Team 39 Sprint 1 Planning Document

Enjoyagoals

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

Sprint Overview

During this sprint, we hope to implement the basic functionality of our project that can fulfill the expected user stories. Since this is the foundation for our entire project, we aim to make everything bug-free, efficient, and well-organized. The frontend and backend need to be robust on their own, and interactions between them should be introduced seamlessly and function quickly.

Scrum Master: Prahas Pattem

Meeting Plan: Tuesdays/Thursdays @ 9:00pm

Risks and Challenges:

One significant challenge is that none of the team is familiar with the technology we intend to use, so we will have to spend time learning how to use them. It will be hard to gauge how long exactly this will take, but it will likely take a significant chunk of our time, leaving less to be spent on working on the project. Also, linking all of the separate components together and testing each functionality will also take a large amount of time. Again, given that this is the first sprint and all of our future work will be built around what we finish now, it is of crucial importance that we implement everything successfully, which means we will likely have to dedicate more time to debugging and testing edge cases for our future sprints to be successful.

Current Sprint Detail

User Story #1

As an individual user, I would like to be able to register for an Enjoyagoals account.

#	Description	Estimated Time	Owner
1	Create UI panel for user registration	2 Hrs	Ryan, Prahas
2	Create an algorithm to obtain all the necessary information from the new user to successfully register	2 Hrs	Karim, Nabi, Nick
3	Debug and test the algorithm that obtains all the necessary information from the UI	3 Hrs	Karim, Nabi, Nick
4	Create an algorithm to send the new user data to the database	2 Hrs	Karim, Nabi, Nick
5	Debug and test the algorithm that sends the data to the database	3 Hrs	Karim, Nabi, Nick
6	Connect the algorithms to the UI so new users can register successfully	2 Hrs	Karim, Nabi, Nick

Acceptance Criteria:

- Given that the UI for registration is correctly implemented, when the user accesses the registration page, they can start the registration process.
- Given that the username validation process is correctly implemented, when the user inputs a certain username, the program will check whether or not the username is taken.
- Given that the authentication of the email address is correctly implemented, when the user inputs the email address, the application will verify the email address' genuinity.
- Given that the user has correctly inputted all fields for registration, when the user clicks the register button, the application will send email confirmation to the email address of the user and direct the user to the homepage.

User Story #2

As an individual user, I would like to be able to login and manage my Enjoyagoals account, and reset my password.

#	Description	Estimated Time	Owner
1	Create UI panel for login screen & account management screen	3 Hrs	Ryan, Prahas
2	Create UI screen for resetting password	2 Hr	Ryan, Prahas
3	Test and debug calls to backend	2 Hrs	Karim, Nabi, Nick
4	Create algorithm to manage requests for login and account management	4 Hrs	Karim, Nick, Nabi
5	Implement hashing function to handle passwords in database & login attempts	3 Hrs	Nick
6	Create algorithm to send a password recovery email to user	3 Hrs	Karim, Nabi, Nick
7	Test and debug backend algorithms	5 Hrs	Karim, Nabi, Nick

Acceptance Criteria:

- Given that the login screen is implemented properly, when the user enters correct login details, they will be signed into their account and brought to their home page.
- Given that the login screen is implemented properly, when the user enters incorrect login details, they will be shown a message that incorrect details were entered and they need to try again.
- Given that the login screen is implemented properly, when the user clicks the “forgot password?” button, they will be prompted for the email associated with their account and then sent an email with a link to a page they can reset their password from.

User Story #3

As an individual user, I would like to be able to register using my google account if I do not have an Enjoyagoals account.

#	Description	Estimated Time	Owner
1	Add the “register using google account” option to login screen	0.5 Hrs	Ryan, Prahas
2	Create algorithm to interface with Google Login API	2 Hrs	Karim, Nabi, Nick
3	Create algorithm to populate database with user information from linked Google account	2 Hrs	Karim, Nabi, Nick
4	Test and debug algorithms	3 Hrs	Karim, Nabi, Nick

Acceptance Criteria:

- Given that the “register using google account” button is implemented properly, when the user clicks the button, they will be connected to Google’s login interface via Google’s API.
- Given that logging in via Google account is implemented properly, when the user successfully logs in using the Google login interface, they will then be logged into their Enjoyagoals account and brought to their home page.
- Given that the backend algorithm that interfaces with the Google API is implemented properly, the user’s information (email, username, etc.) will be populated into the database when the account is first created.

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.

#	Description	Estimated Time	Owner
1	Find set of default profile pictures	0.5 Hrs	Ryan, Prahas
2	Implement ability to upload any image as a profile picture	3 Hrs	Ryan, Prahas
3	Find set of colors for background to use	0.5 Hrs	Ryan, Prahas
4	Implement buttons to change profile picture and background color	2 Hrs	Ryan, Prahas
5	Implement new profile picture and background	1 Hrs	Ryan, Prahas

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.

User Story #5

As an individual user, I would like to be able to display my contribution to the room I am currently in, my email account and phone number on my Enjoyagoals account alongside other personal information for others to see.

