Pete's Plan

Sprint 1 Planning Document

Team 36: Anushka Sharma, Chloe Wang, Cole Doner, Patricia Madalena Magalhaes Casaca, and Shivani Venkatraman

Sprint Overview

During our first sprint, we are looking to set the groundwork for our project by setting up all of the required software, figuring out how to integrate it all together, and setting up some basic functionality for our service. We hope to have a visibly functional product within the first couple weeks where we can view a couple basic pages, and interface with user accounts. Additionally, we will be setting up the back end to store user data and getting it ready for all of the course data, potentially manually pre-loading it with some data for testing purposes.

Scrum Master: Cole Doner

Meeting Plan:

Monday/Thursday 6:00 PM (but Thursday Feb 23rd meeting will be replaced by Friday Feb 24th meeting)

Risks and Challenges:

This is the first time that much of our group has attempted a project like this, and so we anticipate that a lot of the tasks will take a bit longer than expected to complete as we adjust to the technologies we chose to use, as well as working together as a team to implement our solutions. We also expect we will have some tasks with estimated times that are far from the actual time taken due to the aforementioned challenge, though we will get better at planning this as we continue through the first sprint, and on to the later ones

Current Sprint Detail

User Story #1

1.a: As a user, I would like to sign up for an account

#	Description	Estimated Time	Owner
1	Setup frontend with user creation web pages	6 hrs (2-2-2 split)	Chloe, Patricia, Shivani
2	Setup backend to handle user creation	5 hrs	Cole, Anushka
3	Setup mail server to send confirmation emails	2 hrs	Cole
4	Debug and test with valid and invalid inputs	1 hrs (Each)	Cole, Shivani

Acceptance Criteria:

- Given that the webpage is implemented currently, when the user attempts to access the webpage, then they will be greeted with a form to log in or create an account.
- Given that the account creation menu is set up, when the user selects “create account”, they will be directed to a page where they can enter their university email address.
- Given that the email is for a valid university domain and the email verification system is working, when the user submits the email address, an email will be sent to their account to verify their identity.
- Given that the email verification address is set up, when a user clicks the link in their email, they will be prompted to enter a password.
- Given that the backend user system is implemented properly, when the user submits their password, their account will be added to the user database.
- Given that the login screen has been implemented, when the user finishes creating their account, they will be redirected to the login screen.

User Story #2

1.b: As a user, I would like to log in/out of my account

#	Description	Estimated Time	Owner
1	Create user login web page	5 hrs	Patricia
2	Create page template w/ logout button	6 hrs	Shivani

3	Setup backend to handle user login	2 hrs	Chloe
4	Setup backend to handle user logout	2 hrs	Chloe
5	Debugging and tests with valid and invalid inputs	2 hrs (Each)	Chloe, Anushka

Acceptance Criteria:

- Given that the webpage is implemented currently, when the user attempts to access the webpage, then they will be greeted with a form to log in or create an account.
- Given that the backend user system is implemented properly, then the user will be able to log in to their account when providing the correct email, password combination.
- Given that the backend user system is implemented properly, then the user will not be able to log in to an account when providing an incorrect or invalid email, password combination.
- Given that the account creation menu and backend user system is implemented correctly, when the user is logged in then they will not see the “create account” option.
- Given that the webpage and backend user system is implemented correctly, when the user is logged in then they will see the “logout” option.

User Story #3

1.c: As a user, I would like the page to remember me so I don’t have sign in every time

#	Description	Estimated Time	Owner
1	Add remember me checkbox to login page	1 hrs	Patricia
2	Add backend logic for remember me token expiry	2 hrs	Chloe
3	Debugging and tests with valid and invalid inputs	2 hrs (Each)	Patricia, Chloe

Acceptance Criteria:

- Given that the webpage is implemented currently, when the user visits the login page, then they will see a “Remember Me” checkbox.
- Given that the backend user system is implemented properly, when the user checks the “remember me” option, then they will stay logged in across sessions until the expiry date.
- Given that the backend user system is implemented properly, when the user does not check the “remember me” option, then they will not stay logged in across browser sessions.
- Given that the backend user system is implemented properly, when the user’s token expires, then they will not be permitted to make authenticated API requests.
- Given that the backend user system is implemented correctly, when the user logs out

manually, then their active token will be prematurely expired.

