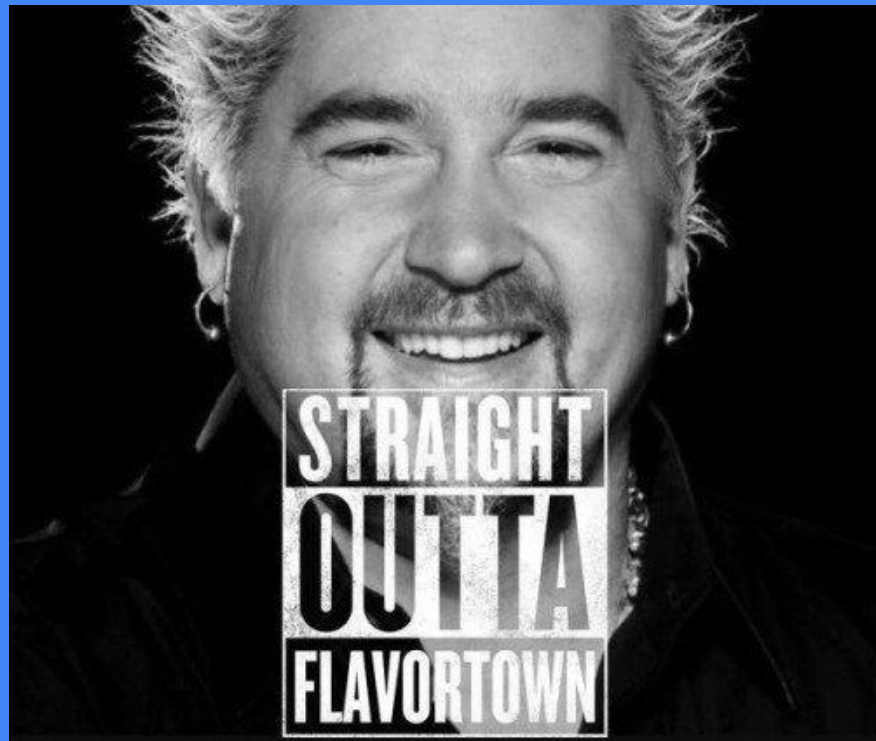# Flavor Town Requirements

Helping you find your new favorite meal

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

Requirements Outline

- Overall Description
- Functional Requirements
- Nonfunctional Requirements
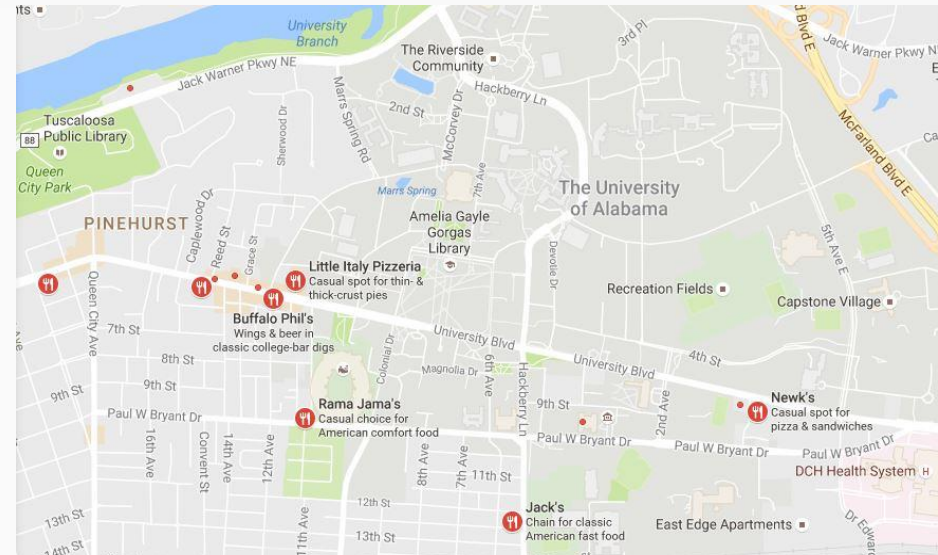- Supporting Diagrams
- Features/Issues

Project Idea:
To create a platform for people to try, rate, and share new foods

The problem

How often do you go to a new restaurant (or an old favorite) and not know what you want to get?

