Team 39 Sprint 2 Retrospective

# Enjoyagoals

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

# What went well?

During this sprint, we made much better progress, and for the most part we were able to catch up on everything we did not finish in Sprint 1, as well as complete almost all of our goals for Sprint 2. Users can now join, switch between, and leave rooms, see tasks for each specific room, see a calendar with their task deadlines, view their level and their position on a room's leaderboard, submit tasks as complete and provide comments and upload an accompanying file, and view the progress of their currently selected room.

## 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 |

**Completed:**

There is a button in the header that displays the current date, and when a User clicks on it, they are shown a calendar that highlights days they have task deadlines for in red, which they can then click to view a list of those tasks.

## 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 |

**Completed:**

Users are awarded points when they submit a task as “Completed,” which is tracked in the database. Then, when a user looks at the Leaderboard on the right side of the homepage, they will see the usernames and corresponding levels of the users in the room they are currently viewing. Additionally, when a User views their Profile page, their level is also shown beneath their profile picture.

## 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 |

**Completed:**

The header contains a progress bar that indicates what percentage of all of the tasks in a Room are marked as “Complete,” showing Users the overall progress towards their goal(s). Under this progress bar, the number of remaining tasks is shown. There is also a “refresh” button next to the progress bar that a User can click to ensure that it is updated.

## 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 |

**Completed:**

When a User goes to their Pending tasks list, clicks on a task, and indicates that they want to submit it as complete by clicking the “Submit” button, a prompt shows up that requires a User to upload at least 1 file.

## 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 |

**Completed:**

When a task is created, a User can assign a point value to the task.

## 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 member 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 an algorithm to modify the database when a task is marked as complete | 2 Hrs | Karim, Nabi, Nick |
| 4 | Debug and test algorithm for modifying the database | 2 Hrs | Karim, Nabi, Nick |

**Completed:**

When a User clicks on a “Completed” task to view its details, they can see the comments that the User who submitted the task entered when they submitted it.

## 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 |

**Completed:**

When a User indicates that they want to submit one of their “Pending” tasks as completed, a prompt shows up that requires a User to write comments on the work they’ve done (minimum 7 characters). These comments are then displayed whenever any User in the room views the details of the Completed task.

## 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 |

**Completed:**

In any of the 3 lists of tasks (Completed, Pending, and Team), a User can click on any of the tasks and view their details.

## 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 | Create an algorithm to check if the task passed the deadline | 2 Hrs | Karim, Nabi, Nick |
| 5 | Debug and test algorithm for checking the deadline of each task | 2 hrs | Karim, Nabi, Nick |
| 6 | If the task is valid, add the task to the “completed tasks” database | 2 Hrs | Karim, Nabi, Nick |

**Completed:**

After a User accepts a task, they then see the task in their “Pending” tasks list. From here, they can click on a task, and click the “Submit” button to indicate that they wish to mark this task as complete. Additionally, there is a check to see when the task was completed and when the deadline for the task is, and if the task was not completed before the deadline, it is distinguished from “Completed” tasks by being marked as “Missed.”

## 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 |

**Completed:**

A User can see the tasks that have been completed in the current room by either clicking the “Completed” button or the “Team” button and seeing the tasks that are marked in green.

## 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 |

**Completed:**

When a User views the “Team” tasks list, they can easily see the color-coded statuses of all of the tasks of the current room, including which tasks are “Pending.” They can then click on the “Pending” tasks to see the details of the task, including which User is assigned to work on it.

## 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 |

**Completed:**

There is a “Completed” button on the homepage that allows a User to view a list of all of the tasks in the current room that have been marked as “Complete.” When a User clicks on one of these tasks, they are shown an information panel that shows the details of the completed task, including the deadline, the date of completion, the User that marked the task as completed, and the comments that the User entered when they submitted the task.

## 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 |

**Completed:**

There is a Leaderboard for each room that is shown on the right side of the homepage, which displays the usernames and corresponding ranks & levels of the Users in the current room.

## Sprint 1 - 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 |

**Completed:**

There is now a frontend view for getting a Room's invite code to invite new Users, as well as a frontend view to see all created tasks for the current room.

## Sprint 1 - 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 |

**Completed:**

Users can now view all of the tasks that have been created in the current room.

## Sprint 1 - 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 |

**Completed:**

Users can now view the details of all of the tasks that have been created in their current room.

## Sprint 1 - 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 |

**Completed:**

The sidebar has a button that brings up a panel with a text field for a Room code. If a User inputs a valid, unique Room code, they will be added to that Room and it will appear in their sidebar. Users can click on the icons corresponding to their rooms to switch the room they are currently viewing details of, which will update the information shown in the lists displayed when clicking on the Completed, Pending, and Team buttons on the right side of the homepage. Finally, there is a button in the Header near the Log Out button that allows a User to leave the room they are currently viewing.

## Sprint 1 - 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 |

**Completed:**

Users can now assign themselves to a created, “Unassigned” task by clicking on the task and clicking the “Accept” button. When this is done, other Users can see that the task is now marked as “Pending,” and can see who is assigned to the task by clicking on it and viewing its details.

# What did not go well?

In general, we did not distribute our work evenly over the three week period. We spent a disproportionate amount of time in the final week of sprint 2 because we did not make as much progress as we wanted to during the first two weeks of the sprint. Additionally, we could have distributed work better among all the team members to better balance our overall pacing and individual workloads.

# How should you improve?

Overall, while our second sprint went much smoother than the first, there were still some points for improvement.

First, while our time management as a group was significantly better than the first, we could still improve. As mentioned before, many of us worked for up to twice as long as usual on the project during the week leading up to the second sprint review. Had we paced ourselves better during the first two weeks, this would not have been as much of an issue. Next time, we could try to tackle harder user stories earlier in the sprint rather than the low hanging fruit to lessen the workload in the later stages of development.

The second thing we could improve is our organization of our Sprint Planning document, as we ended up re-submitting it, and even then, we demoed our application out of order compared to the order of our user stories in our planning document. By better organizing this, we can save time and effort that we have had to spend on reorganizing what we need to get done and in what order. Additionally, this will make our planning document easier to follow when we demo our application.