

# **PRODUCT BACKLOG - TEAM 16**

## **Campus Reviews**

### **Team Members:**

Anirudh Seela, Arnav Gupta, Darsh Dalal, Rohith Rajashekarbabu, Shreyas Kharbanda, Tejas Prakash.

### **Problem Statement:**

Every year thousands of students leave the comfort of their homes behind and enter a complex educational ecosystem to chase their futures. Bombarded with a new environment, responsibilities, and socio-cultural landscape, it takes many students hours of clueless wandering and guessing that most often lead to misguided decisions and unwanted outcomes. We want to reduce this learning curve by creating a one-stop-shop for all things Purdue. Through a peer-based network, new students can learn about and understand the campus through the experiences of their peers.

### **Background Information:**

#### **Audience:**

Incoming students have tremendous confusion, concerns, and questions about the onboarding process at Purdue due to information overload. Current students are also unsure about their upcoming classes and professors. Our application targets this population and offers a solution with a centralized and reliable location for any Purdue related information.

#### **Similar Platforms and Limitations:**

Reddit, RateMyProfessor, and Yelp are good comparisons for the product that we plan to develop. We aim to blend these existing platforms into a more reliable, and smooth college-based software. Specifically speaking about Reddit, although the platform offers great communication between users there is no true credibility or authentication of answers. To battle this issue, PCR allows solely the students/faculty of the university to respond to questions about classes and dorms. RateMyProfessor as an idea is very useful to prospective students, but we look to improve upon the engagement from students to procure more accurate and objective reviews. Another issue with RateMyProfessor is that the mobile application is not very user-friendly, something we want to improve on. We plan to incorporate a more college-based yelp system into the app as well, this will help us recommend places to visit to our users while also limiting the review of nearby buildings and restaurants to Purdue students.

## **Requirements:**

### **Functional Requirements:**

1. As a user, I would like to view an onboarding screen that allows me to understand the features of the application.
2. As a user, I would like to register for an account.
3. As a user, I would like to login to my account.
4. As a user, I would like to continue being logged into my account unless I explicitly log out from the application.
5. As a user, I would like to delete my account and data.
6. As a user, I would like to change the password for my account in case I forget it or my account gets breached.
7. As a user, I would like to create a custom username.
8. As a user, I would like to edit my custom username.
9. As a user, I would like to easily navigate to all the different product screens.
10. As a user, I would like to download the application on either an iOS or an Android device without loss of functionality and user experience.
11. As a user, I would like access to a help/FAQ page if I have trouble with the app.
12. As a user, I would like to have access to a search engine functionality so that I am easily able to find a class.
13. As a user, I would like to upvote or downvote a post or review.
14. As a user, I would like to have a fair metric to assess the credibility of other users.
15. As a user, I would like to choose to see more credible reviews first.
16. As a user, I would like to choose to see the recent reviews first.
17. As a user, I would like to choose to see the oldest reviews first.
18. As a user, I would like to choose to see the recent posts first.
19. As a user, I would like to choose to see the oldest posts first.
20. As a user, I would like to search for reviews by the relevancy of my query.
21. As a user, I would like to search for posts by the relevancy of my query.
22. As a user, I would like to see community approved reviews at the top of the review feed for easier access.
23. As a user, I would like to see the credibility score of another user.
24. As a user, I would like to read others' reviews for classes and professors.
25. As a user, I would like to read others' reviews for on-campus dining facilities and residence halls.
26. As a user, I would like to write my reviews for classes and professors.
27. As a user, I would like to write my reviews for on-campus dining facilities and residence halls.
28. As a user, I would like to edit my reviews for classes and professors.
29. As a user, I would like to edit my reviews for on-campus dining facilities and residence halls.
30. As a user, I would like to delete my reviews for classes and professors.
31. As a user, I would like to delete my reviews for on-campus dining facilities and residence halls.
32. As a user, I would like to write my posts for classes and professors.
33. As a user, I would like to write my posts for on-campus dining facilities and residence halls.

