



Sprint 2 Planning Document

Team 22: Sruthi Bhamidipati, Yana Sheoran, Vihaan Khajanchi, Jack Kenny

Sprint Overview

This sprint will build upon the foundation set in the previous sprint by introducing key enhancements to task management and collaboration features. The primary focus is to expand functionality, improve usability, and ensure a seamless user experience. This sprint will introduce a subtask system, allowing users to create, edit, delete, and mark subtasks as complete, ensuring better task organization and visibility. Additionally, a drag-and-drop feature will be implemented to enable dynamic task reordering, giving users more control over their workflow. Task dependencies will be introduced to establish prerequisite completion rules, ensuring that tasks are completed in the correct order.

Beyond task organization, this sprint aims to enhance task tracking and motivation tools. A time-tracking feature will allow users to log the time spent on individual tasks, helping them better manage productivity. A streak tracker will also be introduced, encouraging users to maintain consistent task completion over time. These enhancements will provide users with greater insight into their progress while fostering engagement with the system.

Collaboration tools will also see significant improvements. Users will gain the ability to create teams, search for other users, and manage team memberships efficiently. The team management feature will enable users to assign tasks within teams, improving workflow efficiency and coordination. Additionally, Sprint 2 will address unfinished user stories from Sprint 1, such as implementing an undo feature for task deletion and allowing users to upload and update their profile pictures.

To ensure these new features work reliably, the sprint will focus on optimizing data persistence, performance, and seamless integration across all components. Extensive testing and debugging efforts will be conducted to guarantee stability, responsiveness, and cross-platform compatibility. By the end of Sprint 2, the system will offer a more robust and efficient task management experience, allowing users to plan, track, and collaborate more effectively.

Scrum Master: Sruthi Bhamidipati

Meeting Plan: T/Th 3:30-4:00 pm

Risks and Challenges:

- UI Complexity: Implementing drag-and-drop and dependency visualization requires a user-friendly interface that integrates seamlessly with existing components. Ensuring an intuitive experience without overwhelming the user is a key challenge.
- Data Persistence Issues: Ensuring task order, dependencies, and archived tasks persist across sessions requires robust database handling. Bugs related to task retrieval and synchronization may arise if not properly managed.
- Performance Concerns: Features like task dependencies, streak tracking, and time logging involve real-time updates. Performance bottlenecks may occur if the backend does not efficiently handle multiple requests simultaneously.
- Conflict Resolution in Collaboration: Managing multiple users editing and modifying team-based tasks simultaneously introduces synchronization challenges, requiring real-time conflict resolution mechanisms.
- Backend Stability: Adding numerous database interactions, such as tracking task history, logging time spent, and managing teams, can lead to database strain if not optimized efficiently.
- Feature Interdependence: Some features rely on others (e.g., dependencies and task completion rules), increasing integration complexity. Breaking changes in one feature may impact others, requiring rigorous testing.

Current Sprint Detail

User Story #1

As a user, I would like to be able to create subtasks within a task.

#	Description	Estimated Time	Owner
1	Create UI for displaying subtasks under a parent task.	3 hrs	Sruthi
2	Implement logic to allow users to add subtasks dynamically.	3 hrs	Sruthi
3	Implement functionality for editing or deleting subtasks.	3 hrs	Sruthi
4	Ensure subtasks can be marked as complete, and reflect their status in the parent task.	2 hrs	Sruthi
5	Debug and test with unit tests to verify correct behavior.	2 hrs	Sruthi

Acceptance Criteria:

- Given the user is on a task page, when they click the “Add Subtask” button, then a new input field should appear for them to enter a subtask name.
- Given the user has entered a subtask name and pressed enter (or clicked confirm), when the action is successful, then the new subtask should appear under the task with its default status.
- Given the user is viewing an existing subtask, when they click the “Edit” button, then an input field should appear allowing them to modify the subtask name.
- Given the user has modified a subtask name and confirmed the change, when the action is successful, then the updated subtask name should be displayed.
- Given the user is viewing an existing subtask, when they click the “Delete” button, then a confirmation modal should appear asking for confirmation.
- Given the user confirms the deletion of a subtask, when the action is successful, then the subtask should be permanently removed from the task list.
- Given the user is viewing a subtask, when they click the checkbox (or toggle) to mark it as complete, then the subtask should visually indicate completion (e.g., strikethrough or a checkmark).
- Given all subtasks are marked as complete, when the parent task is viewed, then the parent task will be marked as complete.