User Story #4

1.d: As a user, I would like to reset my password via email

#	Description	Estimated Time	Owner
1	Create reset password page (I forgot my password)	3 hrs	Shivani
2	Add backend for generating and sending reset token	4 hrs	Cole
3	Add backend for handling new password submission	2 hrs	Cole
4	Debugging and testing with valid/invalid inputs	1 hrs (Each)	Cole, Shivani

Acceptance Criteria:

- Given the user system is implemented correctly, when a user selects to reset their password and enters their email, then they will be emailed a unique link to the reset page.
- Given the user system is implemented correctly, when a user attempts to reset the password for an account that does not exist, then no error will be returned.
- Given the password reset link works, when the user clicks on it, then they will be directed to a page where they can enter a new password.
- Given the user system works correctly, when the user enters their new password, then the account password will be updated and the user will be redirected to the login page.
- Given the user system is implemented correctly, when the account is successfully updated, then all tokens will be prematurely expired.

User Story #5

11.a: As an administrator, I would like the server to automatically update course descriptions

#	Description	Estimated Time	Owner
1	Setup Course storage in backend and database	5 hrs	Anushka
2	Create script to fetch, parse, and store course descriptions	20 hrs	Anushka, Chloe, Cole
3	Debugging and testing the scripts	4 hrs (Each)	Chloe, Cole

Acceptance Criteria:

- Given the script is implemented properly, when it is run, then it should pull course data from the course catalog.
- Given the course catalog data is successfully loaded, when each item is acquired, then the backend should parse the data and add it to the database or update existing entries.
- Given the server is running, when a set amount of time elapses, then the server should run the script during off-hours automatically before waiting for the delay again.

User Story #6

2.a: As a user, I would like to enter my current plan of study

#	Description	Estimated Time	Owner
1	Setup plans of study storage in database	4 hrs	Cole, Chloe
2	Add backend for storing user's plan of study	3 hrs	Cole, Chloe
3	Add backend for retrieving user's plans of study	1 hr	Cole, Chloe
4	Create webpage to historical record of classes	6 hrs	Shivani
5	Create webpage to record future plans of study	6 hrs	Patricia
6	Debugging and testing backend and frontend	2 hrs (Each)	Cole, Chloe, Shivani

Acceptance Criteria:

- Given the user is logged in, when they save their plan of study, then a request will be sent to the backend to store the updated information.
- Given the user has an account, when they log in, then they will get a list of their degree plans
- Given the user is logged in, when they select a degree plan, then the plan's data will be retrieved and displayed for viewing/editing.

User Story #7

2.b: As a user, I would like to modify my current plan of study

#	Description	Estimated Time	Owner
1	Add backend for modifying user's plans of study	2 hrs	Chloe
2	Add plan of study modification to webpage	2 hrs	Patricia

3	Add record of classes modification to webpage	2 hrs	Shivani
4	Debugging and testing backend and frontend	2 hrs (Each)	Chloe

Acceptance Criteria:

- Given the user is logged in, when they save changes, then the backend will be sent a request to modify the user's data in the database.
- Given the user has updated their degree plan, when they view their degree plan, then it should reflect the most recent changes to their courses.
- Given the user did not save changes, when they leave the page, then the changes to the course plan will not be saved, and will be reverted to the original plan.

User Story #8

2.c: As a user, I would like to enter my degrees

#	Description	Estimated Time	Owner
1	Setup degree storage in database	2 hrs	Anushka
2	Preload some degree data - enter info from Purdue degree manually - 2 majors and 2 minors for prototype	4 hrs	Anushka
3	Add degree selection in plan modification page	3 hrs	Patricia
3	Test both frontend and backend	1 hrs (Each)	Patricia

Acceptance Criteria:

- Given the user is logged in, when they save their plan of study, then a request will be sent to the backend to store the updated information.
- Given the degree has been added to the database, when the user starts entering a degree name, then they should be shown a list of degrees that match the entry.
- Given multiple degrees have been added to the database, when the user chooses to modify their degrees, then they should be allowed to select multiple options.

User Story #9

5.a: As a user, I would like to access course descriptions

#	Description	Estimated Time	Owner

1	Add backend for retrieving course descriptions	1 hr	Anushka
2	Create a page for viewing course details	3 hrs	Patricia
3	Modify other pages to allow redirection to details	1 hr	Patricia, Shivani
4	Test both frontend and backend	1 hr	Patricia

Acceptance Criteria:

- Given a course is loaded into the database, when an API request is made for the course, then the backend should return that course's data.
- Given the user can see a course listed, when they click on the course name, then they should be directed to the details for that course.
- Given the course is not in the database, when a request for the details are made, then the backend will respond with an error.

User Story #10

5.b: As a user, I would like to access course prerequisites

#	Description	Estimated Time	Owner
1	Setup course requirement structure in database	5 hr	Anushka
2	Add method for requesting course prerequisites	2 hr	Anushka
3	Add prerequisite display to course detail page	2 hrs	Patricia
4	Test frontend and backend	1 hr (Each)	Anushka, Patricia

Acceptance Criteria:

- Given the course has prerequisites, when the user views course details, then the prerequisites should be included as part of that.
- Given the prerequisites are in the database, when the user clicks on one of them,then they should be directed to that course's details page.
- Given the course has prerequisites, when the user views them, then they should be able to see which requirements they have satisfied and if they meet the minimum standard.

