Team 39 Sprint 2 Planning Document

Enjoyagoals

Team Members
Prahas Pattem, Nick Norton, Karim Mammadli, Nabi Nabiyev, Ryan Chang

Sprint Overview

During this sprint, we will first finish and link the components of all of the left over user stories from Sprint 1. Some of our previous work may also need modification and rewriting to better coordinate between the front and backend. We then hope to implement the necessary functionality to fulfill the expected user stories. As the work done during this Sprint will also form the foundation of Sprint 3, we will aim to make everything as bug free and efficient as we can during the allotted time.

Scrum Master: Nick Norton

Meeting Plan: Tuesdays/Thursdays @ 9:00pm

Risks and Challenges:

One very significant challenge is that there are many user stories from Sprint 1 that we failed to fully implement due to time constraints. The implementation of our selected user stories in Sprint 2 also depends on successfully finishing the user stories from Sprint 1, so we will need to finish the incomplete tasks as soon as possible. Implementing the incomplete frontend, connecting it to the existing backend code, and testing it for bugs will severely cut into our time and will almost certainly be a major setback to the progress of our project and could potentially leave us with more incomplete stories to deal with in Sprint 3. Additionally, the next sprint will also depend on the completion of our current user stories, so we will need to make sure the implementation is complete, functional, and mostly bug-free. Testing each user story to ensure this will also likely take up a large amount of our time.

Current Sprint Detail

User Story #1

As an individual user, I would like to be able to view a calendar with the project and task deadlines listed clearly.

#	Description	Estimated Time	Owner
1	Create UI for displaying a calendar with the project and task deadlines	2 Hrs	Prahas & Ryan
2	Construct HTTP request(s) to get necessary data from backend	2 Hrs	Prahas & Ryan
3	Create an algorithm to parse data and populate the calendar with deadlines	2 Hrs	Prahas & Ryan
4	Debug and test the algorithm that obtains all the necessary information	2 Hrs	Prahas & Ryan
5	Create/update corresponding backend schema and routes	2 Hrs	Nick, Nabi, & Karim
6	Debug and test backend routes	2 Hrs	Nick, Nabi, & Karim

Acceptance Criteria:

- Given that the necessary schema and routes are updated, the backend will respond to requests for a User’s data or specifically a User’s deadlines with JSON that includes all the deadlines.
- Given that the UI for displaying the calendar is correctly implemented, when the user clicks on the calendar button, they should be able to view a calendar with the project and task deadlines.
- Given that the frontend can request level data from the backend and then parse and display it, users will see a calendar that is stored in the database on their homepage.

User Story #2

As an individual user, I would like to be able to view my level based on the number of tasks I have completed.

#	Description	Estimated Time	Owner
1	Update profile UI to include level	2 Hrs	Prahas & Ryan
2	Construct HTTP requests to backend to get necessary data	2 Hrs	Prahas & Ryan
3	Create algorithm to parse backend data and populate UI	2 Hrs	Prahas & Ryan
4	Debug/test HTTP request and parsing algorithms	2 Hrs	Prahas & Ryan
5	Create/update corresponding backend schema and routes	3 Hrs	Nick, Nabi, & Karim
6	Debug and test backend schema and routes	2 Hrs	Nick, Nabi, & Karim

Acceptance Criteria:

- Given that the necessary schema and routes are updated, the backend will respond to requests for a User's data or specifically a User's level with JSON that includes the User's level data.
- Given that the level UI element is implemented, users will see a level on their profile.
- Given that the frontend can request level data from the backend and then parse and display it, users will see their level that is stored in the database on their profile.

User Story #3

As an individual user, I would like to be able to view a progress bar visually showing progress made towards the end goal.