User Story #2

As a user, I would like to drag and drop tasks so that I can easily visualize my task order.

#	Description	Estimated Time	Owner
1	Design and implement the UI to allow dragging and dropping tasks within a task list.	4 hrs	Sruthi
2	Implement logic to update task positions dynamically when a task is moved.	3 hrs	Sruthi
3	Ensure task order persists after refreshing or navigating away from the page.	3 hrs	Sruthi
4	Add visual feedback (e.g., highlight, placeholder) while	1 hr	Sruthi

	dragging a task.		
5	Debug and test drag-and-drop interactions with unit tests.	1 hr	Sruthi

Acceptance Criteria:

- Given the user is on the task page, when they click and hold a task, then the task should be draggable.
- Given the user is dragging a task, when they hover over a new position, then a visual indicator (e.g., placeholder, highlight) should appear to show where the task will be dropped.
- Given the user drops a task in a new position, when the drop is completed, then the task should be placed in its new order and the change should be reflected in the UI.
- Given the user is dragging a task, when they release it without changing its position (e.g., dropping it back at its original spot or outside a valid drop zone), then the task should return to its original position with no changes made to the order.
- Given the user refreshes the page, when the task list loads, then the tasks should appear in the last saved order.

User Story #3

As a user, I would like to be able to set task dependencies to organize connected tasks better.

#	Description	Estimated Time	Owner
1	Design and implement UI for setting dependencies between tasks.	3 hrs	Sruthi
2	Implement logic to allow users to assign one task as dependent on another.	3 hrs	Sruthi
3	Ensure tasks with dependencies cannot be marked as complete until prerequisite tasks are completed.	2 hrs	Sruthi
4	Visually represent dependencies.	1 hr	Sruthi
5	Ensure task dependencies persist after refreshing or navigating away from the page.	2 hrs	Sruthi
6	Debug and test task dependency interactions with unit tests.	2 hrs	Sruthi

Acceptance Criteria:

- Given the user is on the task page, when they open the task dependency menu, then they should see a list of existing tasks to choose from.
- Given the user selects a task as a dependency, when the selection is confirmed, then the dependency should be visually represented using arrows, indentation, or labels to indicate order.
- Given a task has incomplete dependencies, when the user attempts to mark it as complete, then the system should prevent completion and display a message indicating that prerequisite tasks must be completed first.
- Given the user removes a dependency from a task, when the change is saved, then the task should be able to be completed independently.
- Given multiple tasks have dependencies, when they are displayed in the UI, then they should be visually arranged in a way that clearly indicates the sequence.
- Given the user refreshes the page, when the task list loads, then all task dependencies and their visual indicators should be displayed correctly as they were set before refreshing.

User Story #4

As a user, I would like to archive completed tasks so that I can keep my workspace clutter-free.

#	Description	Estimated Time	Owner
1	Implement a "Mark as Complete" button/check for tasks.	1 Hr	Jack
2	Implement change in task to make the task noticeably complete.	1 Hr	Jack
3	Add an "Archive" section where completed tasks can be stored and accessed later.	3 Hrs	Jack
4	Add an "Archive" button next to completed tasks	1 Hr	Jack
5	Implement an auto-archive function that moves completed tasks after a set time.	3 Hrs	Jack
6	Ensure that archived tasks are retrievable but do not clutter the active task board.	1 Hr	Jack
7	Add a dropdown to set how long a task stays until archival.	2 Hrs	Jack
8	Add a time of day button next to the auto archive dropdown	1 Hr	Jack
9	Add time of day UI	1 Hr	Jack
10	Sync time of day with auto archival dropdown functionality	2 Hrs	Jack
11	Debug and test with unit tests.	2 Hrs	Jack

