Team 11

Purdue Eats

Team Members:

Aniket Agnihotri

Anisha Sinha

Eric Thompson

Mark Jin

Sean Joo

Vaastav Arora

Sprint 3 Planning Document

Overview

During this sprint, we hope to successfully implement features including login persistence, tailored meal recommendations, wait times, dining court and meal reviews, weekly summaries/semesterly wrap ups of eating history, and general accessibility features. One significant portion of this sprint on the backend is that we will be implementing all the DevOps requirements for our application, including setting up continuous integration and deployment as well as containerizing our application and hosting them as independent microservices. Overall, this sprint has a lot of challenging components for each team member and it will take a lot of communication and hours to accomplish.

Scrum Master: Vaastav Arora

Meeting Schedule: Tuesdays/Saturdays @ 2:00pm, Thursdays @ 1:00pm

Risks and Challenges:

Despite our successful past sprints, we faced slight difficulties thoroughly testing our implementations due to time constraints. It is important, especially during our final sprint, to be aware of the difficulty we had in the past and to be more proactive in this last stretch. Moreover, as this is the last sprint, we need to wrap up our deployment setup and turn our application into several microservices. This could require extensive refactoring of the backend codebase as we need to decouple functionality and containerize all the services. As the services are turned into containers before deployment, we may also face significant challenges with testing and debugging issues in the backend as containerizing the application will turn it somewhat opaque causing the tracing of code path to be very laborious. Additionally, once the application is decoupled into microservices, each component will run asynchronously opening us up to several concurrency related bugs that we need to watch out for.

Current Sprint Detail

User Story #1

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 that I would like to use PurdueEats.

#	Description	Estimated Time	Owner
1	Implement a function to store a user's data for retrieval on a subsequent reload of PurdueEats' frontend.	4 hrs	Aniket
2	Implement a function to fetch a user's stored data and automatically log them into PurdueEats.	4 hrs	Aniket
3	Implement a function to delete a stored user's data to prevent unauthorized access to PurdueEats.	4 hrs	Aniket
4	Manually/Unit test the storing and retrieving functions and the data associated with each.	2 hr	Aniket
5	**Total hours**	14 hrs	

Acceptance Criteria:

- Given that the function to store a user's data for retrieval is implemented correctly, a user's data will be stored for easy access by the application on a subsequent reload of the Expo emulator.
- Given that the function to fetch a user's stored data is implemented correctly, the correct data will be loaded to the application and executed.
- Given that the login functionality is implemented correctly, the user whose credentials were previously stored will automatically receive access to PurdueEats.
- Given that the delete stored data functionality is implemented correctly, the next user to use PurdueEats on the same device or emulator will be required to login again if the previous user logged out of the application.

User Story #2
As a user I would like to be recommended meals to eat based on my eating habits.

#	Description	Estimated Time	Owner
1	Create an UI panel to view five meals on the homepage juxtaposed to the eating habits UI panel.	4 hrs	Aniket
2	Implement a corresponding API route for processing and sending five recommended meals based on a user's eating history.	5 hrs	Vaastav
3	Manually/Unit test that these meals change as a user's eating habits change.	2 hr	Aniket
4	**Total hours**	11 hrs	

Acceptance Criteria:

- Given that the UI panel is implemented correctly, a user will be able to view five recommended meals.
- Given that the UI panel is implemented correctly, a user will be able to tap on each of these meals to view their associated nutritional information.
- Given that the API route is implemented correctly, the five meals sent to the frontend will be timely and accurate based on the eating history of a particular user.

User Story #3

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.

#	Description	Estimated Time	Owner
1	Create a UI panel to view the cumulative eating habits of all PurdueEats users.	5 hrs	Aniket
2	Implement a corresponding API route for processing and sending aggregate data on user eating habits.	4 hrs	Vaastav
3	Manually/Unit test meal cumulative weekly summary functionality.	2 hrs	Aniket
4	**Total hours**	11 hrs	

Acceptance Criteria:

- Given that the UI panel is implemented correctly, a user will be able to view the previous week's summary of aggregate eating habits at any time from the home page.
- Given that the UI panel is implemented correctly, a user will be able to see the top five most-picked meals from the previous week.
- Given that the API route is implemented correctly, the weekly aggregate summary of eating habits will be updated every week at a specified time to reflect the previous week.

User Story #4

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.

