ProjExpress README

Contents:

Proposed Level of Achievement

Apollo 11

Motivation

Whenever a project is given as an assignment in class, most groups form a group chat and do all the discussion online or on Google Docs. At the same time, the teacher who may be mentoring the project is often left out of the project's progress, and if there are doubts on the details of the project, or even poor time management from the students, they may end up submitting work that is not acceptable.

Ideally, groups of students should be able to coordinate on a common platform without relying on each individual to remember their own workload or timetable. Furthermore, teachers mentoring the project should be allowed to be part of the process, to ensure students are on the right track and doubts can be clarified quickly.

Especially for projects that may span months and require close mentorship, it would be great if the whole process was kept together in a single central area and broken down into simpler parts so everyone is clear of the progress of the project at any time, including the mentors themselves.

In a classroom setting, it would also be ideal if mentors could place all project groups under them into a common group.

Aim

We aim to consolidate the process of completing a project into a single platform so that it is easy for a mentor to provide guidance and easy for group members to keep on track at all times, maintaining communication between the mentor and the group.

Changelog

We have added the following features for Milestone 3, in addition to improvements on our features from Milestone 2 below.

Comments

Group members and mentors can add comments to a task. Each comment consists of a title and the main body of text, and will help group members communicate and give feedback on their own tasks.

Subtasks

Group members can also add subtasks to a main task given by their mentor. These subtasks will ideally be used to help break down the given task into smaller chunks for members to distribute and complete among themselves.

Back to top

Only up to 1 layer of subtasks are allowed, or in other words, subtasks cannot have their own subtasks.

Closing a class

Upon completion of a class, the creator of the class can choose to close the class, officially marking the class as completed successfully. Upon doing so, the class will be locked and can no longer be modified in any way. For example, no new users, groups, tasks, and announcements can be added to the class.

However, the class data will still be stored in the database, and users will still be able to navigate into the class to view the details and submissions.

Pagination

We have added pagination on all lists available, such as the user's list of classes, groups, and each class's list of users, etc.

This should help reduce long lists down to simpler pages of smaller size, reducing the need to scroll while allowing a user to navigate through the list easily without being intimidated by the long list of items.

Adding / Removing of users to classes and groups

Previously, mentors could only add users one at a time to the class. Now, mentors are able to use a csv file containing all user emails that they want to add to the class, greatly reducing the amount of time needed to add students.

The format for adding users to groups has also been changed. Now, you no longer need to state the user role when adding a user to a group (their group role would just be their class role). Instead, to make things easier, students/mentors that are eligible to be added to the group will be listed in a menu, so mentors will no longer need to check if a student is already in another group before adding them. Also, this new format allows adding of multiple users at a time, unlike

Invite Code

Additionally, users can also join classes via invite codes. There are two invite codes per class: one invite code for students, and another for mentors, so depending on the invite code the user inputs they can join as either a student or mentor. These codes are only visible to mentors of the class, so mentors will be able to choose how to distribute these codes. (Via email, a messaging app, or other forms of communication)

Also, mentors can choose to generate new invite codes if needed. (For example, if the invite code is leaked to users that are not supposed to be in the class)

New group creation method

Previously, users could only create groups one at a time. Now, mentors are able to create up to 100 groups in a single click. These groups will be automatically created, with the name format of “Group X”, where X is some number such that all created groups have a unique name. (No two groups will have the same group number)

Distribution of users (students/mentors) into groups

On class creation, you can now specify the size of all groups within the class (from 1 to 10 group members). This will limit how many students can be added to each group.

In addition, mentors can now automatically distribute users into groups. If there are not enough groups for students to be distributed into, an error message will be displayed, mentioning how many more students do not have a group.

User Settings

We have also added a simple profile page for each user, which displays their username. From the profile page, users can also change their username, as long as it is unique.

We also wanted users to be able to change their password, but due to time constraints we focused on other features instead.

Features

These features were completed in Milestone 2, but we have also made improvements to them. In summary, classes have an invite code and can be closed, groups now have an auto-distribution function for students, and users can now add subtasks and comments to their tasks.