How often do you vacation somewhere and want to try the local food?

The solution

1. Pull up local restaurants on the app.
2. Display the menu of your restaurant of choice.
3. Sort recommendations based off of other user's ratings.
4. Order your food.
5. Enjoy!

Functional Requirements

High Priority

- Search Restaurant
- Obtain Menu
- Display Ranked Food Items
- Rate Food
- Update Database
- Calculate Ranking

Medium Priority

- Has Location Tracking
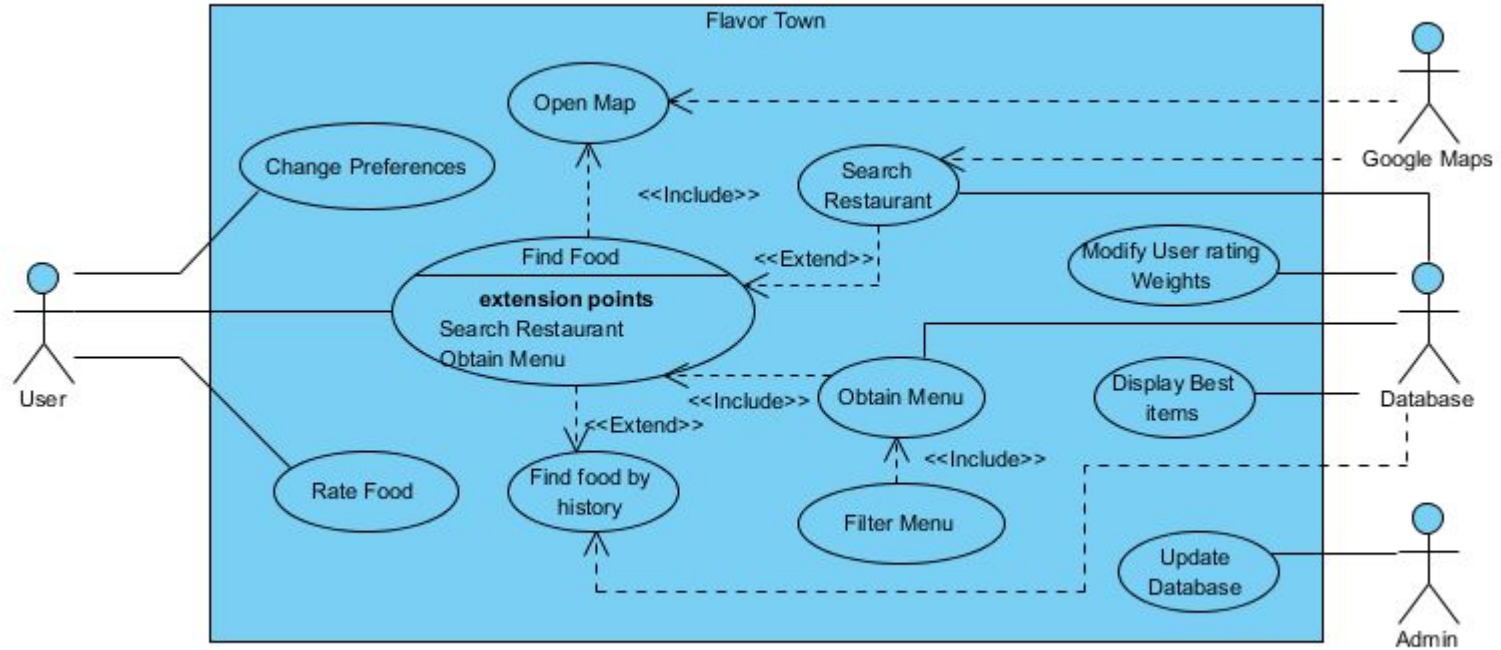- List Top Items in Area
- Filter Items by Category