#	Description	Estimated Time	Owner
1	Create button on Menu page to redirect to the wait-times UI panel of Purdue's dining facilities	1 hr	Sean
2	Create a UI panel displaying the wait-times of Purdue's dining facilities	6 hrs	Sean
3	Create API route containing wait-times of dining facilities	4 hrs	Vaastav
4	Fetch and properly parse the predicted wait-times of each dining facility	4 hrs	Sean
5	Create manual tests to evaluate the functionality of wait-times.	1 hr	Sean
6	**Total hours**	16 hrs	

Acceptance Criteria:

- Given that the button is implemented correctly, a user will be able to successfully be redirected to the wait-times of Purdue's dining facilities when the button is clicked.
- Given that the UI panel is implemented correctly, the user should view the wait-times of Purdue's dining facilities as a graph.
- Given that the fetch is implemented correctly, the correct predicted times should be fetched for each dining facility.

User Story #5

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.

#	Description	Estimated Time	Owner
1	Create a button to instruct users on how to activate dark mode	1 hr	Sean
2	Create a pop-up describing how to access dark mode	2 hrs	Sean
3	Implement dark mode depending on user's settings	6 hrs	Sean
4	Create manual tests to evaluate the functionality of dark mode and light mode.	1 hr	Sean
5	**Total hours**	10 hrs	

Acceptance Criteria:

- Given that the button is implemented correctly, a user will be able to successfully be redirected to the pop up when the button is clicked.
- Given that the pop-up is implemented correctly, a user should be informed on how to activate light mode and dark mode.
- Given that the light mode and dark mode has been implemented correctly, a user should be able to turn their screen to whatever mode they want.

User Story #6

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.

#	Description	Estimated Time	Owner
1	Create a banner that has the Purdue fun fact	3 hrs	Eric
2	Create an API route containings all of the fun facts and refreshes daily	2 hrs	Vaastav
3	Connect backend and frontend of Purdue Fun Fact together for GET requests	5 hrs	Eric
4	Create manual tests to evaluate the functionality of Purdue fun fact displaying successfully.	1 hr	Eric
5	**Total hours**	11 hrs	

Acceptance Criteria:

- Given that the banner has been implemented correctly, the Purdue fun fact should be displayed.
- Given that the GET request has been implemented correctly, the fetch should return a Purdue fun fact.
- Given that the API has been implemented correctly, the Purdue fun fact will update daily.

User Story #7

As a user, I would like to be able to write reviews about dining courts so that I can share my experiences with others.

#	Description	Estimated Time	Owner
1	Create a UI page for where the user will write reviews	1 hr	Anisha
2	Create UI components for the user to enter their reviews in along with a submit button.	2 hrs	Anisha
3	Implement functions to ensure all parts of the review are filled out and that there are no issues with submitting the review.	2 hrs	Anisha
4	Implement navigation so dining court id's are saved on the page the user is writing the review on so the review is associated with the right dining court.	1 hr	Anisha
5	Create an API route to send the review data to the back end.	2 hrs	Vaastav
6	Implement a function that uses the API route to send the data to the backend	1 hr	Anisha
7	Create manual tests to test the ability to write a review	1 hr	Anisha
8	**Total hours**	10 hrs	

Acceptance Criteria:

- Given that the UI page is implemented correctly, there will be a place for the user to enter the title of their review as well as the body of their review.
- Given that the API route is implemented correctly, a Toast message will appear confirming that the user's review has been submitted and stored in the backend
- Given that either the title or the body of the review are empty, a toast message will appear telling the user to enter all parts of the review and try again.

User Story #8

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.

#	Description	Estimated Time	Owner
1	Create a dynamically rendering component to render information from the review	2 hrs	Anisha
2	Implement a function to fetch all the user reviews about a given dining court.	3 hrs	Anisha
3	Implement functionality to store all reviews in an array for easier access	1 hr	Anisha
4	Create an API route to get reviews from the database.	2 hrs	Vaastav
5	Create manual tests to view rendering of the reviews as well as for the API route.	1 hr	Anisha
6	**Total hours**	9 hrs	

Acceptance Criteria:

- Given that the API route is implemented correctly, all of the reviews that are associated with a given dining court in the database will be fetched.
- Given that the UI page is implemented correctly, all of the reviews fetched will be rendered for the user to read.
- Given that the function to get reviews is implemented correctly, only the reviews of that given dining court will be rendered.

