Pete's Plan

Team 36 - Product Backlog

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

Problem Statement

Purdue University offers 6,700 courses in 200 specializations, making it nearly impossible for students to navigate all the requirements. Additionally, degree plan information is widely dispersed and difficult to find, not only for new students, but also upperclassmen. Pete's Plan will allow students to make degree plan choices more effectively and in a well-informed manner by pulling together resources such as plans of study, course descriptions, prerequisites, course scheduling, and more. This eliminates the need for hours of browsing through an enormous number of tabs, prevents students from asking their advisor the wrong questions, and provides guidance to students who are feeling overwhelmed.

Background

Audience

Many colleges, including Purdue, offer a wide variety of courses. College students have many degree paths that they may choose from, with complicated course requirements and prerequisites. Yet, this information is frequently scattered across several different sources, making course scheduling and degree planning difficult. We want to give college students (especially freshman/prospective students) an easy-to-use platform that guides them in planning their college experience.

Similar Applications

Several sources of course information already exist. In particular with Purdue, there is myPurduePlan, UniTime scheduling assistant, and the Purdue University Catalog for course information and prerequisites alone. Information regarding degree requirements, transfer course credits, course ratings, and course difficulty is scattered among various different sites and webpages.

Limitations

A major limitation of all these platforms and sources of information is decentralization. This information is in different formats on different sites. For example, degree requirements for a Purdue Computer Science degree are in an entirely different format and location than the requirements for a Purdue Engineering Degree. Additionally, many existing sites such as the Purdue University Catalog and myPurduePlan have bad usability. Our goal is to combine this information into a single, unifying application that is intuitive and easy to use. None of these sites allow for degree-specific course planning. We intend to import degree information into our application to allow for degree planning and looking ahead to degree possibilities.

Functional Requirements

1. As a user, I would like to create an account so that I can store my information.
2. As a user, I would like the page to remember me so I don't have to log in every time.
3. As a user, I would like to be able to reset my password via email.
4. As a user, I would like to be able to delete my account.
5. As a user, I would like to be able to enter my past classes to show up in the degree plan and be considered for requirements.
6. As a user, I would like to be able to enter my current classes to show up in the degree plan and be considered for requirements.
7. As a user, I would like to be able to enter my grades so that my GPA can be calculated and so pass/fail is considered in course requisites.
8. As a user, I would like the website to recommend courses to me based on my selected degree(s) to fill out my degree plan.
9. As a user, I would like to have a visual plan of study that is dynamic.
10. As a user, I would like to have a visual plan that can change based on what track I select.
11. As a user, I would like to have a visual plan that shows a hover bubble of courses catalog.
12. As a user, I would like to have a visual plan that is sorted by semesters.
13. As a user, I would like the courses to be clearly color coded by completion.
14. As a user, I would like to be able to approximate my grade in a particular course by using the grade calculator form.
15. As a user, I would like to access major requirements through the site.
16. As a user, I would like to access minor requirements through the site.
17. As a user, I would like to access information on course descriptions.
18. As a user, I would like to access information on course prerequisites.
19. As a user, I would like to access scheduling information on a course: on/off semesters, capacity, etc.
20. As a user, I would like to be able to input AP credit to fulfill major requirements and class prerequisites.
21. As a user, I would like to be able to override all previous functionality under specific user conditions.
22. As a user, I would like to create multiple degree plans.
23. As a user, I would like to switch between multiple degree plans.
24. As an administrator, I would like the site to automatically update with new courses.
25. As an administrator, I would like the site to automatically update information from other services.
26. As a user, I would like to access the website from any of my devices.
27. As a user, I would like to view information on course-specific test-out options such as CLEP tests if time allows.
28. As a user, I would like to view information on non-course-based degree requirements such as a civics literacy test if time allows.

29. As a user, I would like to be able to access past grade history when viewing a course, provided from a resource such as BoilerGrades if time allows.
30. As a user, I would like to be able to input my feedback on classes when viewing a course if time allows.
31. As a user, I would like to be able to input my feedback on teachers when viewing a course if time allows.
32. As a user, I would like to be able to access student feedback on classes when viewing a course if time allows.
33. As a user, I would like to be able to access student feedback on teachers when viewing a course if time allows.
34. As a user, I would like this website to be applicable for schools outside of Purdue University if time allows.
35. As a user, I would like to be able to export specific data to be used so it can be shared among users if time allows.
36. As a user, I would like to be able to import specific data to be used so it can be shared among users if time allows.
37. As a user, I would like to be able to use the website without being logged in (knowing my data would not be saved) if time allows.
38. As a user, I would like to be able to be promoted to access the admin panel if time allows.
39. As an admin, I would like to manually update course catalog data if time allows.
40. As an admin, I would like to manually update necessary important information if time allows.
41. As a user, I would like to be able to access a visually accessible version of the site if time allows.
42. As a user, I would like to opt in to having my contact information (email, phone, etc.) shared with students in the same class to help form study groups if time allows.
43. As a user, I would like to be able to chat with other users from my classes within the app if time allows.
44. As a user, I would like to see ratings for professors in the courses I'm taking from RateMyProfessor if time allows.
45. As a user, I would like to access a calendar view of my classes for any given semester if time allows.

Non-Functional Requirements

Architecture
We want to make sure that our front-end and back-end are separated so that we can encapsulate the tasks relevant to either end. For the front-end, we want to use ReactJS and TypeScript, and we want usability to be a focus. In the back-end, we want to use a NodeJS backend that manages requests, houses course catalog/scheduling information, and supports refresh of this information. All of our accounts, degree-related information, and course-relevant data, will be stored using MongoDB, a NoSQL database.

Performance
Our application should be able to load or update the webpage in under 5 seconds. Page navigation should be fast, taking less than 1 second. UI updates such as button presses should be responsive, taking less than 100ms to update (or displaying a loading icon if more time is needed). API responses should be fast: small amounts of data (<1Mb) should take less than 1s and larger amounts should take less than 5s. Our backend should have high uptime (>99.9%), it should be free from bugs that cause it to crash.

Security
Security is an important part of our application, since we will be storing users' private data. We will hash passwords using a common password hashing mechanism such as bcrypt. Then, we can implement authentication and access control on our API to ensure that users can only access data that is relevant to themselves. Additionally, we can utilize rate limiting to prevent systematic attacks. We can use an object-relational mapping to simplify database accesses and reduce potential attack surfaces to our database.

Usability
The interface of the website should be intuitive and easy to understand. As the website offers a variety of different features, each will be categorized based on area and represented in an unique way. The website's graphical representations also have the option to be represented in text for accessibility. Additionally, specific data can be imported and exported out of the service, allowing it to be shared easily by students to relevant individuals.

Testing
Testing is an important part of projects. We can implement separate front-end tests using a front-end testing framework with React such as Jest, and separate back-end tests using a back-end testing framework. We will also need to implement end-to-end testing using a tool such as Protractor.

Hosting

An important part of the project is how the website will be hosted and accessed. For development purposes, NodeJS and React can be run on a local machine with Docker to host a local database. For testing purposes, several of our team members have a publicly-accessible server that can be used as a testing website. Public hosting of this project is likely outside of this class' scope, but this hosting is similar to hosting the testing server.