User Story #11

4.a: As a user, I would like to access major requirements

#	Description	Estimated Time	Owner
1	Store information about major requirements in Degree object	1 hr	Anushka
2	Accept request from user to view particular degree requirements and send back info	0.5 hr	Anushka
4	Display requirements on web page	3 hrs	Shivani

Acceptance Criteria:

- Given the user selects a valid major, when the user enters a major, then the user should view the correct major course requirements on the graduation requirements page.
- Given the user selects a required course of their major, when the user selects a course, then they should be redirected to the course page for that major.
- Given the user selects a different major, when the user inputs the changed major, then the changed major's requirement courses should be reflected on the graduation requirements page.

User Story #12

4.b: As a user, I would like to access minor requirements

#	Description	Estimated Time	Owner
1	Store information about minor requirements in Degree object	0.5 hr	Anushka
2	Accept request from user to view particular minor(s) requirements and send back info	0.5 hr	Anushka
3	Display requirements on web page	3 hrs	Shivani

Acceptance Criteria:

- Given the user selects a valid minor, when the user enters a minor, then the user should view the correct minor course requirements on the graduation requirements page.
- Given the user selects a required course of their minor, when the user selects a course, then they should be redirected to the course page for that minor.
- Given the user selects a different minor, when the user inputs the changed minor, then the changed minor's requirement courses should be reflected on the graduation requirements page.

User Story #13

6.a: As a user, I would like to enter my previous grades.

#	Description	Estimated Time	Owner
1	Accept a grade for each course in degree plan	1 hr	Cole
2	Modification to course record to show grades	1 hr	Patricia

Acceptance Criteria:

- Given that the UserCourse object is constructed correctly, when the user clicks ‘submit grade’ for a specific course, then the UserCourse object should update with this new grade.
- Given that the user has pre-entered grades for a course, when the user clicks ‘submit grade’ for each course, then they should see the field populate with their grade.
- Given that the grade input is invalid, when the user clicks ‘submit grade’, then they will be prompted to resubmit a grade.

User Story #14

1.e: As a user, I would like to modify/delete my account

#	Description	Estimated Time	Owner
1	Create user profile page	2 hrs	Patricia
2	Create a profile modification page + delete button	2 hrs	Patricia
3	Save profile changes back to database	1 hr	Anushka
4	Delete account in database	1 hr	Anushka

Acceptance Criteria:

- Given that the user has signed in, when the user navigates to said profile page, then they will view a profile page that contains basic information about them: name and logout button.
- Given that the user is signed in, when the user navigates to the profile page, then they will see the opportunity to make edits to their profile information.
- Given that the user is signed in, when the user edits a profile field, then they will see it update on the UI and it will update in the database.
- Given that the user is signed in, when the user presses the delete account button, then

they will see that their profile has been deleted on the UI (and it will be updated in the database) and be redirected to the sign in page.

User Story #15

6.b: As a user, I would like to be able to input AP credit information.

#	Description	Estimated Time	Owner
1	Intake information from Purdue AP Credit Page	3 hrs	Chloe
2	Implement AP course selection UI	2 hrs	Shivani
3	Send AP information to the backend and save information	1 hr	Chloe

Acceptance Criteria:

- Given the AP course data is constructed properly, when the user opens the AP selection UI then they will be able to view a list of AP courses.
- Given the user has a valid AP course, when the user selects the AP course then they will be able to view Purdue course credit equivalent(s) for the given AP course.
- Given the user has a valid AP course and score combo, when the user selects the AP course then they will have the option to save their AP course score and corresponding course equivalencies such that it can be viewed across sessions.

Remaining Backlog

Functional Requirements

* = if time allows

1. Users can manage their account

   As a user,

   ~~a. I would like to sign up for an account~~
   ~~b. I would like to log in/out of my account~~
   ~~c. I would like the page to remember me so I don't have sign in every time~~
   ~~d. I would like to reset my password via email~~
   ~~e. I would like to modify/delete my account~~
   f. *I would like to use the site without being logged in
   g. *I would like to be promoted to an admin
   h. *I would like to sign up for Universities other than Purdue University WL

2. Users can enter/modify their course plan

   As a user,

   ~~a. I would like to enter my current plan of study~~
   ~~b. I would like to modify my current plan of study~~
   ~~c. I would like to enter my degrees~~
   d. I would like to enter transfer credits/placement tests
   ~~e. I would like to enter AP scores~~
   f. I would like to create and navigate multiple degree plans
   g. I would like to be notified if I am missing a prerequisite
   h. I would like to override prerequisite warnings
   i. *I would like to download an offline copy of my data
   j. *I would like to upload a previously downloaded data file to my account