Acceptance Criteria:

- Given that a task has been completed, when the user clicks "Mark as Complete," then the mark as complete checkmark will be ticked and the task will change to a completed state.
- Given that the task changing state is functional, when the task has been changed to a completed state, then the tasks color will be a different shade of green exclusive to completed tasks and the word "completed" will appear on the task.
- Given that the archive button is working, when a user marks a task as complete, an archive task button should appear under the complete task check mark.
- Given that the archive feature is functional, when a user archives a completed task, then it should move to the archive section when the archive button is pressed.
- Given that automatic archival is implemented correctly, when a user is in the archived section, then there should be an auto archival in the top right corner that has a check box, period drop down, and time of day input.
- Given that automatic archival is enabled, when a task remains completed for a set period, then it should automatically move to the archive.
- Given that automatic archival is disabled, when a user is done with a task, then they will have to archive it themselves and it will remain on the main taskpage.
- Given that the auto archival period dropdown is implemented, when a user clicks the dropdown, then they should be presented with options (e.g. every hour, day, week, month) that they can select.

- Given that the auto archival period dropdown is implemented, when a user clicks the dropdown, then they should be presented with options (e.g. every hour, day, week, month) that they can select.
- Given that the time button works, when a user clicks the calendar, then they should be presented with a UI to enter a time.
- Given that the time UI is functional, when a user inputs their time, then the auto archival period will base when to do the next archival on the time the user input.
- Given that the time UI is functional and robust, when a user inputs a time that doesn't exist, then they will be shown an error saying that it is incorrect.

User Story #5

As a user, I would like to jot notes down about tasks that I have or have not completed so that I can reference thoughts for later.

#	Description	Estimated Time	Owner
1	Implement a "notes" button for each task.	1 Hr	Jack
2	Implement an interactive UI that allows users to put notes in.	1 Hrs	Jack
3	Implement text editing interface within notes UI with basic formatting button options.	2 Hr	Jack
4	Add a delete notes button in the notes UI.	1 Hr	Jack
5	Make notes persist across multiple sessions and have them saved in the database.	2 Hrs	Jack
6	Debug and test notes features with unit tests.	2 Hrs	Jack

Acceptance Criteria:

- Given that the notes button is implemented, when a user views the main task page, there will be a notes button beside each task.
- Given that the notes button is working, when a user clicks "Notes", another UI will appear with all the users notes.
- Given that the notes UI is implemented correctly, when a user adds a note, they will be able to type their notes into the UI
- Given that the user is able to type their notes properly, when a user adds a note, they will be presented with basic formatting options (bullet points, bold, italics)
- Given that the bullet points option is presented, when a user clicks the bullet points button, then a bullet point will appear that can be typed after, it will also create new bullet points when the user enters.
- Given that the bold option is presented, when a user clicks the bold button, then all text that the user enters within the note will be bolded until the button is unclicked.
- Given that the italics option is presented, when a user clicks the italics button, then all text that the user enters within the note will be italicized until the button is unclicked.
- Given that the notes UI is implemented correctly, when a user adds a note that goes beyond the UI, the UI will be scrollable so that the user can continue to see all their notes
- Given that the delete note button is functional, when a user clicks the delete button, all the notes for that task should disappear and nothing should remain inside.
- Given that the notes persist, when a user refreshes or logs out, then when they are back in the main task page the notes will still be there.

User Story #6

As a user, I would like to set default task templates so that I can quickly create repetitive tasks.

