Team Members:
Tyler Dionne (tdionne2021@my.fit.edu), Kendall Kelly (kelly2021@my.fit.edu), Braden Corkum (corkumb2013@my.fit.edu)

Project Advisor:
Philip Chan, pkc@fit.edu

Project Title:
Tasteful Panthers: Food Recommendation at Dining Halls
Client: Philip Chan

Requirement Document

1. Functional Requirements

1.1 Personalized Recommendations

Description:
The system provides daily personalized meal recommendations based on user reviews and flavor profiles.

Scenario 1:
- Sample Input:
    - User profile has preference for spicy food and history of positive reviews for spicy food.
    - Review data shows that this user, we'll call user A, likes spicy food and a user they've deemed to have similar flavor profile with, user B, likes sweet and spicy food.

- Sample Output
    - Correct: "Today's recommendation: Spicy Chicken Sandwich" (highly rated by users with similar taste profiles).
    - Incorrect: "Today's recommendation: Sweet BBQ Pulled Pork Sandwich" (not aligned with user's flavor profile).

Behavior:
This system matches users that have similar flavor profiles and suggests meals to each individual user based on the meals that similar people liked as well as their own personal flavor profile and reviews.Recommendations will be updated daily based on the most recent reviews.

1.2 Review System

Description: Users can submit, search, and view reviews. These reviews will include star ratings, text, images, videos, and tags.

Scenario 1:

- Sample Input:
    - Review: 4 stars, "Great flavor!", Image of the meal, Tag: "good for studying".

- Sample Output:
    - Correct: Display the review with 4 stars, "Great flavor!", image, and tag on the meal's review page.
    - Incorrect: Display review with missing text and 3 star rating instead of 4.

Scenario 2:

- Sample Input:
    - A user searches for reviews with the "good for playing sports" tag.

- Sample Output:
    - Correct: Displays all reviews with the tag "good for playing sports"
    - Incorrect: Displays either no reviews with the tag "good for playing sports" or does display these reviews, but also includes reviews that don't have this tag.

Behavior:

Users can rate meals from 1 to 5 stars, write text reviews, upload images and videos, and add tags. The users will also be able to search for reviews by tags and keywords.

1.3 GPS-Based Notifications

Description:

The app sends notifications based on the user's location within the dining hall.

Scenario 1:

- Sample Input:
    - The user's location is entering the dining hall. This triggers the notification that the user has entered the dining hall.

- Sample Output:
    - Correct: "Recommendation of the Day: Spicy Chicken Sandwich Don’t forget to leave a review after your meal!"
    - Incorrect: Notifications sent when leaving the dining hall or not at all.

Behavior:
Notifications are triggered when users enter the dining hall to remind them of their recommended meal and to leave a review. Another reminder to leave a review is sent 30 minutes after the user’s arrival.

1.4 Contests and Leaderboard

Description:
The app features daily polls as well as weekly review contests with rewards.

Scenario 1:
- Sample Input:
    - Poll Topic: "Which meal will be rated the highest on tomorrow’s menu?"

- Sample Output:
    - Correct: Display poll results and announce the highest rated meal the next day.
    - Incorrect: Incorrect poll results or never posting the most highly rated meal.

Scenario 2:
- Sample Input:
    - Review Contest Topic: "good for studying'."

- Sample Output:
    - Correct: Display top three people who made the most reviews under this tag, as well as their rewards.

Behavior:
Users participate in daily polls as well as weekly review contests. There are no rewards for the polls. However, the top three people who make the most reviews under the specified tag for that week's contests will receive rewards based on their participation and review quality.

1.5 Meal Suggestions

Description: Users can make meal suggestions for the following week’s menu.

Scenario 1:
- Sample Input:
    - User Suggestion: "Add Chicken Alfredo to next week’s menu."

- Sample Output
    - Correct: "Your suggestion has been submitted. Check back next week to see if it’s been added to the menu!"
    - Incorrect: Suggestio is not recorded/submitted or no message to notify the user their request was submitted.

Behavior:
Users can submit meal suggestions weekly and popular suggestions may be considered to add to next week’s menu.

1.6 Staff Interaction with Reviews

Description: Kitchen staff can view, search, and comment on user reviews.

