

Sprint 1 Planning Document

Eric Aguilera, Justin Bonner, Carson Harmon, Dan Zheng

1 Sprint Overview

For our initial sprint, we will focus primarily on the back-end of our web application. We will set up the back-end for our application by creating a Node.js server, deploying it to Heroku, and linking it with a MySQL database. We will also create database models for users and groups and set up user authentication. Simple signup/login and profile views will be created so that our site is functional.

By the end of our sprint, users will be able to create an account using email and password. They will be able to update their profile information and password. Users can search for Purdue courses and add them to their account. After adding a course, they can view other users in the course.

If time allows, we will work ahead on the messaging feature that will be delivered in the second sprint. We predict that the messaging feature will be complicated and we want to get started on it early if possible so that we can fulfill our requirements for the second sprint. In particular, we will work on the database model for messages, write controllers for sending/receiving messages, and create front-end views for viewing messages.

Meeting Plans

Our scrum master will be Dan Zheng. We plan on meeting up to three times per week. At least once a week, we will make an effort to have an in-person meeting. In-person meetings are important for allowing us to communicate issues and help each other more effectively. In order to accommodate everyone's busy schedules, we will also utilize Google Hangouts for online meetings. We plan to hold our meetings on MWF, 11:30 am - 12:20 pm.

Risks and Challenges

Since not all members of our team are familiar with the technologies we are using (Javascript, Node.js, Heroku, MySQL, Socket.io, Passport.js), it is important that we help them get up to speed. We predict that the learning curve of our stack will be our biggest obstacle for our first sprint. To address this challenge, our team will need to meet frequently and help each other learn the technologies.

2 Current Sprint Detail

User story #1

As a user, I would like to create a Meetchu account using a valid email and password.

#	Task description	Estimated Time	Owner
1	Set up back end server and deploy to Heroku	1 hr	Dan
2	Set up database	1 hr	Dan
3	Create database model for users	2 hrs	Eric
4	Add back-end controller for user account creation with input validation	2 hrs	Justin
5	Add front-end view for user account creation with input validation	2 hrs	Carson

Acceptance Criteria

- Given that the front-end view for user account creation is completed, when a user navigates to the page for creating an account, they will see a form containing fields for email and password. When they submit the form, the fields will be validated and the data will be sent to the back-end controller.
- Given that the back-end controller for user account creation is completed, when a request is sent to create an account with email and password fields, then the fields will be validated (email must be unique and password must be at least 5 characters) and the user account will be added to the database with a hashed password.

User Story #2

As a user, I would like to login to my account.

#	Task description	Estimated Time	Owner
1	Implement user authentication using username and password	3 hrs	Justin
2	Add back-end authentication middleware	3 hrs	Dan
3	Add front-end view for login	2 hrs	Eric

Acceptance Criteria

- Given that the front-end view for login is completed, when a user navigates to the page for logging in, they will see a form containing fields for email and password. When they submit the form, the fields will be validated and the data will be sent to the back-end controller.

- Given that the back-end controller for user authentication is completed, when a login request is sent, the email and password fields are validated and matched with the database. If a match is found, the user is successfully authenticated for their session.

User Story #3

As a user, I would like to recover my password using my email.

#	Task description	Estimated Time	Owner
1	Create database model for recovery tokens	1 hr	Eric
2	Add back-end controller for recovery process	3 hrs	Dan
3	Add front-end interfaces for recovery process	3 hrs	Justin

Acceptance Criteria

- Given that the front-end interface for the recovery process is completed, when a user requests to recover their account (with email field), then they will be presented with a password reset form asking for their user email.
- Given that the back-end controller is completed, when a user submits the password reset form, the user will be sent an email containing a url with a unique token to trigger the account creation process.
- Given that the back-end controller is completed, when the user visits the url, they will be prompted to enter a new password. When they submit, the database will be updated with the hash of their new password immediately.

User Story #4

As a user, I would like to add profile information (name, photo, major, phone number, short bio) to my account and update it at any time.

