Group 14 Sprint 1 Planning

OURPURDUEPLAN

Team Members: Neil Van Eikema Hommes, Cooper Chiappetta, Michael Dimitrov, Jack Roscoe

Sprint Overview

During this sprint we hope to implement account creation, user profiles, logging in/out, and the basics of plan creation. At the minimum, a user should be able to create multiple plans with branching and drag and drop. We will be setting up the Phoenix API to send and verify web-tokens to handle authenticating requests, as well as the basic database structures.

Scrum Master: Jack Roscoe

Meeting Plan:
- Tuesday @ 8:30 PM on a voice call
- Thursday @ 6:30 PM with our Project Coordinator

Risks and Challenges: We could encounter some delays at the beginning as we learn and get familiar with Elixir and Phoenix, as we have not used it before. We will also need to learn and get proficient with React, as not everyone on our team has used it. We will also have to learn to coordinate between different members, especially the people working on the frontend and backend, so that we can fully integrate our frontend with our backend in order to have a finished incremental version at the end of the sprint.

Current sprint detail

User Story #1:

As a user, I would like to register my account and set my user type (student/advisor). In the case of an advisor account, I want my account creation request to be sent to the admin for approval.

#	Description	Time	Team	Owner
1	Text input for Email, Username, Password	2	Front	Michael
2	Redirect between account creation and login	1	Front	Michael
3	Interface between front and backend for account creation	4	Front	Michael
4	Write API controller logic for account creation	3	Back	Neil
5	Store the user in the database	2	Back	Neil
6	Create Unit tests for unique creating accounts	4	Back	Neil

Acceptance Criteria:

- Given that a username doesn't exist, when a user creates an account, they should be redirected to the forum page and have their account username stored client side for future requests.
- Given that a username already exists, when a user creates an account, an error should pop up indicating as such.
- Given that a user is logged in, when a user tries to access the register page, they get redirected to the forum page.
- Given that the confirm password field matches the password field, the account should be created, otherwise it should show an error.
- Given the password reaches the minimum required length, the account should be created, otherwise it should show an error.

User Story #2:

As a user, I would like to login and logout with a username and password to access my account.

#	Description	Time	Team	Owner
1	Build login page	3	Front	Michael

2	Build logout button	1	Front	Michael
3	Server needs to send email if forgot password	2	Back	Jack
4	Display appropriate error messages if incorrect username/password	2	Front	Michael
5	Redirect to the sign in page if not logged in	1	Front	Michael
6	Store the authentication token on the client and send it with all future requests	2	Front	Michael
7	Api route to login and get token	2	Back	Jack
8	Verify token in privileged api routes	2	Back	Jack
8	Implement Unit tests	3	Back	Jack

Acceptance Criteria:

- Given that a user is logged out, when a user tries to access the logout page they should be redirected to the login page.
- Given that a user is logged in, when a user tries to access the login page, they should be redirected to the forum page.
- Given that a user is logged out, when a user tries to login with invalid username or password, an error message indicating invalid credentials should be displayed.
- Given that a user successfully logs in, their authentication token should be appropriately set and they should be redirected to the forum page.

User Story #3:

As a user, I would like to be able to have personal information such as my name and profile picture.

#	Description	Time	Team	Owner
1	UI to view settings	2	Front	Cooper
2	UI to update settings	2	Front	Cooper
3	UI to view a user's profile	3	Front	Cooper
4	API Request to get settings and profile data	2	Back	Neil
5	API controller logic for setting and changing profile	4	Back	Neil

6	Photo upload for profile picture	2	Back	Neil

Acceptance Criteria

- Given that a user is logged in, when a user wants to set their profile picture, they should upload a file and the image will be displayed appropriately.
- Given that a user is logged in, when a user wants to update their email, they should provide the new email and the database should be updated.
- Given that a user is logged out, when a user accesses this page they should be redirected to the login page.
- Given the user goes to any other page, and is logged in, the profile picture in the top right should be updated accordingly.

User Story #4:

As a user, I would like to be able to view a list of all my plans.

