Flavor Town Design Document

Ian Campbell, Tyler Gabriel, Austyn Trull, and Joe Xu

Fall 2016

Flavor Town Change History

Version Summary Author Date

0.1 Initial Wireframe Joe Xu 9/18/2016

0.2 Reviewed and commented wireframe Ian Campbell 9/19/2016

0.3 Filled in descriptions Joe Xu 9/21/2016

0.4 Added Main Activity Diagram to the document Tyler Gabriel 9/21/2016

0.5 Added Use Case Diagrams Austyn Trull 9/21/2016

0.6 Added Class Diagrams Ian Campbell 9/21/2016

0.7 Added Activity for “User Adds Comment” Tyler Gabriel 9/26/2016

0.8 Added Detailed Class Diagram Austyn Trull 10/17/2016

0.9 Filled in description of Activity Diagrams Joe Xu 10/18/2016

1.0 Added Sequence Diagrams and Descriptions All Team Members 10/19/2016

1.1 Formatted Document to have uniform properties throughout Tyler Gabriel 10/19/2016

1.2 Reassessed Functional Requirements (removed guess preferences/recommendations) All Team Members 10/19/2016

Table of Contents

Requirements Document

Requirements Documents

1. Introduction

1.1 Purpose

The purpose of this document is to outline the requirements specifications for the Flavor Town food app. This documents will give an overall descriptions of the project. The documents will further list the functional and nonfunctional requirements. Attached will also be UML documentation for specific use cases, user activity or interaction flows, and an overarching class diagram.

1.2 Goal

This application will attempt to create a platform for people to try, rate, and share new foods in their local area.

1.3 Scope

The project will be developed on android. Users will hopefully find this app to be a innovative and engaging way to find the best food item at a restaurant.or in an area. This app should be the first point of reference when someone wants to go eat something new.

1.4 Definitions

Flavor Town - The name of of android application being developed

Rating - A scale from 1-5 that allows users to rate their food

2. Overall Description

2.1 Product Perspective

Finding new food has always been an ordeal, especially when at a new restaurant facing a new menu loaded with options. Hard earned money is at stake and ordering the wrong item can ruin one’s day. Today there exists applications that will lists what restaurants are good, but none go as far as to tell items on that restaurant's menu that makes that restaurant great. This need gives FlavorTown the opportunity to truly give the user a good idea of what food to order.

2.2 Product Features

This will be a crowd sourced application with user rating as the backbone of the app. The user will be able to open up the app and locate restaurants in the area, either through the interactive map or through a list of nearby restaurants. Once having reached the locale, and having tried an item, the user is then able to rate the food. This rating will be aggregated and computed on the

back end. Theses calculation will allow future eaters to be better informed about the food that they will choose to eat.

2.3 Android Application - User Interaction

The main method users will use this application is through their Android phone. The application will use standard touchscreen interfaces that should be intuitive enough for casual users to quickly understand. The application will try to pull resources from the user's phone such as GPS.

2.4 Database

To store the ratings for our application we will use a database. The database will associate users with their ratings. These ratings will be aggregated to find the best food for particular restaurants. The database will hold all the data on the restaurants, the menus, and the users.

2.5 User Accounts

Each user will create an account upon first use of the app. This will help keep track of the user’s voting, preferences, and other things associated directly with the user.

2.6 Assumptions and Dependencies

The minimum SDK version the application will be developed on “Ice Cream Sandwich”. The user would need the minimum version supported by this SDK to run the application. The application will require data connection in order to update maps and receive the aggregated ratings from the database.

3. Functional Requirements

3.1 Main Feature

Feature	Description	Priority
Search Restaurants	The restaurant is able to be found through the map or through a search, based on what restaurant is listed in the backend.	High
Obtain Menu Display	The items within the restaurants will be ranked. These items will allow the user to see what food item is worth considering.	High
Display Best Item	The best items within the restaurants will be ranked based the the ratings that the user gives.	High

3.2 User Rating

Feature	Description	Priority
Rate Food	Ratings of the menu items are gathered from the users. These ratings are to be stored in a database and will return a calculated list of ranked food items.	High

3.3 Map

Feature	Description	Priority
Has Location Tracking	The map will have the option of tracking the user as they move about, calculating the most relevant restaurants in the area.	Medium
Calculate Best Item in Area	The app will be able to return the best food item within a given radius of the user.	Medium

3.4 Database

Feature	Description	Priority
Database Updates	The database stores user accounts, restaurants, restaurant menu items, user ratings, and rating comments.	High
Calculate Ratings	The restaurants and menu items are associated in the database along with the rating for the item. The database updates according to new information	High
Show Restaurant Hours	The app is able return food recommendations based on what restaurants are open	Low

3.5 User Preferences

Feature	Description	Priority
Food Filter Preferences	The user is allowed to save their food preferences. Future searches will be filtered on these preferences	Low

User history	The user is able to find previously rated food through a history feature.	Low

4. Nonfunctional Requirements

4.1 Usability

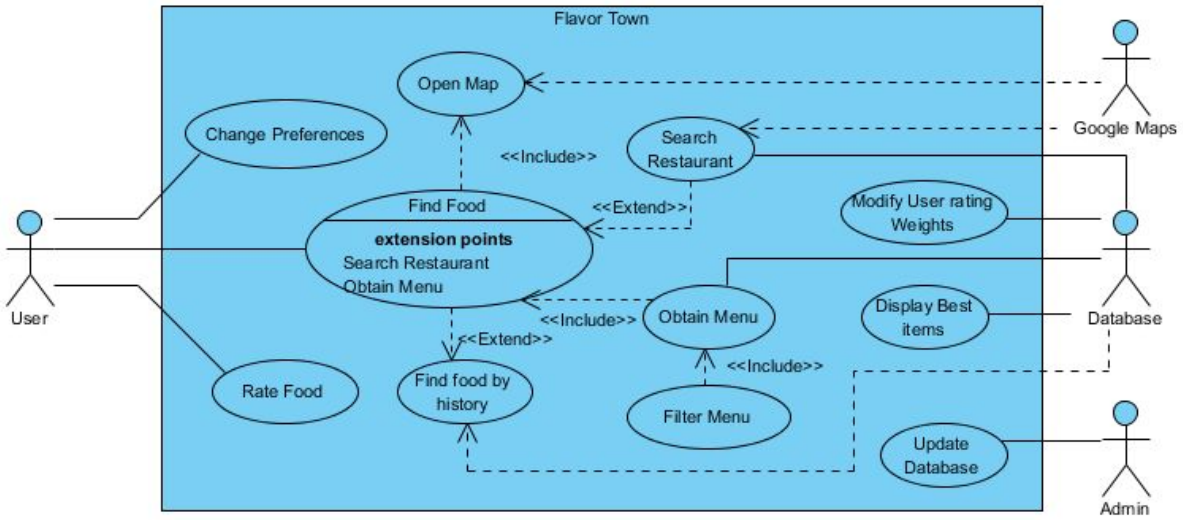
- FlavorTown should be intuitive to the user - user should be able to see recommendations and rate food with relative ease.
- The application should be non-intrusive, allowing users to browse and rate foods at their own pace.
- The application should have fast response times where the calculations done on the backend are unnoticed to the user.

4.2 Scalability

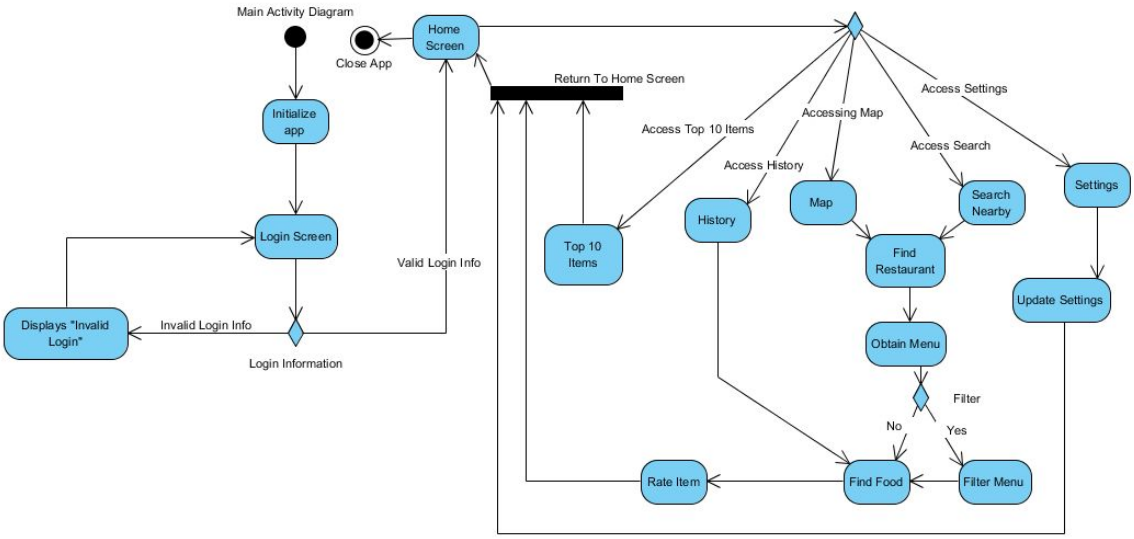
- Popular enough for users to populate database, improving the reliability and trustworthiness of the application