Scenario 1:
- Sample Input:
    - Student review says, "The pasta was overcooked". Staff see this and comment, "We apologize and will ensure our pasta is no longer overcooked."

- Sample Output:
    - Correct: The staff’s response comment should be seen under the review.
    - Incorrect: Comment is not displayed under the review.

Behavior:
Staff can search for and view reviews as well as comment on reviews to address user feedback.

2. Non-Functional Requirements

2.1 Performance Requirements

- Response Time: The application should be able to load recommendations and reviews within five seconds and notifications within ten seconds. We want the application to function at a fairly fast rate in order to create a better experience for the users.

- Accuracy: Recommendations should be relevant to the user’s flavor profile and previous reviews with a minimum of 80% accuracy. Ensuring accuracy will contribute to a positive experience when using the application as well as encourage current users to keep using the app and new users to join.

- Concurrency: This application will need to be able to handle multiple people using the application at one time. Several users will be submitting reviews, searching reviews, and setting off the notifications all at the same time. Our application will need to be able to handle this kind of traffic.

2.2 Usability Requirements

- User Interface: The app must have a user-friendly interface. It should be clear how to switch between screens as well as how to navigate throughout the entire application. The key functions such as submitting/searching reviews, polls, recommended meals, and leaderboards should be displayed in a way that is very clear to the user.

2.3 Reliability

- Availability: The app should not crash often. The app should be at least 90% functional at any given time. The 10% would include issues with loading pages and other things of this nature.

- Error Handling: The app should be able to handle errors quickly and provide corresponding messages to users. For example, if there is some issue when trying to save a user's response to a poll, an error message will be displayed saying something like “Unfortunately, we are unable to save your response at this time. Our apologies for the inconvenience!” .

2.4 Security

- Authentication: When signing up, users must sign up with their @fit.edu emails, whether they be student or staff. Kitchen staff will have their own admin login which will have different access. For example, users can submit reviews but kitchen staff cannot.

2.5 Compatibility

- Platforms: This application will be compatible with Android mobile phones.

- Devices: The app must be tested on a screen that matches the average screen size of any given Android mobile phone.

3. Interface Requirements

3.1 User Interface

- Home Screen: This is the page that users will see whenever they open the app. This page should display the daily personalized meal recommendations as well as current polls and contests. This will include leaderboard information, results of polls, or pending polls.

- Personal Profile Screen: This screen will allow users to view their personal information such as their name, email, and flavor profile. The personal profile screen will also display past reviews/ratings and any contest information if there is any.

- Review Screen: This screen allows users to submit and view reviews for different meals. This page must be able to upload pictures/videos, tag reviews, and have a search bar so that you are able to search for reviews by tags.

- Notifications: There will be GPS-based alerts and reminders that will be sent to the user's phone. When the application is opened, the notification will be displayed in the app.

3.2 System Interface

- Backend API: This will manage communication between the mobile app and server. It is important for getting profiles, generating recommendations, processing any contests or polls, and submitting reviews. The backend API will also be essential for updating our databases in real time.

- Database: This will store all important information for the application such as user profiles, reviews, meal recommendations, and leaderboard data.
It also must be structured for queries to handle information about the user's flavor profile or what tags have been placed on what reviews. This will be essential for matching profiles, creating recommendations, and searching for specific reviews.

4. Specific Requirements

- When a user clicks the button to submit a review, a confirmation message will be displayed on the screen within two seconds in order to confirm that the review has been submitted successfully.

- When a user submits a review, the review should be stored in the database and should be visible under reviews for that meal within five seconds.

- When searching for reviews, users should also include tags related to dietary restrictions such as, gluten free, vegan, dairy free, etc. They should also be able to use keywords related to the name of the dish, such as chicken or pasta.

- The system should send a notification to the user when he/she is within 100 feet of the panther dining hall.

- The system should send a notification 30 minutes after the user is within 100 feet of the panther dining hall to remind them to review their meal.

- The system should have both a student dashboard and a kitchen staff dashboard. The dashboard will be loaded based on the account type at sign in.

- The system should not allow students to create staff accounts.

- The system should not allow staff to create student accounts.