#	Description	Time	Team	Owner
1	UI to display all plans a user has created in a grid	3	Front	Cooper
2	Be able to search and filter through plans the user created	3	Front	Cooper
3	Clicking on a plan opens the plan builder / editor	2	Front	Cooper
4	API route for getting detailed plan information	2	Back	Neil
5*	Plan preview to see the plan at a glance before going to editor	3	Front	Cooper

Acceptance Criteria

- Given that a user is logged in, a user should be able to view a list of all plans they created on the plans page.
- Given that a user is not logged in, when they try to view this page they get redirected to the login page.
- Given that a user is logged in, when a user deletes or updates a plan's name / description, that name and description should be updated accordingly on the plan viewing page.
- *Given that a user has selected a plan, they should be able to see a preview of the plan without going into the editor.*

User Story #5:

As a student, I would like to be able to create and delete multiple plans of study and select the different majors and minors that it corresponds to.

#	Description	Time	Team	Owner
1	Create plan creation screen form with title, majors, minors	2	Front	Cooper
2	Create dropdown list components to search and append majors and minors	2	Front	Cooper
3	Logic to send payload data and navigate to the plan builder screen	4	Front	Cooper
4	Api route for creating a plan	2	Back	Jack
5	Store the plan of study in the database	4	Back	Jack
6	Create unit tests for creating/deleting plans	4	Back	Jack

Acceptance Criteria

- Given that the user has navigated to the "my plans" page and hit the create plan button, they should be able to enter plan details, hit create, and be brought to the plan creator.
- Given that the user has navigated to the "my plans" page, they should be able to delete plans of study.
- Given that the user is in the create plan screen, they should be able to search for majors and minors and add them to a list.

User Story #6:

As a student, I would like to be able to use the plan creator to access existing courses and add/remove to/from different semesters.

#	Description	Time	Team	Owner
1	Create plan creator screen with a way to add and remove classes from each different semester	3	Front	Cooper
2	Drag and drop courses into and between semesters	3	Front	Michael
3	Search existing class list to add to plans	3	Front	Cooper

4	Create buttons to add and delete semesters	2	Front	Cooper
5	API endpoint to get plan data	1	Back	Neil
6	API to add or move a class to a different semester	4	Back	Neil

Acceptance Criteria

- Given that a user is logged in, a user can navigate to the plan creator and create college plans from a list of existing courses.
- Given that a user is not logged in, a user should be redirected to the login page.
- Given that a user is logged in, they should be able to view and update plans they have already made in the past.

User Story #7:

As a user, I would like all the classes in my plan to have their prerequisite classes checked.

#	Description	Time	Team	Owner
1	Display a warning message if a prerequisite is missing	2	Front	Michael
2	Query the backend whenever a plan is saved for missing prerequisites	3	Front	Michael
3	Create logic to check if all prerequisites in a plan are satisfied, and report ones which are not	5	Back	Jack
4	Save and retrieve the prerequisite classes in the database	3	Back	Jack
5	Add unit tests to ensure prerequisites are met	3	Back	Jack

Acceptance Criteria:

- Given a plan has all of its prerequisites satisfied, do not report any warnings to the user
- Given a class is missing a prerequisite, show a warning box to the user displaying which prerequisite is not satisfied
- Given that a plan does not satisfy prerequisites, when a user tries to create a plan, it should still let them, but warn that the plan does not match prereqs.

User Story #8:

As a student, I would like to be able to see degree requirements as I plan for them.

#	Description	Time	Team	Owner
1	Display warning if user doesn't meet degree requirements	2	Front	Michael
2	Query the backend whenever a plan is saved for degree requirements	2	Front	Michael
3	Check if the plan is following degree requirements for the specified majors and minors	3	Back	Jack
4	Add unit tests to ensure degree requirements are met	3	Back	Jack

Acceptance Criteria:
- Given the plan meets all degree requirements, do not show warnings to the user
- Given the plan is missing classes, show a warning indicating which class needs to be integrated into the plan
- Given that a plan does not meet all degree requirements, the server needs to recalculate which classes are missing and send them to the client, each time a plan is visited.
- Given that a plan does not satisfy degree requirements, when a user clicks to create, it should still let them, but warn that the plan does not meet requirements for their major or minor.

User Story #9:

As a student, I would like to view information about a course.

#	Description	Time	Team	Owner
1	Page which displays relevant information about a course	3	Front	Michael
2	Api route to fetch information about a course	3	Back	Neil
3	Query database to retrieve information about a class	2	Back	Neil

Acceptance Criteria:
- Given that a user navigates to a course webpage, the user is able to view the course title.
- Given that a user navigates to a course webpage, the user is able to view the course description.
- Given that a user navigates to a course webpage, the user is able to view the course credit hours.