Low Priority

- Show Food History
- Enable User Submit Comments
- List Restaurant Hours
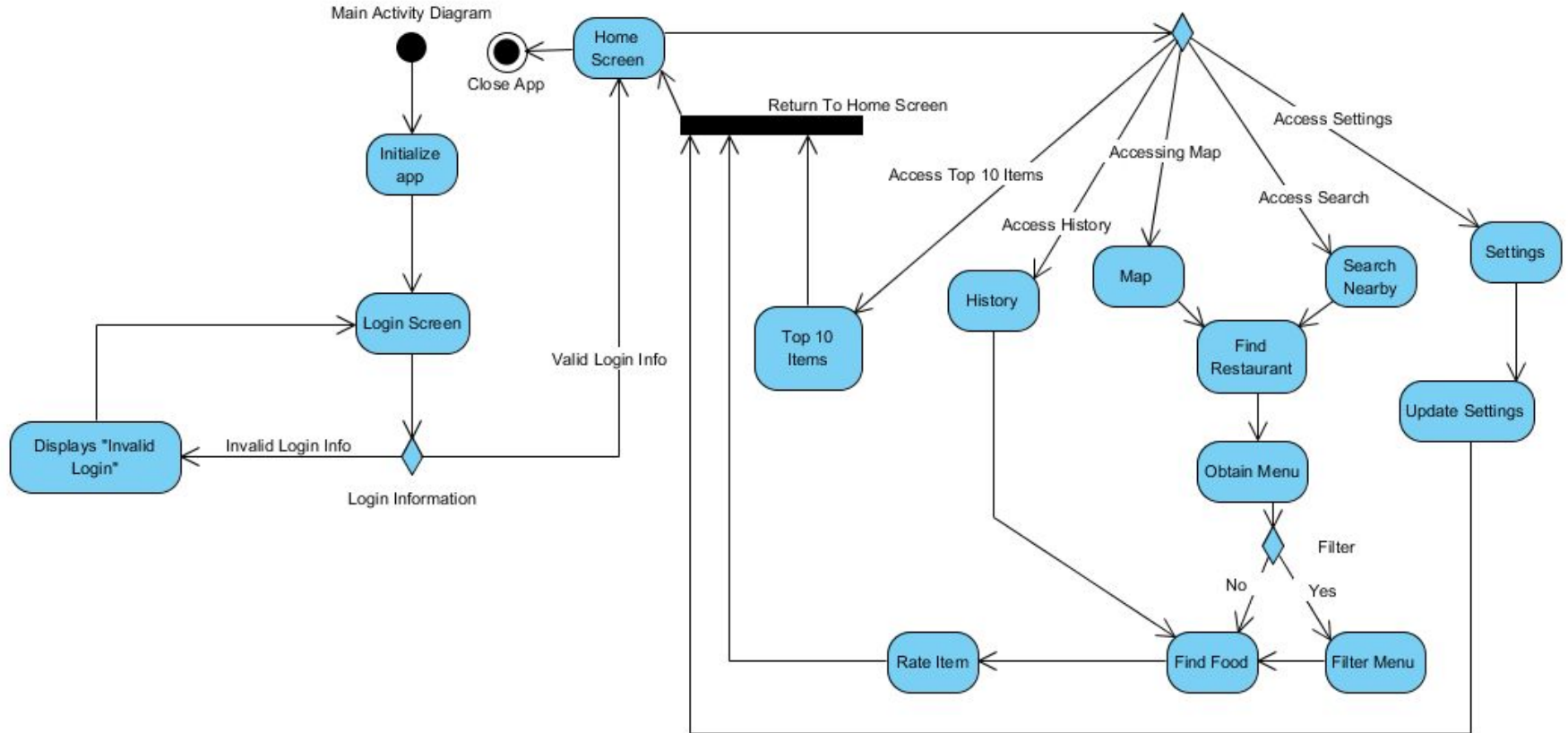- Guess User Preference with Suggestions