5. UML Diagrams

5.1 Use Case Diagram



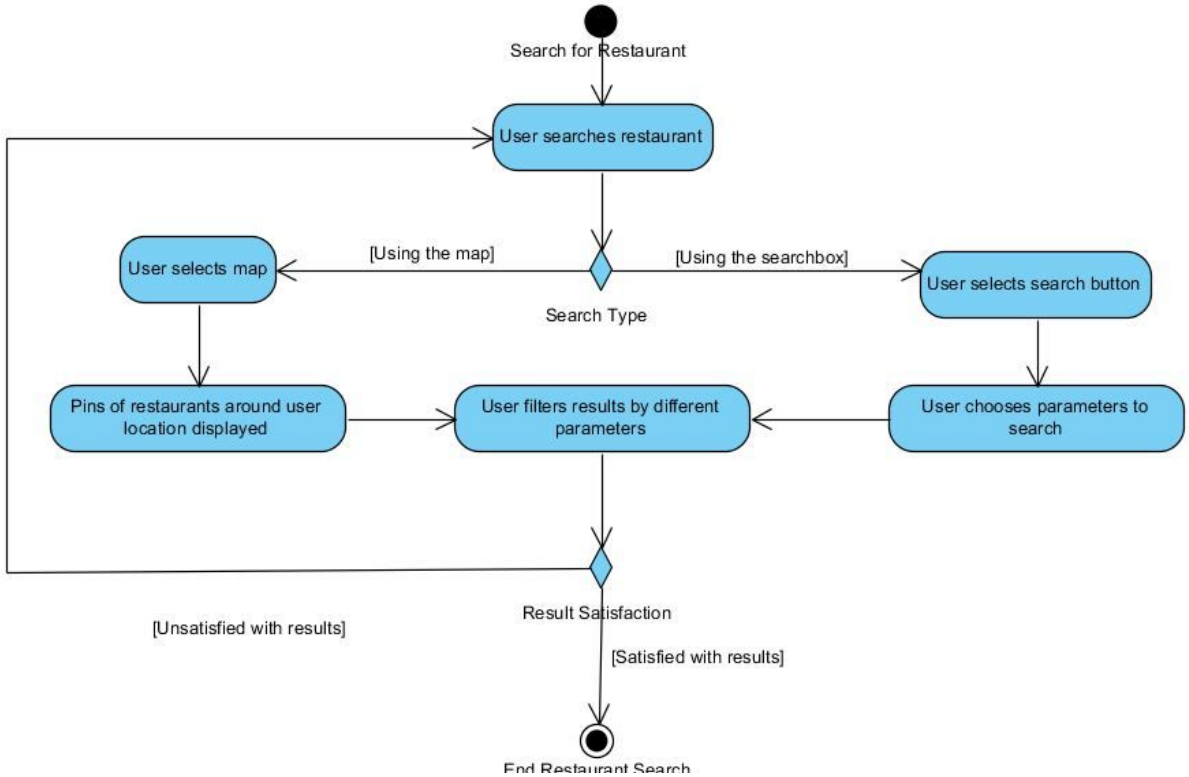
5.2 Main Activity Diagrams



Activity Diagram - Main Activity Diagram

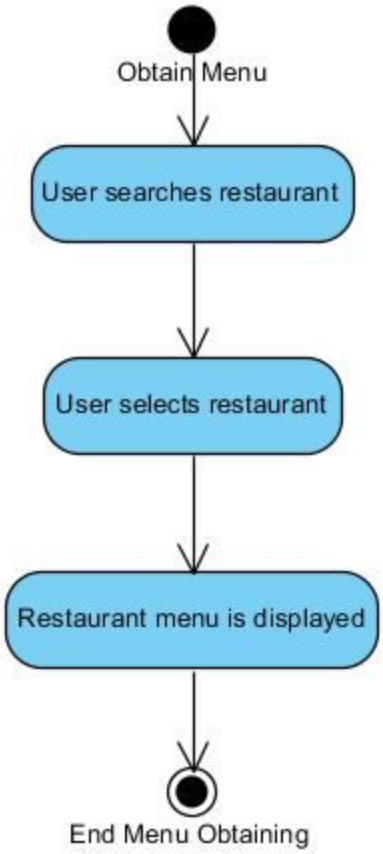
Details the basic functions of the application.

5.3 Individual Activity Diagrams



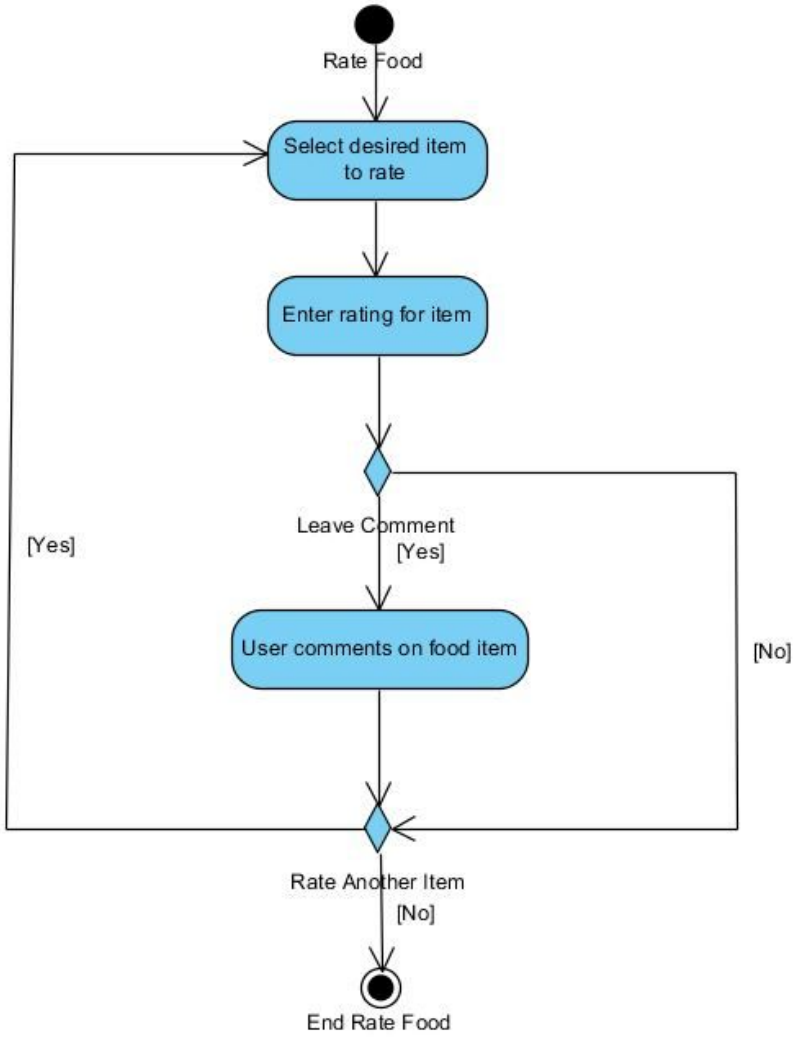
Use Case Scenario - Find Restaurant

The user should be able to find a restaurant through our app. The user would first search for a restaurant. They can either find the restaurant through using the map using pins or through a search box. Filters will be available for the user to pick the location of their choice.



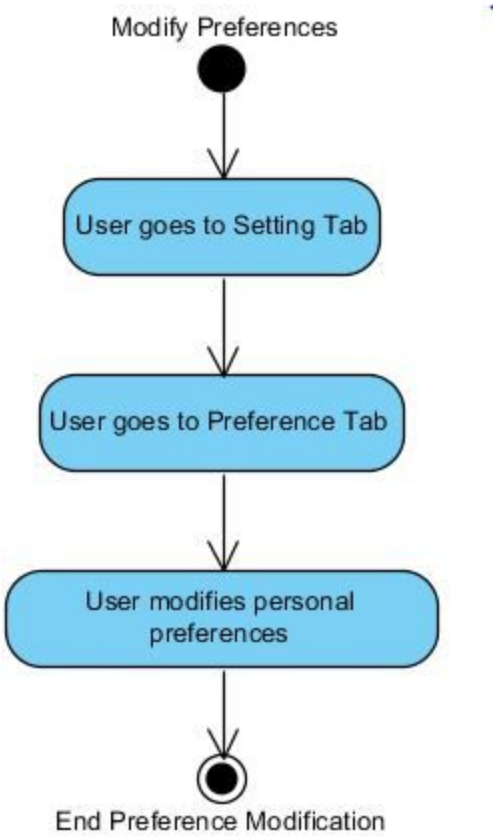
Use Case Scenario - Obtain Menu

The user is able to obtain the restaurant menu from the application. The user would first find a restaurant. From that restaurant, the user is able to view the menu of that particular restaurant.



