

## Requirements Workshop

0.

a. Team Name: **Group 0**

b. Nishesh Saikrishna ([nisheshsai9@vt.edu](mailto:nisheshsai9@vt.edu))

Peter Murphy ([pmurphy26@vt.edu](mailto:pmurphy26@vt.edu))

Benjamin Hurt ([benjaminhurt@vt.edu](mailto:benjaminhurt@vt.edu))

Joel Buba ([joelb1@vt.edu](mailto:joelb1@vt.edu))

c. Taskboard Manager

1. Non functional requirements:

1. Must use Java (Implementation/Constraints)
2. Server response time is less than 2 seconds for most accesses (Performance)
3. Use OOP so that the functions and features can be easily added and removed based on user feedback/requirements (Supportability)
4. When adding a task there should be a 24 hour MTBF (Reliability)
5. The system should allow for easy, self explanatory use with labels and a clear process for how to add a task (Usability)

2. Functional requirements:

1. Should let users add tasks to the taskboard
2. Should allow users to assign certain users to certain tasks
3. Should be able to mark tasks as complete/deleting
4. Categorize tasks as not started, in progress, and completed, or by custom groups
5. Can be able to add notes/smaller details to tasks

3. Task:

For Non-Functional requirements: Make sure everyone has java installed on their computer, Make sure server being used is fast, increase text size and use a clear font, add error checking to minimize failures, and add proper documentation to help users use the product

For Functional requirements: Create an add button to add tasks on taskboard, Allow for an option within each specific task to add a user to the task, add buttons allowing the user to mark a task as completed and delete a task, allow users to group tasks, add an option to add notes within a specific task

4. User Stories:

- a. A small software team is using the app to manage the tasks they have to complete to finish their product. The team manager should login to the app and add the other team members to the taskboard. From there they are able to add tasks to the board, assigning them as a general task or assigning them to a specific person(s). The other members of the team should be able to login to the app, be able to select the new taskboard they have been added to, and see their own tasks along with everybody else's. There should

be the option to change a task to in-progress, and then change it to completed once they have completed the task.

- b. A classroom group has to work on a software project together for their semester credit. They take the project milestones and break them into tasks that each member can work on. They will be able to see the overall project progress by looking at categorized completed tasks, as well as see any notes on milestones that other members have left.
- c. A solo developer is working on an individual project and needs good organization to be able to work through towards a final product. He uses a taskboard to map out the ideas and features he wants to add, as well as grouping them by certain milestones and difficulties. As he works, he goes one task at a time and checks off his completed tasks. This gives him one area at a time to focus on and complete.

#### 5. Examples of risk:

- 1. Lack of security can be a potential risk to users, with their data being accessed unknowingly. We can add a user authentication system in order to mitigate this risk.
- 2. Code issues/bugs are another risk that may be encountered and can easily cause usability issues. Making sure that each part of our code has been thoroughly tested and making sure that bugs don't appear when interviewing/allowing clients to use the app to avoid this from happening in the final product will be key.

6. We would interview our clients and learn what their preferences for the product are, including information about how the user interface should look like and what functions the product should have. This would be an efficient approach, as we can hear directly from the client about their requirements for the product.

## Requirements Analysis

Use Case: Assign task to user

1 Preconditions

Tasks must be created

2 Main Flow

User will select a task [S1]. Bot will ask the user for the deadline for the assignee to complete the task [S2].

3 Subflows

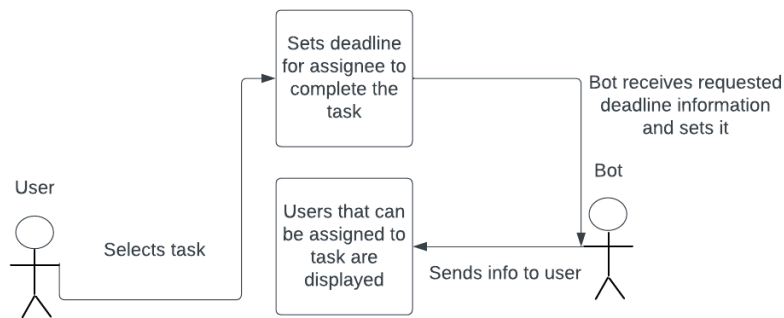
[S1] Bot displays the users that can potentially be assigned the task