Login / Registration

Users can create an account with an email address. From there, they can start creating classrooms for themselves and their students, or simply wait to be added to a classroom by their mentor.

Dashboard

Upon login, users will be greeted by a dashboard which showcases a list of tasks (from all classrooms) that they have due soon. At the same time, they will also be able to see recent announcements made by their mentors and any comments made on their tasks.

Classrooms

Any user can view all classrooms that they are involved in. From there, they can also create their own classroom if needed. If they do so, they will automatically be considered a mentor of that class.

Back to top

Users can also join any class, provided they have an invite code. Each class has 2 different invite codes, one for inviting students, and the other for inviting mentors. Only mentors will be able to see the codes.

From the many tabs of a class page, a user can view all announcements, groups, tasks and other users that are members of the class. If the user is a mentor of that class, they have access to the task framework (see below), and are able to make announcements, manage groups, and manage users of the class. They will also be able to access the class settings, which contain the class invite code and the ability to close the class after it is completed.

On the other hand, if the user is a student, they can only view announcements and other members of the class, and they will be redirected to a page for their group if they are in one.

Announcements

Mentors can make announcements to pass information quickly to all students from the class announcements tab. Mentors simply need to fill in a title and the main body of their announcement, and can post it to all users in the class.

For students, they will simply be able to view a list of announcements related to that class made by their mentors, sorted by newest to oldest. Mentors will also be able to see these announcements.

Groups

Mentors can create groups in a class through the Project Groups tab. From the Project Groups tab, a mentor can view a list of all groups that a class has, and assign students to a group, either by manually adding them or by using the auto-distribution function, which will distribute all students without a group by assigning them to existing groups that are not full yet. Note that any student cannot be in multiple groups in the same class.

For students, the group page will simply be a summary of the group that they belong to, provided they are assigned one.

Task Framework

The task framework allows mentors of a classroom to create a task framework for the class, which is a list of tasks made for every group to complete. Mentors can set a simple task name, give detailed instructions in the description, and a due date for all groups to follow. This allows mentors to set a pace while ensuring all groups (and by extension students) know what they should be doing by when.

Once a mentor has created a task framework, all groups will receive their own copy of the list of tasks based on that framework given.

Tasks

A task is simply an assignment that has a name, a description (which should ideally contain detailed instructions on how to complete it), and a due date. Mentors will provide tasks for students to complete through the task framework.

Students cannot create their own tasks, but they can view those tasks in the task page of that classroom.

From the task list within their group page, each student will see the list of tasks from that class that are due. Mentors can also do the same by going into any group's main page.

From there, users can add subtasks, make comments, and add submission links to each task. Students will be able to mark any subtask or task as completed by simply clicking on their respective checkbox. Note that a task cannot be marked as completed until all subtasks below it are marked as completed.

Application

Hosted on https://projexpress-orbital-2021.herokuapp.com/
Note: There may be a small amount of lag as the app is hosted in the US.

Github link for local copy: https://github.com/danielsimre/orbital-2021-app/ If you have signed up before, please sign up again as the database was reset during development.

Software Engineering Practices

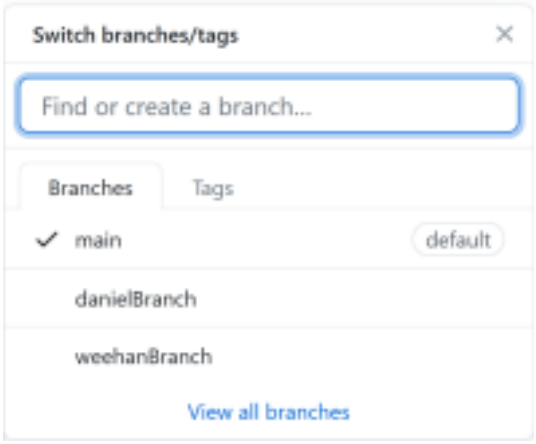
Delegation of Tasks

To manage our work, we used a Google Document for documentation and recording. Tasks to be done are segregated by meetings and are assigned under each person during the meeting, where we discuss the next features to implement, bugs to be fixed and merge our work on Github.