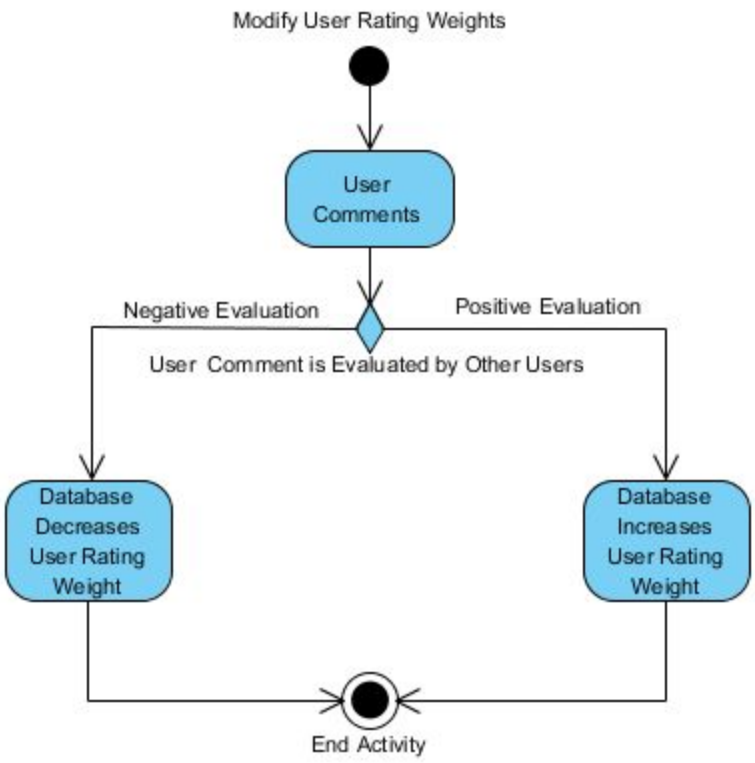
Use Case Scenario - Rate Food

Users are able to rate the food from a particular restaurant. From the menu of a restaurant, the user is able to select the food that they want to rate. The user would rate the item based on a fixed scale. The user would then have the option of leaving a comment to justify their rating. After choosing the leave a comment for a food or not, the user would be given the option to rate another food.



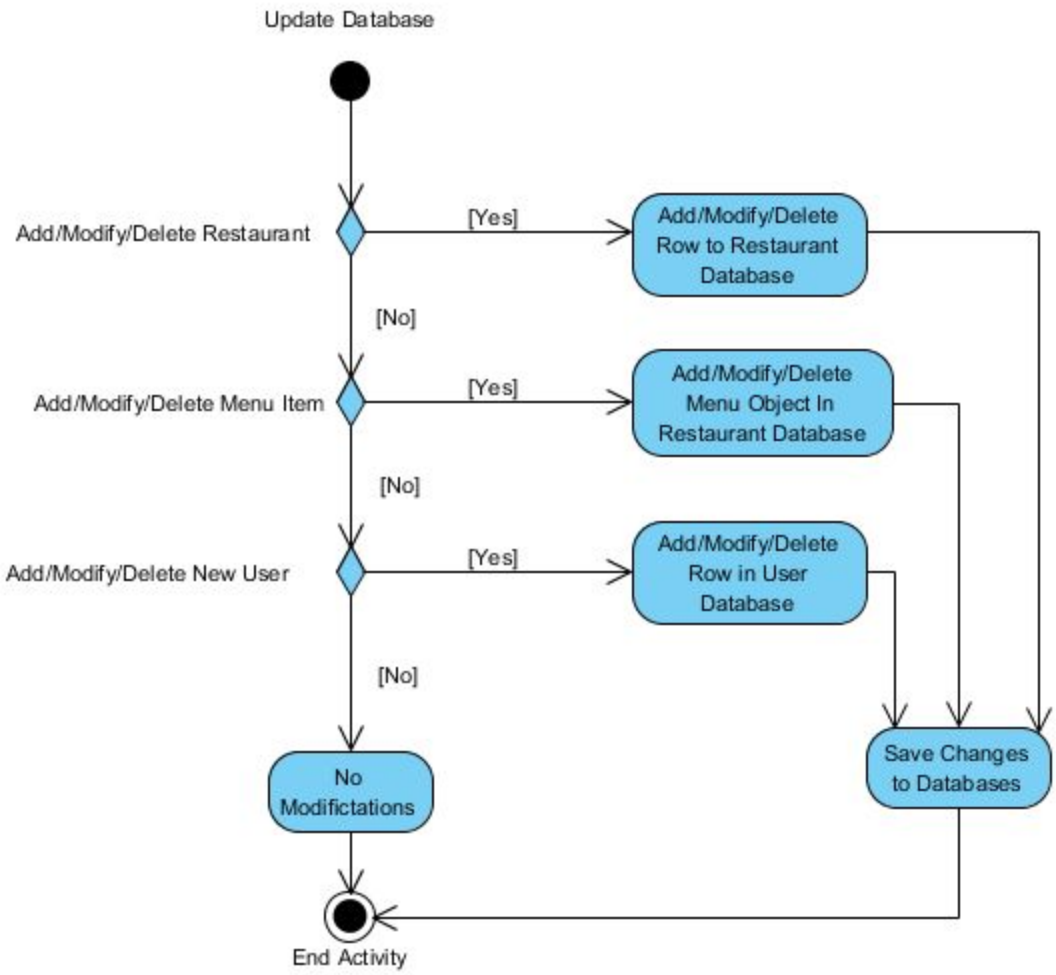
Use Case Scenario - Modify Preferences

The user is able to modify their food preferences based on their dietary needs or personal tastes. The user would go to the settings tab. In the settings the user would be able to access a preferences tab. From the preferences tab, the user would be able to modify their own personal preferences.



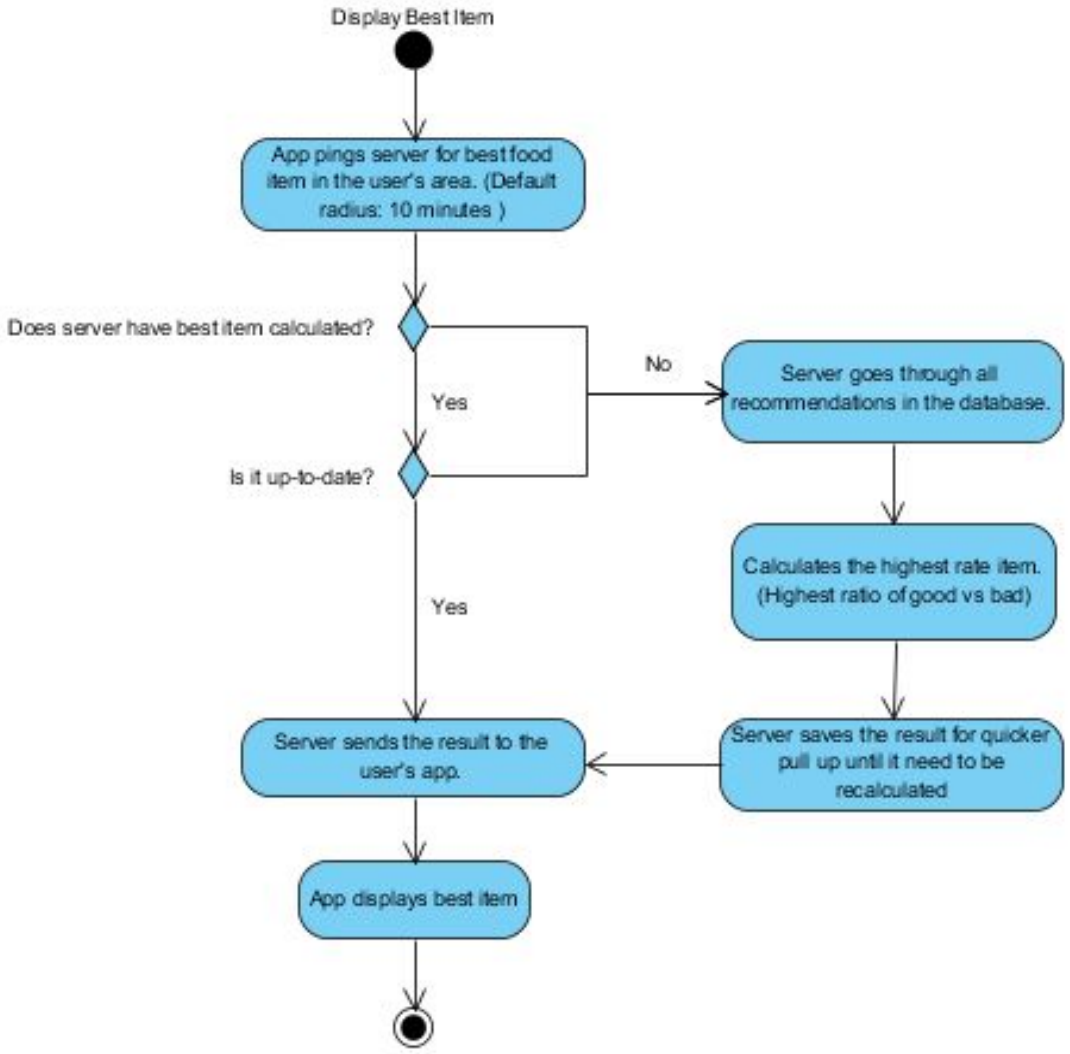
Use Case Scenario - Modify User Weights

The system is able to weight the ratings of users based on user feedback. The system calculates how weighted the user is based on how their comments are rated. The system would increase or decrease the ratings for the food based on the evaluation.



