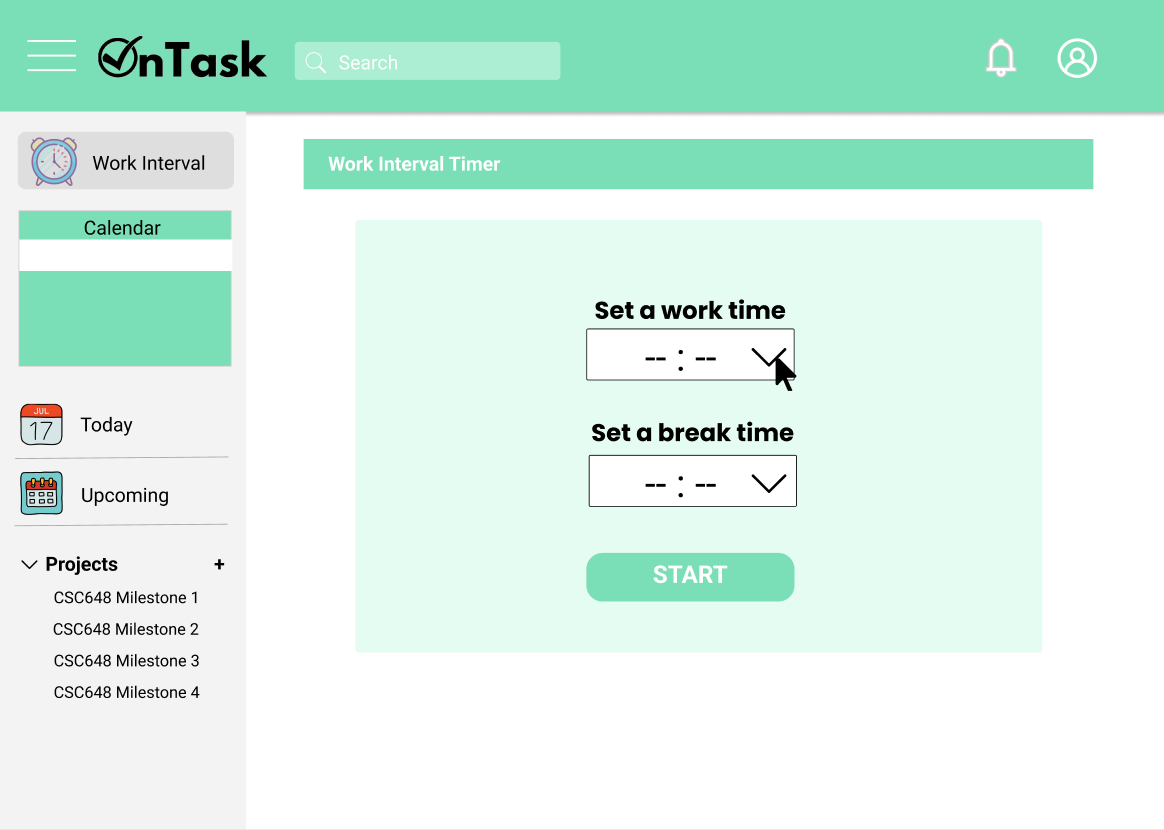
****

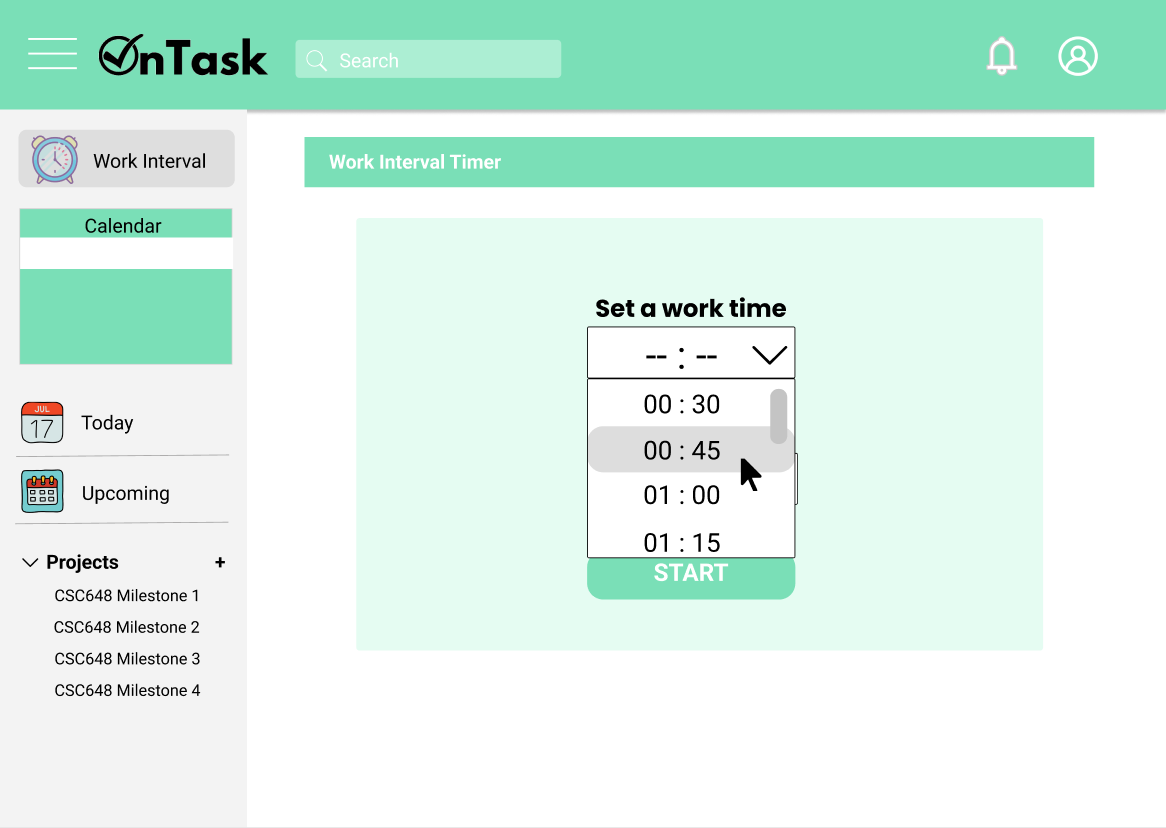


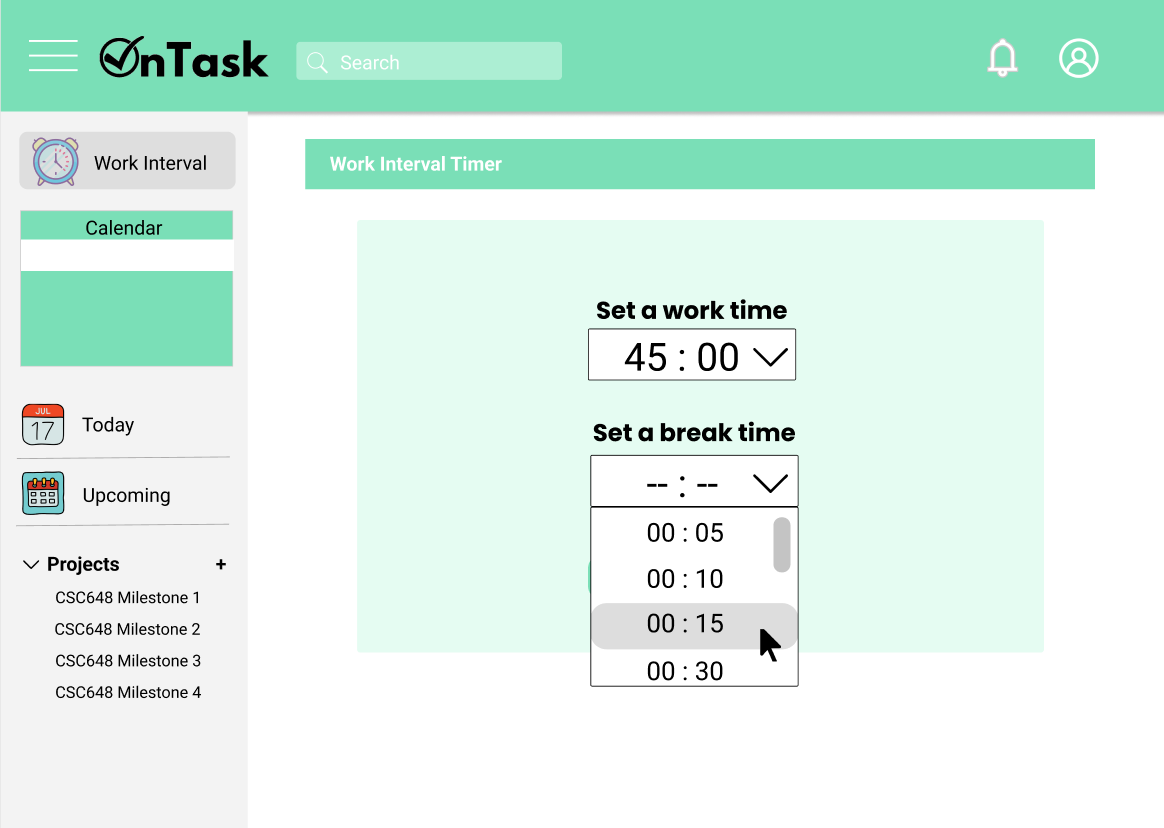


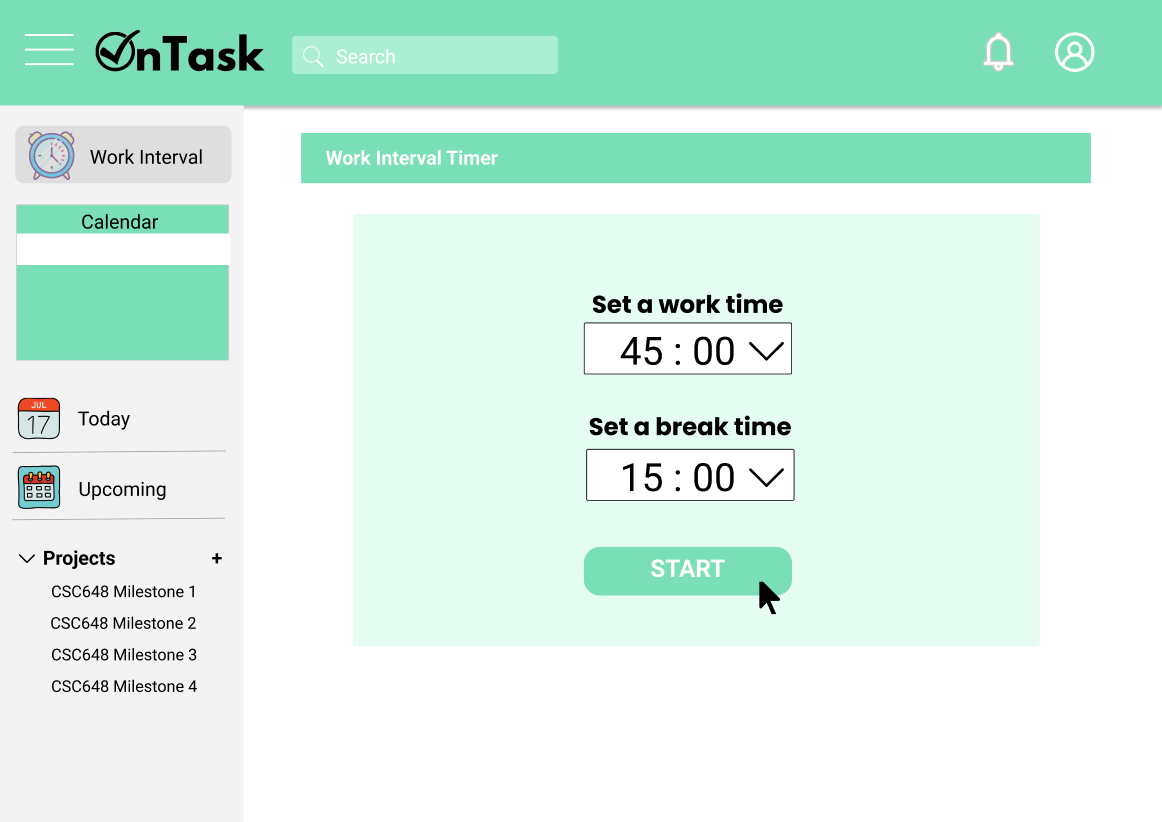


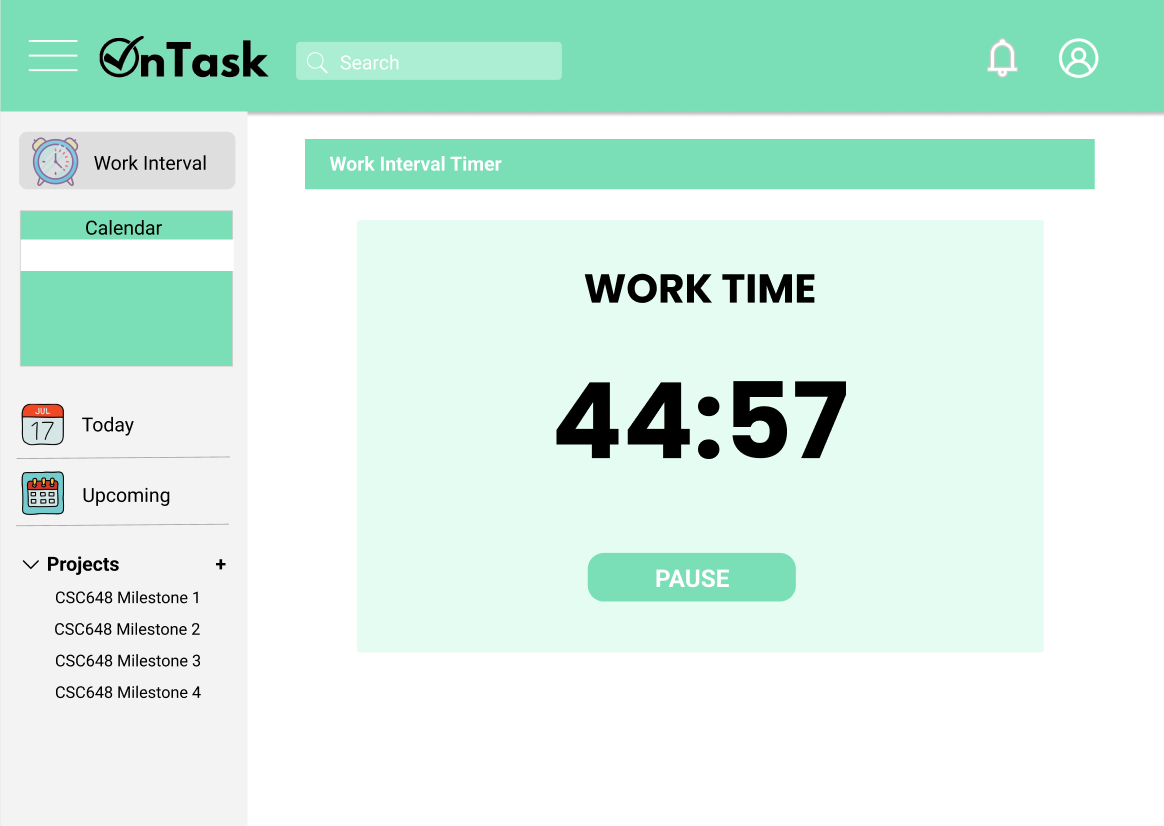
**3.7: Work Interval Function**

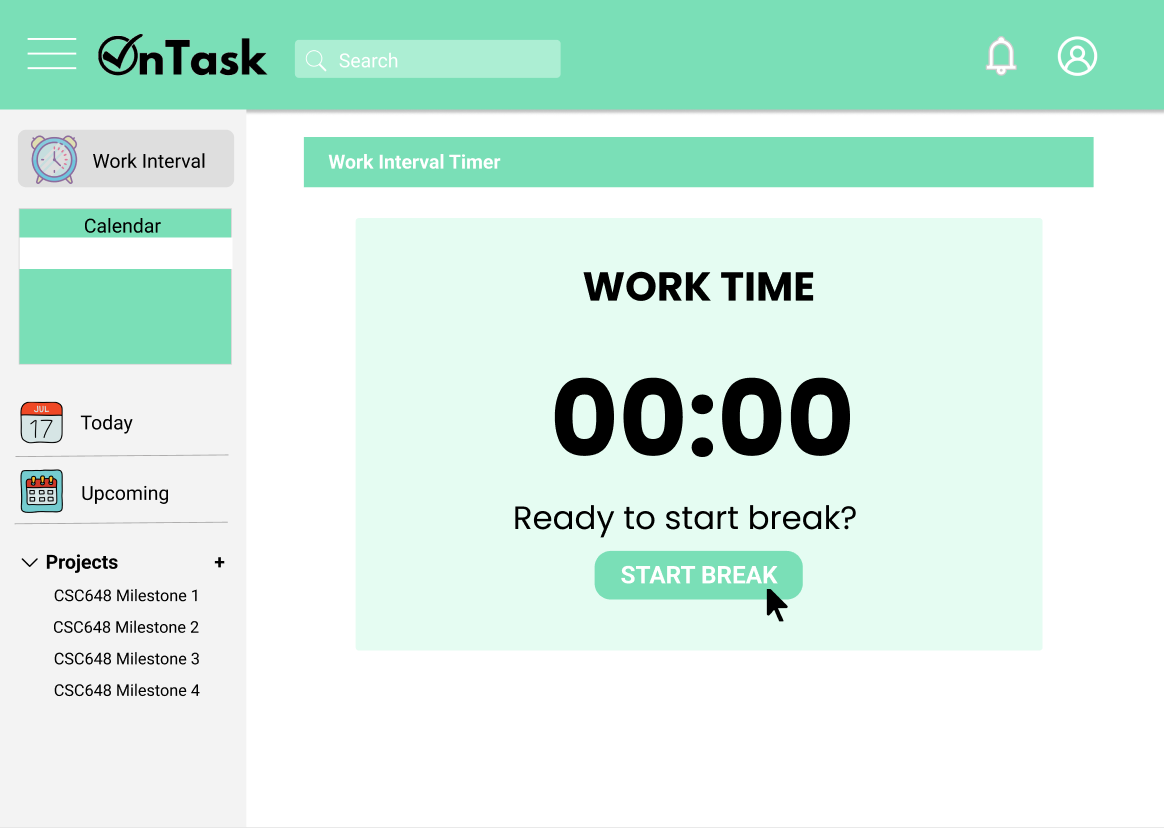