#	Description	Estimated Time	Owner
1	Create UI for displaying progress bar	2 Hrs	Ryan, Prahas
2	Create UI for brief explanation of progress bar information	2 Hrs	Ryan, Prahas
3	Construct HTTP requests to backend to get necessary data	2 Hrs	Karim, Nabi, Nick
4	Create algorithm to parse backend data and calculate the progress bar	3 Hrs	Karim, Nabi, Nick
5	Debug/test HTTP request and parsing algorithms	2 Hrs	Karim, Nabi, Nick
6	Create/update corresponding backend schema and routes	2 Hrs	Karim, Nabi, Nick
7	Debug and test backend schema and routes	2 Hrs	Karim, Nabi, Nick

Acceptance Criteria:

- Given that the UI for the progress bar is implemented, users will be able to view this progress bar on the homepage.
- Given that the UI for the brief explanation of progress bar is implemented, users will be able to view the number of tasks they need to finish to reach the end goal of the project.
- Given that the frontend can request progress bar data from the backend and then parse and display it, every user will be able to see this progress bar.

User Story #4

As an individual user, I would like to be able to upload files from my computer to my Enjoyagoals group(s) when I have completed a task.

#	Description	Estimated Time	Owner
1	Create UI for uploading files upon completing a task	2 Hrs	Prahas & Ryan
2	Create an algorithm to accept local files from the computer	2 Hrs	Prahas & Ryan
3	Debug and test algorithm for accepting local files	2 Hrs	Prahas & Ryan
4	Construct request to send file(s) to backend	2 Hrs	Prahas & Ryan
5	Create backend route & schema to be able to accept files from frontend and store them in database	4 Hrs	Nick, Nabi, & Karim

Acceptance Criteria:

- Given that the necessary schema and routes are implemented, the backend will respond to valid requests to store a file in the database with a 200/201/202 HTTP status response.
- Given that the backend can properly accept and store files, the backend will respond to GET requests for stored files with the corresponding files.
- Given that the UI for uploading files is implemented, users will be able to select files from their computer to send to the server.
- Given that the UI for uploading files is properly connected to the backend, users will be able to upload files from their computer and have them stored in the database so that they are able to retrieve them later.

User Story #5

As a team member, assign point values to each task created.

#	Description	Estimated Time	Owner
1	Add UI element to adjust point values of pending tasks to existing task UI	2 Hrs	Prahas & Ryan
2	Create algorithm to accept intended point value	2 Hrs	Prahas & Ryan
3	Construct request to send assigned point value & the ID of the corresponding task to backend	2 Hrs	Prahas & Ryan
4	Test & debug algorithm and request	2 Hrs	Prahas & Ryan
5	Update schema and routes to differentiate between tasks that have an assigned point value and those that don’t yet	2 Hrs	Nick, Nabi, & Karim
6	Debug and test backend routes and algorithms	2 Hrs	Nick, Nabi, & Karim

Acceptance Criteria:

- Given that the assigning point value UI is implemented, a team member will be able to input an intended point value for a task.
- Given that the backend schema and routes are updated, the backend will respond to a POST request to assign a point value to a task.
- Given that the backend team member check is implemented, the point value of a task will not be updated in the database if the request is not validated as being from a team member from another room.

User Story #6

As a team member, easily review any finished tasks to ensure proper completion.

#	Description	Estimated Time	Owner
1	Create UI to allow a leader to view all the finished tasks	2 Hrs	Ryan, Prahas
2	Construct HTTP requests to backend to get necessary data	2 Hrs	Karim, Nabi, Nick
3	Create UI with a description of a finished task with buttons to approve or deny a finished task	2 Hrs	Ryan, Prahas
4	Create an algorithm to modify the database when a task is marked as complete	2 Hrs	Karim, Nabi, Nick
5	Debug and test algorithm for modifying the database	2 Hrs	Karim, Nabi, Nick

Acceptance Criteria:

- Given that the UI for all the finished tasks is implemented, a team member will be able to view all the finished tasks and pick which task to review.
- Given that the UI with a description of a finished task is implemented,a team member will be able to view the task name and the user the task was assigned to.
- Given that an algorithm to modify the database is implemented, a task won’t be visible on the finished task page unless it’s marked as completed.