#	Description	Estimated Time	Owner
1	Create UI to display all the personal information	4 Hrs	Ryan, Prado
2	Create an algorithm to obtain all the information you need to display	3 Hrs	Karim, Nabi, Nick
3	Debug and test algorithm	3 Hrs	Karim

Acceptance Criteria:

- Given that the UI to display all the personal information is created, a user will be able to see the personal information email account of every other member.
- Given that the UI to display all the personal information is created, a user will be able to see how many tasks did every other team member complete.
- Given that the UI to display all the personal information is created, a user will be able to see what level every other team is at.

User Story #6

As a team leader, I would like to be able to invite team members to my team at any point.

#	Description	Estimated Time	Owner
1	Create a user controller to see if a team member is a leader	0.5 Hrs	Nabi, Nick, Karim
2	Create algorithm to generate a unique link for invitation	4 Hrs	Nabi, Nick, Karim
3	Expire the link after the specific period of time or if the user joined the team	3 Hrs	Nabi, Nick, Karim
4	Create frontend view for invitation link	1 Hrs	Ryan, Prahas
5	Debug and test algorithm	4 Hrs	Nabi, Nick, Karim

Acceptance Criteria:

- Given that the referral link algorithm is created, people who are not referred won't be able to join the group.
- Given that the user controller is implemented, only the team leader will be able to generate a link for a referral.
- Given that the frontend view for invitation links is implemented, the team leader will be able to extract the referral link using a copy button.

User Story #7

As a team leader, I would like to be able to create tasks to be completed with the details of each task clearly displayed, such as the date & time created, who is assigned the task, the deadline, the point value, etc.

#	Description	Estimated Time	Owner
1	Create controller logic for task creation	3 Hrs	Nabi, Nick, Karim
2	Create frontend view for task post	3 Hrs	Ryan, Prahas
3	Make the created task visible on main feed	2 Hrs	Nabi, Nick, Karim
4	Create a user controller to see if a member of team is a leader or not	1 Hrs	Nabi, Nick, Karim
5	Add the task details with a unique id to the database	2 Hrs	Nabi, Nick, Karim

Acceptance Criteria:

- Given that the UI for the task page is implemented, users will be able to scroll down the page for access if the information is too big for the screen to hold.
- Given that the User controller is implemented, only the team leader will be able to create tasks.
- Given that the Task controller is implemented, the team leader won't be able to create a task if some of the required information is left blank.

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.

#	Description	Estimated Time	Owner
1	Create UI to create a new task and propose it	3 Hrs	Ryan, Prahas
2	Create an algorithm to send the proposed task to the database	2 Hrs	Nabi, Nick, Karim
3	Debug and test algorithm	4 Hrs	Nabi, Nick, Karim
4	Connect the algorithm to the UI to display all the information	2 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 verify or reject.
- Given that a leveling system is implemented properly, completed tasks should grant the assigned point values to the users who completed it and contribute to their level progression.

User Story #9

As an individual user, I would like to be able to view all of the tasks I need to finish in a comprehensive list after logging in.

#	Description	Estimated Time	Owner
1	Create UI panel to display every task in a list	2 Hrs	Ryan, Prahas
2	Create algorithm to obtain information about each task from the database	3 Hrs	Nabi, Nick, Karim
3	Debug and test algorithm with calls to the backend	4 Hrs	Nabi, Nick, Karim
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 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.
- Given that the UI for the Room is implemented correctly, a button should be able to filter out tasks that are completed and tasks that the user has not accepted.

User Story #10

As an individual user, I would like to be able to easily check the details of each individual task (description, date created, deadline, point value).

#	Description	Estimated Time	Owner
1	Create UI to display detailed information about a task	1 Hr	Ryan, Prahas
2	Create algorithm to obtain information about the task from the database	4 Hrs	Nabi, Nick, Karim
3	Debug and test algorithm	4 Hrs	Nabi, Nick, Karim
4	Connect algorithm to UI that displays information about the individual task	1 Hrs	Nabi, Nick, Karim

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 #11

As a team member, I would like to be able to view the tasks to be completed, their point values, and their status.

#	Description	Estimated Time	Owner
1	Create UI to view all the tasks that need to be completed, their point values, and their status	1 Hrs	Ryan, Prahas
2	Create an algorithm to obtain information about all the tasks	4 Hrs	Nabi, Nick, Karim
3	Debug and test the algorithm	4 Hrs	Nabi, Nick, Karim
4	Connect UI to algorithm that displays all the tasks to be completed and all the relevant information	4 Hrs	Nabi, Nick, Karim

Acceptance Criteria:

- 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.
- Given that the UI for the Room is implemented correctly, a button should be able to filter out tasks that are completed.

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.

#	Description	Estimated Time	Owner
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
5	Create an algorithm so the database is updated whenever a user joins a new room or withdraws from a room	2 Hrs	Nabi, Nick, Karim
6	Test and debug algorithm	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 accepts an invitation from another user or 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.
- Given that the algorithm of the database is implemented, all other team members will get notified whenever a user joins a new room or withdraws from it.

User Story #13

As a team member, I would like to be able to 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.

#	Description	Estimated Time	Owner
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
4	Create an algorithm to send the new task list to the database	2 Hrs	Nabi, Nick, Karim
5	Debug and test the algorithm to send the new list to the database	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.
- Given that the add notes functionality is implemented, any other team member will be able to see notes that the person made.

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.