User Story #9

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.

#	Description	Estimated Time	Owner
1	Create a UI component to upvote or downvote each review.	2 hrs	Anisha
2	Create API routes to get the upvotes and downvotes as well as store the upvotes and downvotes of a review.	2 hrs	Vaastav
3	Implement a function to post the upvote or downvote associated with a review using an API route.	2 hrs	Anisha
4	Implement a function to get if a user has upvoted or downvoted a review using an API route	2 hrs	Anisha
5	Create a UI change if the user has already upvoted or downvoted a review.	1 hr	Anisha
6	Create manual tests to test upvoting and downvoting capacity.	1 hr	Anisha
7	**Total hours**	10 hrs	

Acceptance Criteria:

- Given that the UI is implemented correctly, each review will have a UI component rendered with it to either upvote or downvote it.
- Given that the API route is implemented correctly, if the user upvotes or downvotes a review it will be stored in the database.
- Given that the UI changes are implemented correctly, if the review is upvoted or downvoted there will be a change in the UI to indicate to the user they have already voted.

User Story #10

As a user, I would like to be able to report reviews that are spam or inappropriate so that they can be taken down.

#	Description	Estimated Time	Owner
1	Create a UI component to report reviews	2 hrs	Anisha
2	Create an API route to tag a report as either spam or inappropriate.	2 hrs	Anisha
3	Implement function to send report information using API route	2 hours	Anisha
4	Implement a function to display a toast message upon submitting a report.	2 hrs	Anisha
5	Create manual tests to test reporting capacity	1 hr	Anisha
6	**Total hours**	9 hrs	

Acceptance Criteria:

- Given that the UI is implemented correctly, each review will have a UI component rendered with it to report it as spam or inappropriate.
- Given that the review is reported as spam and the API route is implemented correctly, a toast message will appear stating the review has been marked as spam and their report is submitted.
- Given that the review is reported as inappropriate and the API route is implemented correctly, a toast message will appear stating the review has been marked as inappropriate and their report is submitted.

User Story #11

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.

#	Description	Estimated Time	Owner
1	Create SQL queries to get all user-inputted meal-consumed information	2 hrs	Mark
2	Create function to sum macronutrients in consumed meals	1 hrs	Mark
3	Perform data analysis on individual user dietary habits	5 hrs	Mark
4	Create and post visualizations of dietary habits for user	4 hrs	Mark
5	Create UI panel to display visualizations and relevant information to app	3 hr	Aniket
6	Manual test summary visualization for user end	1 hr	Mark
	Total hours	16 hrs	

Acceptance Criteria:

- Given the API route is implemented successfully, a developer should have all dining activity history of each given user.
- Given the data analysis is performed on the data appropriately, functions should return plots and graphs of sums of meals consumed, macronutrients consumed, and other insights
- Given the visualizations on the app are completed, the app user should see the aforementioned plots and graphs on the front-end.

User Story #12

As a developer, I would like to perform data analytics on user dining trends and habits so that I can use such insights to identify potential problems.

#	Description	Estimated Time	Owner
1	Create SQL queries to retrieve all user information from database	4 hrs	Mark
2	Manually populate users and activity if sample population is insufficient	1 hr	Mark
3	Clean, sort, manipulate data onto custom database for aggregate analysis	3 hrs	Mark
4	Perform clustering/segmentation analysis to obtain dining trends and habits	5 hrs	Mark
5	Analyze and create visualizations of results	4 hrs	Mark
	Total hours	17 hrs	

Acceptance Criteria:

- Given the API route is implemented successfully, a developer should have meta-data of all user activity stored in the database.
- Given the data is sorted and manipulated, a developer should be able to run an analysis model using the dataset
- Given the analysis is successfully performed, a developer should be able to read results of aggregate user activity trends and obtain insights from said results.

User Story #13

As a developer, I would like to view API interface documentation so that front-end and back-end developers can work independently.

#	Description	Estimated Time	Owner
1	Obtain API routes on swagger, comment all routes to include function, input, output and status codes	3 hrs	Mark
2	Format and generate documentation based on comments on swagger	3 hrs	Mark
	Total hours	6 hrs	

Acceptance Criteria:

- Given that the documentation is written correctly, each route should have a detailed description that includes route function, input, output and status codes.
- Given that the documentation is compiled correctly, all of the route documentation would be collected into a single directory
- Given that the interface documentation is implemented correctly, front-end and back-end developers can easily access the documentation.