User Story #7

As a team member, I would like to be able to provide feedback for every task completed.

#	Description	Estimated Time	Owner
1	Create a comment section for each finished task page	3 Hrs	Ryan, Prahas
2	Construct HTTP requests to backend to get necessary data	1 Hrs	Karim, Nabi, Nick
3	Create algorithm to send feedback to database	3 Hrs	Karim, Nabi, Nick
4	Create a comment section controller	2 Hrs	Karim, Nabi, Nick
5	Debug and test algorithm for controlling the database	2 Hrs	Karim, Nabi, Nick

Acceptance Criteria:

- Given that the comment section is implemented, a team member will be able to provide feedback on tasks.
- Given that the comment section controller is implemented, a team member won’t be able to provide feedback if a section is left blank or if it doesn’t meet the minimum word count.
- Given that the algorithm to send feedback is implemented, a receiver will be able to view a team member’s comments.

User Story #8

As a team member, I would like to be able to view a task’s status.

#	Description	Estimated Time	Owner
1	Create an algorithm to check the status of a task	2 Hrs	Karim, Nabi, Nick
2	Debug and test algorithm for the task completion	2 Hrs	Karim, Nabi, Nick
3	Make all task posts clickable	2 Hrs	Ryan, Prahas
4	Create a UI to view a task’s status	2 Hrs	Ryan, Prahas

Acceptance Criteria:

- Given that the algorithm to check the status of a task is implemented, a team member will be able to see whether the task is completed or pending.
- Given that the UI is implemented properly, a team member will be able to click on a task and see current progress and the user that is working on it.
- Given that all task posts are clickable, a team member will be able to easily view each task's status.

User Story #9

As a team member, I would like to be able to indicate that I have completed a task.

#	Description	Estimated Time	Owner
1	Create a “completed” button for all task pages of a specified team member	2 Hrs	Ryan, Prahas
2	Create an algorithm to detect all tasks that a logged-in user is assigned	2 Hrs	Karim, Nabi, Nick
3	Debug and test algorithm for detecting tasks	2 Hrs	Karim, Nabi, Nick
4	Make the “complete” button unclickable if the deadline has passed	2 Hrs	Ryan, Prahas
5	Create an algorithm to check if the task passed the deadline	2 Hrs	Karim, Nabi, Nick
6	Debug and test algorithm for checking the deadline of each task	2 hrs	Karim, Nabi, Nick
7	If the task is valid, add the task to the “completed tasks” database	2 Hrs	Karim, Nabi, Nick

Acceptance Criteria:

- Given that the algorithm for detecting all the tasks users are assigned to is implemented, users will be able to know which task they shall complete.
- Given that the UI and algorithm for the “complete” button is implemented, users will be able to submit completed tasks before deadline by clicking on the button, but if the deadline is past, clicking the button will not mark the task as completed past the deadline, and this task will be marked as missed
- Given that the algorithm for checking whether or not a task is valid, tasks will be added to the database as “completed tasks” if the task is valid, otherwise they will not be added to the database.

User Story #10

As a team member, I would like to be able to view the tasks that have been completed.

#	Description	Estimated Time	Owner
1	Create UI for displaying the completed tasks	2 Hrs	Ryan, Prahas
2	Create UI for displaying additional task details when clicked on a task from the completed tasks list	2 Hrs	Ryan, Prahas
3	Construct HTTP requests to backend to get necessary data	2 Hrs	Ryan, Prahas
4	Create algorithm to parse backend data and display the completed tasks	2 Hrs	Karim, Nabi, Nick
5	Debug/test HTTP request and parsing algorithms	2 Hrs	Karim, Nabi, Nick

Acceptance Criteria:

- Given that the necessary schema and routes are updated, the backend will respond to requests for room’s completed task lists.
- Given that the UI panel for the Room is correctly implemented, if there is too much information for the panel to hold, the user should be able to scroll through for access.
- Given that the frontend can request list data from the backend and then parse and display it, users will see the completed task list that is stored in the database on their homepage.

