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

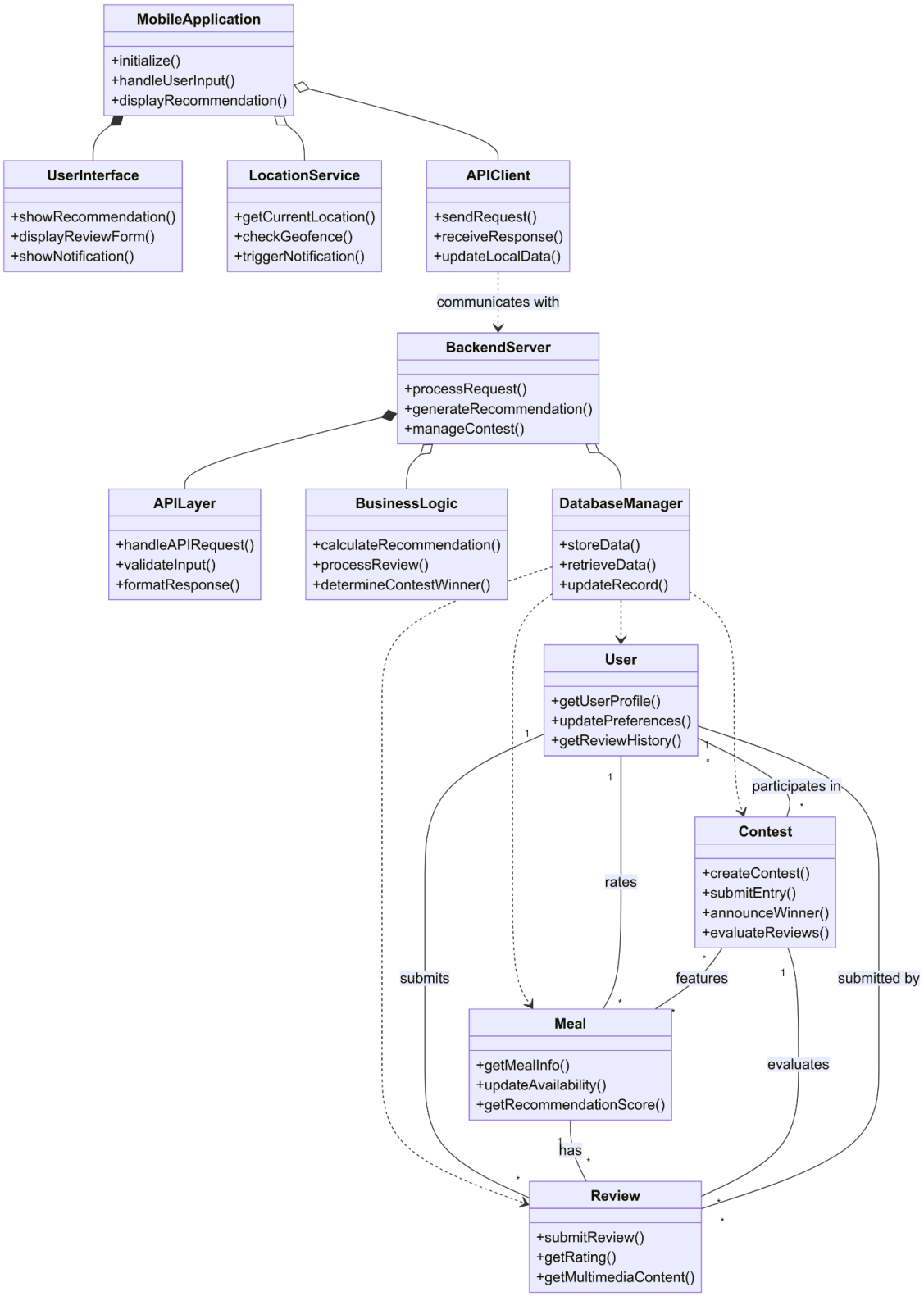
Introduction

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, GPS-based notifications, 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:
+handleAPIRequest(): 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
+retreiveData(): 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
+updatePreferencees(): 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

