Group 14 Sprint 3 Planning

OURPURDUEPLAN

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

Sprint Overview

During this sprint we hope to implement the final parts of our app, which focus on admin operations, advisor functionality, and branching. Additionally, if time allows, we would like to clean up other parts of the app to make it run better, such as UI cleaning and unification (some parts of our UI currently do not match up very well).

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: There is still risk when working with and around code written in sprints 1 and 2. While permissions and advisor work might not be bad, branching works around the most fundamental part of the app: plans. Plans and the plan editor are very complex, and adding branching functionality will only increase that complexity. While it's possible it might go smoothly, there is a lot of code refactoring that will need to occur to allow branching to work properly. Additionally, adding admin permissions might require adding functionality to almost

every page which could add complexity if not done carefully. As always, there is always room for error when integrating the frontend with the API in its entirety.

Current sprint detail

User Story #1:

As a user, I would like to be able to be able to branch off of semesters to avoid creating plans with duplicate starting semesters.

#	Description	Time	Team	Owner
1	Update plan editor to be able to create branches	3hr	Frontend	Cooper
2	Update plan editor to be able to cycle through branches with the arrow buttons	3hr	Frontend	Cooper
3	Update plan editor with a delete button that deletes an entire branch	2hr	Frontend	Cooper
4	Branching API integration	3hr	Frontend	Cooper
5	Backend API for creating branches	3hr	Backend	Jack
6	Store branches in database	4hr	Backend	Jack
7	Unit tests for branching	2hr	Backend	Neil

Acceptance Criteria:

- Given that a user is on the plan editor, they are able to create a new branch off of any semester.
- Given that a user is on the plan editor, they are able to use the arrow buttons to cycle between 1 or more branches.
- Given that a user is on the plan editor, they are able to delete a branch by clicking a delete button that is easily accessible.

User Story #2:

As a user, I would like to be able to post a plan to the forum that has branches by choosing which branches to display.

#	Description	Time	Team	Owner
1	Update post creation form with an option to select branches if there are any	3hr	Frontend	Cooper
2	API integration update to send the branch IDs when creating a new post	2hr	Frontend	Cooper
3	API logic to create a post with a hard copy of a plan	2hr	Backend	Jack
4	API logic to choose specific branches of that plan to create the hard copy	1hr	Backend	Jack
5	API logic to update copying a plan with these rules	1hr	Backend	Jack
6	Unit tests for creating hard copies of plans from choosing branches	2hr	Backend	Neil

Acceptance Criteria:

- Given that a user is on the post creation screen, they should be able to choose a plan to post, and if that plan has branches, choose a sequence of branches.
- Given that the user creates a post after choosing a plan with branches, the resulting post contains a plan without any branches.
- Given that the user copies a plan that had branches before on the forum, they should receive an exact copy of the plan from that post (without branches).

User Story #3:

As an admin, I would like to be able to moderate forum posts and comments and delete unwanted ones.

#	Description	Time	Team	Owner
1	Update post page to have a delete button for admins	1hr	Front	Cooper
2	Update comments to have delete buttons for admins	1hr	Front	Cooper
3	API integration to send delete requests for posts and comments	2hr	Front	Cooper
4	API logic to delete a post or comment	2hr	Backend	Jack
5	API logic to only allow deletion if the user is an admin	2hr	Backend	Jack
6	API logic to send permissions data to the frontend when getting your own profile data to know if the current user is an admin	2hr	Backend	Jack
7	Unit tests for deleting posts and comments with/without admin permissions	2hr	Backend	Neil

Acceptance Criteria:

- Given that an admin is viewing a post, they should be able to click a delete button to remove that post from the forum.
- Given that an admin is viewing a post with comments, they should be able to click a delete button to remove comments from the post.
- Given that a non-admin user is viewing a post, they should not be able to see a delete button and in turn delete items.

User Story #4:

As an admin, I would like to be able to add, edit, and delete courses from the database.

#	Description	Time	Team	Owner
1	Update course page with a delete button that only appears for admins	1hr	Front	Cooper
2	Update course page with an edit button that only appears for admins	1hr	Front	Cooper
3	Update the courses page with an add course button that only appears for admins	1hr	Front	Cooper
4	Craft a create course form	2hr	Front	Cooper
5	Update the course page to be editable after an admin clicks the edit button	3hr	Front	Cooper
6	API integration to support adding editing and deleting courses	3hr	Front	Cooper
7	API integration to support resetting a course's difficulty	1hr	Backend	Neil
8	API logic to support create, update, delete on courses	2hr	Backend	Neil
9	Unit tests for creating, updating, and deleting courses with and without admin permissions	2hr	Backend	Neil

Acceptance Criteria:

- Given that an admin is viewing a course, they should be able to click a delete button to remove that course from the database.
- Given that an admin is viewing a course, they should be able to click an edit button to alter the information about that course and save it.
- Given that an admin is viewing the courses page, they should be able to click an add button to be sent to the create course form.
- Given that an admin creates a course, the course should now be searchable for all users.

User Story #5

As a user I would like to be able to see prerequisites for courses on the course page, and click a link to visit that prerequisites course page.

#	Description	Time	Team	Owner
1	Add links to the course page of each prerequisite	1hr	Frontend	Michael
2	Remove the "prerequisites" section if there are no prerequisites	1hr	Frontend	Michael
3	Request the prerequisites for each course from the server	2hrs	Frontend	Michael
4	Backend API logic to return the prerequisites of a course	3hrs	Backend	Jack
5	Unit tests for course prerequisites	2hrs	Backend	Neil

Acceptance Criteria

- Given that a user is on the course page, they should see the list of prerequisites for a course.
- Given that a user is on the course page, and the course has no prerequisites, they should see an appropriate message
- Given that a user taps one of the prerequisites, they should be navigated to that prerequisites course page.