User Story #11

As a team member, I would like to be able to view the tasks that are being completed by other team members.

#	Description	Estimated Time	Owner
1	Create UI for displaying the tasks that are being completed	2 Hrs	Prahas, Ryan
2	Create UI for displaying additional task details when clicked on a task from the tasks that are being completed list	2 Hrs	Ryan, Prahas
2	Construct HTTP requests to backend to get necessary data	2 Hrs	Prahas, Ryan
3	Create algorithm to parse backend data and display the tasks that are being completed	2 Hrs	Prahas, Ryan
4	Debug/test HTTP request and parsing algorithms	2 Hrs	Karim, Nabi, Nick

Acceptance Criteria:

- Given that the necessary schema and routes are updated, the backend will respond to requests for room's tasks that are being completed.
- Given that the UI panel for the Room is correctly implemented, if there is too much information for the panel to hold, the user should be able to scroll through for access.
- Given that the frontend can request list data from the backend and then parse and display it, users will see the list of tasks being completed that is stored in the database on their homepage.

User Story #12

As a team member, I would like to be able to view the details of each completed task, such as the date & time marked as complete, who completed the task, the point value, the difficulty, the deadline of the completed task

#	Description	Estimated Time	Owner
1	Create button for viewing completed tasks	2 Hrs	Ryan, Prahas
2	Create UI for viewing completed tasks	2 Hrs	Ryan, Prahas
3	Create UI for viewing description of a certain completed task	2 Hrs	Ryan, Prahas
4	Create algorithm to correctly display date & time, the name of contributor, point value, difficulty, deadline of the completed task	3 Hrs	Karim, Nabi, Nick
5	Debug/test HTTP request and parsing algorithms	2 Hrs	Karim, Nabi, Nick
6	Create/update corresponding backend schema and routes	2 Hrs	Karim, Nabi, Nick
7	Debug and test backend schema and routes	2 Hrs	Karim, Nabi, Nick

Acceptance Criteria:

- Given that the UI for viewing the completed tasks is implemented, users will be able to view which tasks were completed.
- Given that the UI and the algorithm for viewing details of a certain completed task is implemented, users will be able to view who finished a certain task.
- Given that the UI and the algorithm for viewing details of a certain completed task is implemented, users will be able to view which task at what time has been completed and its point value, difficulty, deadline.

User Story #13

As a team member, I would like to be able to view the leaderboard that shows the other team members' levels & rankings.

#	Description	Estimated Time	Owner
1	Create UI for users to view leaderboard of levels and rankings	5 Hrs	Prahas, Ryan
2	Create algorithm to obtain relevant user data of other team members in room	2 Hrs	Karim, Nabi, Nick
3	Connect algorithm to UI so users will be displayed on leaderboard	2 Hrs	Karim, Nabi, Nick

Acceptance Criteria:

- Given that the UI for the leaderboard was completed, users will be able to view the profiles of other members by clicking their entry in the leaderboard, including their own.
- Given that the UI for the leaderboard is properly implemented, when the list of users is too large for the screen to hold, they should be able to scroll through the page.
- Given that the UI for the leaderboard is properly implemented, when the list of users is displayed, their level will be highlighted and displayed differently than the other team members'.

Functional Requirements:

- **As an individual user, I would like to be able to:**
    1. ~~register for an Enjoyagoals account.~~
    2. ~~login and manage my Enjoyagoals account, and reset my password.~~
    3. ~~register using my google account if I do not have an Enjoyagoals account.~~
    4. ~~easily set a custom profile picture and customize my profile page's theme and banner.~~
    5. ~~display my contribution to the room I am currently in, email account and phone number on my Enjoyagoals account alongside other personal information for others to see.~~
    6. ~~view all of the tasks I need to finish in a comprehensive list after logging in~~
    7. ~~easily check the details of each individual task (description, date created, deadline, point value).~~
    8. ~~easily view all of my completed tasks in a list along with my total points accumulated and number of tasks completed in a graphic visualization.~~
    9. ~~join, switch between, or withdraw from different project rooms at any time after signing in.~~
    10. purchase exclusive profile pictures or emotes with points that I have earned from tasks to modify my user experience.
    11. ~~view a calendar with the project and task deadlines listed clearly.~~
    12. receive notifications through email and/or text for approaching deadlines and disable them if I want.
    13. ~~view my level based on the number of tasks I have completed.~~
    14. easily join a team with either a designated team leader delegating tasks to others or a team without a designated leader democratically deciding what tasks should be done.
    15. ~~view a progress bar visually showing progress made towards the end goal.~~
    16. (if time allows) view a rough estimation of the time to complete a task.
- **As a team member, I would like to be able to:**
    1. ~~view tasks to be completed, their point values, and their status~~.
    2. ~~indicate that I am currently working on a task and then add notes if I feel like I need to share information with my team~~.
    3. ~~upload files from my computer to my Enjoyagoals group(s) when I have completed a task.~~
    4. ~~indicate that I have completed a task.~~
    5. ~~view the number of the tasks that have been completed.~~
    6. ~~view the tasks that are being completed by other team members.~~
    7. ~~view the details of each completed task, such as the date & time marked as complete, who completed the task, the point value, etc.~~
    8. communicate with other team members through a messaging system.
    9. start and participate in a voting system in case the team needs to kick a member.
    10. view the leaderboard that shows the other team members' levels & rankings.
    11. query the leaderboard.
    12. generate a pdf file that includes a progress report for every team member.

13. (if time allows) implement a coup d'etat system to remove the leader.

- **As a team member with a team leader, I would like to be able to:**
    1. request that the team leader reviews my work & provides either points or feedback.
- **As a team member without a team leader, I would like to be able to:**
    1. ~~propose tasks to be completed as well as point values for the tasks to be completed.~~
    2. vote on each task proposal and each point value proposal.
    3. request that the rest of the team review my work.
    4. receive feedback and points for the tasks I have completed.
    5. view a leaderboard showing everyone else's contributions, levels, and rankings.
- **As a team leader, I would like to be able to:**
    1. ~~invite team members to my team at any point.~~
    2. ~~create tasks to be completed with the details of each task clearly displayed, such as the date & time marked as completed, who completed the task, the point value, etc.~~
    3. ~~assign point values to each task created.~~
    4. ~~provide feedback for every task completed.~~
    5. ~~view a task's status.~~
    6. forcibly remove a team member from a task they are currently working on.
    7. easily review any finished tasks to ensure proper completion.
    8. view a leaderboard graphically visualizing each team members' contributions, levels, and rankings.
    9. create announcements that will notify all of the members in my room through email and/or text.
    10. easily transfer my leader role to another member if I leave the room.
    11. assign moderators with less authority than me but more than the other members to my rooms.
    12. edit pending tasks assigned by the rest of the team members before approving or ignoring them.
    13. (if time allows) deduct points from team members if a task was not completed properly.

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

User Story #4

As an individual user, I would like to be able to easily set a custom profile picture and customize my profile page’s theme and banner.

4	1 Implement buttons to change profile picture and background color	2 Hrs	Ryan, Prahas
4	2 Create frontend view for invitation link	1 Hrs	Ryan, Prahas
2	3 Create frontend view for task post	3 Hrs	Ryan, Prahas
3	4 Make the created task visible on main feed	2 Hrs	Nabi, Nick, Karim

Acceptance Criteria:

- Given that the settings are implemented properly, when the user clicks their profile picture, they will be able to select one out of a list of new profile pictures, including any pictures they uploaded themselves.
- Given that the settings are implemented properly, when the user clicks on a color button, their UI will change to reflect the new color.
- Given that saving the new selections is implemented properly, when the user closes and reopens Enjoyagoals, their UI and profile picture will be the same as when they left.
- Given that the changes are made immediately, when the user clicks between profile pictures and colors, the picture and color should change on the screen.
- Given that the frontend view for invitation links is implemented, team members will be able to extract the room code using a copy button.