User Story #14

As a user, I would like to be able to access a settings page so that I can manage application settings.

#	Description	Estimated Time	Owner
1	Create button that allows user to access settings page from other pages	2 hrs	Eric
2	Create and organize UI layout that will allow user to edit the settings options that they choose	5 hrs	Eric
3	Implement settings so that it tells you if you are currently using light mode or dark mode for the application.	2 hrs	Eric
4	Manual testing for all of the settings buttons.	2 hrs	Eric
	Total hours	11 hrs	

Acceptance Criteria:

- Given that the UI is implemented correctly, each setting will have a button or slider that allows the user to update the settings of their choice.
- Given that the UI is implemented correctly, changing settings will change them throughout the entire app.
- Given the UI is implemented correctly, the user will be able to access the settings page from the profile page.

User Story #15

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.

#	Description	Estimated Time	Owner
1	Create button that allows user to access feedback page from the settings page	2 hrs	Eric
2	Implement UI so that the user can view the answers to commonly asked questions before submitting feedback.	2 hrs	Eric
3	Create UI for feedback page that has text input for user to type their feedback into	2 hrs	Eric
4	Implement route so that the users' input can be sent to the database and be accessed by the developers	3 hrs	Vaastav
5	Implement UI so that when the information is sent the user is notified that the submission was successful	3 hrs	Eric
6	Manual testing for feedback.	2 hrs	Eric
	Total hours	14 hrs	

Acceptance Criteria:

- Given that the UI is implemented correctly, the user will be able to input feedback into a textbox.
- Given that the UI is implemented correctly, the user will be able to view answers to commonly asked questions before submitting feedback to the database.
- Given the route is implemented correctly, feedback from the user will be sent to the database where the developers can view it.
- Given the UI is implemented correctly, the user should get a response from the database once their feedback is successfully submitted.

User Story #16

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.

#	Description	Estimated Time	Owner
1	Create an API route to fetch feedback data from database	2 hrs	Vaastav
2	Create a Javascript webpage that fetched and displays feedback information from fetch request	5 hrs	Sean
3	Implement a GET fetch call to receive the feedback to display	2 hrs	Sean
4	Create an admin user that is authenticated allowed to fetch user feedback information from the route	2 hrs	Sean
5	Create manual/unit tests for the API route and display page	2 hr	Sean
	Total hours	13 hrs	

Acceptance Criteria:

- Given that the fetch call has been implemented correctly, the feedback should be received.
- Given that the Javascript website has been implemented correctly, the user's feedback should be displayed in an orderly form.
- Given that the authentication is implemented correctly, only admins should be able to access the feedback.

User Story #17

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.

#	Description	Estimated Time	Owner
1	Create Dockerfile to generate docker image	4 hrs	Vaastav
2	Implement network configuration of application in dockerfile	2 hrs	Vaastav
3	Setup image hosting on GCP container registry	2 hrs	Vaastav
4	Host container on appropriate URL route	2 hrs	Vaastav
	Total hours	10 hrs	

Acceptance Criteria:

- Given that the Dockerfile is implemented correctly, it should be useable to create a docker image for the application
- Given that the network configurations are implemented correctly, the application should be accessible at port 8080
- Given that the docker image is hosted correctly, it should be accessible through GCP container registry and usable through cloud run

User Story #18

As a developer, I would like the application to be hosted as three different microservices so that I can decouple the application to make future maintenance easier.

#	Description	Estimated Time	Owner
1	Implement config setup for app engine service	2 hrs	Vaastav
2	Implement config setup for GNN service	2 hrs	Vaastav
3	Implement config setup for XGBoost service	2 hrs	Vaastav
4	Host all three services on cloud run	4 hrs	Vaastav
	Total hours	10 hrs	

Acceptance Criteria:

- Given config file for app engine is implemented correctly, it should be successfully deployed on GCP as a microservice
- Given config file for GNN module is implemented correctly, it should be successfully deployed on GCP as a microservice
- Given config file for XGBoost module is implemented correctly, it should be successfully deployed on GCP as a microservice

User Story #19

As a developer, I would like to setup continuous integration and deployment for all microservices so that they can automatically sync with the code base and redeploy whenever there is an update to the codebase