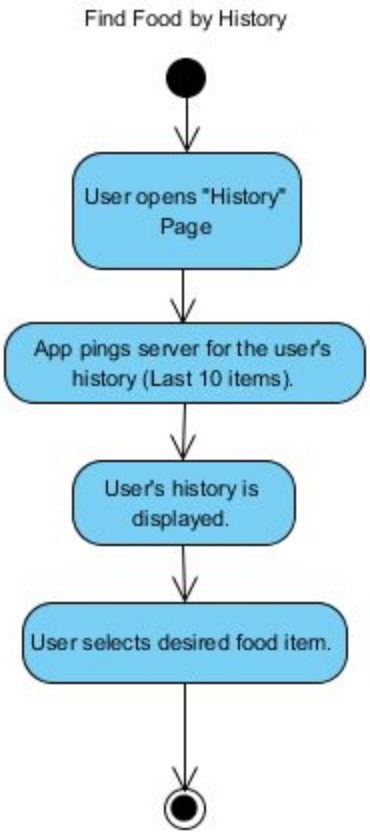
Use Case Scenario - Update Database

The database is able to add/modify/delete restaurants, menu items, or users. During an update of the database the system would figure out what fields need to be edited. The corresponding data would be changed in the database. The database would then be saved.



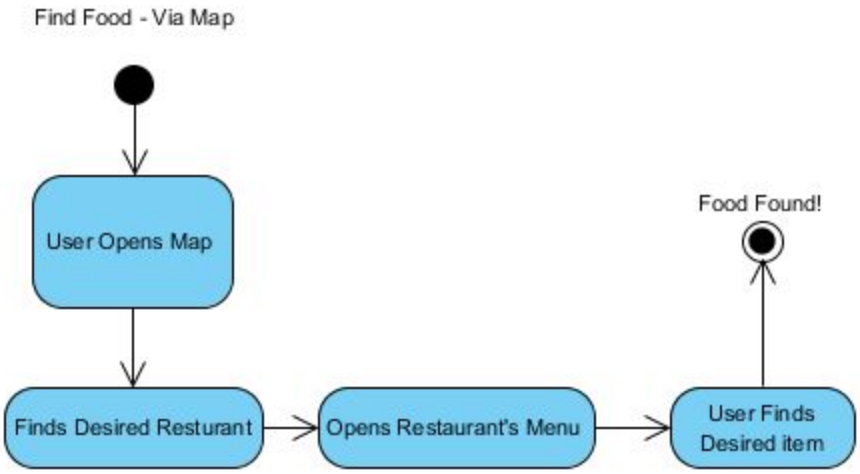
Use Case Scenario - Display Best Item

The system is able to display the best item of a particular restaurant to the user. The application, on command, would figure out the best item within the radius of an area or of a particular restaurant. The server would figure out how to get the best item calculated and up to date. The server would would do the calculation and save the results for faster access. The application would then receive the information from the server. The use will be able to see the best item displayed by the application.



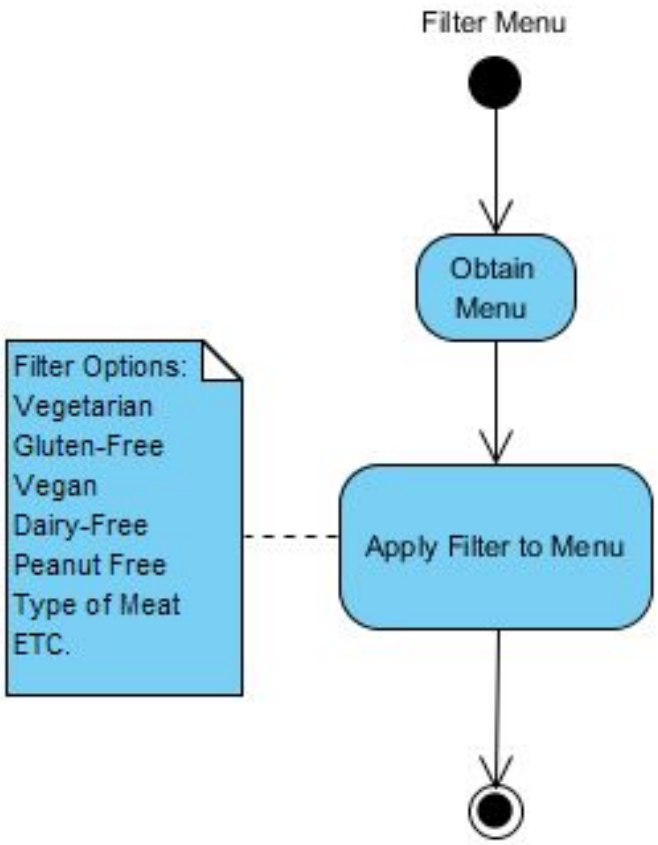
Use Case Scenario - Find Food by History

The user is able to find a food they like from their history. The user opens a history of their food ratings. The user is capable of viewing the list of items. The user would then select the food that they desire.



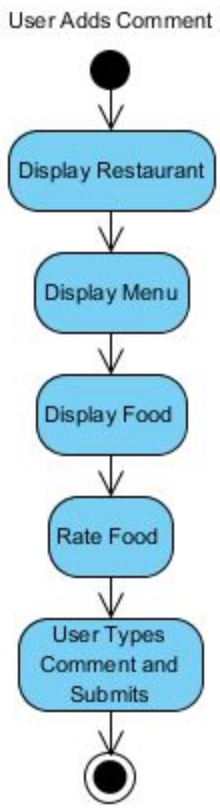
Use Case Scenario - Find Food using Map

The user is able to find food via the map. The user can open a map in the application. They can then locate a desired restaurant. The user would then go about opening the restaurant's menu. From the menu, the user is able to find desired food item.



Use Case Scenario - Filter Menu

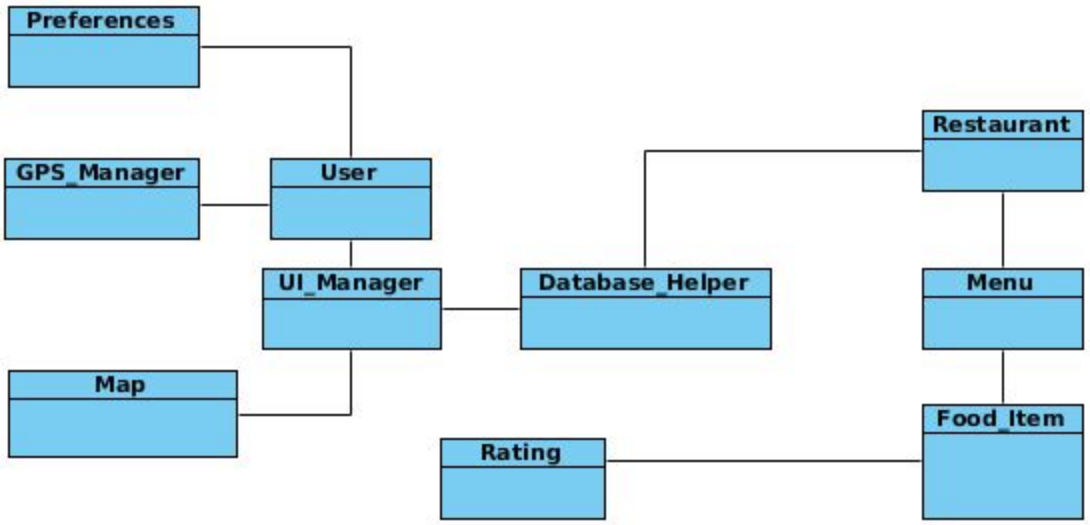
The user is able to filter the menu based on their preferences. The user would first access a menu. Then the user is able to apply a filter to the menu.



Use Case Scenario - User Adds Comment

User navigates to the desired food item. The user does this by searching for the restaurant either by Map or the Search bar. After rating the desired food, the user then clicks a button to add a comment, types out the comment and then submits the comment with the rating.