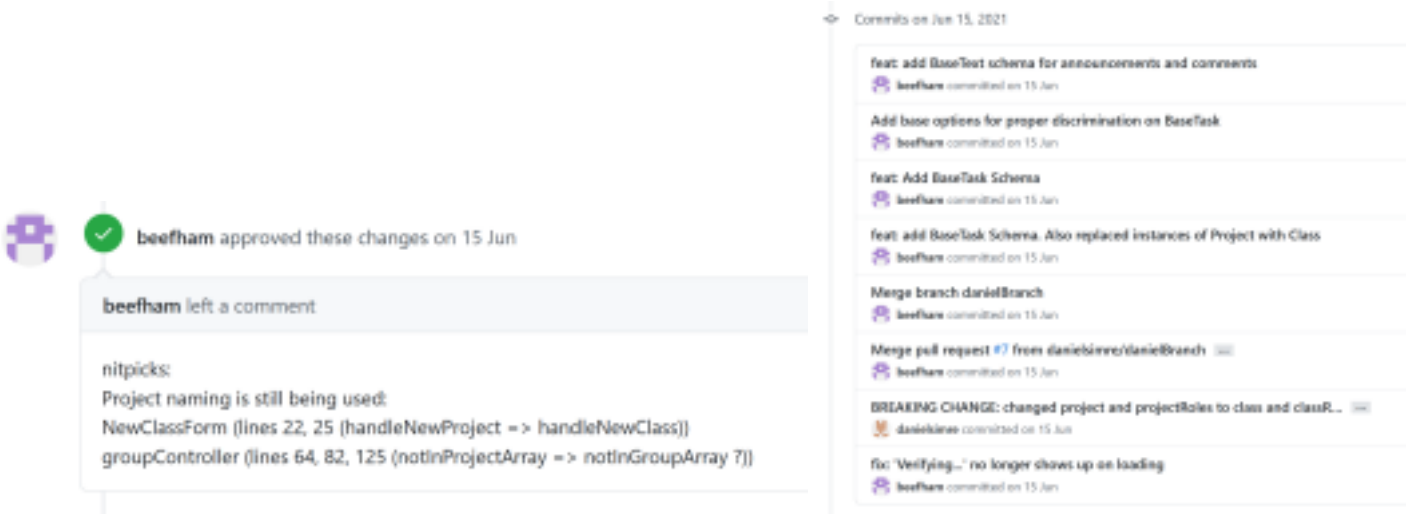
From there, we can easily go back to the shared document to view our own tasks and make any notes if needed, while the other person can also view the changes and notes made.

Back to top

Version Control



We created a GitHub repository for our application for version control. We decided to manage our work by pushing each member's code to separate branches (danielBranch and weehanBranch) so that changes that one person makes will not affect the other. To update the main branch, each member creates a pull request, and then the other member reviews the code and merges the pull request to update the main branch. This prevents buggy code from being pushed to the main branch, as well as ensuring that both members are aware of all changes that are being made to the application. From there, we can pull from the main branch and continue our work as usual.