#	Description	Estimated Time	Owner
1	Make default task creation button	1 Hr	Jack
2	Develop and implement UI to hold all task templates	2 Hrs	Jack
3	Create task template editing, creation, and deletion buttons.	1 Hr	Jack
4	Create and implement “favorite” feature for default task templates	2 Hrs	Jack
5	Implement “most recent” button	1 Hr	Jack
6	Ensure that the task templates persist across sessions and are saved in the database.	2 Hrs	Jack
7	Debug the task template with unit tests	1 Hr	Jack

Acceptance Criteria:

- Given that the default task creation button is functional, when a user clicks that button, then the UI for default task creation will appear next to the main task page.
- Given that the task template UI is implemented, when the user is viewing the UI, then they will see a list of tasks similar to the main task page.
- Given that the task template UI is functional, when the user has more tasks than space in the UI, then they will be able to scroll to see the other default templates.
- Given that the create and edit buttons are functional, when a user clicks them, then it will be the same create and edit interface as the normal tasks.
- Given that the delete button is functional, when a user deletes a task, then it will be given a time based undo like for normal tasks but it will not store the last 10 deleted tasks.
- Given that the favorite feature is implemented, when a user sees a task, they will be given the option to click the favorite icon(a heart) that will put that template at the top of the list of templates.
- Given that the favorite icon is working, when a user clicks the icon, then they will see that there is a noticeable difference between the icon associated with a “favorited” template and a template that is not favorited.
- Given that the most recent button works, when a user has already used a default task template, then they will be presented with a button that will make the most recently used template when clicked.

User Story #7

As a user, I would like to schedule tasks for a future start date so that I can plan ahead without cluttering my current task list.

#	Description	Estimated Time	Owner
1	Add a “Start Date” field to the task creation form	1 Hr	Yana
2	Modify the database schema to store future task start dates	1 Hr	Yana
3	Update task retrieval logic to exclude tasks with a future start date	2 Hrs	Yana
4	Implement a mechanism (cron job or scheduled check) to move tasks to active when the start date is reached	2 Hrs	Yana

5	Enable users to manually start a scheduled task before its start date	2 Hr	Yana
6	Test and validate scheduling functionality	1 Hr	Yana

Acceptance Criteria:

- Given the user creates a new task with a future start date, then the task should not appear in the current task list until the start date is reached.
- Given the user schedules a task with a future start date, then the task should be saved and visible in a "Scheduled" or equivalent section.
- Given the current date reaches the scheduled start date of a task, then the task should automatically move to the active task list.
- Given the user edits a scheduled task before its start date, then they should be able to modify the task details, including the start date.
- Given the user edits the start date to the current date then the task will immediately move to the active tasks list.
- Given the user deletes a scheduled task before its start date, then the task should be removed from the system and should not appear in the task list.
- Given the user has completed a scheduled task before its start date, then the user should be able to select an option in the scheduled list to switch the task to the completed section, which would edit the start and finish date as well.
- Given the user sets a recurring task with a future start date, then the recurrence should begin only after the start date is reached.
- Given the user enters a past date for the task start date, then the system should display an error message.

User Story #8

As a user, I would like to log time spent on tasks so that I can keep track of progress.

#	Description	Estimated Time	Owner
1	Add a "Time Spent" field to the task model in the database	1 Hr	Yana
2	Update the UI to allow users to manually enter time spent on a task	1 Hrs	Yana
3	Store multiple time log entries per task and sum the total time	1 Hrs	Yana
4	Display logged time in task details and task summary	2 Hrs	Yana
5	Allow users to edit or delete logged time entries	1 Hrs	Yana
6	Test and validate time-tracking functionality	1 Hr	Yana

Acceptance Criteria:

- Given the user views a task, then they should see a field for logging time spent.
- Given the user manually enters time spent on a task, then the system should save and display the logged time.
- Given the user logs multiple time entries for a task, then the system should sum the total time spent.
- Given the user edits a logged time entry, then the system should update the total time accordingly.
- Given the user deletes a logged time entry, then the total time should be adjusted.
- Given the user views a task summary, then they should see the total time spent across all tasks.
- Given the user enters a past date for the task start date, then the system should display an error message.

- Given the user tries to log an unreasonably large time entry, then the system should display a validation error.
- Given the user tries to add text to the number logging field, then it should display an error message asking the user to enter a valid input.

User Story #9

As a user, I would like to see a streak tracker to visualize my consistent task completion.