5.4 Class Diagram



High-level Class Diagram – Android FlavorTown App

Design Document

6. Scope and Features

6.1 Design Document Introduction

This design document will solidify the outline proposed for the design of the FlavorTown app. It will include diagrams and descriptions detailing how the FlavorTown app will function. Class diagrams, database diagrams, and sequence diagrams will further flesh out the structure and logic behind the app's systems.

6.2 Changes in Scope

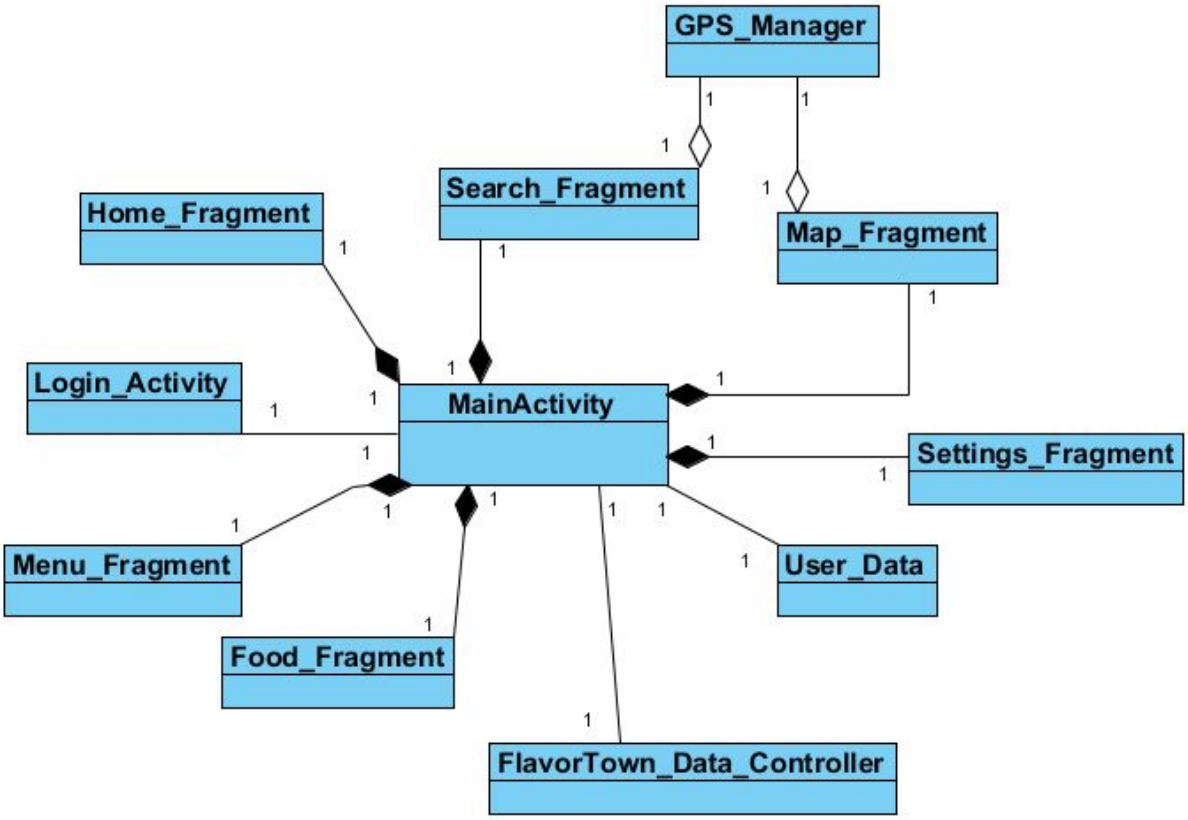
The application will still attempt to create a platform for people to try, rate, and find new foods. FlavorTown will allow the user to locate and browse restaurants, new and unfamiliar. The user will be able to find the best food items to try using the FlavorTown app.

6.3 Changes in Features

The ability for the system to guess a user's preference and further recommend food items shall be shelved until further notice. This due to time constraints, unneeded complexity in the base design, and the feature being low priority.

7. Class Diagrams

7.1 High Level Class Diagram



High Level Class Diagram – Android FlavorTown App.

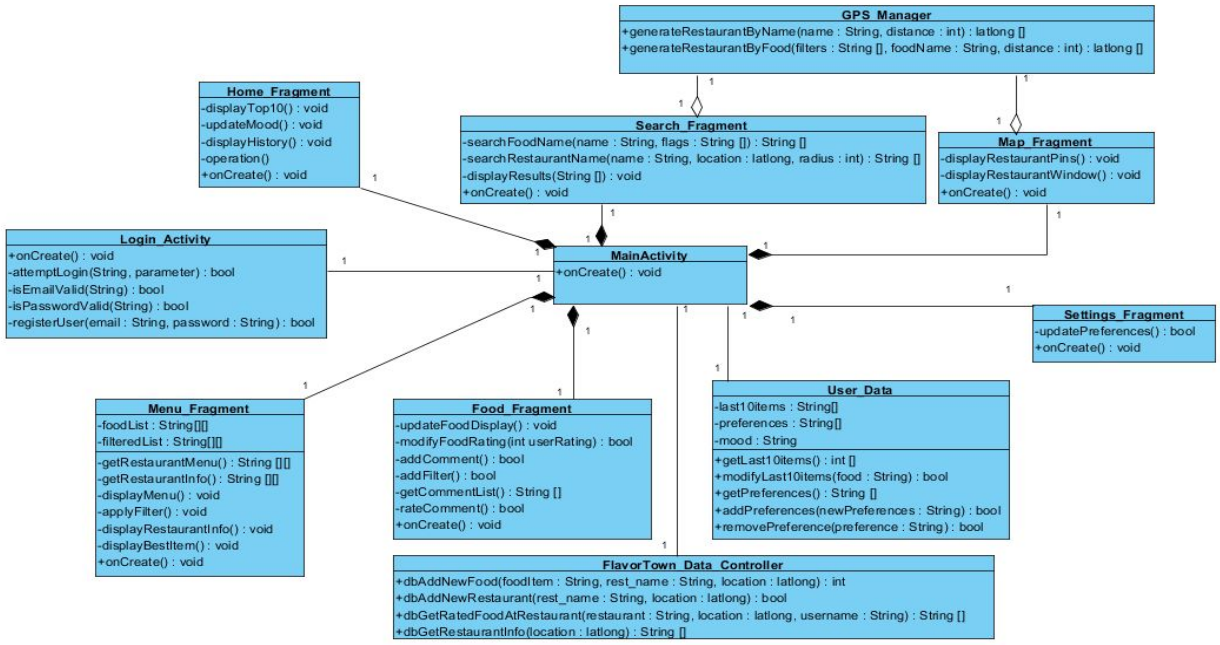
7.2 High Level Class Description

This diagram shows the relationships between the classes. We chose to split up the class diagrams between a high level diagram and separate images of the detailed class diagram so that it is easier to read. The abstract class diagram provides a way to view the connections between the classes more clearly.

The first activity, *Login_Activity*, is the first activity that the user will interact with and its only use is to facilitate the login procedure and upon successful login, creates the second activity, *Main_Acitivity*. *MainActivity* is what the user will interact with for the remainder of their session, and its main purpose is to control which fragments are currently being displayed. *MainActivity*

has six composite members: *Home_Fragment, Search_Fragment, Map_Fragment, Settings_Fragment, Menu_Fragment,* and *Food_Fragment*.. *Search_Fragment* and *Map_Fragment* are also unique in that they have a composite member in the form of *GPS_Manager*.

7.3 Detailed Class Diagrams



7.4 Class Diagram Description

This diagrams show how the classes of the Android application FlavorTown interact with one another.

Login_Activity is the first of two activities the user will interact with. Users must login to access the app and this class simply controls the display of the login screen and verifies the user has entered valid login information.