Nonfunctional Requirements

- Ease of use
- Low levels of intrusion for the user
- Fast response times
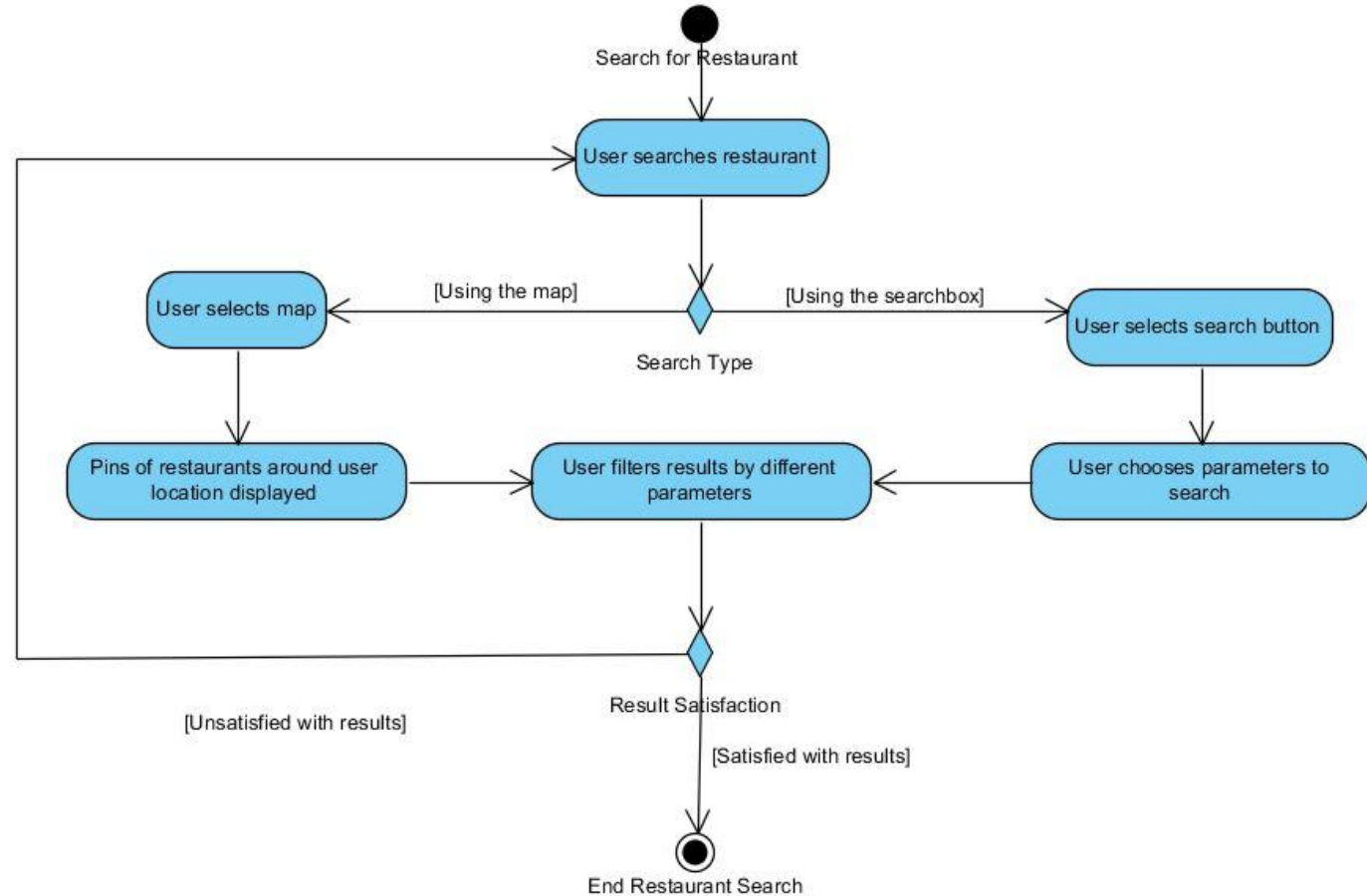- Popular (used widely enough to populate the database with ratings)
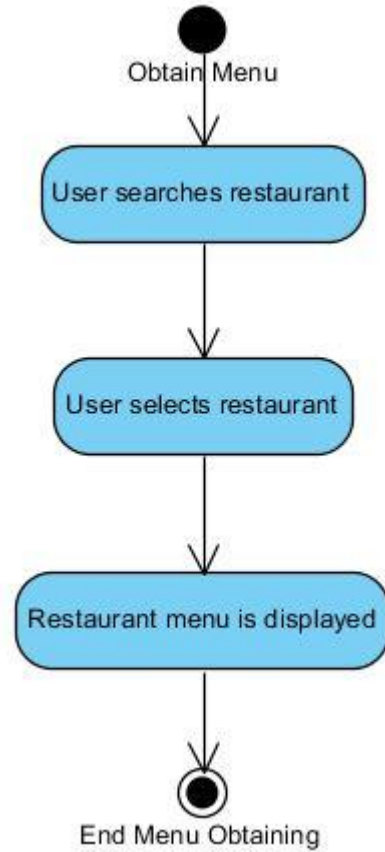
Use Case Diagram