Total Hours Per Person

Cooper	34
Michael	31
Neil	30
Jack	36

Remaining Backlog

1. ~~As a user, I would like to register my account and set my user type (student/advisor). In the case of an advisor account, I want my account creation request to be sent to the admin for approval.~~
2. ~~As a user, I would like to login and logout with a username and password to access my account.~~
3. ~~As a user, I would like to be able to have personal information such as my name and profile picture.~~
4. ~~As a student, I would like to be able to create multiple plans of study.~~
5. ~~As a student, I would like to be able to create and delete plans of study with confirmation before doing a dangerous action~~
6. ~~As a student, I would like to be able to create plans by semester to accommodate graduating in less or more than 4 years.~~
7. As a student, I would like to be able to share my plans of study to other people by clicking a button to copy a link to send to other users.
8. As a student, I would like to be able to branch from a semester so that I don't have to create a bunch of plans with identical n semesters.
9. ~~As a student, I would like to have the option to pick from existing classes to add to my plans.~~
10. ~~As a student, I would like to be able to select majors and minors that a plan satisfies.~~
11. ~~As a student, I would like to be able to drag and drop courses between semesters.~~
12. As a student, I would like to be able to post a plan of study to the forum for other people to see.
13. As a student, I would like to be able to view the forum to see other people's posted plans of study.
14. As a student, I would like to be able to filter plans of study on the forum by major and minor.
15. As a student, I would like to be able to rate a course based on perceived difficulty and see other ratings of it.

16. As a student, I would like to be able to sort plans of study on the forum by a difficulty score calculated by the system, and by upvotes.
17. As a student, I would like to be able to comment and see the comments on forum posts.
18. As a student, I would like to be able to "mention" a course in my comment, which highlights and links the course.
19. As a student, I would like to be able to copy a plan from the forum into my plans of study.
20. As a student, I would like to be able to see statistics about plans, including popular classes, majors/minors, and more.
21. ~~As a student, I would like to know when a course cannot be taken because there is a missing prerequisite.~~
22. As a student, I would like to be able to see degree requirements as I plan for them.
23. As an admin, I would like to be able to review new classes.
24. As an admin, I would like to be able to add and remove new classes from the database.
25. As an admin, I would like to be able to delete comments on plans.
26. As an admin, I would like to delete obviously malformed plans.
27. As an admin, I would like to give popular plans a verified badge.
28. As an admin, I would like to give graduated students a graduated badge.
29. As an advisor, I would like to know when my account is approved via an email.
30. As an advisor, I would like to save a student's plan to recommend to others.
31. As an advisor, I would like to have students be able to send their degree plans for me.
32. As an advisor, I would like to give feedback on a student's degree plan.
33. As an advisor, I would like to recommend alternative classes for my students.
34. As an advisor, I would like to be able to categorize my students by major/minor
35. As an advisor, I would like to be able to send announcements to all students linked to me.
36. As an advisor, I would like to be able to easily share common plans with my students.
37. ~~As a student, I would like to view information about a course.~~

Non-Functional Requirements

As developers, we would like the following items:

1. **Architecture and Performance**

   a. I would like OurPurduePlan to operate without unexpected behavior.

   b. I would like OurPurduePlan to be responsive, with response times less than 2 seconds in the worst case.

   c. I would like errors to be handled elegantly and displayed to the user so they can correct them.

2. **Security**

    a. I would like data to be secure.

    b. I would like permission checking to ensure that users can only access data that they are allowed to see.

3. Usability

    a. I would like to create a user interface that is intuitive to use and caters to students of any familiarity with technology.

    b. I would like to create a user interface that looks visually appealing and displays the data in an effective manner.

    c. I would like to create a plan creator that makes adding and deleting courses between semesters easy.

    d. I would like to display branches in a way that makes it easy to change between them.

    e. I would like to have navigation that allows users to change between the creator, forum, and profile in one click.

    f. I would like to provide a set of courses that users can use out of the box in their plans.

4. Hosting/Development

    a. I would like to deploy the frontend and backend servers on DigitalOcean instances.

5. Server

    a. I would like to store information with the most efficient schema on the server.