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

Test Document

1. Introduction

1.1. Document identifier

Test Plan for Tasteful Panthers: Food Recommendation at Dining Halls

1.2. Scope

The scope of this test plan is to outline the testing process for the Tasteful Panthers mobile application. This will include all required features for the Tasteful Panthers application. We need to understand the testing process and what is to be expected in order to properly meet all of the requirements for our app.

1.3. References

[e.g. page 53 of IEEE Standard for Software Test Documentation]

1.4. Context

Tasteful Panthers is a mobile application designed to provide personalized meal recommendations, allow users to leave, search, and view reviews, as well as participate in contests and offer meal suggestions. Testing will cover the six key features of the app, including the personalized recommendation system, enter/view/search reviews, GPS notification system, leaderboard contests, student meal suggestions, and kitchen staff searching/viewing/commenting on reviews as well as a few other features.

1.5. Notation for description

a. Test case identifier
b. Objective
c. Inputs
d. Outcome(s)
e. Environmental needs
f. Special procedural requirements
g. Intercase dependencies

2. Details

2.1 Application GUI

a. Test case identifier: GUI Test Case

b. Objective: Ensure the UI is easy to use and all main features are easily accessible.

c. Inputs:

1. Switch between different screens
2. Load data such as reviews or recommendations

d. Outcome(s):

1. Correct page is loaded and displayed.
2. Incorrect or wrong page displayed.
3. Proper loading of reviews for an item or personalized recommendation.
4. Wrong data loaded or no data/no response from app.

e. Environmental Needs: Needs to work in an Android mobile environment.

f. Special procedural requirements: N/A

g. Intercase dependencies: We must have a functional login system in order for users to access all of the available pages

2.2 Login System

a. Test case identifier: Login Test Case

b. Objective: Users can log into the app.

c. Inputs:

1. Diner login something like “dineremail@my.fit.edu”
2. Kitchen staff login something like “kitchenstaff@fit.edu”

d. Outcome(s):

1. Diners can access all required pages, can enter/search/view reviews, compete in contests, leave suggestions, view their own profile, and view personal recommendations.
2. Diners can access kitchen staff information/pages and/or cannot access student pages.
3. Kitchen staff can view and comment on reviews as well as see meal suggestions.
4. Kitchen staff cannot comment on reviews/see suggested meals or they are able to access student pages and enter reviews.

e. Environmental needs: We will need a database to store all of this information.

f. Special procedural requirements: Account validation to ensure user/staff have access to correct pages/functionalities.

g. Intercase dependencies: we will need a backend API for login validation.

2.3 Personalized Recommendations

a. Test case identifier: Recommendation Test Cases

b. Objective: Ensure personalized recommendations are determined by users flavor profiles as well as other profiles they’ve been determined to have similar flavor profiles to.

c. Inputs:

1. User 1 likes Spicy food

2. User 2 likes sweet and spicy food
3. Both likes meals A and B, but User 1 has not tried C

d. Outcome(s):
1. User 1 is recommended meal C
2. User 1 is recommended a meal they've already reviewed, no new recommendation, or a meal that only has a sweet flavor and no spicy flavor.

e. Environmental needs: Backend data focusing on user profiles and their review history.

f. Special procedural requirements: Test a variety of profiles with a variety of flavor profiles.

g. Intercase dependencies: This will require a functional review system as well as databases to store user info and the reviews.

2.4 Review System

a. Test case identifier: Review System Test cases

b. Objective: Ensure users can submit and search for reviews which should include tags, images/videos, and star ratings, as well as view these reviews.

c. Inputs:
1. Search for reviews tagged "good for studying"
2. Enter a review

d. Outcome(s):
1. Search results with all reviews tagged with "good for studying" and the user is able to view each review. Each review should also include all of its key attributes.
2. No reviews, or reviews that don't have the correct tag are shown. Key attributes could also be missing from reviews.
3. Review is processed once entered and is able to be viewed.
4. Review is not found in the system, is missing key attributes, or cannot be viewed.

e. Environmental needs: Reviews need to be stored and to be able to be retrieved.

f. Special procedural requirements: Test video links can be stored and accessed.

g. Intercase dependencies: Need databases to access reviews.

2.5 GPS Notifications

a. Test case identifier: GPS Test cases

b. Objective: Ensure notifications are triggered at the correct location/times, such as when a diner enters a dining hall and 30 minutes after arriving.

c. Inputs:
1. User enters the dining hall.
2. 30 minutes after the user enters the dining hall.

d. Outcome(s):
1. Notification to remind users of recommended meals when entering the dining hall.
2. No notification or wrong notification when entering the dining hall.
3. User gets reminder to leave review 30min after entering dining hall.

4. User never gets a reminder to leave a notification.

e. Environmental needs: GPS tracking needs to be enabled on the device of the user.

f. Special procedural requirements: Test for how accurate the GPS's accuracy is.

g. Intercase dependencies: We will need to have built the actual notification system for this to work.

2.6 Contests/Leaderboards

a. Test case identifier: Contest Test Cases

b. Objective: Ensure contests and leaderboards update in real time based off of the users participation.

c. Inputs:

1. Poll Query = "Which meal will be the most popular on the menu this week?"
2. Review contest with tag "good for studying"

d. Outcome(s):

1. Result of the poll should be displayed to all users. Ex. "Chicken Sandwich"
2. No result displayed for the poll, or wrong meal displayed.
3. Top 3 users with the most reviews tagged with "good for studying" are shown on leaderboard in live time.
4. The leaderboard is not updated in real time or displays the wrong winners at the end of the contest.

e. Environmental needs: Database needed to track users activity and reviews.

f. Special procedural requirements: Leaderboard must update in real time.

g. Intercase dependencies: Review system must work as well as a system for the polls.

2.7 User Suggestions for Meals

a. Test case identifier: User Suggestion Test Cases

b. Objective: Ensure users are able to submit meal suggestions for next week's menu and receive confirmation the suggestion has been received.

c. Inputs:

1. User suggests "Lasagna" for next week's menu.

d. Outcome(s):

1. User receives a confirmation message that the suggestion has been received and it is put into the suggestions for kitchen staff to see.
2. User receives no confirmation and suggestion is never processed.

e. Environmental needs: Database is needed to store meal suggestions.

f. Special procedural requirements: Test multiple different meals to make sure they're being processed properly.

g. Intercase dependencies: N/A

2.8 Staff Interactions with Reviews

a. Test case identifier: Staff Review Test Cases

b. Objective: Ensure kitchen staff can search/view/comment on user reviews.

c. Inputs:

1. Staff comment on poor review about a meal

2. Staff searches for reviews tagged “good for health”

d. Outcome(s):

1. Staff review can clearly be seen underneath the students' review.
2. Staff review cannot be seen or is under the wrong review.
3. Staff can see all reviews with the desired tag.
4. Staff cannot see any reviews, only some, or reviews with the wrong tags.

e. Environmental needs: There needs to be an admin page for staff.

f. Special procedural requirements: Ensure staff are not able to submit reviews.

g. Intercase dependencies: Requires the review system to be functional and staff logins.