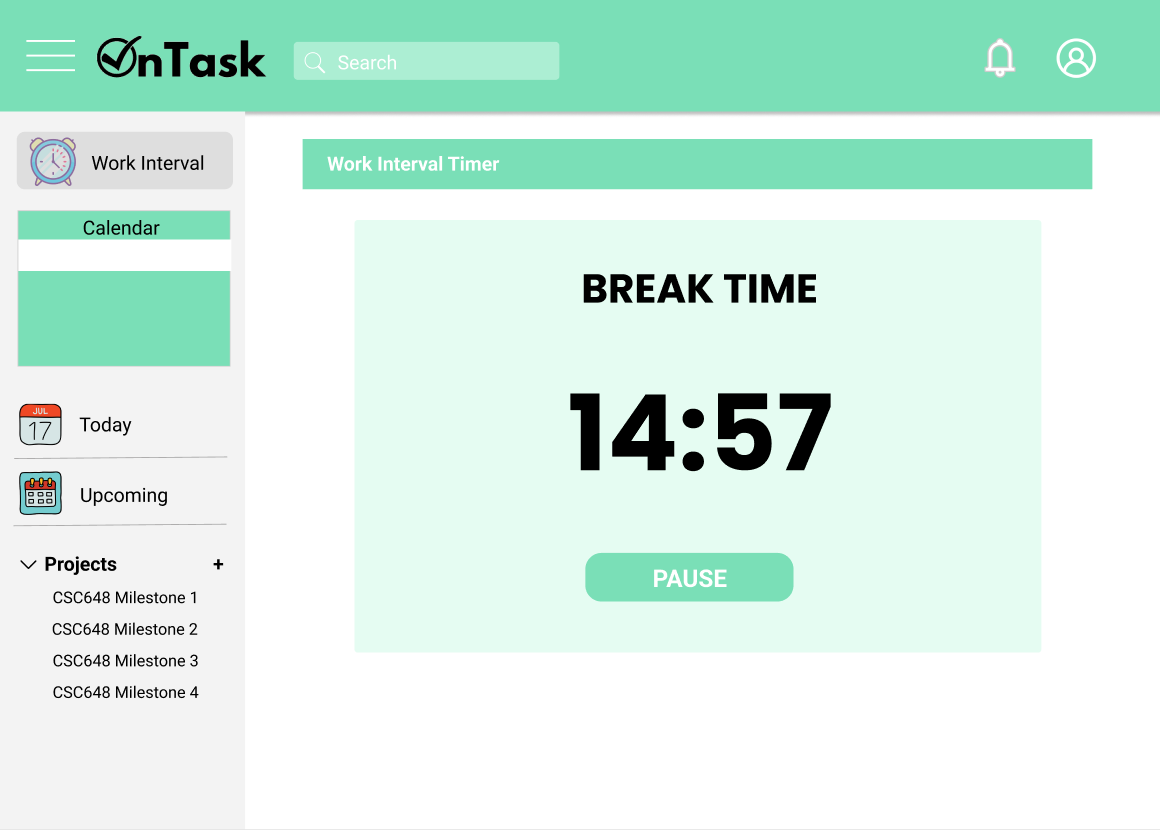


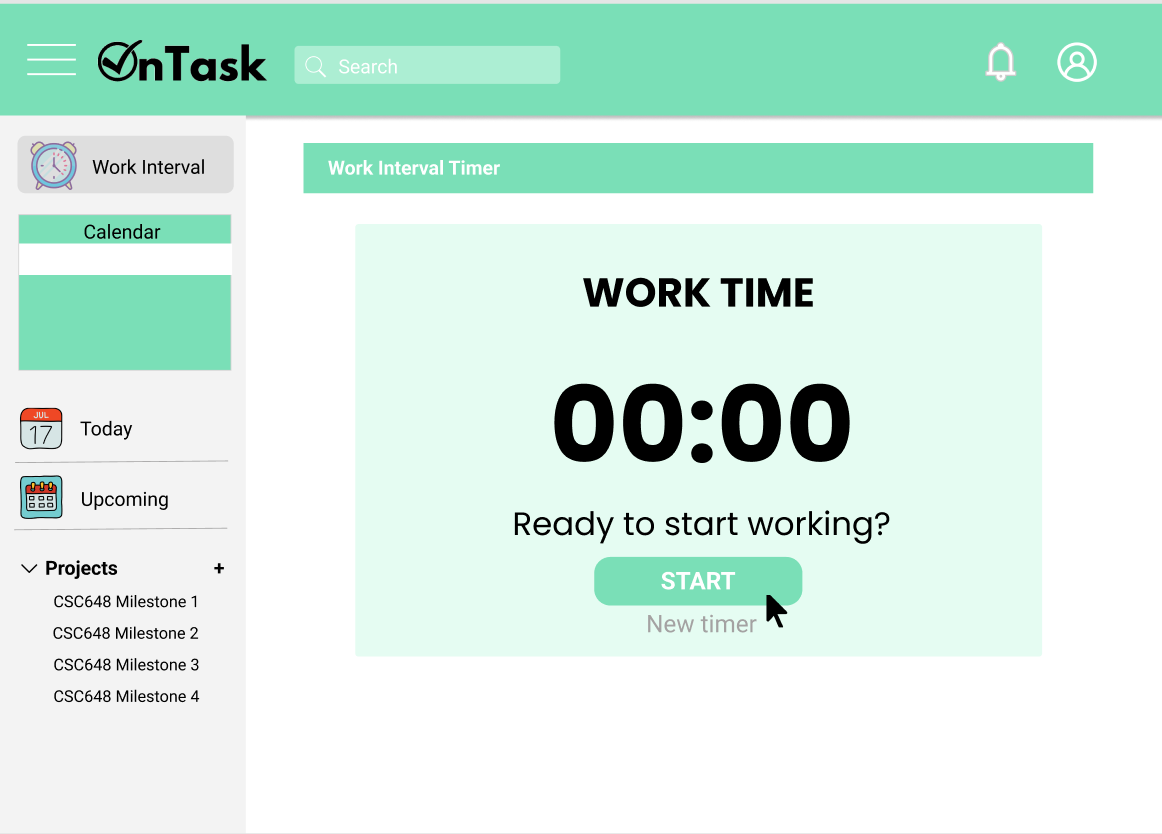












**4. High level Architecture, Database Organization**

**Table 1: Users**

* Name
* Email address
* Password
* To-do list
* Friends

**Table 2: Task**

* Task name
* Task description
* Date created
* Due date