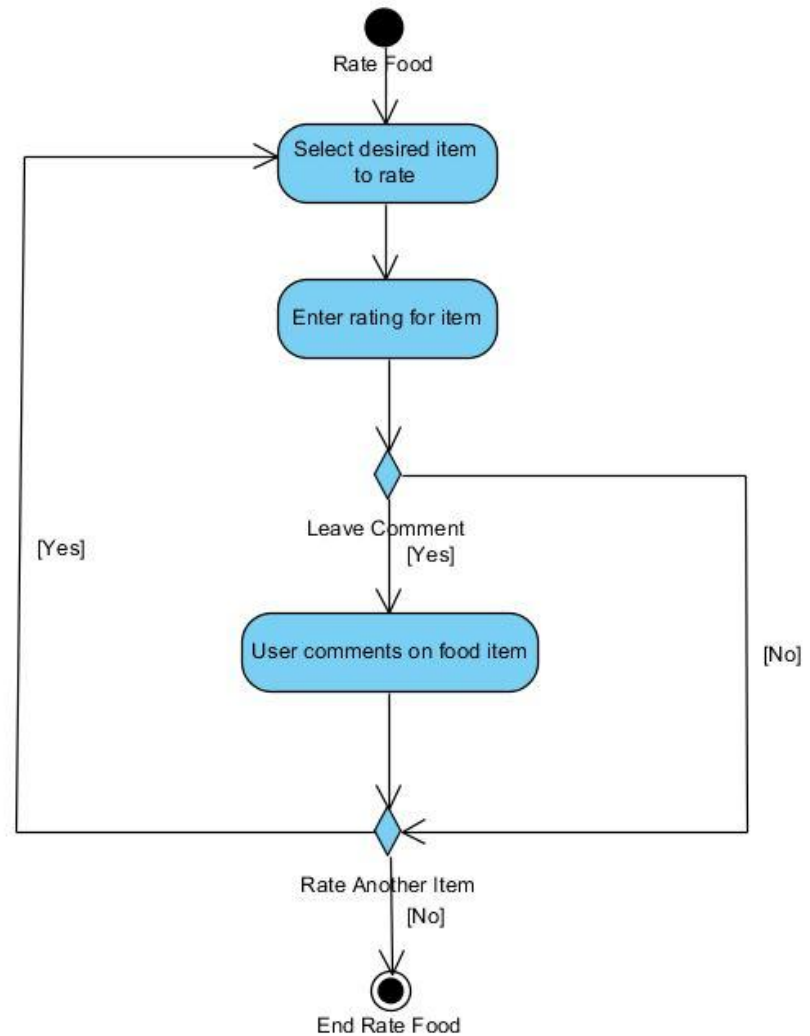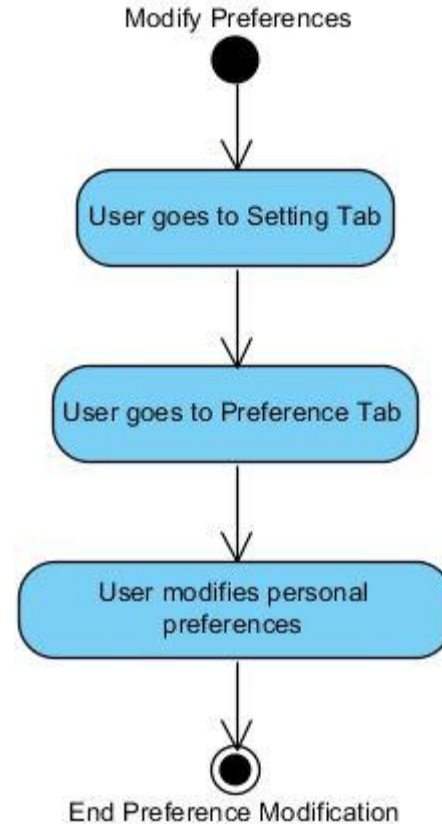# Main Activity Diagram