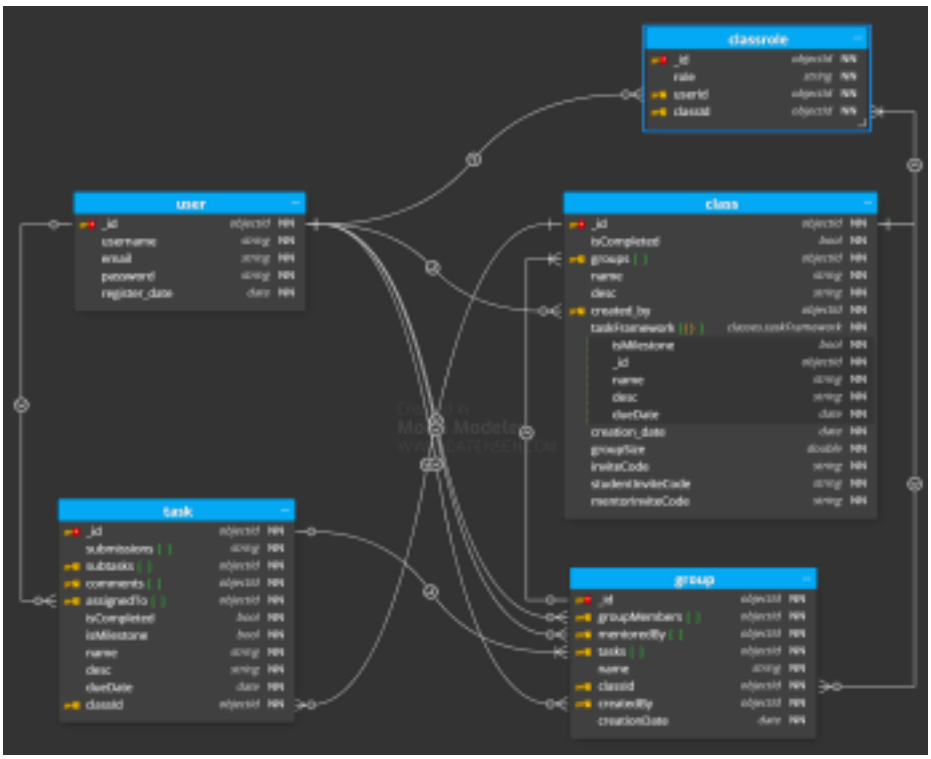
In addition, we decided to adhere to the Conventional Commits (https://www.conventionalcommits.org/en/v1.0.0/) and Conventional Comments (https://conventionalcomments.org/) standard. This makes it clearer for both members of the

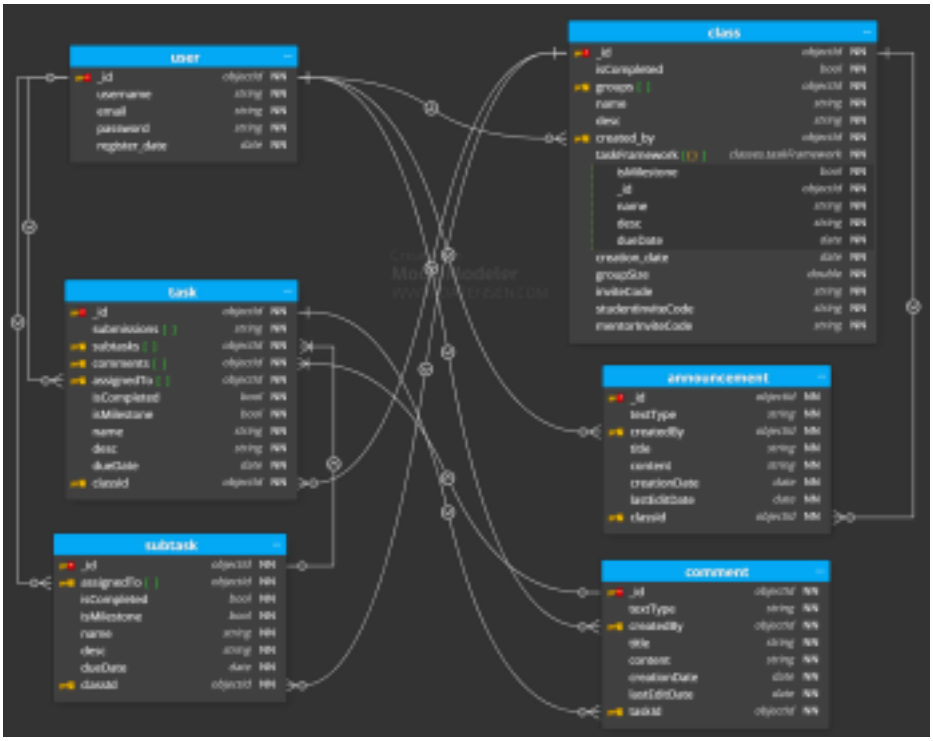
Back to top

group to understand what changes are being made with each commit, and differentiating urgent requests from suggestions on potential improvements for comments.