#	Description	Estimated Time	Owner
1	Implement CI/CD configuration for app engine	2 hrs	Vaastav
2	Implement CI/CD configuration for GNN service	2 hrs	Vaastav
3	Implement CI/CD configuration for XGBoost service	2 hrs	Vaastav
4	Integrate and redeploy all services too cloud run	4 hrs	Vaastav
	Total hours	10 hrs	

Acceptance Criteria:

- Given that CI/CD has been correctly setup for app engine, it will automatically sync with the most recent iteration of origin/main and redeploy to the latest version
- Given that CI/CD has been correctly setup for GNN module, it will automatically sync with the most recent iteration of origin/main and redeploy to the latest version
- Given that CI/CD has been correctly setup for XGBoost module, it will automatically sync with the most recent iteration of origin/main and redeploy to the latest version

Remaining Backlog

Functional

1. **Account Registration, Login, and Management**
    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 enter my meal preferences so that I can have better meal recommendations based on my current likings.~~
    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 that I would like to open the app.~~
2. **Application Navigation**
    1. ~~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.~~
3. **Dining Facilities and Menus**
    1. ~~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.~~
    2. ~~As a user, I would like to be able to filter meals based on if they are vegetarian or contain specific allergens so that I can be sure that what I am eating fits my dietary needs.~~
    3. ~~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.~~
    4. ~~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.~~
    5. ~~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.~~
    6. ~~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.~~
    7. ~~As a user, I would like to be able to remove favorite meals when my preference changes.~~

8. ~~As a user, I would like to customize my favorited meals to receive specific notifications.~~
9. ~~As a user, I would like to be able to rate meals so that I get better meal recommendations based on my preferences.~~
10. ~~As a user, I would like to use a search bar to search through the dining court menus on any given day.~~
11. ~~As a user, I would like to be able to view the specific nutritional information for menu items.~~
12. ~~As a user, I would like to be able to see the correct menu items so that I can view what is offered at dining courts, record the right meal preferences, and record my meals.~~
13. ~~As a user I would like to be recommended meals to eat based on my eating habits.~~

4. Meal Recommendation

1. ~~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.~~
2. ~~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.~~

5. Buddy System

1. 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.
2. 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.
3. 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.

6. User Eating Habits

1. ~~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.~~
2. ~~As a user, I would like to be able to view my past eating history so that I can keep track of my diet.~~

7. Meal Plan Manager

1. ~~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.~~
2. ~~As a user, I would like to be able to view my dining dollars balance so that I can track my transactions.~~
3. ~~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.~~

8. Meal Reviews and Ratings

1. ~~As a user, I would like to be able to write reviews about dining courts so that I can share my experiences with others.~~
2. ~~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.~~
3. ~~As a user, I would like to be able to report reviews that are spam or inappropriate so that they can be taken down.~~

4. ~~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.~~

9. **Accessibility**
    1. 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.
    2. ~~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.~~
    3. ~~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.~~
    4. ~~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.~~
    5. ~~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.~~
    6. ~~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.~~
    7. ~~As a user, I would like to disable the display of a Purdue fun fact so that I have a more minimalistic app experience.~~
10. **Summaries and Trends**
    1. ~~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.~~
    2. ~~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.~~

Non-functional

1. ~~As a developer, I would like to create a dataset with dining menu logs and wait times so that I can use the data in our wait-time prediction modeling dataset.~~
2. ~~As a developer, as part of the XGBoost modeling process, I would like to generate an importance matrix to identify the most important influencing factors affecting dining court wait-times so that I may use these factors for wait-time prediction.~~
3. ~~As a developer, I would like to fine-tune the XGBoost model to accurately predict wait-times per dining court.~~
4. ~~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.~~
5. ~~As a developer, I would like to have custom GNN layer and model assets so that I can assemble a high performance GNN model engine~~
6. ~~As a developer, I would like to construct a data preprocessing pipeline for our GNN model~~
7. ~~As a developer, I would like to construct a data preprocessing pipeline for our GNN model~~
8. ~~As a developer, I would like to be able to view database analytics so that I can track and scale the database.~~
9. 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~~.~~

10. ~~As a developer, I would like to view user data analytics so that I can improve user experience over time.~~
11. ~~As a developer, I would like to view API interface documentation so that front-end and back-end developers can work independently.~~
12. ~~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.~~
13. ~~As a developer, I would like the application to be hosted as three different microservices so that I can decouple the application to make future maintenance easier.~~

-