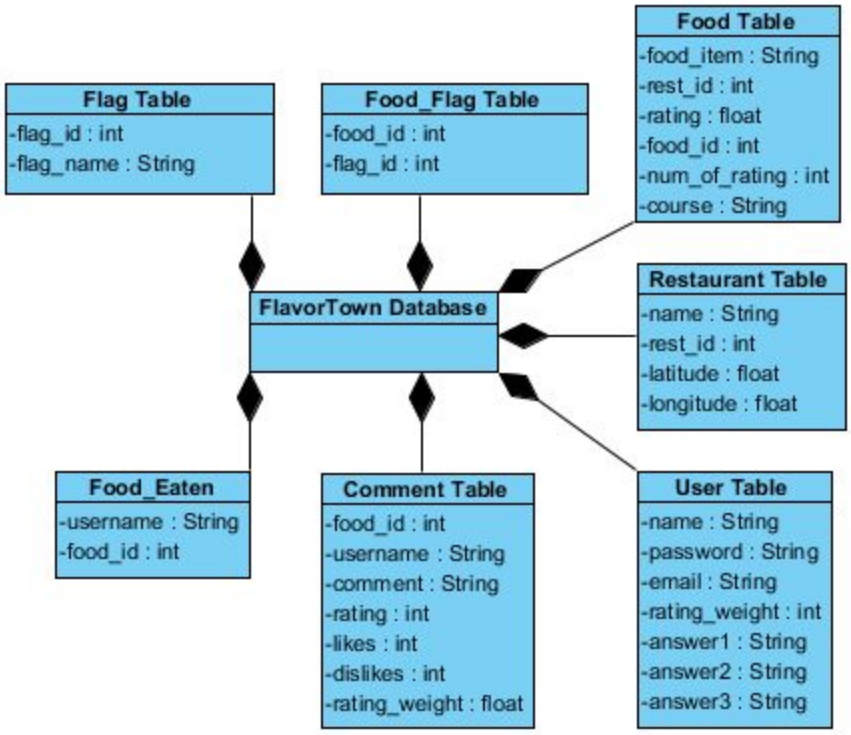
MainActivity is the second and only other activity the user interacts with. Its role is to control which fragments are currently being displayed on the app. The login window creates this class when a successful login is entered.

Home_Fragment, Search_Fragment, Map_Fragment, Settings_Fragment, Menu_Fragment, and *Food_Fragment* are simply fragment objects and control what they display on their own respective fragment. They also contain all the necessary functionality for displaying the necessary information.

GPS_Manager does calculations for *Search_Fragment* and *Map_Fragment*, assisting them provide restaurant locations that are within the search radius.

8. Database Diagram

8.1 Database Diagram

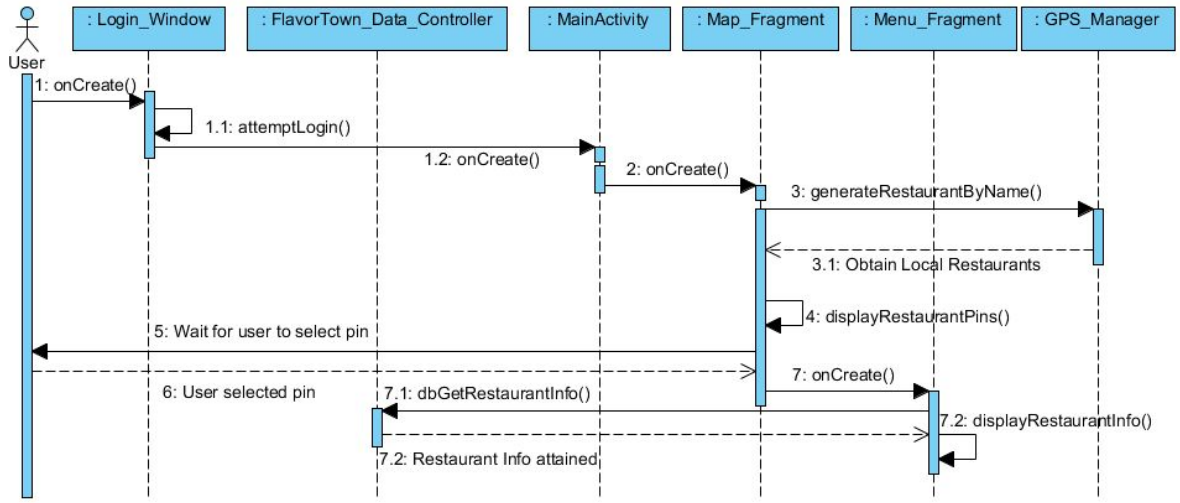


8.2 Database Description

The FlavorTown Database is pictured above in UML format to get a general layout of what the tables and their columns will look like.

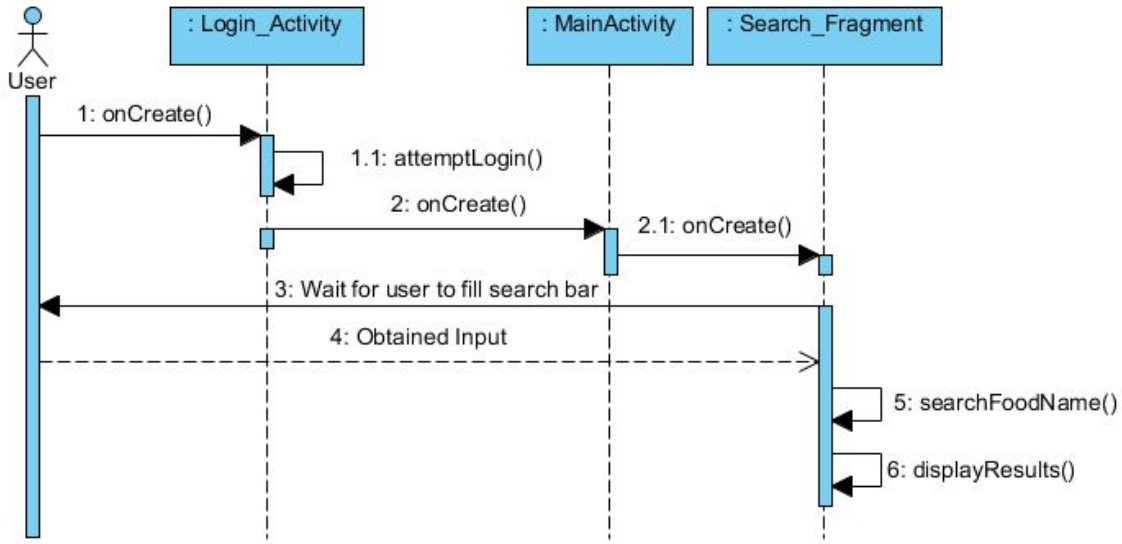
9. Sequence Diagrams

9.1 Sequence Diagrams



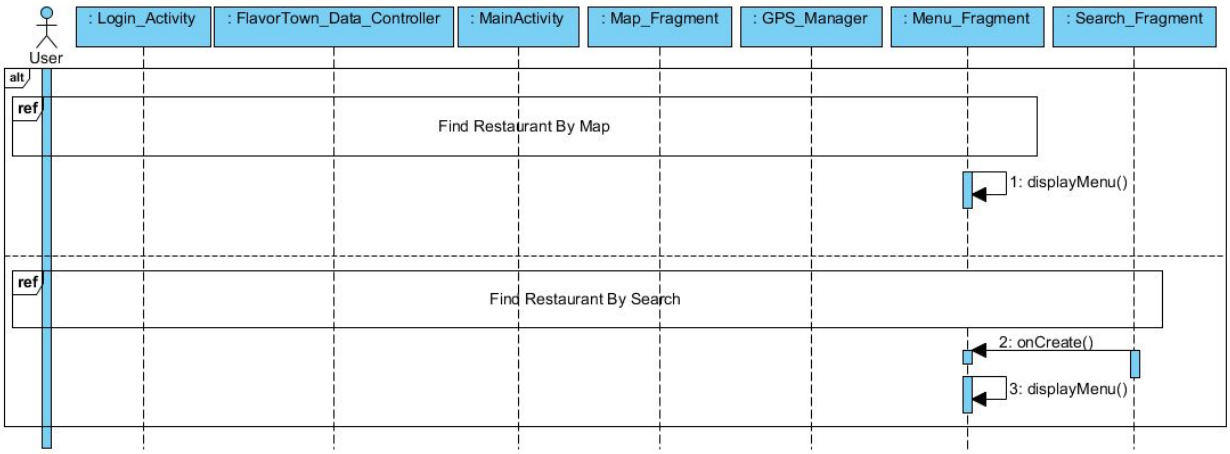
Sequence Diagram – Find Restaurant By Map

The user is able to find a restaurant through the use of searching on the map. The user would log in to the activity. From the home page, the user would would select the map button which would then display pins of all restaurants in a radius around the user's current location. The user could then select a pin to display a small window above the pin, requiring them to tap a button on the small window to confirm they wish to know more about the restaurant.