User Story #6

As an advisor, I would like to convert my account to an advisor account, and know when my account is approved via an email.

#	Description	Time	Team	Owner
1	Add an option in settings to convert to an advisor account, if they are not already an admin	1hr	Frontend	Michael
2	If user is an admin put a badge next to their profile picture	1hr	Frontend	Michael
3	Have a confirmation message when the user tries to convert advisor account	1hr	Frontend	Michael
4	Send a request to the server to change account to advisor account	1hr	Frontend	Michael
5	Backend API logic to convert an account to an advisor account	2hr	Backend	Neil
6	Send an email confirmation when an advisor account is approved/denied	2hr	Backend	Neil
7	Unit tests for advisor accounts.	2hr	Backend	Neil
8	Admin dashboard to view pending users and accept them	1hr	Frontend	Cooper
9	Fetching/accepting pending users API integration	1hr	Frontend	Cooper

Acceptance Criteria:

- Given that an advisor profile picture is being displayed, there should be a badge next to the profile picture.
- Given that a user who is not an advisor is viewing their settings, they should see the option to convert to an advisor account.
- Given that an advisor is viewing their settings, they should not see the option to convert to an advisor account.
- Given that an advisor account was approved, they should receive an email to the email associated with their account.
- Given that an advisor account was denied, they should get an email saying they were denied.
- Given that an admin has navigated to the advisor acceptance dashboard, they should be able to accept or deny incoming requests to become an advisor.

User Story #7

As an advisor, I would like to save a student's plan to "My Student's Plans".

#	Description	Time	Team	Owner
1	Create a new section of the plans page for plans an advisor has saved	2hrs	Frontend	Michael
2	Add error message on saved plans page for when the advisor has saved no plans	1hr	Frontend	Michael
3	Add a button on posts that advisors only can see, so they can save plans	2hr	Frontend	Michael
4	Add functionality to save plans when the user taps the save plan button	2hrs	Frontend	Michael
5	Navigate to the advisors "Saved plans" page	1hr	Frontend	Michael
6	API route to fetch all an advisors student plans	2hr	Backend	Jack
7	API route to save a student's plan to my students plans	2hr	Backend	Jack
8	Store which plans are in the advisors student plans in the database	2hr	Backend	Jack
9	Unit tests for advisors storing plans	2hr	Backend	Neil

Acceptance Criteria:
- Given that an advisor taps the save plan button on a post, that post's plan should be copied to the advisor's saved plans, and the advisor should be navigated there.
- Given that an advisor is on the saved plans section of the plans page, they should see a list of all the plans they have saved
- Given that an advisor is on the saved plans section of the plans page, and has not saved any plans, they should see an appropriate error message.
- Given that a user is viewing the forum or post page they should only see a save plan button if they are an advisor

User Story #8

As an advisor I would like to be able to search for users to request to add them as my student.

#	Description	Time	Team	Owner
1	Add a user search page to search for users by username	2hrs	Frontend	Michael
2	Display an appropriate error message if no users are found	1hr	Frontend	Michael
3	Show a list of usernames that match the search text	2hrs	Frontend	Michael
4	Frontend logic to search for users	2hrs	Frontend	Michael
5	Button to request user to be advisor's student	2hrs	Frontend	Michael
6	API route to fetch and search for users	2hrs	Backend	Neil
7	API route to add user to advisors student list	2hrs	Backend	Neil
8	Unit tests for adding students to advisor's list	2hrs	Backend	Neil

Acceptance Criteria

- Given the advisor is on the user search page, they should be able to search for users by username.
- Given the advisor is on the user search page, they should be able to click on one of the results to request the user be added to their student list.
- Given the advisor's search does not return any results, the advisor should see an appropriate error message.

User Story #9

As an advisor I would like to be able to share plans with my students.

#	Description	Time	Team	Owner
1	Add a button on the advisor's page to share plans with a selected student from a dropdown of advisor's students	1hr	Frontend	Michael
2	Add a dropdown selector for advisor's students	1hr	Frontend	Michael
3	Frontend logic to add a plan to users shared plans section	2hrs	Frontend	Michael
4	Backend logic to add a plan to users shared plans	3hrs	Backend	Jack
5	Unit tests for adding a plan to users shared plans	2hrs	Backend	Jack

Acceptance Criteria

- Given that an advisor selects a student and clicks to share a plan with them, the plan should be added to the user's shared plans.
- Given that an advisor clicks the student dropdown, they should be presented with a list of their students.
- Given that an advisor has no students attached to their account, the dropdown should present an appropriate error message.

User Story #10

As a student I would like to be able to view plans that my advisor shared with me

#	Description	Time	Team	Owner
1	Frontend page to view all plans that an advisor has shared with a student.	2hrs	Frontend	Michael
2	Frontend logic to display the recommended plan	1hr	Frontend	Michael
3	Frontend button to accept / reject the recommended plan	1hr	Frontend	Michael
4	API route to get a list of plans that an advisor shared with a student	2hr	Backend	Neil
5	API route for a student to accept a plan that an advisor shared	2hr	Backend	Neil
6	Unit testing for students accepting advisor recommended plans	2hr	Backend	Neil

Acceptance Criteria

- Given that an advisor shares a plan with a student, the student should receive that plan and have a choice whether or not to accept that plan into their plans.
- Given that a student accepts a plan recommended to them by an advisor, the plan should be added to their list of plans.
- Given that a student declines a plan recommended to them by an advisor, the plan does not show up in the student's list of plans.

Total Hours Per Person

Cooper	33
Michael	33
Neil	33
Jack	31

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 post creation page that makes it easy to create a post with your selected plan
   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.