User Story #8

As a team member without a team leader, I would like to be able to propose tasks to be completed as well as point values for the tasks to be completed.

1	Create UI to create a new task and propose it	3 Hrs	Ryan, Prahas
4	Connect the algorithm to the UI to display all the information	2 Hrs	Nabi, Nick, Karim
1	Create UI panel to display every task in a list	2 Hrs	Ryan, Prahas
4	Connect algorithm to UI that displays each task	2 Hrs	Nabi, Nick, Karim
5	Implement button to show only tasks user needs to finish	1 Hrs	Nabi, Nick, Karim

Acceptance Criteria:

- Given that the UI for the task page is implemented, the user will be able to scroll down the page for access if the information is too big for the screen to hold.
- Given that the UI for the task page is implemented, the proposed task should appear on the main page for others to view.
- Given that the UI for the Room is implemented correctly, when the list of tasks is too large for the screen to hold, they should be able to scroll through the page.
- Given that the UI for the Room is implemented correctly, the list of tasks should initially be sorted by date of creation.

User Story #10

As a user, I would like to be able to view the tasks that need to be completed and their details (description, date created, deadline, point value).

1	Create UI to display detailed information about a task	1 Hr	Ryan, Prahas, Nick
4	Connect algorithm to UI that displays information about the individual task	1 Hrs	Ryan, Prahas, Nick

Acceptance Criteria:

- Given that the connection between the Room and the UI panel is correctly implemented correctly, when the user clicks on a task, a popup panel will appear.
- Given that the algorithm for retrieving task data is correctly implemented, the popup panel will display detailed information about the task.
- Given that the UI panel for the task's information is correctly implemented, if there is too much information for the panel to hold, the user should be able to scroll through the popup for access.

User Story #12

As an individual user, I would like to be able to join, switch between, or withdraw from different project rooms at any time after signing in.

1	Create UI to make a user input a unique room code or accept an invitation to join a room	2 Hrs	Ryan, Prahas
2	Create UI so users can switch between rooms	2 Hrs	Ryan, Prahas
3	Create UI so users can withdraw from rooms	2 Hrs	Ryan, Prahas
4	Create an algorithm so users can join, switch between, and withdraw from rooms	3 Hrs	Nabi, Nick, Karim
7	Connect the algorithm to UI so users can join, switch between, and withdraw from rooms	2 Hrs	Nabi, Nick, Karim

Acceptance Criteria:

- Given that the UI and algorithm for switching between rooms are correctly implemented, when the user clicks on other rooms, the application will direct the user to the UI of the new room.
- Given that the UI and algorithm for joining rooms are correctly implemented, when the user inputs a unique room code, the application will direct the user to the UI of the room.
- Given that the algorithm for withdrawing from rooms is correctly implemented, when the user clicks the leave button, the user will no longer participate in the room and be returned to the mainpage.

User Story #13

As a team member, I would like to be able to indicate that I am currently working on a task

1	Create UI to indicate that a task is currently being worked on by a team member	2 Hrs	Ryan, Prahas
2	Create an algorithm to update the task list so the task a team member chooses cannot be completed by someone else	2 Hrs	Nabi, Nick, Karim
3	Debug and test the algorithm to update task list	3 Hrs	Nabi, Nick, Karim
6	Connect all the algorithms to the UI	2 Hrs	Nabi, Nick, Karim

Acceptance Criteria:

- Given that the UI to indicate that a task is currently being worked on by a team member is correctly implemented, when a team member assigns themselves a task, then the task list will be updated so the task that is picked by the user cannot be completed by someone else.
- Given that the UI is correctly implemented, upon selecting a task to work on, the task should show up when the user filters their main UI to only show tasks they are working on.