3. Users can enter specific course data

   As a user,

   ~~a. I would like to enter my final score for each course~~
   b. I would like to select a professor for each course
   c. I would like to select course times for each course

4. Users can get degree information

   As a user,

   ~~a. I would like to access major requirements~~
   ~~b. I would like to access minor requirements~~
   c. I would like to see degree requirement overlap
   d. I would like to see the total requirements for multiple degrees
   e. I would like courses to be recommended to me based on selected degrees and remaining required courses

    f. *I would like to see non-course degree requirements (e.g. Civics Literacy)
5. Users can get course information

   As a user,

    ~~a. I would like to access course descriptions~~
    ~~b. I would like to access course prerequisites~~
    c. I would like to access previous course scheduling
    d. I would like to view previous course instructors
    e. I would like to see instructor ratings on Rate My Professor
    f. I would like to see classes in Boiler Grades
    g. I would like to see instructors in Boiler Grades
    h. *I would like to see options to test out of the course (e.g. CLEP Tests)
6. Users can view a visual degree plan

   As a user,

    a. I would like the plan to change dynamically with entered information
    b. I would like to see a block describing the course when I hover over it
    c. I would like the degree plan to be sorted by semester
    d. I would like courses to be clearly color coded by completion
    e. *I would like to see a calendar view of my classes each semester
    f. *I would like to export my course schedule to another calendar app
7. Users can calculate grades

   As a user,

    a. I would like to see my cumulative GPA
    b. I would like to see my semester GPA
    c. I would like to see my major GPA
    d. I would like to approximate grades based on assignment grades and category weights
8. *Course and instructor rating system

   As a student,

    a. *I would like to view reviews for instructors
    b. *I would like to leave reviews on instructors I have had
    c. *I would like to view reviews for courses
    d. *I would like to leave reviews for courses I have taken
9. *Admin Panel

   As an administrator,

    a. *I would like to manually update course data
    b. *I would like to update other important information
10. *Communicate with other users

    As a user,

    a. *I would like to optionally have contact information shared by default
    b. *I would like to optionally have contact information shared by course

c. *I would like to view the contact information of other students enrolled in the same course as me
d. *I would like to communicate with classmates in the same course as me through the website

11. Course data automatically updates

As an administrator,

a. ~~I would like the server to automatically update course descriptions~~
b. I would like the server to automatically update previous/current scheduling
c. *I would like the server to fetch Rate My Professor data as needed
d. *I would like the server to fetch Boiler Grade data as needed

Non-Functional Requirements

1. Architecture

As a developer,

a. I would like the front-end and back-end separated to encapsulate the application's relevant tasks.
b. I would like to use ReactJS and TypeScript for the front-end of the application.
c. I would like to use a NodeJS back-end in order to manage requests, house course catalog/scheduling information, and support information refresh.
d. I would like to use a MongoDB (NoSQL) database to store account, degree-related, and course-relevant information.
e. I would like to focus on the usability of the application.

2. Performance

As a developer,

a. I would like our application to load/update the webpage in under 5 seconds.
b. I would like page navigation to take less than 1 second.
c. I would like UI updates (ex: button presses) to take less than 100ms to update (or provide a loading icon).
d. I would like fast API responses, with <1Mb of data taking less than 1s and larger amounts taking no more than 5s.
e. I would like our backend to have high uptime >99.9% and to be free from bugs that would cause crashing.

3. Security

As a developer,

a. I would like to hash passwords using a common password hashing mechanism (ex: bcrypt).
b. I would like to implement authentication and access control on our API.
c. I would like to use rate limiting to prevent systematic attacks.
d. I would like to use object-relational mapping to simplify database accesses and reduce potential attack surfaces.

4. Usability

   As a developer,

   a. I would like the website interface to be intuitive and easy to understand.
   b. I would like the website's different features to be categorized by area and represented uniquely.
   c. (If time allows) I would like the website's graphical representations to also be represented in plain text for accessibility.
   d. (If time allows) I would like the lab to be accessible from mobile clients
   e. I would like specific data to be imported and exported out of the service to allow for information sharing.

5. Testing

   As a developer,

   a. I would like to implement separate front-end tests using a front-end testing framework (ex: Jest).
   b. I would like to implement separate back-end tests using a back-end testing framework.
   c. I would like to implement end-to-end testing using an end-to-end testing tool (ex: Protractor).

6. Hosting

   As a developer,

   a. I would like to run NodeJS and React on a local machine, utilizing Docker to host a local database.
   b. I would like to utilize a publicly accessible server to be used as a testing website for development purposes.