Entity Relationship Diagram

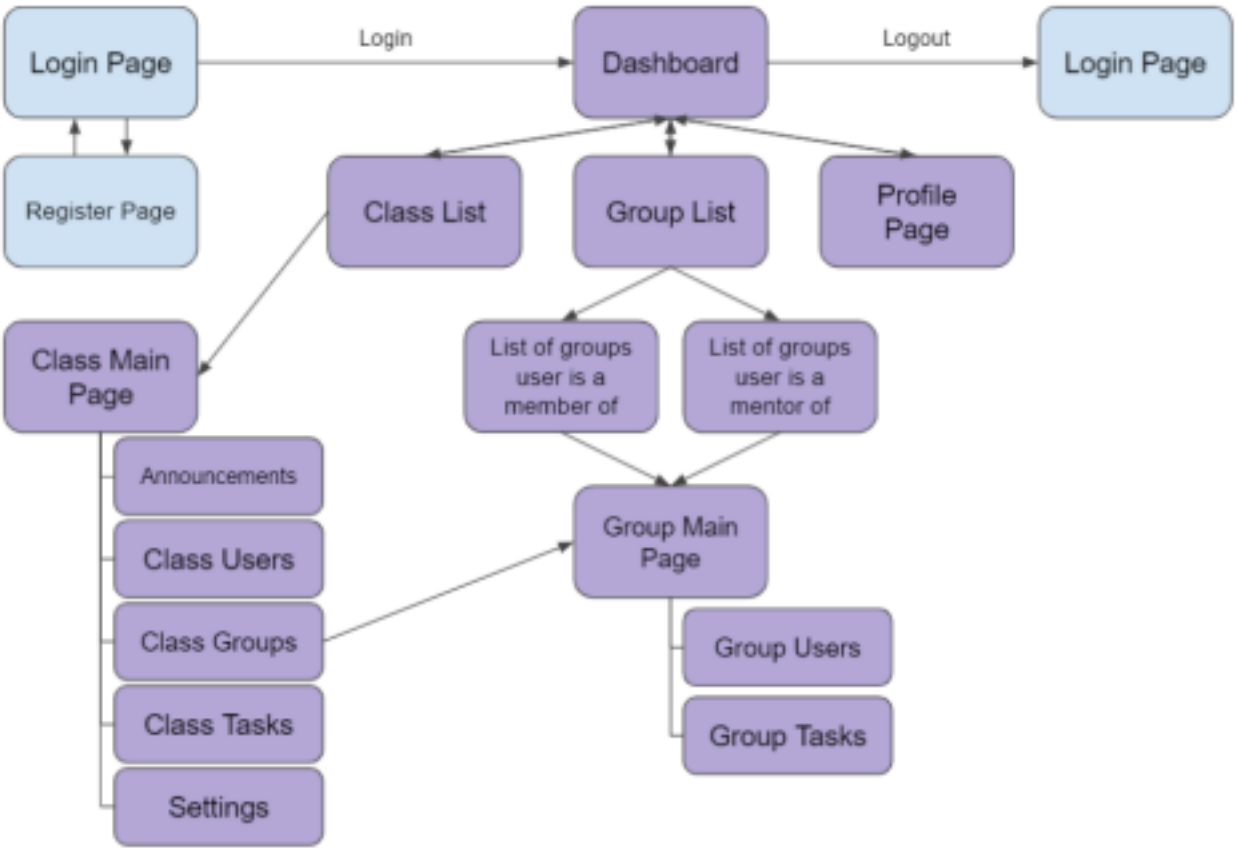
Note: Diagram is split into parts for ease of viewing (All entities are in the same database)





Back to top

User Flow (Page layout)



Back to top

Testing

We have performed manual integration and system testing on both the front-end and back-end of our application. To simplify this document, we have placed both logs on separate documents

each, linked below.

Backend API

We have used Postman to help us test our API. The sheet below is a log of the different endpoints that we have tested and their results.