UC1: Search Restaurant

UC2: Obtain Menu

UC3: Rate Food

UC4: Modify Preferences

Modify Preferences

User goes to Setting Tab

User goes to Preference Tab

User modifies personal preferences

End Preference Modification

UC5: Modify Rating Weights

Modify User Rating Weights

○ (initial node)

User Comments

Negative Evaluation ◇ Positive Evaluation

User Comment is Evaluated by Other Users

Database Decreases User Rating Weight

Database Increases User Rating Weight

● End Activity
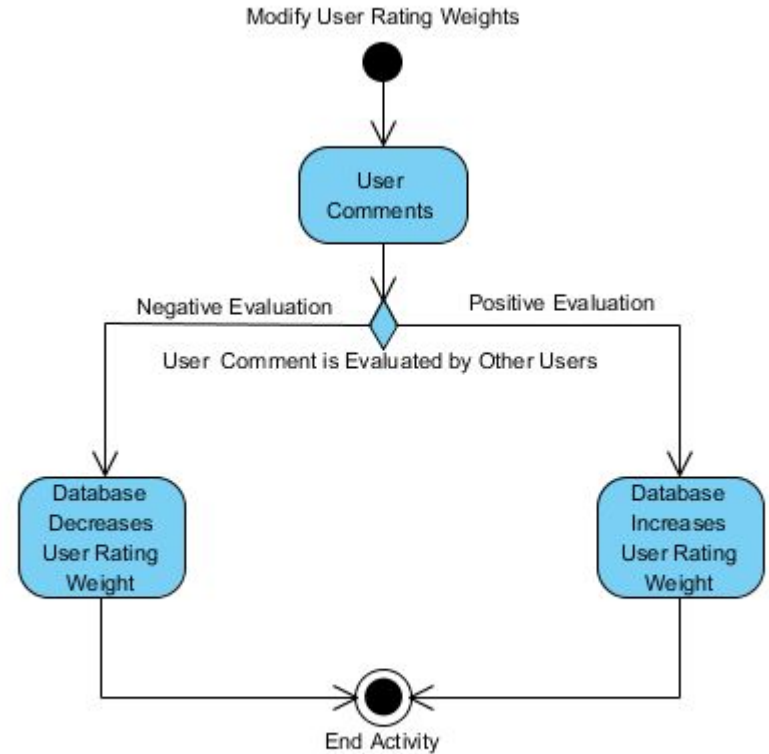
UC6: Update Database

UC7: Display Best Item

Display Best Item

App pings server for best food item in the user's area. (Default radius: 10 minutes)

Does server have best item calculated?

No → Server goes through all recommendations in the database.

Yes

Is it up-to-date?

Yes

Calculates the highest rate item. (Highest ratio of good vs bad)

Server sends the result to the user's app.

