

Sprint 3 Planning Document

Eric Aguilera, Justin Bonner, Carson Harmon, Dan Zheng

1 Sprint Overview

For our third sprint, we will focus on improving our frontend, refactoring our backend, and adding some small additional features. Currently, the primary features of our application are completed, but our app lacks polish and “glue”: some parts of our application that logically should be connected are not.

By the end of our sprint, users will be able to use a cleaner, more robust front-end interface. Our codebase will also be much more organized and modular.

2 Meeting Plans

Our SCRUM master will be Dan Zheng. We plan on meeting at least twice per week. We will meet starting at 11:30 on MWF for 1-2 hours.

3 Risks and Challenges

Like the previous sprint, the biggest risk of this sprint is the amount of work that needs to be done. Refactoring the entire application involves a significant amount of time and effort, and may break or negatively impact previously completed features of our application.

However, we believe in the importance of high-quality, organized code and that refactoring is a meaningful, worthwhile endeavor.

4 Current Sprint Detail

Since we have completed the functional requirements for our project, the bulk of the work for this sprint involves refactoring or rewriting previous code. In previous sprints, we focused on creating a minimal viable product and writing code that worked. However, much of our code could be improved: we plan on refactoring our back-end code, particularly asynchronous functions, to be cleaner, as well as rewriting the front-end code to be more systematic and modular instead of being ad-hoc and performing direct DOM manipulations. Thus, some of our tasks are not related to particular user stories, but are related to user experience.

Refactor Frontend

#	Task description	Estimated Time	Owner
1	Set up webpack/Vue.js for client code separation	5 hrs	Dan
2	Set up Vue.js environment (views, router, store)	5 hrs	Dan
3	Rewrite basic front-end views using Vue	5 hrs	Dan, Carson
4	Rewrite chat/meetings views using Vue components	15 hrs	Dan, Justin

Acceptance Criteria

- We will consider this task completed when our project front-end has been rewritten using Vue.js. The front-end code will be modular, with clear separation of components, routes, and stores. Various front-end views, particular chats and meetings, will be rewritten.
- From a user's experience, pages/actions will load more quickly. The chat interface will feature real-time data, as messages are sent and received.

Refactor Backend

#	Task description	Estimated Time	Owner
1	Rewrite database queries to improve efficiency	15 hrs	Eric, Justin
2	Rewrite database schemas to improve efficiency	5 hrs	Eric
3	Cleanup callbacks and prefer using Promises or Async/Await	5 hrs	Eric

Acceptance Criteria

- Generally, changes in back-end code will not directly affect the user experience. However, our codebase should become cleaner, more modular, and easier to maintain.
- We will consider this task completed when our database schemas/queries written using the Sequelize ORM have been refactored into cleaner MySQL and when functions containing nested functions/Promises have been rewritten using cleaner Promises or using async/await.

User story #1

As a user, I would like to view links to public profiles of other users whenever possible.

#	Task description	Estimated Time	Owner
1	Add public profile links	2 hrs	Carson
2	Rewrite database schemas to improve efficiency	5 hrs	Eric

Acceptance Criteria

- Given that the links to public profiles of other users has been completed, when users visit any page with view elements containing information about other users, they should be able to click on the elements to view public profiles.

User story #2

As a user, I want to invite students in my courses to chat.

#	Task description	Estimated Time	Owner
1	Create back-end controller for creating a chat and inviting students	7 hrs	Carson, Justin
2	Create front-end interface for inviting students to chat	7 hrs	Carson, Justin

Acceptance Criteria

- Given that the front-end interface for inviting course students to chat is completed, when a user views a course page, they should be able to create a group chat by selecting students in the course and clicking a button.
- Given that the back-end controller for inviting course students to chat is completed, when a user requests to invite students to a group chat, a new chat should be created in the database with the requester as the chat creator and all other students as chat members.

5 Remaining Backlog

Note: we have completed all “required” tasks in the backlog. We may attempt the optional tasks if there is time after our application refactor.

As a user, I would like to:

1. ~~create a Meetchu account using a valid email and password.~~
2. ~~recover my password using my email.~~
3. ~~add profile information (name, photo, major, phone number, short bio) to my account.~~
4. ~~update my account information at any time.~~
5. ~~search for courses by title and number.~~
6. ~~add a course to my account.~~
7. ~~view people in my courses.~~
8. ~~view a list of my courses.~~
9. ~~create and join study groups with people in my classes.~~
10. ~~send personal messages to any other user.~~
11. ~~view message history with other users.~~
12. see users who are online (if time allows)
13. ~~link my account with and authenticate using Google and Facebook (if time allows).~~
14. view my class schedule (if time allows).

15. search for people in my classes by name or email (if time allows).
16. view the schedules of members in my study groups (if time allows).

As a meeting creator, I would like to:

1. ~~create a one time meeting for a study group.~~
2. ~~create a personal one time meeting with selected members.~~
3. ~~suggest possible dates and times for a meeting.~~
4. ~~add a title, purpose, and location to a meeting.~~
5. ~~add or remove people from a meeting and notify them.~~
6. ~~cancel a meeting and notify participants.~~
7. ~~finalize the date and time for a meeting and notify participants.~~
8. ~~view participants who have RSVP'd to a meeting.~~
9. add an address to a meeting and view on Google Maps (if time allows).
10. ~~view meeting attendance (if time allows).~~
11. create a recurring meeting for a study group (if time allows).

As a meeting participant, I would like to:

1. ~~view my meetings.~~
2. ~~indicate my preferred times for a meeting.~~
3. ~~RSVP to a meeting after its date and time have been finalized.~~
4. ~~set time preferences for group meetings.~~
5. ~~receive email notifications about upcoming meetings.~~

As a group creator, I would like to:

1. ~~be able to delete the group and notify members.~~

As a group member, I would like to:

1. ~~be able to leave the group.~~
2. ~~invite people to the group.~~
3. ~~send messages to the group.~~
4. ~~view group messages.~~