Backend API Testing Log

User Interface

User Interface Testing Log

Issues Faced / Challenges Encountered

1. We had an issue where refreshing in the app (F5) causes the app to reload and throw the user to the home page (if they are logged in) or login page. This is because the user's login state is lost when the application reloads. We have resolved this issue, and in addition, users will also be correctly redirected if they try to access a class or group that they do not have access to.

2. There are also a few issues with how items are displayed when the user is using the application in a small window, or have zoomed in to very high levels. We will continue to look into possible solutions with regards to that.

3. We faced an issue where upon removing a user from a group, the tasks assigned to the user were not updated. This was due to an oversight on handling the removal of the user from the group, and has been fixed. A similar issue occurs for removing a user from a class, and both issues have been resolved by adding an additional step to query tasks and remove the user to be deleted.

4. We have also faced an issue with the naming system provided when creating groups. For example, if a mentor has created 3 groups, the system would name them Group 1, 2 and 3. If another mentor in the class who does not have access to those groups (and are therefore unable to see these groups) attempts to create groups, the system would attempt to create Group 1, 2, and 3 again, which would cause a naming clash. We have fixed the issue by ensuring that a full list of group names are checked before creating the names of the new groups.

5. We found a bug where users were allowed to enter a space (' ') as an input to most text fields, which would not be caught by the text fields as they were technically not empty. We have fixed the issue by using regex to sanitise such inputs, as well as adding checks and trimming strings on the backend as an extra layer of validation.

6. We faced an issue where some API calls manage to bypass the check of whether the class is completed or not, which allows users to modify class data of a class which has been marked as completed through backend API calls. We found that the problem was

due to then() calls returning too early, and we have resolved this issue.

User Testing / Feedback

We have also tested the features of our app by giving a link to our hosted web application to a group of testers. The users were not given instructions on how to navigate the UI when testing the features in the app. Most users found the website UI to be intuitive and simple to navigate in general. We have also received a lot of valuable feedback and suggestions, some of which is shown below.

- **Ability to add multiple users into a class at once, rather than one at a time.** This is a good suggestion, and we have worked on this by allowing a mentor to submit a csv file of emails to add multiple users at once, or by using the invite code function to get students to join by themselves manually.

- **Long paragraphs of texts like task titles / announcement text mess up the webpage.** This is very useful feedback, and we will work on trimming text that may be too long post Milestone 3.

- **Move class info that belongs to classes that have been closed to another page to prevent clutter.** This is a good suggestion, as it is likely that the user interface will get very cluttered. This is something we will look into.

- **Group creation allowed non-numeric inputs.** This issue has been fixed by sanitizing the inputs via regex, so only integers between 1 and 100 are valid.

- **It is not clear what links are accepted as valid for submissions.** We checked for valid links by using the URL constructor in Javascript, which caused issues with links that do not start with "https://" or “http://”. One possible solution is to append “https://” to links that do not start with it, so that the url constructor will accept it as valid.

- **Some button names are not clear in their purpose / are a bit misleading. (Example: "Create New Task Framework", "Project Groups" in class tabs for a student).** This is a good suggestion, and we will give clearer names that more accurately describe the buttons after Milestone 3.

- **Reloading any page while logged in causes the login page to flicker for a moment, before redirecting to the home page.** This is related to the aforementioned refresh bug, and is currently fixed.

- **If a class is marked as completed/closed, groups from that class could be removed from the View My Groups tab to reduce clutter.** This is a good suggestion, and in fact it can be applied to other things such as tasks, comments and announcements. However, it may be better to put these groups in a separate tab, rather

than not displaying them at all. We aim to implement this after Milestone 3.

- **It would be nice if clicking an announcement or comment from the main dashboard redirects the user to the announcement or comment.** This would be a good quality of life improvement for the user. We aim to implement this after Milestone 3.

User Guide

This is the link to our user guide. Please read it as it contains valuable information and may help resolve some issues you might face. Thank you!

User Stories

Features are categorised into three groups: Must Haves, Should Haves, and Nice to Haves, in order of decreasing priority.

User Roles:
1) Users (User base of the application in general)
2) Mentors (Teachers or teaching assistants in charge of mentoring the class)
3) Group Members (Students enrolled in the class)

---

1. (Must Have) As a user, I want to log in to an account so that I can keep track of classes and tasks that I am involved in.