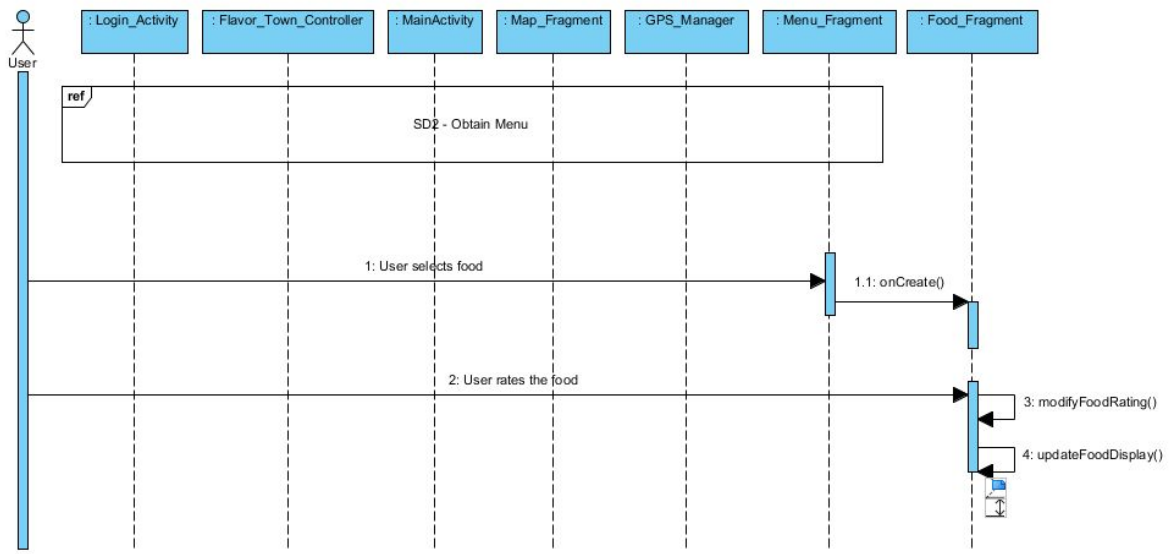
Sequence Diagram – Find Restaurant By Search

As an alternative to searching for a restaurant by using the map function, the app allows users to be able to find a restaurant by using the search function. After a user logs in and is redirected to the home page, the user would select the search button which would then display the search bar. From there they could then enter a restaurant name and receive a list a local restaurants of the same name to chose from.



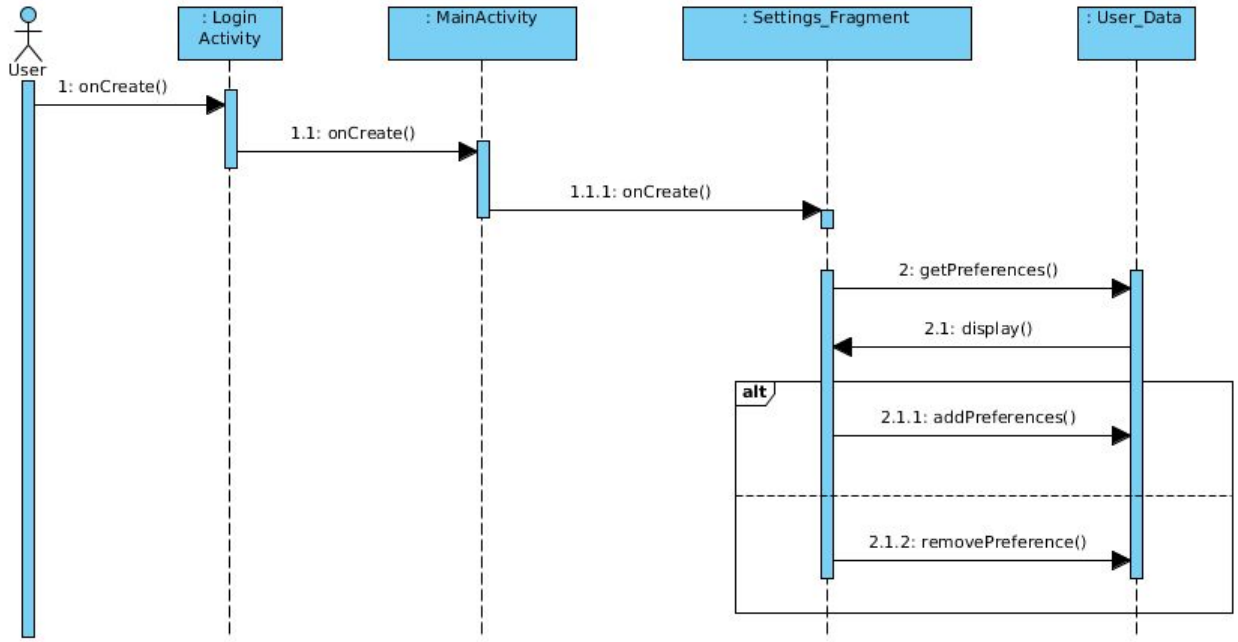
Sequence Diagram – Obtain Menu

The user is able to view the menu of a restaurant of their choosing. After selecting a restaurant from either the map or search functions, the user will be redirect to the “Menu_Fragment”. The ‘Menu_Fragment’ will display the information of the restaurant that was selected, as well as the list of the food offered and each food’s respective rating.



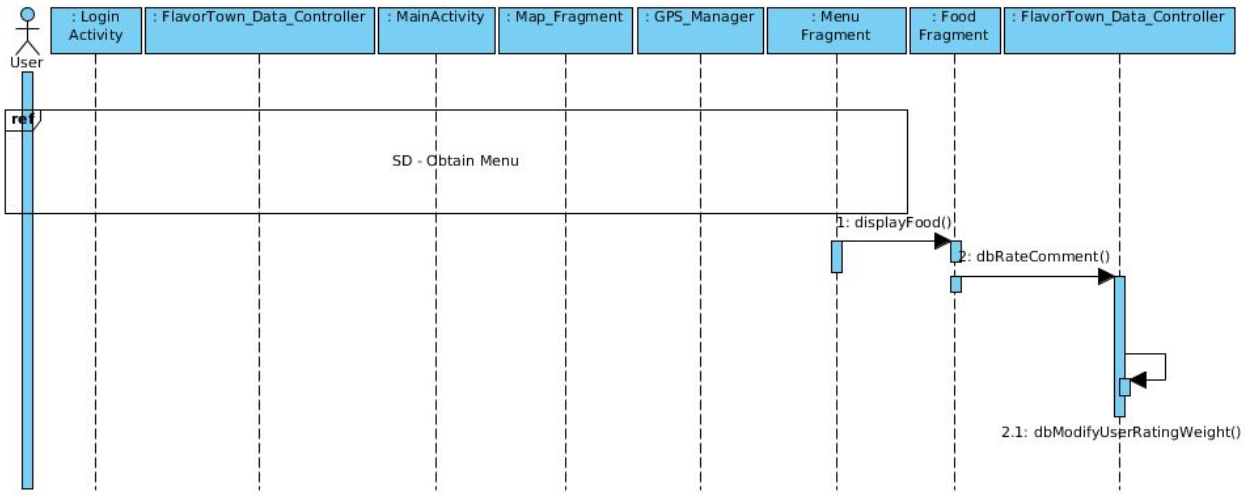
Sequence Diagram – Rate Food

Following from the previous 'Obtain Menu' sequence, users may choose which food they wish to rate. The user selects the food from the list on the 'Menu_Fragment'. The click will open up a new fragment called 'Food_Fragment'. The 'Food_Fragment' contains relevant information about the food the user selected, such as name, rating, comments, and rating on the comments.



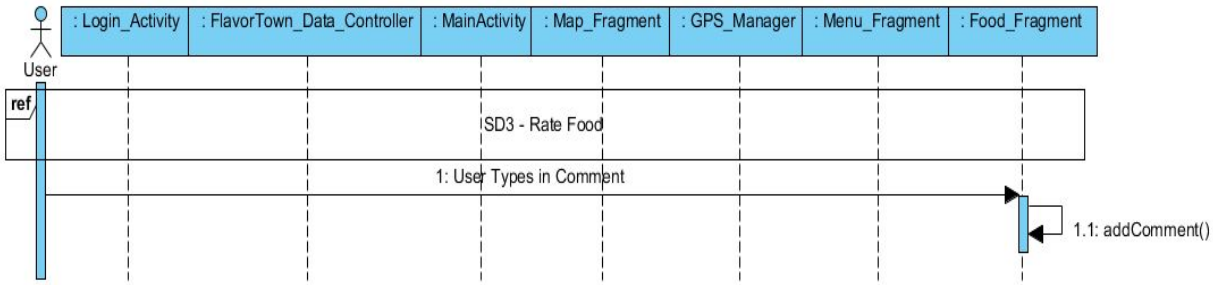
Sequence Diagram – Modify Preferences

The user is able to modify preferences in order to filter their options of food items. The user would login to the activity. Then from the main activity the user would go to the settings. From settings the user will be able to see their preference and able to add/remove preferences.



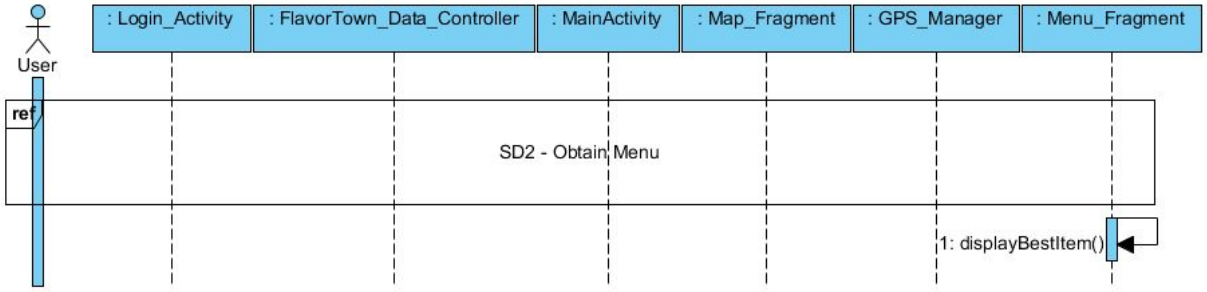
Sequence Diagram – Modify User Weights

The system will be able to adjust the weights of the foods. The user would login, go to main activity, access a restaurant menu, and then select a food. From the “Food_Fragment”, the user is able to rate other user's comments. The database would then modify the rating weight of the user that submitted the comment based on how it was rated.



