

Team Name: Tasteful Panthers

# Tasteful Panthers: Food Recommendation at Dining Halls

Software Design Document

Name(s): Kendall Kelly, Tyler Dionne, Braden Corkum

Section: CSE 4101

Workstation: N/A

Date: 09/30/2024

## **Table of Contents**

1. Introduction
2. System Overview
3. System Architecture Diagram
4. Modules (classes) Functionalities & Interface (methods)
5. Sketch of GUI & Screens
6. Database (ER diagram, tables, keys)

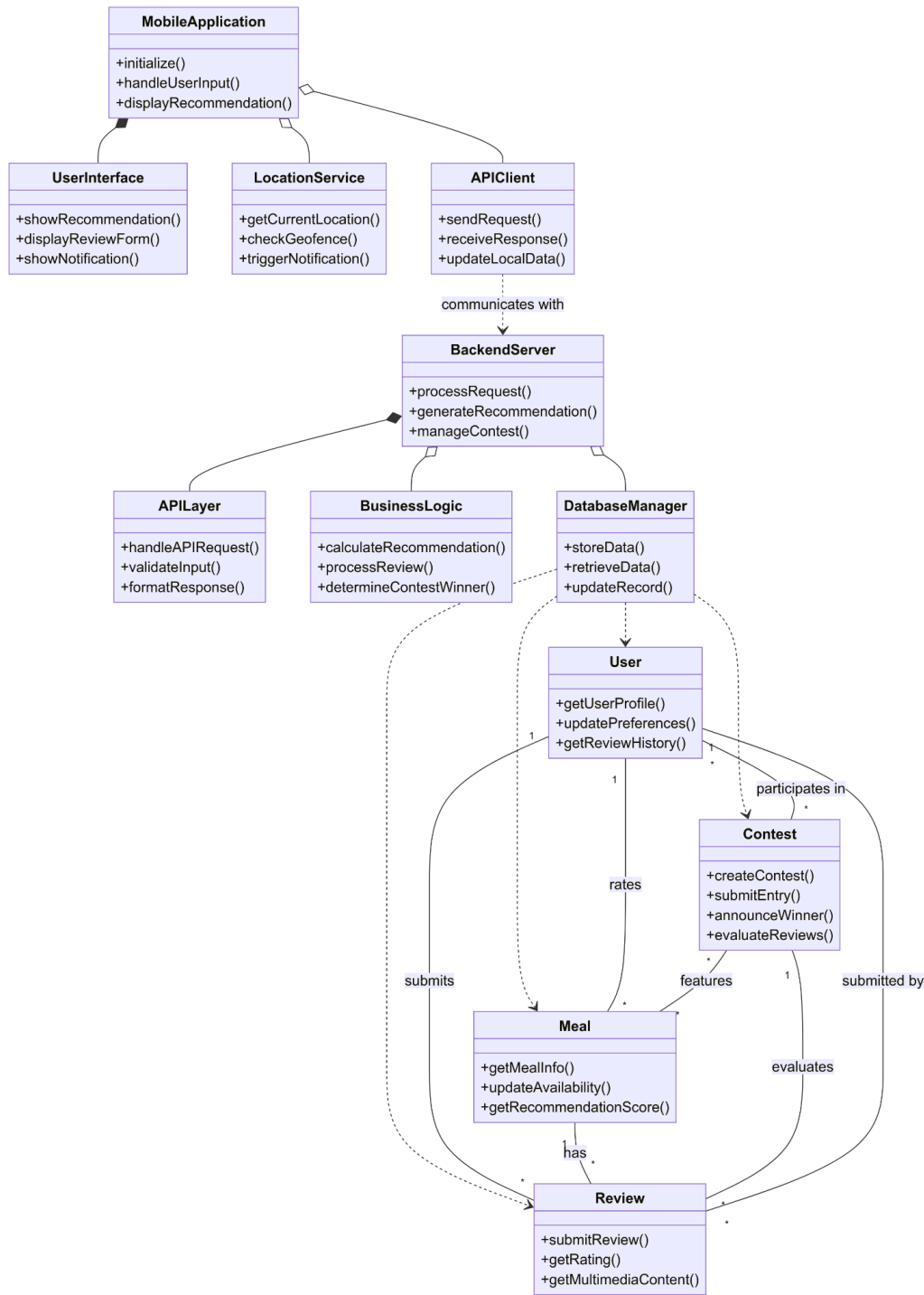
## **Introduction**

This Software Design Document outlines the design approach for the Tasteful Panthers mobile application, which aims to enhance the dining experience at campus dining halls through personalized meal recommendations and user reviews. The document details the system architecture, modules, GUI designs, and database schema required for developing the application.