2. (Must Have) As a mentor, I want to be able to quickly disseminate a list of project tasks/requirements, project milestones/deadlines, as well as announcements to every student in the class.

3. (Must Have) As a mentor, I want to be able to keep track of each groups' progress so that I can help groups that are struggling with the workload or other issues.

4. (Must Have) As a group member, I want to be able to create tasks and ensure the workload can be distributed evenly amongst the group, such that the list of project requirements given by the teacher can be met.

5. (Must Have) As a group member, I want to be able to assign tasks to fellow group members and set a deadline, so that everyone in the group knows what to do and by when.

6. (Must Have) As a group member, I want to be able to keep track of my share of the tasks so I can complete a project on time and as smoothly as possible.

7. (Must Have) As a group member, I want to be able to upload files and consolidate relevant links so that I do not lose track of what has been done, as well as update our teacher/mentor

with what we have done so far.

8. (Must Have) As a user, I want to be able to tell at a glance which tasks are about to be due.

9. (Should have) As a user, I want to be able to tell at a glance if there are any upcoming meetings or events.

10. (Should Have) As a mentor, I want to be able to give feedback at each milestone so that each group is aware of what to fix or modify to ensure that the project is completed satisfactorily.

11. (Nice to Have) As a mentor, I want to be able to automatically create random groups based on certain parameters or requirements, so that I do not need to waste time manually creating each group.

12. (Nice to Have) As a mentor, I want to be able to allow students to create their own groups based on certain parameters or requirements, so that I do not need to waste time manually creating each group.

13. (Nice to Have) As a group member, I want to be able to discuss and chat with my group members about the project.

14. (Nice to Have) As a group member, I want to be able to give feedback on my fellow group members after the completion of a large group project.

15. (Nice to Have) As a mentor, I want to be able to consolidate which groups I am consulting with and when into a single place so that I can prevent scheduling clashes.

16. (Nice to Have) As a group member, I want to be able to easily arrange project consultations with my teacher/mentor according to their given schedule.

17. (Nice to Have) As a user, I would like to have the option to be notified by email about upcoming due tasks, meetings or events.

Back to top

Project Scope

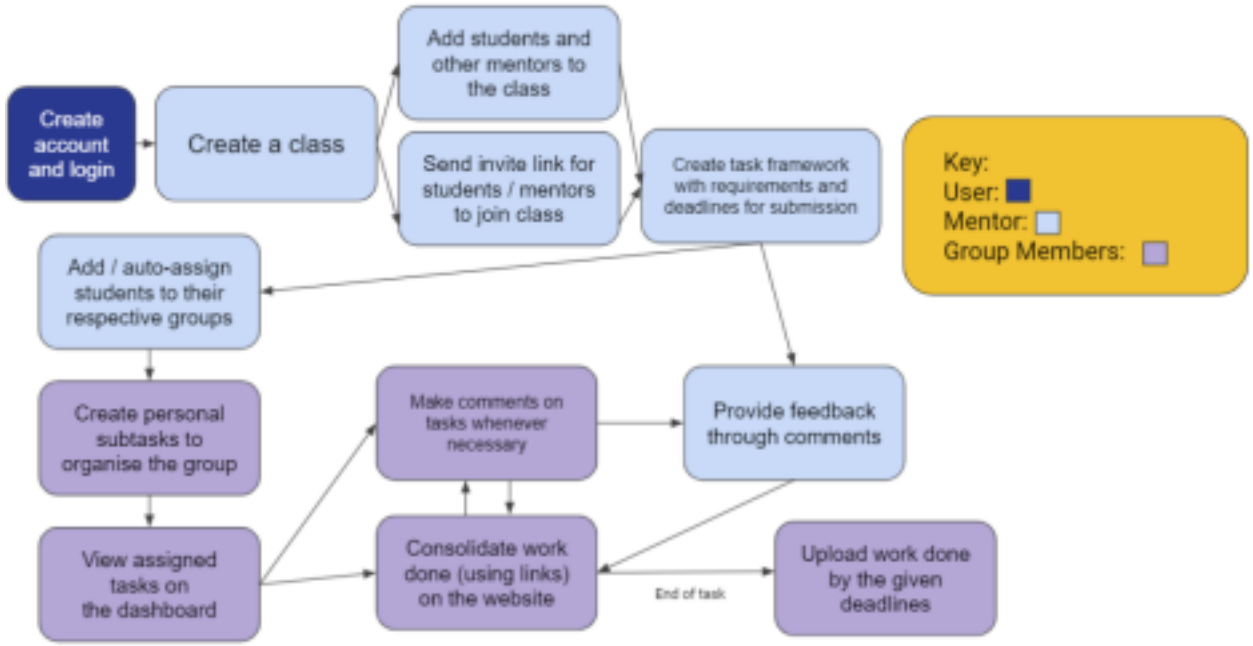
The project is a web-application that provides a platform for a mentor to create a classroom and place their students into groups. Then, groups can coordinate their tasks for a group project assignment and allow their mentor to be able to stay involved and give feedback any time.

