

Team 11

Purdue Eats

Team Members:

Aniket Agnihotri

Anisha Sinha

Eric Thompson

Mark Jin

Sean Joo

Vaastav Arora



Problem Statement:

With the onset of COVID-19, Boilermakers have and continue to experience long wait times at Purdue's dining facilities and struggle to make good dietary decisions. PurdueEats aims to utilize machine learning (ML) model(s) to understand a student's eating habits and provide them with a personalized dining experience at Purdue's dining facilities. PurdueEats is tailored towards user-specific dietary needs and provides more relevant and timely information to users compared to existing applications.

Background Info:

Audience:

Purdue students and faculty have an abundance of dining choices, but information on the quality and convenience of certain food options is often unclear. While there are some existing options for obtaining information on dining experiences, it is still a challenge for students and faculty to make quality diet choices and integrate their meals with their busy class schedules. An application that brings all the necessary information together and personalizes the dining experience does not yet exist. Because of this, Purdue residents often struggle to decide on the best place to spend their meal swipes.

Similar platforms:

There are currently three relevant applications regarding Purdue's dining experience as well as user fitness tracking. The first is Purdue Menus, which provides students with the menus of every dining facility as well as the hours that they serve those respective meals. The second is MyFitnessPal, which helps users keep track of the meals they eat, their nutritional information, alongside the exercise they perform. The last application is Google Maps, particularly an extension, which provides popular times that locations are frequented, including the dining courts on Purdue's campus.

Limitations:

Existing platforms only provide solutions to one facet of the meal planning problem, either in the form of viewing available meals or tracking meal consumption, but not both. Additionally, these services lack integration with individual schedules and fail

to incorporate an understanding of user preferences when it comes to selecting and consuming meals. Through PurdueEats, we hope to combine the many facets of meal planning and add an element of personalization previously unseen, serving it through a unified, intuitive interface in the form of an easy-to-use mobile application.

Functional Requirements:

1. As a user, I would like to be able to register for a PurdueEats account so that all of my information is associated with my account.
2. As a user, I would like to be able to login to my PurdueEats account so that I can track my meals and view wait times.
3. As a user, I would be able to login to my PurdueEats account via two-factor authentication so that my account is more secure.
4. As a user, I would like to be able to change my account name or password so that I have control over my credentials.
5. As a user, I would like to be able to change my registered email id so that I can have flexibility with my account credentials.
6. As a user, I would like my password to be reset when required so that I can still access my account if I forget my password.
7. As a user, I would like to be able to delete my user account so that my user-generated data is removed from the application's database.
8. As a user, I would like to have a profile page so that I can edit my credentials, name, and meal plan as well as access my past meal history.
9. As a user, I would like to upload a profile picture to my profile so that others can see me when finding a lunch buddy through the app.
10. As a user, I would like to have my login persist on exiting and re-entering the application so that I do not need to spend time logging in every time I would like to open the app.
11. As a user, I would like to have a navigation bar so that I can access all of the features of the application from this screen.
12. As a user, I would like to be able to easily view meal options across Purdue's various dining facilities so that I may choose where to eat from the options I'm given.
13. As a user, I would like to be able to filter meals based on if they are vegetarian or contains specific allergens so that I can be sure that what I am eating fits my dietary needs.
14. As a user, I would like to be able to view the wait-times at each of Purdue's dining facilities so that I can decide where it would be quickest for me to eat.
15. As a user, I would like to be able to view the distance from my location to each of Purdue's dining facilities so that I may account for distance to each facility as a factor in my dining decision-making.
16. As a user, I would like to be able to view a map of my current location and each dining court so that I can receive directions to any dining court of choice.
17. As a user, I would like to be able to mark meals as favorites so that I won't miss out on them when they are offered.

18. As a user, I would like to be notified when my favorited meals are available and being served at a particular location so that I can track my preferred meals.
19. As a user, I would like to be able to toggle notifications for each of my favorited meals so that I can customize the meals that I am tracking.
20. As a user, I would like to be able to rate meals so that I get better meal recommendations based on my preferences.
21. As a user, I would like to be able to input times that I am free to eat so that I am only recommended meals during those times.
22. As a user, I would like to be given meal recommendations that are consistent with my past eating habits so that I can stick to eating foods that I enjoy and have eaten in the past.
23. As a user, I would like to be able to match with other users to dine at locations that meet our mutual dining preferences so that I can meet new people while still abiding by COVID-19 regulations.
24. As a user, I would like to be able to share my current location with my matched user so that I can meet up with them and dine together.
25. As a user, I would like to have a "I'm here" button so that I can let my matched user know that I have arrived at the agreed location.
26. As a user, I would like to be able to view a visual representation of nutritional information regarding my meals so that I can have a clear understanding of the health impacts of the food I am eating.
27. As a user, I would like to be able to view my past eating history so that I can keep track of my diet.
28. As a user, I would like to be able to view the number of meal swipes that I have remaining for the week so that I can manage my meal swipe count better.
29. As a user, I would like to be able to view my dining dollars balance so that I can track my transactions.
30. As a user, I would like to be able to update and edit my dining dollars balance so that I can make necessary changes if my meal plan changes.
31. As a user, I would like to be able to write reviews about dining courts so that I can share my experiences with others.
32. As a user, I would like to be able to upvote or downvote reviews from others so that I can promote relevant reviews and demote irrelevant ones.
33. As a user, I would like to be able to report reviews that are spam or inappropriate so that they can be taken down.
34. As a user, I would like to be able to read reviews about dining courts so that I can learn more about a dining court from my fellow students before making a decision on where to eat.
35. As a user, I would like to use a search bar to search through the dining court menus on any given day.
36. As a user, I would like to be able to adjust the app's text size so that I am able to clearly read information from the app.
37. As a user, I would like to be able to choose between a dark mode and a light mode UI so that I can have a color scheme that suits my preferences.