* Completed
* Priority

**Table 3: To-do List**

* Array of tasks
* Category title
* Owner of list
* Shared permissions

**F.1. Sign Out/Log in/Log Out:**

Entries to be added will be the user’s name, email and password.

**F.2. Categorize Tasks:** Entries to be added will be the user’s task’s category (?)

**F.3. Assign Priority:**

Entries to be added to the user’s tasks will be the number in priority.

**F.4. Collaborative To-Do List:**

Entries to be added, deleted, displayed, and searched will be the owner of the to-do list and the collaborator’s IDs of the to-do lists.

**F.5. Mark Task As Complete:**

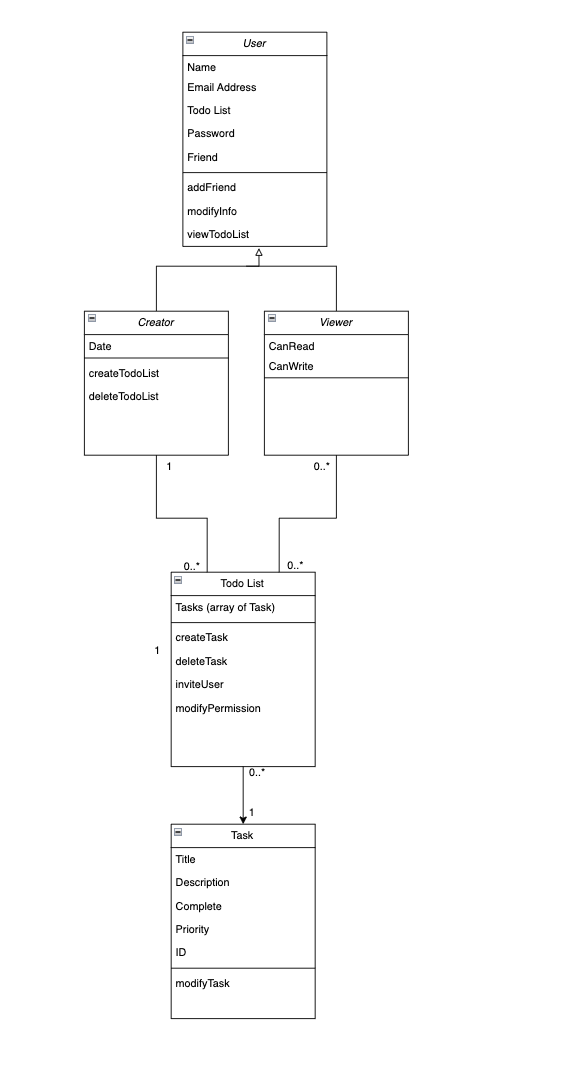
Entries to be added, deleted, displayed will be if the task is or is not complete.

**F.6. Work-Study Intervals:**

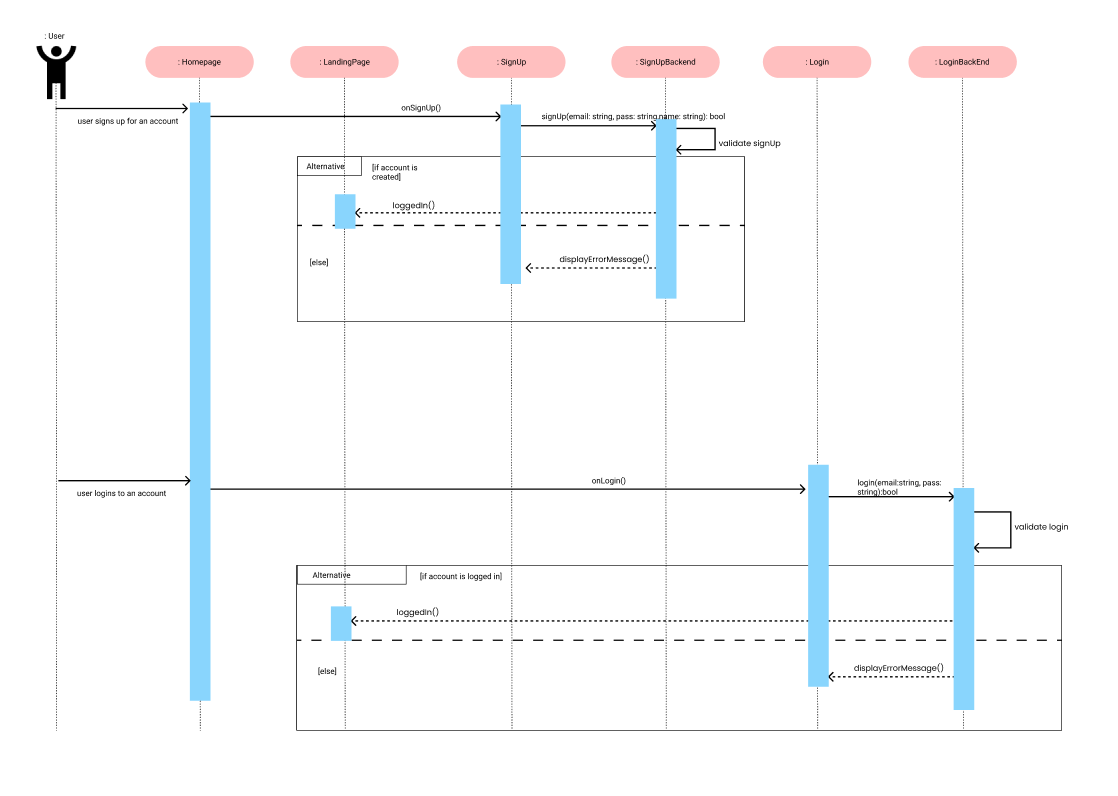
Entries to be added and deleted will be the number of minutes for work time and break time.