## **System Overview**

The Tasteful Panthers app provides students with personalized meal recommendations based on their flavor preferences and previous reviews. The app includes features such as review submission, review searching, image reviews, text reviews, video reviews, kitchen staff feedback, student suggested meals, GPS-based notifications, leaderboards, contests, and meal suggestion functionalities. The system consists of a mobile application interacting with a backend server that handles data processing and storage.

# System Architecture Diagram



## Modules (Classes) Functionalities & Interface (Methods)

## I. Mobile Application

### Methods:

+initialize() - starts the app

+handleUserInput() - handles user input

+displayRecommendation() - shows the recommendation of the day

### Relationships:

“Has” a UserInterface (strong connection)

“Uses” LocationService and APIClient (weak connection)

## II. UserInterface

### Methods:

+showRecommendation(): Displays meal suggestions

+displayReviewForm(): Shows form for writing reviews

+showNotification(): Displays app notifications

### Relationships:

“Part of” MobileApplication

## III. LocationService

### Methods:

+getCurrentLocation(): Finds where the user is

+ checkGeofence(): Checks if user is in given area

+triggerNotification(): Sends location based alert

### Relationships:

“Used by” MobileApplication

## IV. APIClient

### Methods:

+sendRequest(): Asks server for information

+recieveResponse(): Gets answer from server

+updateLocalData(): Update the apps info

### Relationships:

“Used by” MobileApplication

“Talks to” BackendServer

## V. BackendServer

### Methods:

+processRequest(): Handles the apps requests

+generateRecommendation(): Makes meal suggestion using algorithm

+manageContest(): Runs the contest

### Relationships:

“Has a” APILayer (strong connection)

“Uses” BusinessLogic and DatabaseManager (weak connection)

## VI. APILayer

Methods:

+handleRequest(): Handles incoming requests

+validateInput(): Validates input data

+formatResponse(): Formats the data being sent back

Relationships:

“Part of” BackendServer

## VII. Logic

Methods:

+calculateRecommendation(): Decides what meals to suggest

+processReview(): Handle new reviews

+determineContestWinner(): Deals with determining contest winner

Relationships:

“Used by” BackendServer

## VIII. DatabaseManager

Methods:

+storeData(): Saves info

+retrieveData(): Retrieves info

+updateRecord(): Updates saved info

Relationships:

“Used by” BackendServer

“Manages” User, Review, Meal and Contest data

## IX. User

Methods:

+getUserProfile(): Get the users info

+updatePreferences(): Changes the users settings

+getReviewHistory(): Gets users past reviews

Relationships:

“Can” submit “many” Reviews

“Can” rate “many” Meals

“Can” join many “Contests”

## X. Review

Methods

- +submitReview(): Create new review
- +getRating(): Get the reviews rating
- +getMultimediaContent(): Gets photos and videos in the review

Relationships:

- “Belongs” to “one” User
- “Is” about “one” Meal
- “Can” be a part of Contests

## XI. Meal

Methods:

- +getMealInfo(): Gets meal info
- +updateAvailability(): Changes if meal is available or not
- +getRecommendationScore(): Calculates how likely to suggest

Relationships:

- “Can” have many” Reviews
- “Can” be in “many” Contests

## XII. Contest

Methods:

- +createContest(): Starts a new contest
- +submitEntry(): Lets users join contest
- +announceWinner(): Declares who won
- +evaluateReviews(): Checks reviews to find best ones

Relationships:

- “Involves” many Users
- “Features” many Meals
- “Evaluates” many Reviews



## Sketch of GUI & Screens

### GUI Mockup

#### 1. Student Home Screen

Tasteful Panthers

Today's Recommendation:

Burger and Fries

Leave a Review?

Review

Reviews

Contest

Settings

#### 2. Review Submission Screen

Write a review

Meal: Burger and Fries

Rate: \*\*\*

Review Text:

Good!

Add Video/Photo

Submit

#### 3. Reviews Screen

Reviews

Filter v Sort v

John M. **	32 ♥
Burger	
Not good	
Saran H. ****	27 ♥
Pizza	
Fabulous!	