38. As a user, I would like to be able to access a settings page so that I can manage my notifications, light/dark mode UI, text size, and fun fact feature.
39. As a user, I would like to be able to view answers to frequently asked questions so that if I run into issues with the app I can find potential solutions.
40. As a user, I would like to be able to send feedback to the creators of the application within the application so that if there are any issues the developers can fix it.
41. As a user, I would like to view cumulative weekly summaries of what all users are eating through the app so that I can find popular eating trends.
42. As a user, I would like to be able to see a wrap-up of the food that I have eaten over a semester so that I can look back on my eating habits for the semester.
43. As a user, I would like to have a daily Purdue fun fact displayed each day that I log in so that I have an enjoyable user experience.
44. As a user, I would like to disable the display of a Purdue fun fact so that I have a more minimalistic app experience.
45. As a developer, I would like to predict future wait-times for each dining facility based on the historical menu items and wait-time logs.
46. As a developer, I would like to receive and record menu items from Purdue's dining menu website so that I can use the data as a factor in our wait-time prediction modeling dataset.
47. As a developer, I would like to receive and record user-inputted wait-time logs into the database so that I can use the data as a factor in our wait-time prediction modeling dataset.
48. As a developer, I would like to merge the dining menu log and wait-time log into a single dataset so that I can use the merged dataset for the wait-time prediction model to train and predict.
49. As a developer, I would like to be able to visualize the model training process so that I can improve model performance based on new insights.
50. As a developer, I would like to be able to view database analytics so that I can track and scale the database.
51. As a developer, I would like to be able to view user feedback submitted through the application so that I can incrementally remove bugs and improve the application.
52. As a developer, I would like to view user data analytics so that I can improve user experience over time.
53. As a developer, I would like to view API interface documentation so that front-end and back-end developers can work independently.
54. As a developer, I would like the application to be containerized so that I have more control of the permissions and resources used by the application.

Non-Functional Requirements:

Architecture and Performance

To complete our project, we plan to divide our application into two independent development segments, the frontend and backend.

The backend is further divided into two independent segments, the API, modeled as a RESTful API interface, and the ML model based on Graph Convolutional Network (GCN) architecture and XGBoost. The GCN will be built in Python using Tensorflow due to Tensorflow 2.0 flexibility, portability, and performance benefits offered on a multitude of hardware. The XGBoost library will be imported to Python and operate on a merged menu-item and wait-time dataset as training data. Furthermore, the API will be built using Starlette due to the convenience and performance benefits it offer.

The frontend will be implemented using React Native, making it available to both Android and iOS users. The UI will be built using templates from the React Native Material Kit, React Native, and Bootstrap packages. These templates include several prebuilt elements that will assist in easily building a foundation for the UI and allows for customization to fit the application's specific needs. For example, in Material Kit there are navigation bars with built-in notification and profile-registering sections that will be helpful for the PurdueEats application.

Response Time

A user will receive a response from the app instantly (~500 ms), limited only by their internet connection. With these quick response times, users will be able to see which dining facilities are best fit for their current schedules, meal preferences, and locations. Moreover, through the lunch buddy mechanic, PurdueEats will match users on a first come, first serve basis, making response times largely depend on number of unmatched users.

Security

Since our application records and stores personal data, security is one of the highest priorities for our application. Each user can only view their own past dining information, favorited items and submitted ratings and need to be logged in to view their personal data. Regarding API access, users have to be authenticated before accessing any user information. Moreover, each user account is protected through two-factor authentication to further prevent security violations. Additionally, all user data in the backend is stored using BigQuery with a model of "least privilege" access, thus further bolstering the security of the system.

Usability

Our project aims to support a fluid user interface so that users have an easy time locating the various features implemented in our application. Similar applications

often have rigid user interfaces which makes it difficult to obtain dining facility menus and reduces visibility of key information and features. Thus, through our application, we plan on easily informing users on necessary dining information coupled with queueing times and location to increase the visibility of relevant information. Any information that the user needs will be accessible through ≤ 5 screens of traversal from any location in the app. Moreover, our application will be accessible and operational for both Android and iOS devices 24 hours a day, 7 days a week so that we can ensure satisfaction of all users.

Hosting/Deployment

The backend and frontend will be entirely separate, enabling individual deployment and hosting. The backend will be hosted on Google Cloud Platform, given its modular cloud services, integration with BigQuery, and load balancing ability to implicitly manage the scale of our application. The frontend will be deployed as an application to both Android and iOS app stores, making it easily accessible to users.

Scalability

Our backend infrastructure will use parallel and distributed computing-based cloud solutions, allowing us to accommodate up to 40,000 daily users. We will be able to service up to 5,000 server request per minute with a maximum response size of 100 MB per request, limited only by our cloud payment plan. Additionally, as our database uses distributed querying, we are able to consistently guarantee a performance $\leq 4s$ per query for a dataset size up to 200 GB. Our database also has the ability to autoscale up to to a size of 1 TB, once again limited only by our payment plan. Such a scalable infrastructure should help us accommodate Purdue's student body and aid each and every student in their meal planning ventures.