#	Task description	Estimated Time	Owner
1	Add back-end controller for adding/updating profile information	2 hrs	Carson
2	Add front-end view for adding/updating profile information	2 hrs	Carson

Acceptance Criteria

- Given that the front-end view for adding/updating profile information is completed, when a user navigates to the page for changing profile information, they will see a form with various fields (name, email, etc) that they can fill in. If the user has previously submitted profile information, then the old information will be displayed in the form automatically.

- Given that the back-end controller for adding/updating profile information is completed, when a user requests to add/update profile information, their new information will be saved to their account in the database.

User Story #5

As a user, I would like to search for courses by title and number.

#	Task description	Estimated Time	Owner
1	Create database model for courses	1 hr	Eric
2	Add back-end logic that uses Purdue course API (purdue.io) to populate database with courses	3 hrs	Dan
3	Add back-end controller for searching courses	3 hrs	Justin
4	Add front-end view for searching courses	3 hrs	Carson

Acceptance Criteria

- Given that the back-end logic for communicating with the Purdue course API is completed, when the database is started for the first time, it will be populated with course objects.
- Given that the back-end controller and front-end view for searching courses have been completed, when a user searches for a course based on criteria such as department and course number, they should see a list of courses that match the criteria.

User Story #6

As a user, I would like to add a course to my account.

#	Task description	Estimated Time	Owner
1	Add back-end controller for adding a course to account	1 hr	Justin
2	Add front-end interface for adding a course to account	1 hr	Justin

Acceptance Criteria

- Given that the back-end controller and front-end interface for adding a course have been completed, when a user adds a course to their account, it will be saved to their account in the database.

User Story #7

As a user, I would like to view a list of my courses.

#	Task description	Estimated Time	Owner
1	Add back-end controller for returning user courses	1 hr	Eric
2	Add front-end view for viewing user courses	1 hr	Justin

Acceptance Criteria

- Given that the back-end controller and front-end interface for viewing user courses have been completed, when a user visits the courses page, they can view a list of their courses.

User Story #8

As a user, I would like to view other users in my courses.

#	Task description	Estimated Time	Owner
1	Add back-end controller for viewing course students	1 hr	Carson
2	Add front-end interface for viewing course students	1 hr	Carson

Acceptance Criteria

- Given that the back-end controller and front-end interface for viewing course students have been completed, when a user clicks on a course page, they can see other users in their courses.

User Story #9

As a user, I would like to create and join study groups with people in my classes.

#	Task description	Estimated Time	Owner
1	Create database model for groups	1 hr	Eric
2	Add back-end controller for creating a group	3 hrs	Eric
3	Add front-end interface for creating a group	3 hrs	Eric

Acceptance Criteria

- Given that the back-end controller and front-end interface for creating a group have been completed, when a user requests to create a group, they should be able to specify a group name and the group should be saved to the database.

User Story #10

As a group creator, I would like to be able to delete the group and notify members.

#	Task description	Estimated Time	Owner
1	Add back-end controller for deleting a group	1 hr	Carson
2	Add front-end interface for deleting a group	1 hr	Carson

Acceptance Criteria

- Given that the back-end controller and front-end interface for deleting a group have been completed, when a group creator requests to delete a group, then the group should be deleted from the database and members should be notified.

User Story #11

As a group member, I would like to invite people to the group.

#	Task description	Estimated Time	Owner
1	Add back-end controller for inviting people to a group	2 hrs	Dan
2	Add front-end interface for inviting people to a group	2 hrs	Dan

Acceptance Criteria

- Given that the back-end controller and front-end interface for inviting people to a group have been completed, when a group member requests to invite a user to a group, then the user should be added to the group in the database.

User Story #12

As a group member, I would like to be able to leave the group.

#	Task description	Estimated Time	Owner
1	Add back-end controller for leaving a group	1 hr	Dan
2	Add front-end interface for leaving a group	1 hr	Dan

Acceptance Criteria

- Given that the back-end controller and front-end interface for leaving a group has been completed, when a group member requests to leave a group, then they should be removed from the group in the database.

3 Remaining Backlog

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.