Load more

#### 4. Contest Screen

Contest

Top Reviews:

John M.	32 ♥
Burger	
Not good	
Tim J.	22 ♥
Pizza	
Good	

Contest Ends in 06:17:32

#### 5. Kitchen Staff Home Screen

Tasteful Panthers

Student Reviews

Top rated meals

Lowest rated meals

Analytics

#### 6. Kitchen Staff Lowest Rated Meals Screen

Lowest Rated meals

- Pizza Avg: \*
- Burger Avg: \*\*
- chicken Avg: \*\*\*

Student Reviews:

John S *	
Pizza	
Horrible	
Heather J *	
chicken	
Dry, chewy	



## Gvi Mockup 2

### meal suggestion leaderboard

TOP suggested meals	
1. Burger	90 votes
2. Pizza	80 votes
<input type="text" value="Suggest a meal?"/>	

### Review Judging Screen

Thank you for leaving a review!
Would you review the reviews shown below?
Good Pizza ★★★
Good food ★
Great food ★★★★★

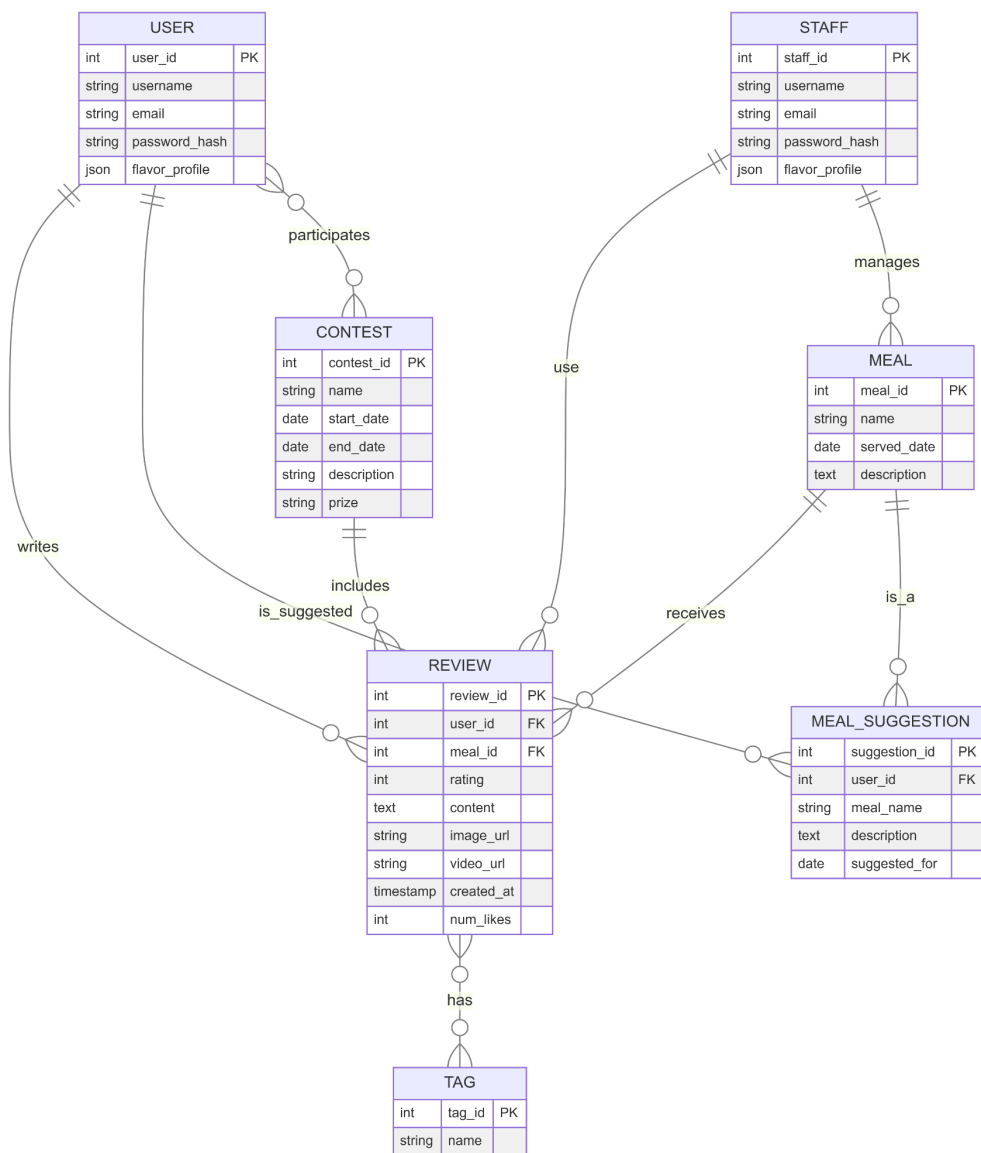
### Review contest leaderboard

Review Contest	
1. Good Pizza	Judges Rated ★★★★★
2. Bad Pizza	Judges Rated ★★★

### Forecasting leaderboard

Forecasting leaderboard	
1. Emily H.	82% accuracy
2. Ron J.	74% accuracy
<input type="text" value="Predict tomorrow's favorite?"/>	

## Database (ER diagram, tables, keys)



### Additions:

Under meal suggestion we also need a slot for a link to the recipe (optionally provided by the user) which would be of type string. We need an int for the number of votes from other students on the meal suggestion and we also need a string for the reasoning.