**5. High Level UML Diagrams**

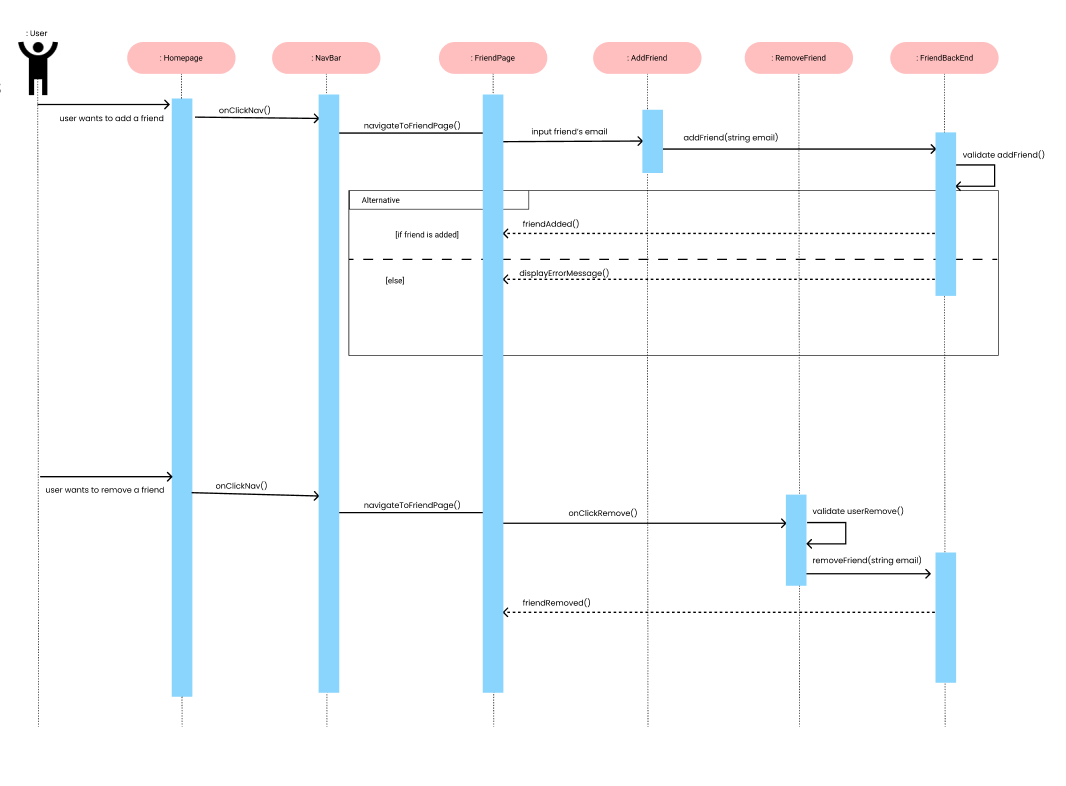
**5.1. Class Diagram**

****

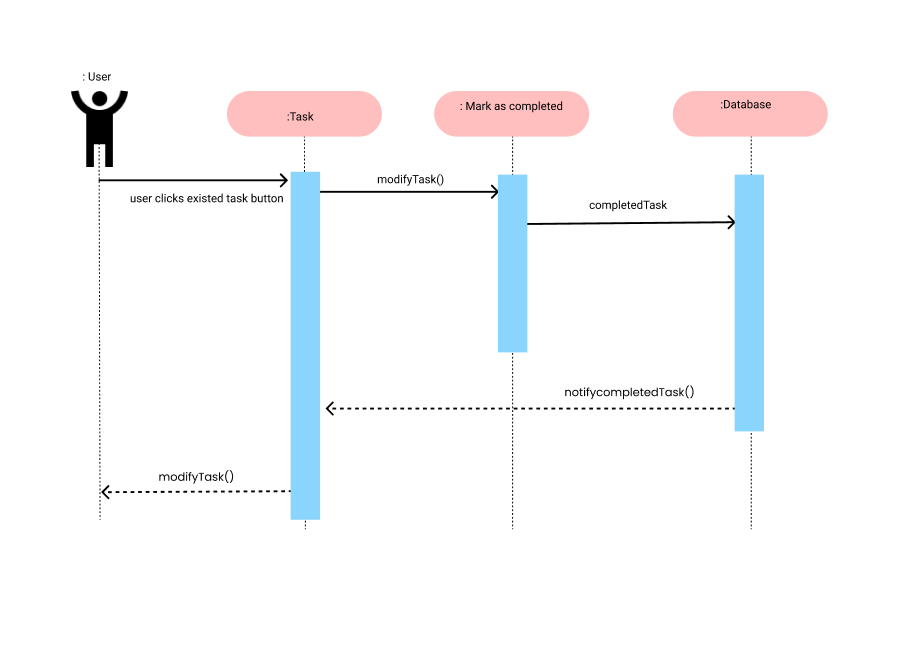
**5.2.1: Sequence Diagram for Login and Signup Function**