#	Description	Estimated Time	Owner
1	Modify the database schema to track task completion dates per user	2 Hrs	Yana
2	Implement logic to calculate and update streak counts based on consecutive days of completion	3 Hrs	Yana
3	Design a UI component to display the streak tracker (e.g., a progress bar or calendar view)	2 Hrs	Yana
4	Show current streak, highest streak, and total completed days in the UI	2 Hrs	Yana
5	Allow users to view past streak history and performance trends	2 Hrs	Yana
6	Test and validate streak tracking functionality	1 Hr	Yana

Acceptance Criteria:

- Given the user completes a task today, then the streak counter should increment by 1.
- Given the user does not complete any task for a day, then the streak counter should reset to 0.
- Given the user completes tasks on multiple consecutive days, then the streak counter should increase accordingly.
- Given the user views their streak tracker, then they should see their current streak, highest streak, and total completed days.
- Given the user views their past performance, then they should see a history of their streaks and trends.
- Given that the user has no active task in a day and nothing to register, then the streak will still not break and will continue over to the next day.
- Given an error occurs while updating the streak count, then the system should notify the user and retry updating the streak.
- Given the user manually deletes a completed task from a streak day, then the system should recalculate the streak accordingly.

User Story #10

As a team leader, I would like to create a “team” so that I can manage group tasks.

#	Description	Estimated Time	Owner
1	Add a new button to navigate to a teams page	30 mins	Vihaan
2	Add a “create team” button that will prompt you to create a name for the team	1 hr	Vihaan
3	Any teams created will show up on the leader’s teams page	1 hrs	Vihaan

4	Can click on a button to open that team and view its details, like members and tasks	3 hrs	Vihaan
5	Within a team, add a button for the team leader to add members to the team	2 hrs	Vihaan
6	Add a button for the team leader to delete their team, erasing all data and members	2 hr	Vihaan
7	Test and debug with unit tests	1 hr	Vihaan

Acceptance Criteria:

- Given the user clicks on the teams button, then they should navigate to a teams page specific to them
- Given the user is on their teams page, then they should see a button to create teams.
- Given the user clicks on the create team button, then they should be prompted to type in a team name and click confirm.
- Given the user created a team, then the team should show up on their teams page.
- Given the user clicks on the team, then they should open a page showing more details about the team, like members and tasks, and see buttons to add members and delete the team.
- Given the user already has teams, then creating a team should still create another unique team.
- Given the team leader clicks delete team, then they should be asked to confirm their choice.
- Given the team leader confirms the delete, then the team will be removed from their task page, as well as any members' pages.
- Given the team leader cancels the delete, nothing will happen.

User Story #11

As a user, I would like to be able to search for other users so that I can use other functionality like teams.

#	Description	Estimated Time	Owner
1	When adding members to a team, add a search bar where you can type in a username and any matching usernames will show up	3 hrs	Vihaan
2	Each username should have a "view" and "add" button next to it	2 hrs	Vihaan
3	Also add the same search feature to home page, except with just "view" button	1 hr	Vihaan
4	"view" button will preview their profile page	3 hrs	Vihaan
5	Test and debug with unit tests	1 hr	Vihaan

Acceptance Criteria:

- Given a team leader clicks the button to add members on their team's page, then they should see a search bar.
- Given a user clicks the search button on the home page, then they should see a search bar.
- Given a search bar is empty, then no users will show up underneath
- Given the search bar has contents, then any usernames matching the pattern will show up underneath the search bar.
- Given the search is being done in the add members section, then each username should have an add and view button.
- Given the search is being done from the home page, then each username should have only a view button.
- Given the user clicks view on any user, then they should see a preview of that user's profile page.

User Story #12

As a team leader, I would like to be able to add users to a team.

#	Description	Estimated Time	Owner
1	When clicking “add”, send an invite to the user, which will show up in their teams page	3 hrs	Vihaan
2	Once added, the option will be greyed out unless the member denies the invite	2 hrs	Vihaan
3	Users can choose to join or deny an invite	2 hrs	Vihaan
4	If joining, the team should show up in their teams page	1 hr	Vihaan
5	Debug and test with unit tests	2 hrs	Vihaan

Acceptance Criteria:

- Given the user clicks the add button after searching, then the receiving user should get an invite in their teams page.
- Given the user already sent an invite, then they should not be able to send another unless the invite is removed by denial.
- Given a user receives an invite, then they should have a noticeable notification on their teams page.
- Given a user sees an invite, then they should be able to click either join or deny.
- Given a user clicks join, then the team should show up on their team page.
- Given a user joins, then their username should show up under the members section for everyone who views it.
- Given a user clicks deny, then the invite should be removed.