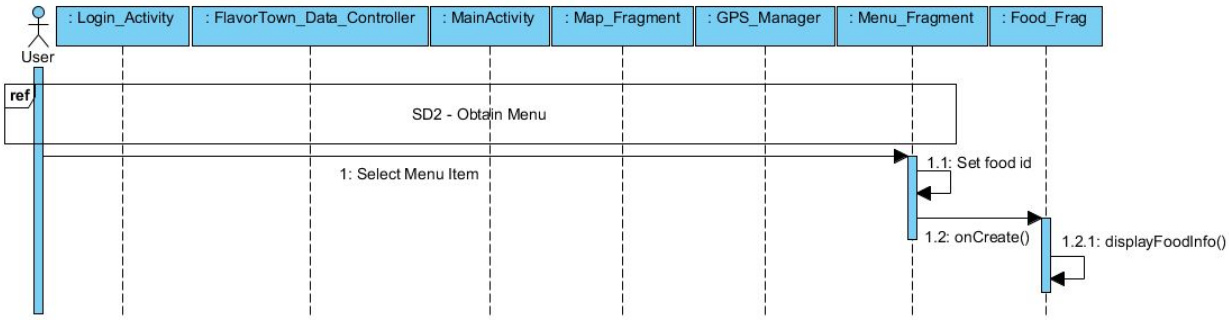
Sequence Diagram – Add Comment

While the user is rating a food item, they have the option to add a comment to their rating. Otherwise the value will be null in the database.



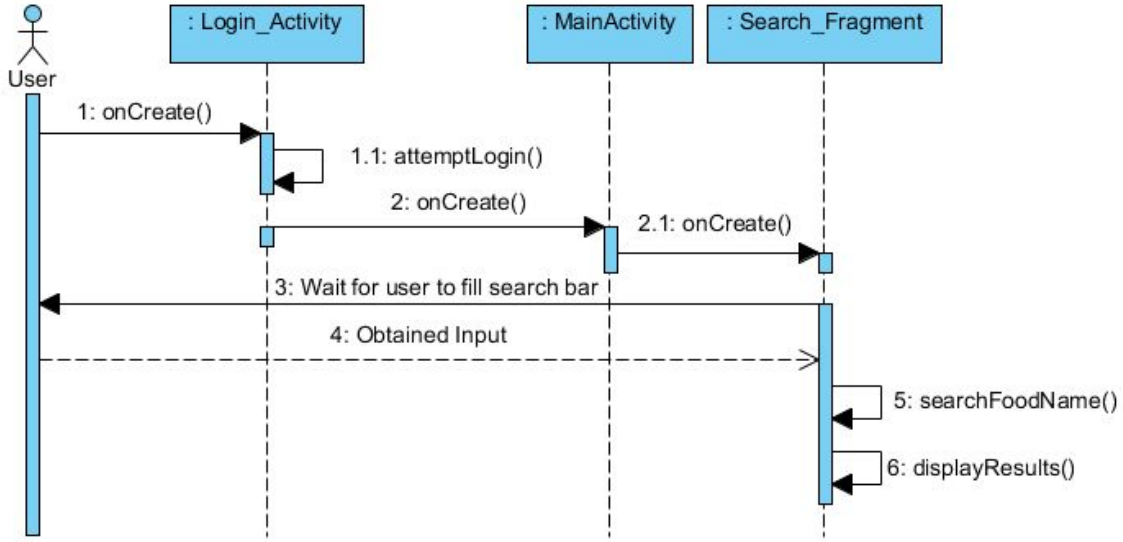
Sequence Diagram – Display Best Item

The user is capable of viewing the best item of a particular restaurant. Continuing on from the 'Obtain Menu' sequence, the user will be able to access the Menu Fragment. On the Menu Fragment, the food is grouped based on what type of meal it is (appetizer, entree, or dessert). In each category, the food is displayed in order based on overall average rating, such that higher rated items are at the top of the list of their respective category and lower rated items appear towards the bottom.



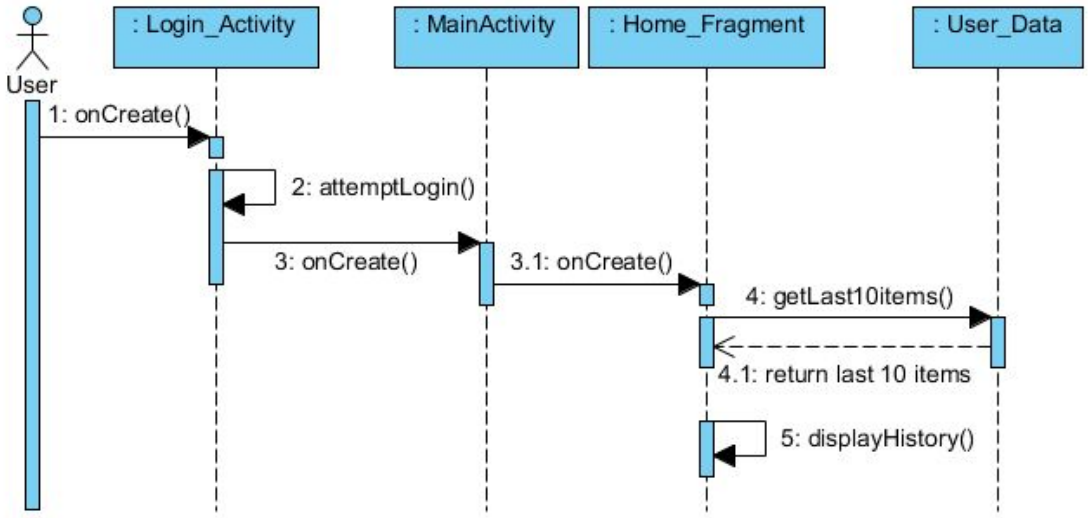
Sequence Diagram – Find Food Off Menu

At any time a user is given a restaurant menu, they can select a food item off that menu to receive more information on that item and potentially rate it. This is accomplished by the user selecting a food item from the menu. The app will then display all related information about that item as well as display its rating, posted comments, and options for the user to rate it.



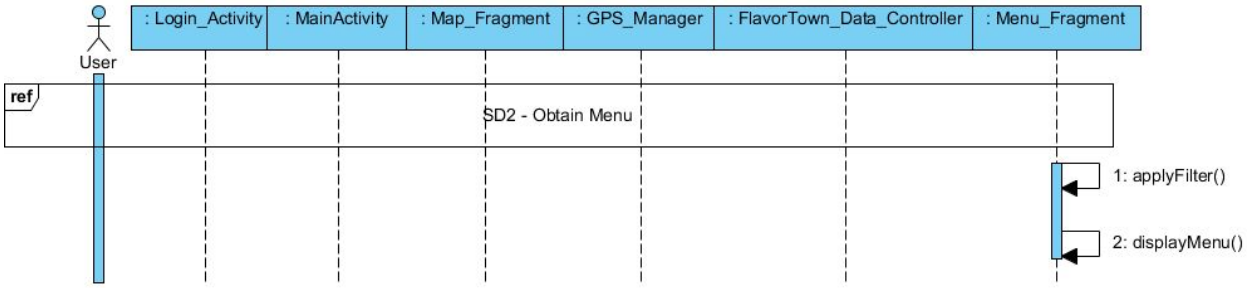
Sequence Diagram – Find Food by Search

The user is capable of finding food via a text search. The user logs in and upon successful login attempt, they would arrive at the home page. The user selects the search button and arrives at the search page. They may then enter the search parameters for the food they are seeking. Once they have entered all the parameters they wish, they may hit the search button. The database is queried with the parameters and a list of all food that meets the parameters. The list is then displayed on the Search Fragment.



Sequence Diagram – Find Food by History

The user is capable of finding food by viewing the last ten items they have rated, i.e. their history. The user logs in and upon a successful login attempt, they would arrive at the home page. From the homepage they will arrive at the History page. The History page will obtain the history from the locally stored data and display the information. The user can then click on the food and access their individual Food Fragments.



Sequence Diagram – Filter Menu

Once the user has obtain the list of menu items for a restaurant, apply one of the predefined filters to the list of food items (the menu) of the restaurant and then display the menu to the user