Server saves the result for quicker pull up until it need to be recalculated

App displays best item

UC8: Find Food By History

Find Food by History

- User opens "History" Page
- App pings server for the user's history (Last 10 items).
- User's history is displayed.
- User selects desired food item.

UC9: Find Food Via Map

Find Food - Via Map

User Opens Map

Finds Desired Restaurant → Opens Restaurant's Menu → User Finds Desired item
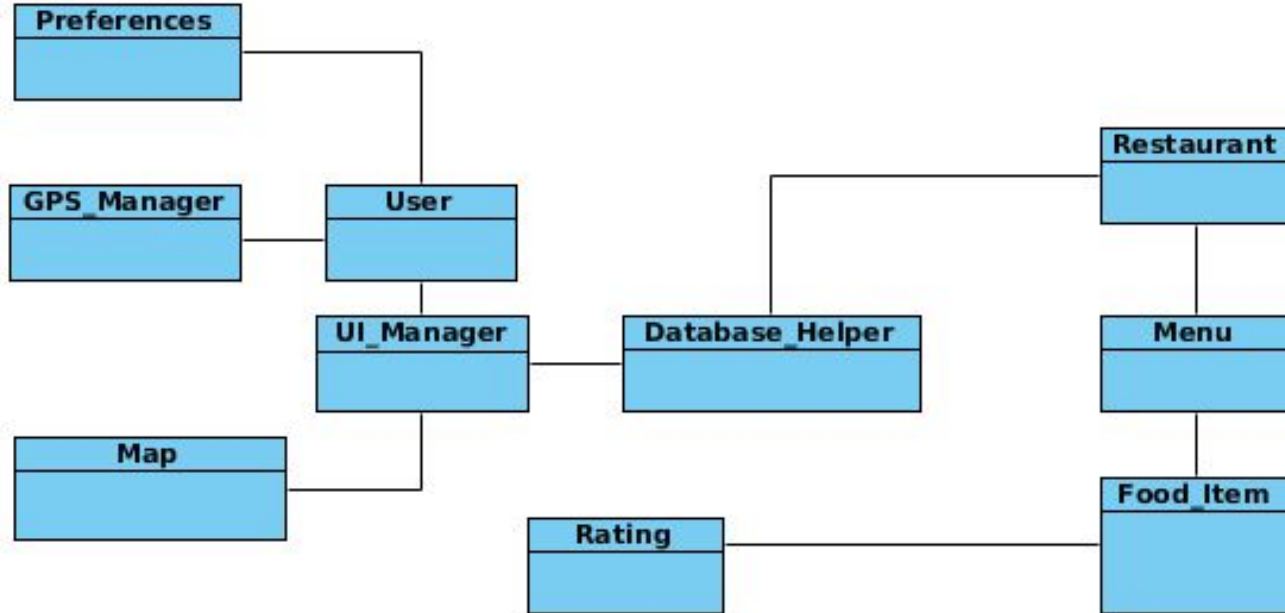
Food Found
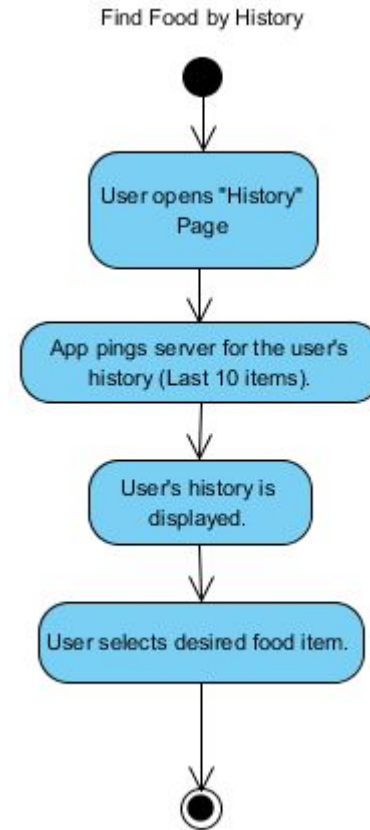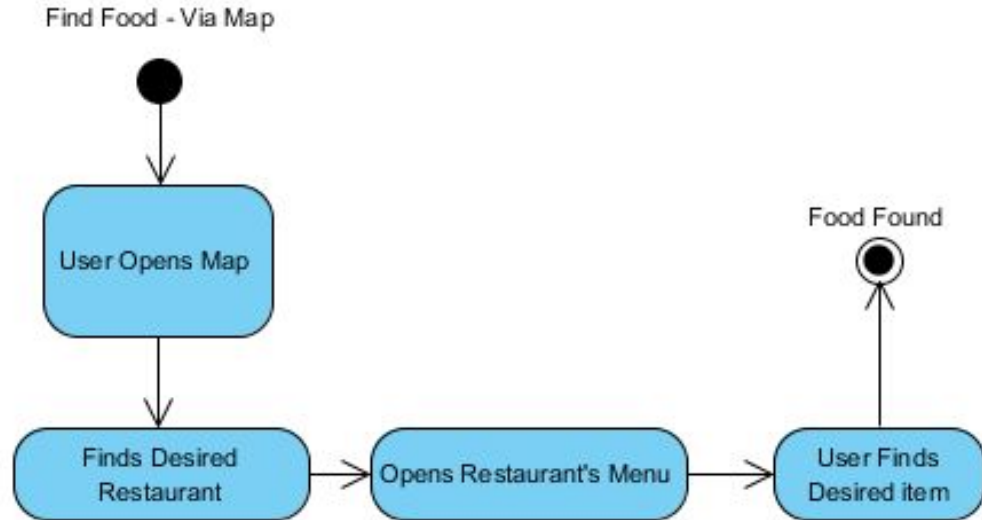