[S2] User sets the deadline for the assignee to complete the task

4 Alternative Flows

[E1] No users are available to assign a task to

Use Case Diagram:



Use Case: Create a Task

1 Preconditions

The system must have users

2 Main Flow

User will click on the create task button [S1]. User will click on the submit task button [S2].

3 Subflows

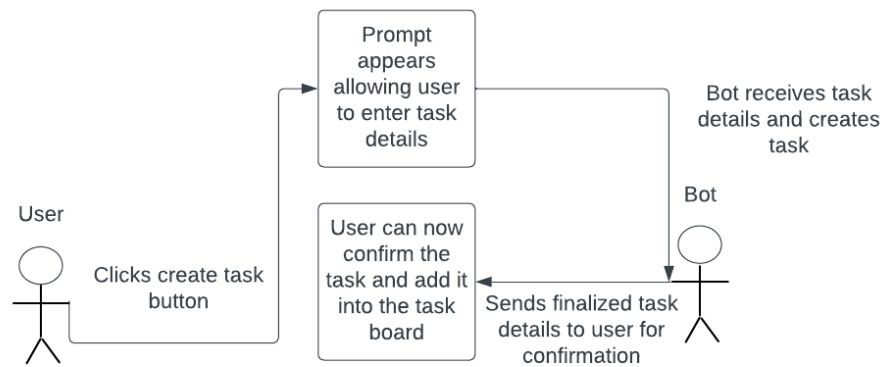
[S1] The system opens up a prompt allowing the user to enter the details of the task.

[S2] The system finalizes the task and presents it on the board of tasks.

4 Alternative Flows

[E1] Board of tasks is full and system cannot add any more tasks

Use Case Diagram:



Use Case: User can designate completed task

1 Preconditions

App installed/account created

Task must be created

2 Main Flow

User will click on task checkmark/button to signify completion [S1]

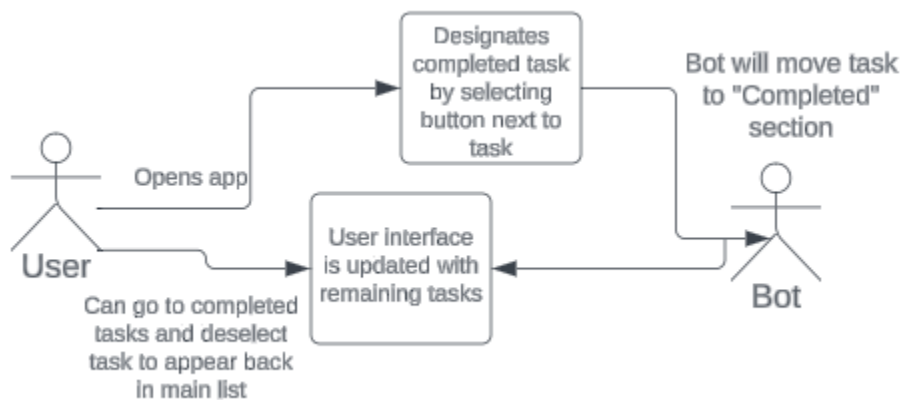
Upon exiting, bot will move task to separate "Completed" section [S2]

3 Subflows

[S1] Bot can accept task completion selection from any member of team

[S2] User can go to Completed section and deselect task

Use Case Diagram:



Use Case: User is alerted of past due task

### 1 Preconditions

Tasks must be created

Deadline must be assigned by user

### 2 Main Flow

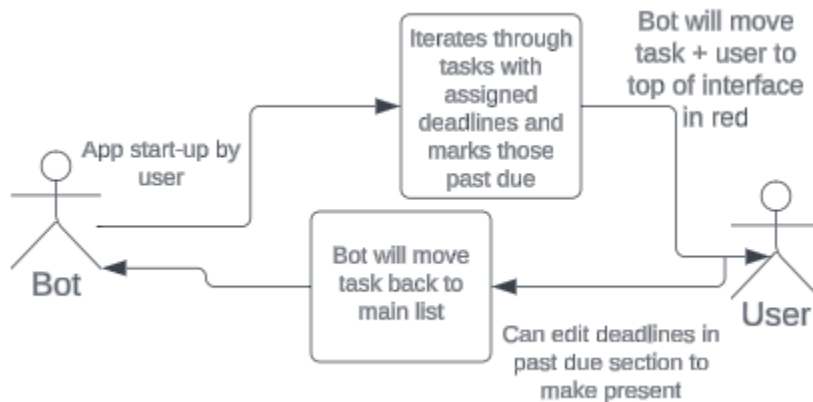
On start-up, bot will iterate through available tasks and highlight those past deadline [S1]

Bot will move past due task(s) to the top of the interface, along with assigned user [S2]

### 3 Subflows

[S1] User can edit past due deadline at which bot will re-iterate and move task back to general list

Use Case Diagram:



Use Case: User adds description to task

### 1 Preconditions

The system must have users

### 2 Main Flow

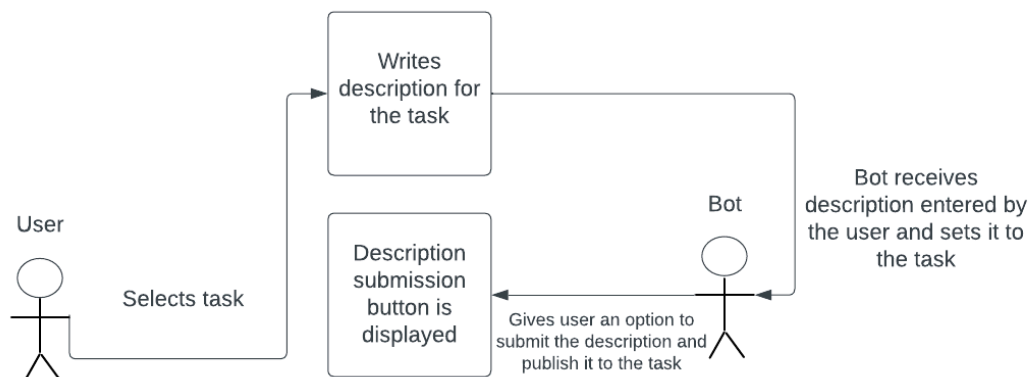
User will select a task [S1]. Bot will ask the user to submit their description [S2].

### 3 Subflows

[S1] Bot will show the user a button to add a description

[S2] User clicks submit button to publish description to the task

Use Case Diagram:



## Process Deliverable - prototype

<https://github.com/pmurphy26/Taskboard>

Task Viewer and Groups Example

User Profile	All tasks	[group1]	[group1, group2]	[ group3, group1]	task1
<div>Add Task</div> <div>Add User</div> <div>Add Task Group</div>	<div><input type="checkbox"/> task1</div> <div><input type="checkbox"/> task2</div> <div><input type="checkbox"/> task3</div> <div><input type="checkbox"/> task4</div> <div><input type="checkbox"/> task5</div> <div><input type="checkbox"/> task6</div> <div><input type="checkbox"/> task7</div>	<div><input type="checkbox"/> task1</div> <div><input type="checkbox"/> task6</div> <div><input type="checkbox"/> task7</div>	<div><input type="checkbox"/> task3</div>	<div><input type="checkbox"/> task4</div> <div><input type="checkbox"/> task5</div>	<div>Due date 1-1-1</div> <div>Assigned Members group1</div> <div>Task Status not started</div> <div>Additional Notes n/a</div>
	<div>edit 0</div>	<div>edit 1</div>	<div>edit 2</div>	<div>edit 3</div>	

3:43 PM 3/15/2024