

# Design Document

## Tasteful Panthers: Food Recommendation at Dining Halls

### Team Members

Alex Laureano, [elaureano2021@my.fit.edu](mailto:elaureano2021@my.fit.edu)  
Jiahao Shu, [jshu2023@my.fit.edu](mailto:jshu2023@my.fit.edu)  
Anthony Hordesky, [ahordesky2022@my.fit.edu](mailto:ahordesky2022@my.fit.edu)  
Lexi Franklin, [afranklin2021@my.fit.edu](mailto:afranklin2021@my.fit.edu)

### Faculty Advisor

Philip Chan, [pkc@cs.fit.edu](mailto:pkc@cs.fit.edu)

## 1. Introduction

### 1.1 Purpose

The purpose of this document is to outline the design and implementation plan for a mobile-based **Food Recommendation System for Panther Dining Hall**. The system shall provide personalized food recommendations, allow users to submit and view reviews, and help dining hall staff address food-related issues efficiently using reviews. The system shall also manage a ranking system for top reviewers, best meal predictors, and top-rated dishes.

### 1.2 Scope

This system aims to:

- Provide up to three daily food recommendations based on user preferences, reviews, and similar likes with other users.
- Allow dining guests to submit reviews with various media formats (text, images, videos, tags).
- Allow users to search and filter food reviews.
- Offer location-based notifications.
- Provide feedback to dining hall staff based on user reviews.

## 2. System Overview

### 2.1 Users

- **Guests:** Students, faculty, or guests who eat at the dining hall.
- **Staff:** Dining hall employees responsible for food preparation and service.

### 2.2 Key Features

- **Personalized Meal Recommendations:** Based on past user reviews, preferences, and similar likes to other users.
- **Review System:** Users can submit text, images, video links, star ratings, and tags.
- **Search & Filter:** Reviews can be searched using tags.
- **Notifications:** Location-based alerts for meal suggestions.
- **Gamification:** Leaderboards and contests for predicting the most popular dishes, suggesting new dishes, and having well liked reviews.
- **Staff Feedback:** Dining hall staff can view reviews with the ability to flip flags on targeted reviews.

### 3. System Architecture

#### 3.1 High-Level Architecture

- **Frontend:** Mobile application (iOS & Android) developed using Flutter.
- **Backend:** Developed using Flutter and XCode with Dart.
- **Database:** Firebase for storing user reviews and recommendations.

#### 3.2 Data Flow

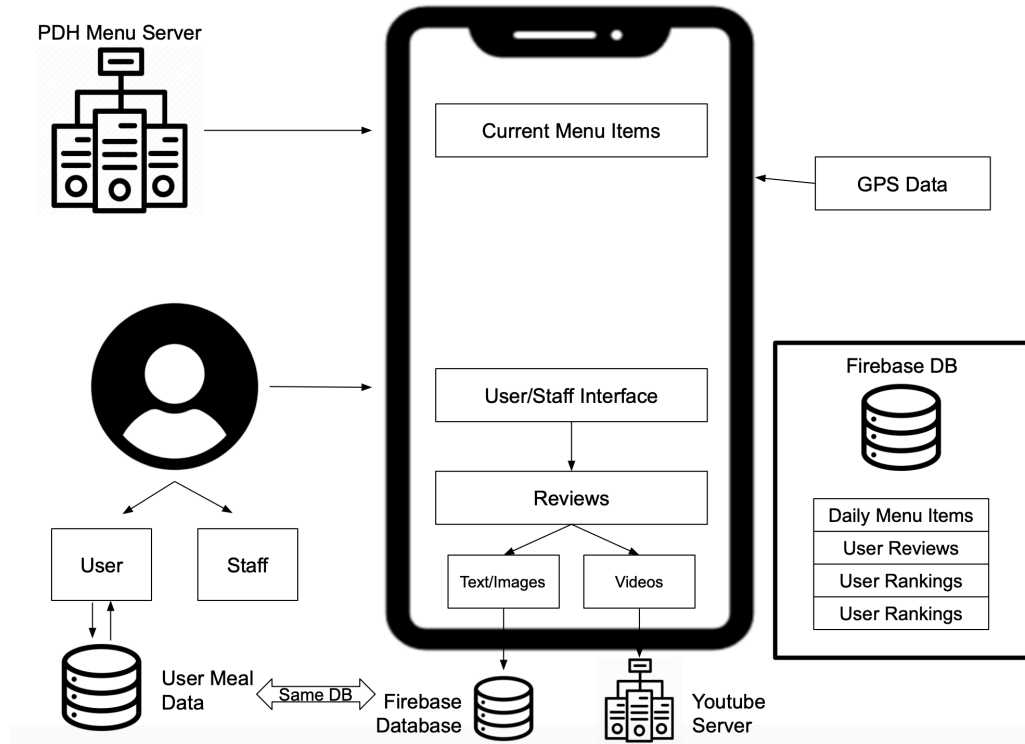
1. Users log in to the app (via PAWS & DUO authentication).
2. They view personalized food recommendations based on their past activity.
3. Users submit reviews, which are processed and stored in the database.
4. The system updates recommendations based on new reviews.
5. Staff members access user feedback and respond as needed.
6. Notifications are triggered based on user location (upon entering dining hall)

### 4. Technology Stack

Component	Technology
Frontend	Flutter
Backend	Flutter, XCode
Database	Firebase
Authentication	PAWS Authentication via DUO