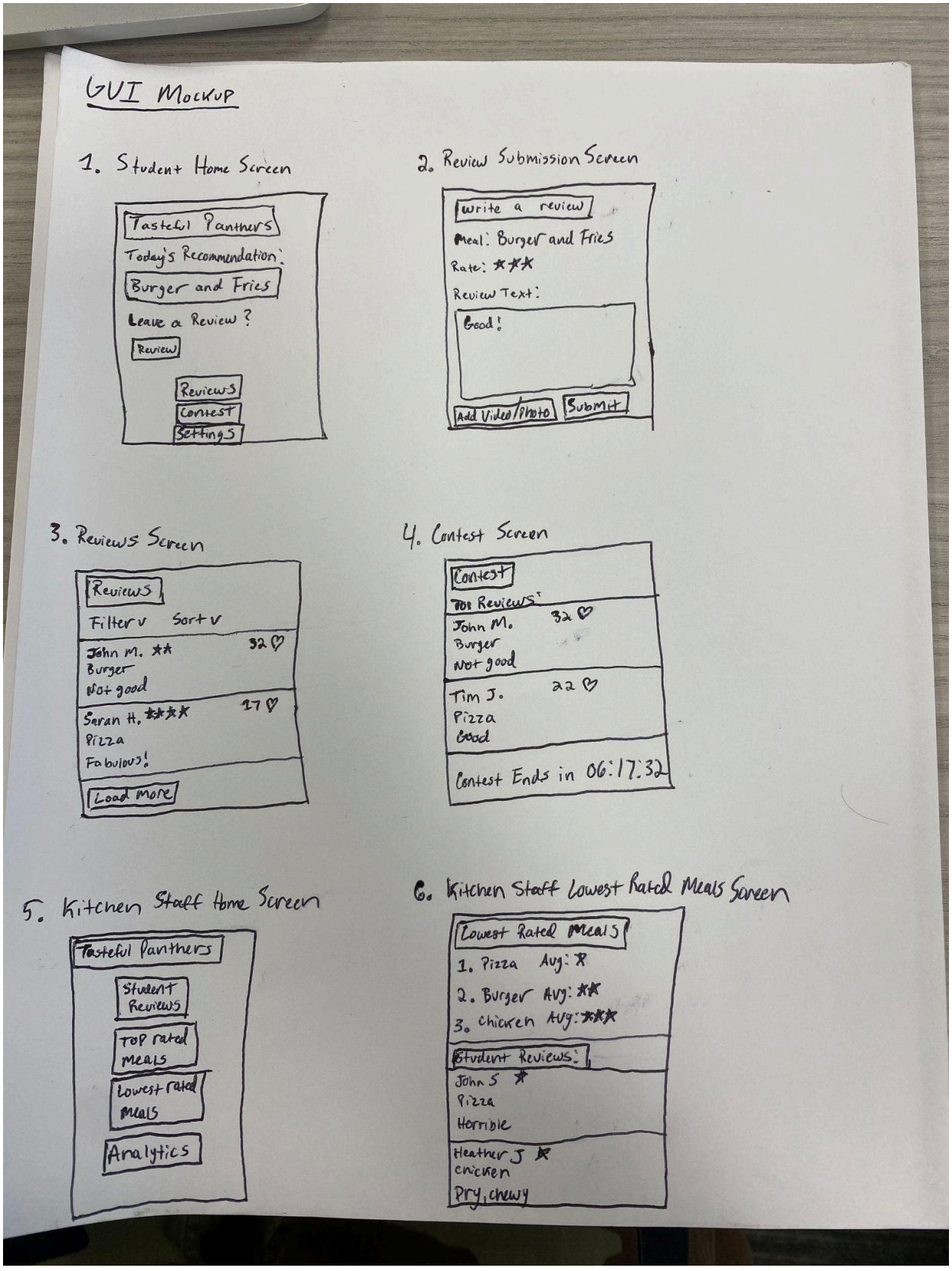
Relationships:

“Involves” many Users

“Features” many Meals

“Evaluates” many Reviews

Sketch of GUI & Screens



Database (ER diagram, tables, keys)