Differences from Competing Platforms/Applications

1. **Google Classroom**
    a. Google Classroom lacks the features of creating groups and group assignments within a class of students
    b. Google Classroom also focuses more on the homework and learning rather than project assignments.
2. **Edmodo**
    a. More marketed towards K-12 students, whereas our app is also targeted towards University students
    b. Does have a task list, but it is an individual one and not a group one
3. **Project Management Apps (e.g. Trello, Basecamp, Clickup)** a. Have a lot of features that are not necessary for the scale of school projects, so it can be intimidating for students to use even if the core project management features are beneficial
    b. Free versions of these typically do not allow for the management of many groups at once

Our app focuses on a few key features that are useful for project management to allow groups to work more efficiently while streamlining the communication process between groups and teachers/mentors.

Program Flow



Project Log

Task Date	Daniel Ng Remarks Sim Wee (hrs) Han (hrs)
Peer Review 3/7	3 3
Team Meeting: Discussion on features for Milestone 3 5/7	1 1
Development of features: Removal of users and pagination of lists 5/7 - 10/7	4 4

S/N

1

2

3

4 Team Meeting: Discussion on features for Milestone 3

5 Development of features: Profile page and invite code

6 Team Meeting: Discussion on features for Milestone 3 and progress check
10/7 1 1

https://github.com/danielsimre/orbital-2021-app/commit/bf7b408d6758aae08d25c6679c753763f62e3a47

10/7-12/7 6 6

12/7 1 1 Merged progress:

7 Development of features: Closing of 12/7-16/7 9 9

Back to top

class and adding of multiple users and groups at once		
Team Meeting: Discussion on features for Milestone 3 and progress check	16/7	1 1 Merged progress: https://github.com/danielsimre/orbital-2021-app/commit/d1195e992d892d2ef62235c8ce21c768f9e8310e
Development of features: Ability to remove users / leave from a group	16/7-18/7	3 3
Team Meeting: Discussion on features and documentation for Milestone 3	18/7	1 1
Development of features: Ability to auto-assign users to groups, look for users to test	18/7-21/7	4 4
Team Meeting: Discussion on documentation, merged work done and pushed to heroku deployment Conducting User Testing 21/7	21/7	1 1 Merged progress: https://github.com/danielsimre/orbital-2021-app/commit/68ff73defa11bc0024f3472ef2dcf98b3ab3ffbd 2 2 Tested other apps and received feedback for ours

Debugging and refactoring of code 21/7-24/7	9 0 Bugs fixed over multiple commits, merged in this commit: https://github.com/danielsimre/orbital-2021-app/commit/d0ed5f700d6ad9e830c5cfc098772108ddf5664c
Application Testing and documentation 21/7-24/7	0 9 Tested front end interface + back end API
Team Meeting: Recording of video and finalizing documentation 26/7	1 1
Hours for Milestone 3:	47 47
Hours for Milestone 1:	57 57
Hours for Milestone 2:	60 60
Hours for all Milestones:	164 164

8

9

10

11

12

13

14

15

16