### 5. Diagrams

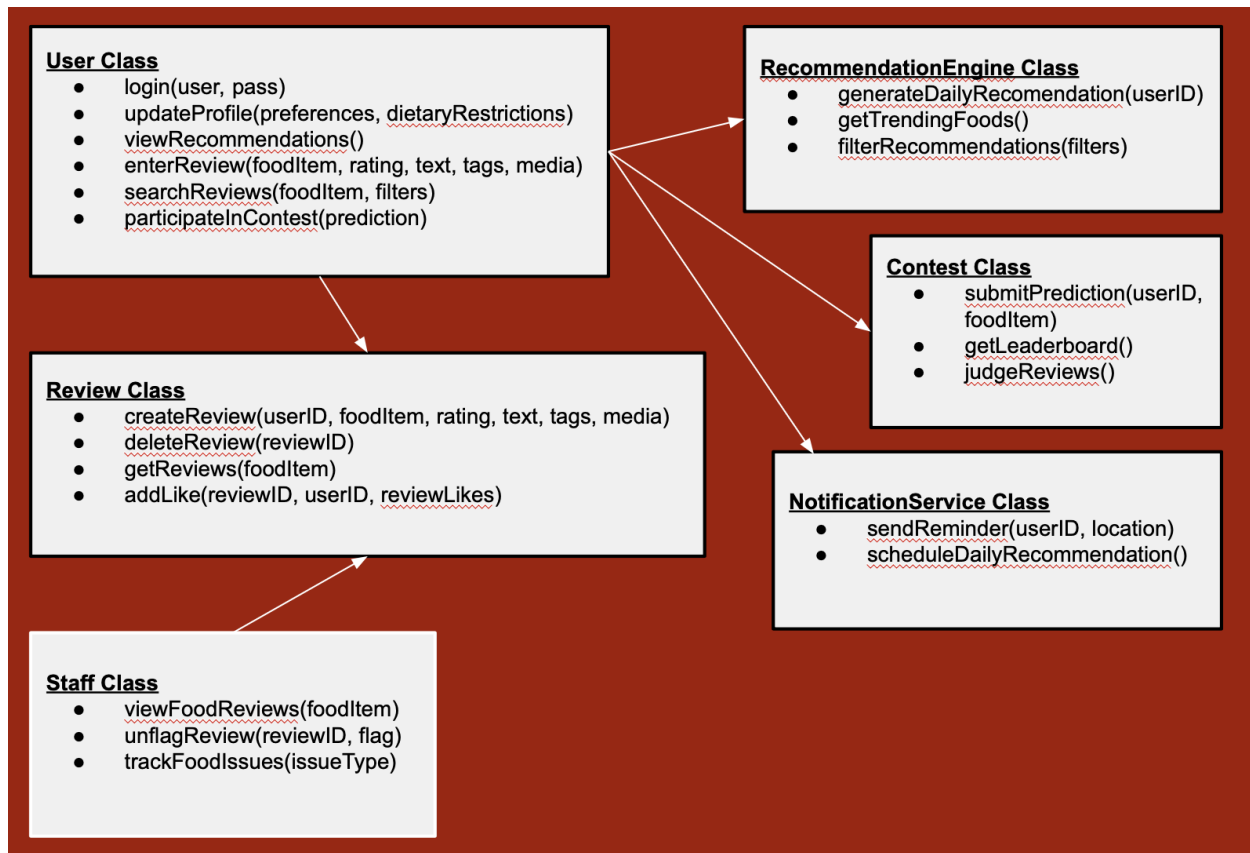
#### 5.1 System Architecture Diagram



## 5.2 GUI Mockup (layout for each screen)

<https://docs.google.com/presentation/d/1BMRA8IfOLtBZLB3beAbVSv7cNA-5ybAzftEDLjATaI4/edit?usp=sharing>

## 5.3 ER Diagram for Database



## 6. Classes and Methods

### 6.1 User Class

Represents diners using the app

- login(username, password): Authenticates the user.
- updateProfile(preferences, dietaryRestrictions): Updates user preferences.
- viewRecommendations(): Retrieves personalized daily food recommendations.
- enterReview(foodItem, rating, text, tags, media): Allows users to submit reviews.
- searchReviews(foodItem, filters): Searches reviews based on filters like tags, rating, or food type.
- participateInContest(prediction): Submits a prediction for contest participation.

### 6.2 Review Class

Handles user-submitted reviews.

- createReview(userID, foodItem, rating, text, tags, media): Creates a new review.
- deleteReview(reviewID): Removes a review.

- `getReviews(foodItem)`: Retrieves all reviews for a specific food item.
- `likeReview(reviewID, userID, reviewLikes)`: Adds a like to a review.

### **6.3 RecommendationEngine Class**

Generates personalized food recommendations.

- `generateDailyRecommendation(userID)`: Provides recommendations based on past reviews and preferences.
- `getTrendingFoods()`: Returns the most popular food items of the day.
- `filterRecommendations(filters)`: Filters recommendations based on user preferences.

### **6.4 Contest Class**

Handles competitions within the app.

- `submitPrediction(userID, foodItem)`: Allows users to predict the highest-rated food.
- `getLeaderboard()`: Retrieves the current leaderboard.
- `judgeReviews()`: Selects the top reviews based on user votes.

### **6.5 Staff Class**

Represents staff who monitor food feedback.

- `viewFoodReviews(foodItem)`: Allows staff to see user feedback.
- `unflagReview(reviewID, flag)`: Lets staff flip the flag on reviews.
- `trackFoodIssues(issueType)`: Monitors reported food-related issues.

### **6.6 NotificationService Class**

Manages user notifications.

- `sendReminder(userID, location)`: Sends a reminder based on GPS.
- `scheduleDailyRecommendation()`: Sends daily food recommendations.

## **7. Future Enhancements**

- **Integration with Nutrition Data**: Display nutritional values of meals.