Sprint 1: Incomplete User Stories, Tasks, & Acceptance Criteria

User Story #4

As a user, I would like to be able to delete tasks.

#	Description	Estimated Time	Owner
3	Implement undo functionality for accidental deletions.	2 Hrs	Jack
4	Debug and test task deletion and persistence with unit tests.	1 Hr	Jack

Acceptance Criteria:

- Given the user clicks "Undo" within a set time frame (e.g., 10 seconds), then the task should reappear in its original state in the task list.
- Given the undo time frame expires, when the user tries to restore the task, then they should no longer be able to undo the deletion.
- Given the user deletes multiple tasks, when they access the undo function, then only the last 10 deleted tasks should be recoverable.
- Given the user deletes more than 10 tasks, when they try to undo, then older deleted tasks (beyond the last 10) should no longer be available for recovery.

User Story #11

As a user, I would like to be able to change my profile picture so that I can customize my account.

#	Description	Estimated Time	Owner
1	Update backend to store the uploaded profile picture in the database	2 Hrs	Yana
2	Test the end-to-end flow to confirm profile picture persistence across sessions	1 Hr	Yana

Acceptance Criteria:

- Given the user uploads a profile picture, then the system should store it persistently.
- Given the user logs out and logs back in, then their profile picture should remain unchanged.
- Given the user uploads a new profile picture, then the previous one should be replaced.
- Given the system fails to load the profile picture when the user logs in, then it should display a default avatar.

Backlog

Functional

Task Management

- ~~1. As a user, I would like a task page so I can view all my tasks.~~
- ~~2. As a user, I would like to be able to create tasks.~~
- ~~3. As a user, I would like to be able to edit tasks.~~
- ~~4. As a user, I would like to be able to delete tasks.~~
- ~~5. As a user, I would like to be able to create subtasks within a task.~~
- ~~6. As a user, I would like to drag and drop tasks so that I can easily visualize my task order.~~
- ~~7. As a user, I would like to categorize tasks by priority, tags, or deadlines so that I can make decisions on which order to take the tasks.~~
- ~~8. As a user, I would like to color code tasks based on categories so that I can quickly identify them.~~
- ~~9. As a user, I would like to set default task templates so that I can quickly create repetitive tasks.~~
- ~~10. As a user, I would like to set different statuses for tasks (in progress, blocked, waiting for review).~~
- ~~11. As a user, I would like to be able to filter tasks by completion status.~~
- ~~12. As a user, I would like to archive completed tasks so that I can keep my workspace clutter free.~~
13. As a user, I would like to be able to roll over incomplete tasks to the following day, either automatically or manually.
- ~~14. As a user, I would like to be able to set task dependencies to organize connected tasks better.~~
- ~~15. As a user, I would like to schedule tasks for a future start date so that I can plan ahead without cluttering my current task list.~~
16. As a user, I would like to receive smart suggestions for breaking up long tasks into smaller subtasks

Goal-Setting & Progress Tracking

- ~~17. As a user, I would like to jot notes down about tasks that I have or have not completed so that I can reference thoughts for later.~~
18. As a user, I would like to track undesirable habits/tasks so that I can check progress on if I'm lowering how unproductive I am (if time allows).

Time-Tracking & Productivity Insights

- ~~19. As a user, I would like to log time spent on tasks so that I can keep track of progress.~~
20. As a user, I would like to be able to visualize how my time is spent across different categories (if time allows).
21. As a user, I would like to set an estimated time and an actual time for each of my tasks, and see the difference between the two.
22. As a user, I would like to receive a weekly summary of my most and least productive days so that I can adjust my workflow.
23. As a user, I would like to compare my productivity across different weeks/months to track my long-term progress.
24. As a user, I would like to check time spent on breaks so that I can maintain a healthy balance.
25. As a user, I would like to be given gentle reminders through pop-ups or alerts suggesting I take breaks after long periods of continuous work.
26. As a user, I would like to receive productivity tips based on my task history (if time allows).