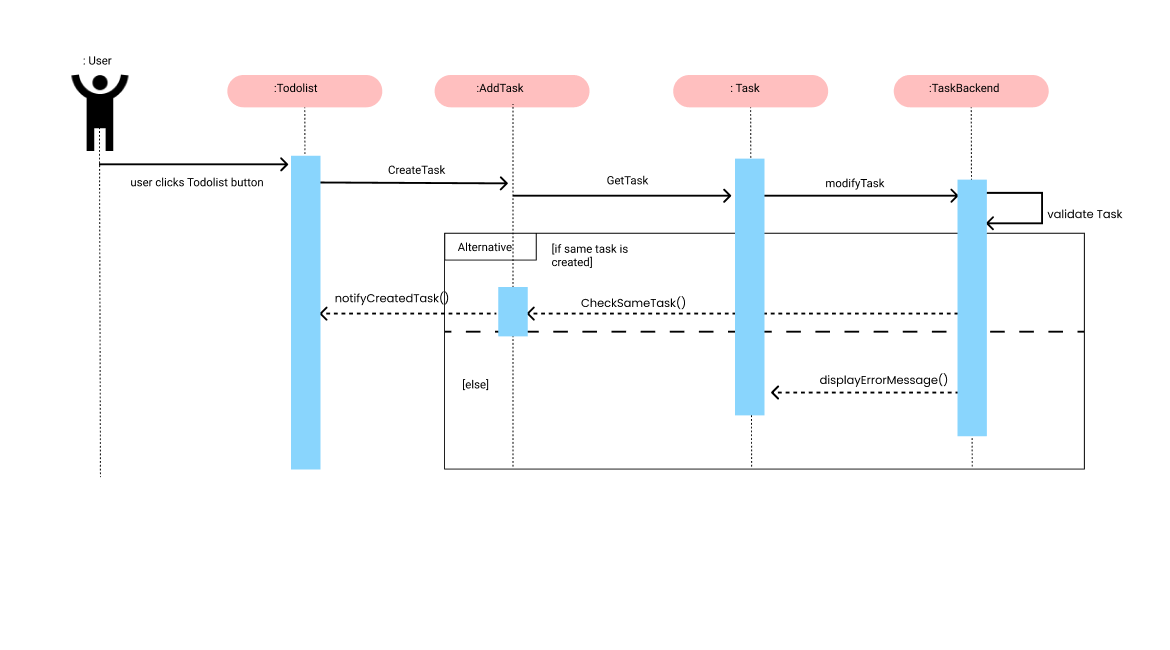
**5.2.2: Sequence Diagram for Adding and Removing Friend**



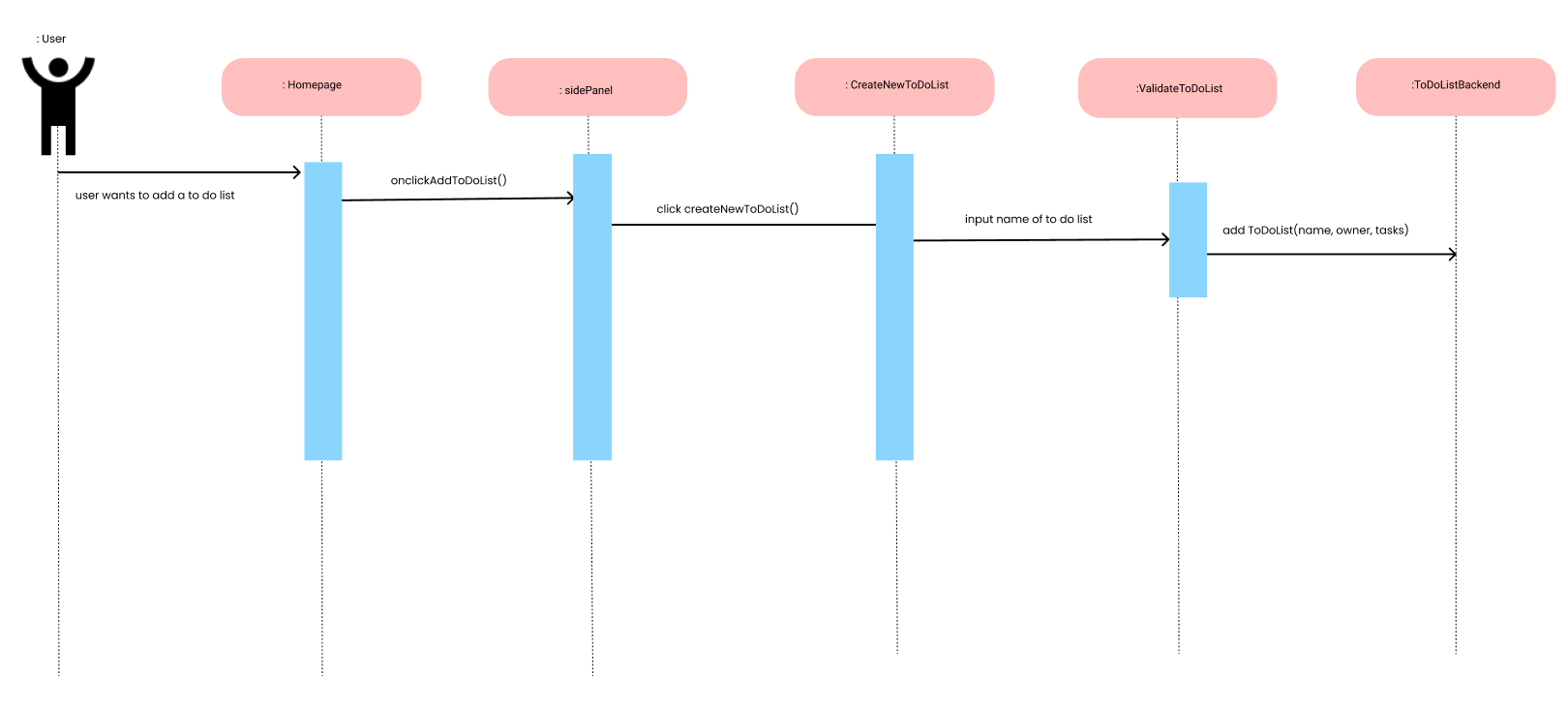
**5.2.3: Sequence Diagram for Marking Task as Complete**