34. As a user, I would like to read others' posts for classes and professors.
35. As a user, I would like to read others' posts for on-campus dining facilities and residence halls.
36. As a user, I would like to edit my posts for classes and professors.
37. As a user, I would like to edit my posts for on-campus dining facilities and residence halls.
38. As a user, I would like to delete my posts for classes and professors.
39. As a user, I would like to delete my posts for on-campus dining facilities and residence halls.
40. As a user, I would like to report users if they violate any of the platform policies.
41. As a user, if I am reported, I should be able to explain my position and have my credibility score taken into consideration before any decision is taken.
42. As a user, I would like to see recommended dining court options.
43. As a user, I would like to see recommended roommate options.
44. As a user, I would like to create a profile and showcase my information.
45. As a user, I would like to see other's profiles in order to select my roommate.
46. As a user, I would like to have a variety of themes for the interface (light/dark mode).
47. As an administrator, I would like to remove posts.
48. As an administrator, I would like to remove reviews.
49. As an administrator, I would like to uphold terms and conditions.
50. As an administrator, I would like to track engagement with the application.
51. As an administrator, I would like to have access to an emergency shutdown option.
52. As an administrator, I would like the ability to push updates to the application.
53. As an administrator, I would like to collect and review feedback from users.
54. As an administrator, I would like to authenticate users through proof of enrollment at Purdue University.
55. As an administrator, I would like the ability to ban accounts along with giving them reasoning through email.
56. (If time allows) As a user, I would like to directly message other users.
57. (If time allows) As a user, I would like to have a privacy setting for message requests. Either can't be messaged, message after an accepted request, or any message.
58. (If time allows) As a user, I would like to block other users from messaging me.
59. (If time allows) As a user, I would like to receive notifications when I am messaged.
60. (If time allows) As a user, I would like to receive notifications when someone has responded to my post.
61. (If time allows) As a user, I would like to have access to a search engine functionality so that I am easily able to find on-campus dining facilities.
62. (If time allows) As a user, I would like to have access to a search engine functionality so that I am easily able to find residence halls.
63. (If time allows) As a user, I would like to have access to a search engine functionality so that I am easily able to find a professor.
64. (If time allows) I would like to be able to change my primary email for login and email purposes.
65. (If time allows) Sentiment analysis of reviews for a more comprehensive credibility score calculation.

## **Non-Functional Requirements:**

### Architecture:

We are going to follow the client-server architecture model. Our highest priority is to build a modular, highly integrable codebase for rapid iteration. Moreover, such an architecture would enable better separation of concerns and an easily navigable codebase for better developer productivity.

Based on the client-server model, we are going to build the client (front-end) and server (back-end) separately. The server will be built modeling RESTful (Representational State Transfer) Application Programming Interface, written in JavaScript using the popular Node.js framework. Node.js is an open-source, cross-platform, back-end JavaScript runtime environment that runs on the Chrome V8 engine and executes JavaScript code outside a web browser. Through the Node Package Manager (NPM), Node.js allows downloading third-party modules that enable faster development and re-usability for commonly required structural building blocks.

Moreover, for data persistence, we will be using the Firestore NoSQL database by Google Firebase. Our system will require heavy data flow to and from the database, we want to have a database that is quick and robust enough to handle the data storage and retrieval. As the prototyping and sprints will lead to new discoveries and understanding of what data we require and the order in which we need to fetch it, having a non-tabular format that can be easily modified will enable the developers to be productive and iterate quickly without any time-consuming overhead of sanitizing the codebase and restructuring the tables that generally comes with SQL databases.

The client will be built using React Native. React Native is an open-source mobile application framework created by Facebook, Inc. It uses the React way of writing re-usable, stateful, and lightweight components that can be used to create a user interface. It uses a bridge, a way to compile applications written with JavaScript into native mobile applications. Thus, it allows the developer to maintain one codebase while deploying the application on all major platforms.

### Performance and Scalability:

We want to focus on performance and scalability throughout our development endeavor to create a seamless user experience. As our application will be a mobile application, it is crucial for the application to be small in size (<50 MB) for easy download on the user's phone and cache only if direly required so that our application does not hoard space on the user's phone.

By following the client-server model, we want to maximize performance by developing a thin client that delegates the computational work to the server to minimize crashes and have a secure place for business logic.

As the server would be the primary engine for handling multiple requests from multiple users at the same time, we want to focus on a non-blocking and asynchronous architecture that can handle I/O intensive requests. On the other hand, as we are building a thin client, we want to focus on compactness and

simplicity. To achieve this, we plan to use a lightweight, stateful, and reusable component-based library that can enable simple computations on the client, such as filtering and sorting, without disturbing the server.

Through using such a non-blocking, asynchronous architecture on the server-side along with the lightweight architecture on the client-side, we can easily scale the application to accommodate as many users as required. Moreover, our database is an on-demand, cloud-based database that can be scaled up or down based on user traction. Such a structure gives us a great headstart when it comes to scalability and performance.

We plan to scale this application for other colleges in the future.

#### Security:

Security and privacy are important and of high-priority to foster an engaged and safeguarded community. We want to maximize security on our platform by minimizing sensitive data. Leveraging this approach, we delegate as much sensitive data storage to well-established, secure services.

The main sensitive data that will most commonly flow through our system is the username and password. We delegate such data security by using Firebase by Google Cloud Platforms. Firebase takes charge of authenticating the user, generating a token, and checking for persistence when accessing private routes, making it a secure, efficient, and scalable option for authenticating users.

#### Usability:

It is crucial for a mobile application to facilitate information aggregation for students to be easily navigable and aesthetically pleasing. For user retention and engagement, the application needs to be easy to understand, intuitive, and devoid of jargon or any other specialized structural system. Moreover, we want our application to run on all different phones without degradation in user experience or crashes.

#### Hosting/Deployment:

The React Native Application will be deployed using Expo to the App Store and the PlayStore. The Node.js backend will be deployed on Heroku or Digital Ocean based on economic and performance feasibility. The Firestore database would be deployed using the Google Cloud Platforms (GCP).

---