UC10: Filter Menu
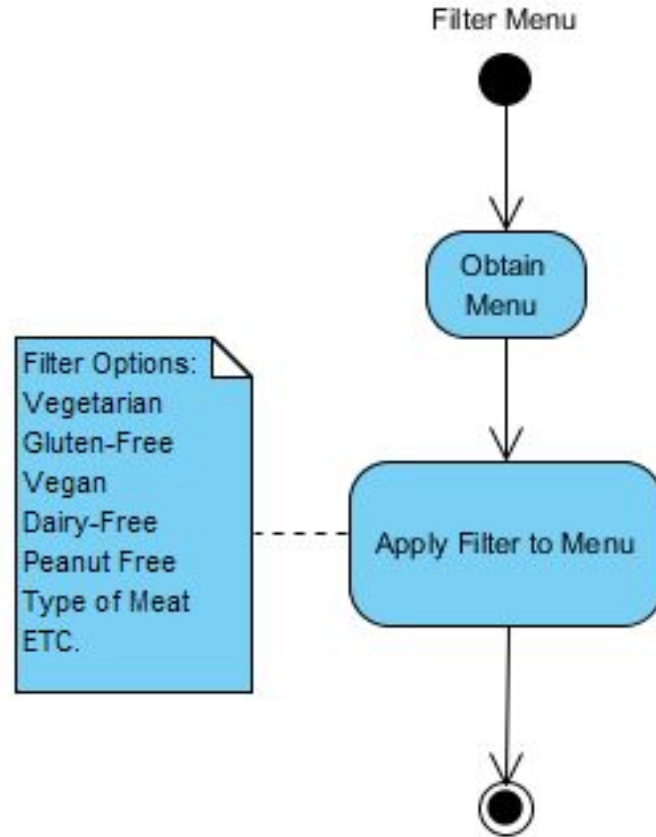
Class Diagram

Potential Risks & Problems

- Database Management
- Preventing artificial ratings
- There might not be enough users to populate database
- Might be difficult to get menu items

Team Member's Discussions

- The project is going to differentiate itself by telling the user what item is best instead of what restaurant is best
- Hopefully people will think this is going to be useful
- Our project should be easy and convenient to use
- Our hope is that this project is adopted and used to some extent

Questions?