Gamification & Team Features

- ~~27. As a user, I would like to see a streak tracker to visualize my consistent task completion.~~
- ~~28. As a user, I'd like to be able to navigate to a profile tab where I can see relevant information about my account.~~
29. As a user, I would like to be able to share or display my streak or other object on my profile.
30. As a team member, I would like to view a productivity leaderboard to foster a sense of competition and teamwork.
31. As a team member, I would like to send and receive kudos for completed tasks to encourage engagement.
32. As a team leader, I would like to assign weighted points to tasks to reflect their complexity.
- ~~33. As a user, I would like to be able to search for other users so that I can use other functionality like teams.~~
34. As a user, I would like to be able to be a contributor (leader or member) for multiple unique teams.

35. As a user, I would like to be able to access a unique team page for each of the teams that I am a part of to see team tasks and contributors.
- ~~36. As a team leader, I would like to create a “team” so that I can manage group tasks.~~
- ~~37. As a team leader, I would like to be able to add users to a team.~~
38. As a team leader, I would like to be able to assign tasks.
39. As a team leader, I would like to set permissions for different team members so that I can control who edits tasks.
40. As a team leader, I would like to be able to view team members’ activity.
41. As a team leader, I would like to track the time my members are spending on tasks.
42. As a team member, I would like to be able to add tasks.
43. As a team member, I would like to be able to claim tasks.
44. As a team member, I would like to mark team tasks as completed.
45. As a team member, I would like to leave comments on tasks so that I can communicate updates.
46. As a team member, I would like to tag people so that they can be notified when a task needs their attention.

User Preferences & Accessibility

- ~~47. As a user, I would like to create an account with unique log in credentials.~~
- ~~48. As a user, I would like to log in to my account when I return.~~
49. As a user, I'd like my data to be synced to my account so that signing in somewhere new shows me my tasks.
50. As a user, I would like to be able to use multiple devices so that I can be productive and check my tasks no matter my situation.
- ~~51. As a user, I would like to access a settings option that allows me to customize my experience, including enabling dark mode or adjusting the app theme, so that I can work in my preferred visual setting.~~
- ~~52. As a user, I would like to change the font size and spacing so that I can improve readability based on my preference.~~
53. As a user, I would like to choose what view I am in (calendar, timeline, etc.) so that it can be in my most preferred state.
54. As a user, I would like to set focus mode where notifications are muted for deep working sessions.
55. As a user, I would like to receive notifications so that I can be reminded of important tasks and stay productive.
56. As a user, I would like to be able to change my display name for different teams to best suit each team.
57. As a user, I would like to be able to change my profile picture so that I can customize my account.
- ~~58. As a user, I would like to be able to delete my account if I no longer want to use the app.~~

Integrations & Data Export

59. As a user, I would like to integrate Planorama with third-party tools like Google Calendar or Slack for streamlined workflow (if time allows).
60. As a user, I would like to export my task data in CSV or PDF format for external analysis (if time allows).

Non-Functional

1. The platform must ensure tasks and insights load within 500ms for an optimal user experience while supporting up to 10,000 simultaneous users with minimal latency.
2. The architecture should be scalable with cloud-based solutions to accommodate a growing user base and new features.
3. A responsive design must ensure compatibility across desktop and mobile devices, while an intuitive and consistent user interface minimizes the learning curve.
4. To ensure data security, sensitive user information must be protected with end-to-end encryption, preventing unauthorized access. Additionally, all user actions should be authenticated using OAuth 2.0 standards.
5. The platform must maintain 99.9% uptime, ensuring high availability and reliability for users. Any downtime should be limited to scheduled maintenance periods, minimizing disruptions and optimizing system performance.