****

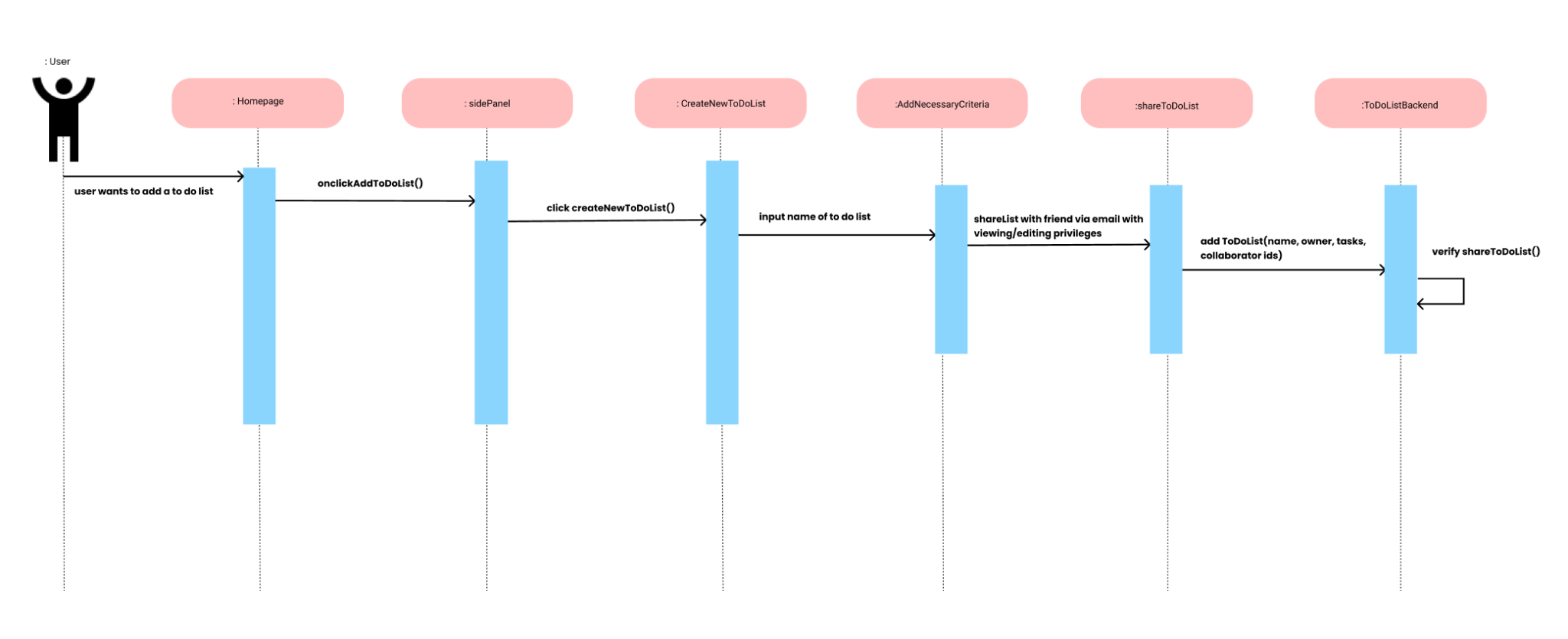
**5.2.4: Sequence Diagram for Adding a Task**



**5.2.5: Sequence Diagram for Creating a To-Do List**



**5.2.6: Sequence Diagram for Creating a Collaborative To-Do List**



**6. Identify actual key risks for your project at this time**

1. **Skills risks and mitigation plan**

Our skill levels for React, Node, Express, MongoDB and Redux, which are important parts of creating our project, vary on average of 2.5 on a scale of 5. To ensure that we become more familiar, we’ve created a study plan to cover all concepts and languages by sending material (videos, documentations, notes) on each concept/language and making sure we set a deadline to familiarize ourself with it. Afterward the deadline, we’ll go over what we learned during our team meetings.

Ex: Front end team to learn Redux by March 9th and will discuss their findings during the team meeting at 2 pm on March 9th.

1. **Schedule risks**

Our team has a schedule to follow once every milestone is released. As a team, we create soft deadlines for when certain parts of our milestone should be due and assign each team member with tasks and make sure to split it up evenly. Our team uses a Google Doc, where our team lead splits up our tasks and lists out every detailed task for each member. If changes happen, it is updated on the Google Doc’s to-do list.

1. **Teamwork risks**

Everyone meets regularly, and if not, members tell us when they’re unable to make it and the scrum master relays the information that they missed during the meeting. Everyone on our team keeps their pace. If not, I check the milestone document regularly to see if someone has not completed their part. If someone hasn’t done their part, I can check in with the team member and see if they need more support and when they think they’ll have their part finished. I usually try to ensure that team members don’t procrastinate until the deadline.

1. **Legal/content risks**

Our team does not see any legal/content risks as of the moment. If there are any content risks, it may be using copyrighted images, but in order to prevent that, we will make sure we use stock/royalty free images.

**7. Project Management**

During scrum meetings, we first discuss what we’ve completed and where our progress is since the last time we met and notes are taken down to make sure we’re clear on what each of us have completed. Afterwards, we discuss what still remains for us to complete and discuss how we should move forward.

For milestone 2, we started off by looking at the milestone 2 document to see what needs to be done. Our team lead creates a to-do list on Google Docs for every milestone and splits up each section evenly while sharing her screen and each member assigns themselves to each task. This to-do list is updated with new changes by every member, marking tasks as complete when finished or adding extra notes to track their progress. As well as the end of each meeting, we take notes and share them with the team to see what deadlines are coming up (ex: front-end team’s wireframe designs are due Monday, March 7 and high fidelity designs are due Monday, March 14th).