



FoodBuddy - Food Recommendation System

Team Members:

Vikhyat Goyal

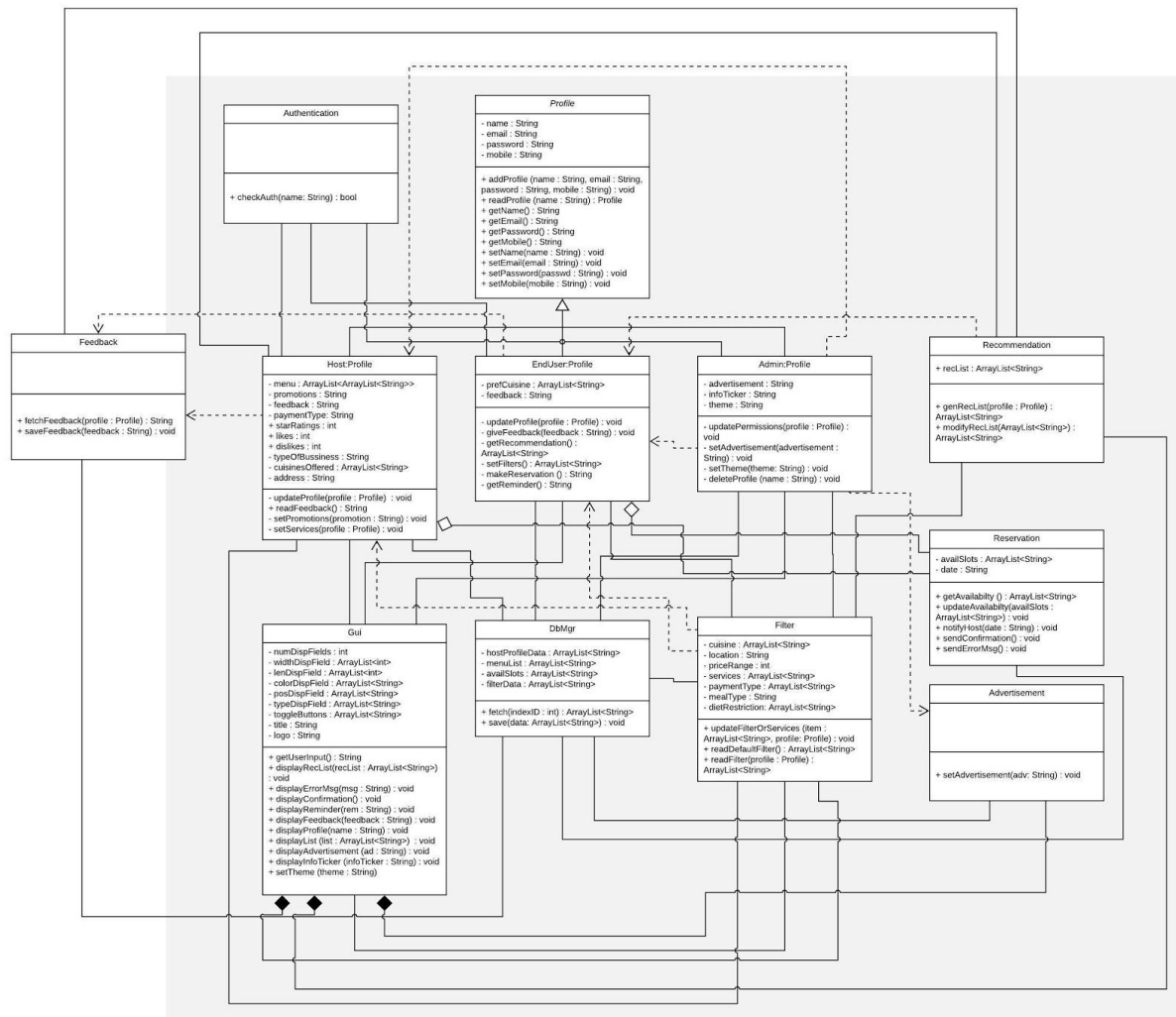
Pavan Dhareshwar

Rishi Soni

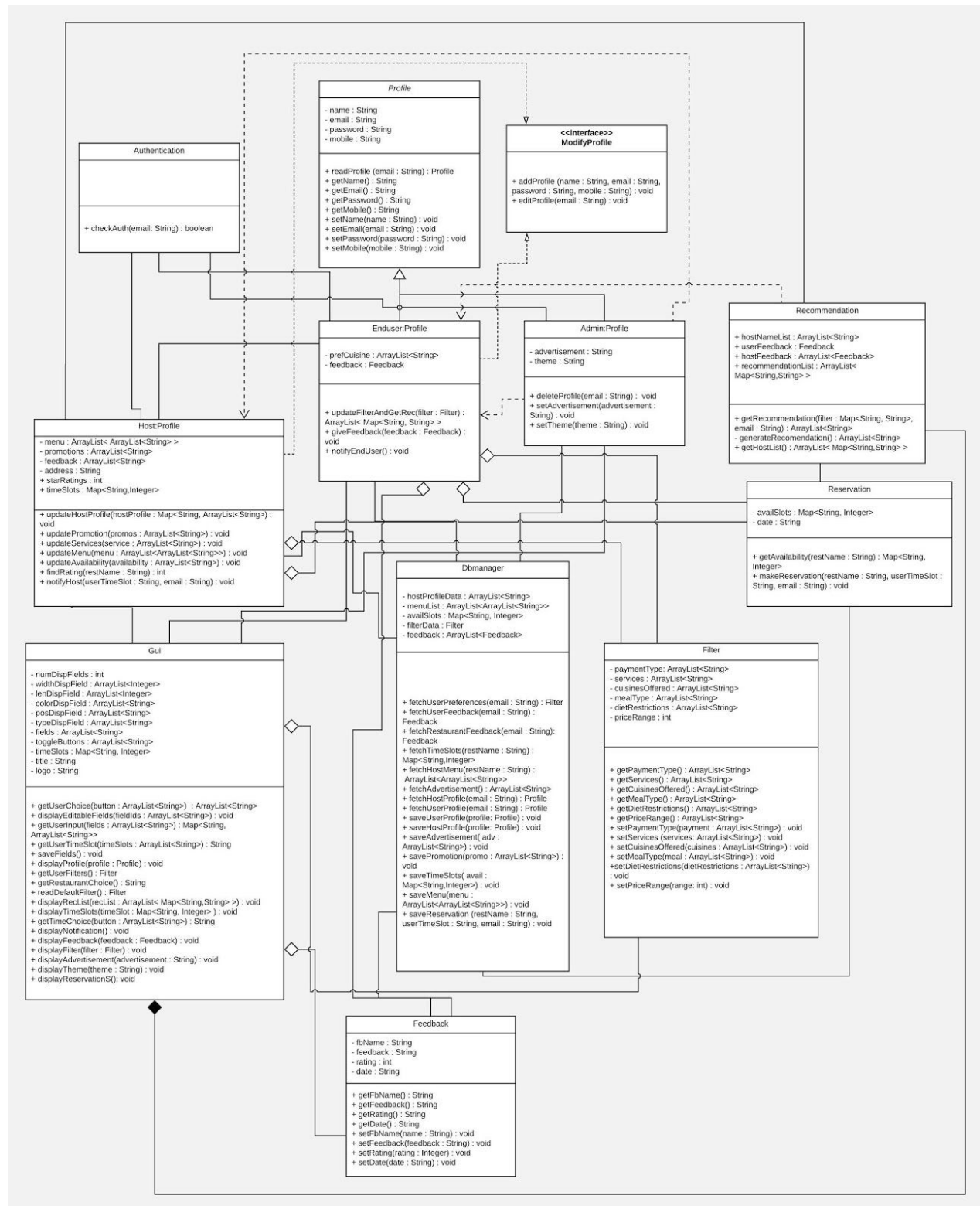
Summary:

Our system for the project will be a website that will enable end users to find new restaurants or food product to try based on their profile, past experiences and random projection. It will also allows new restaurants to get registered and promote their business, old restaurants and food product manufacturers to add new cuisines/dishes and get feedback.

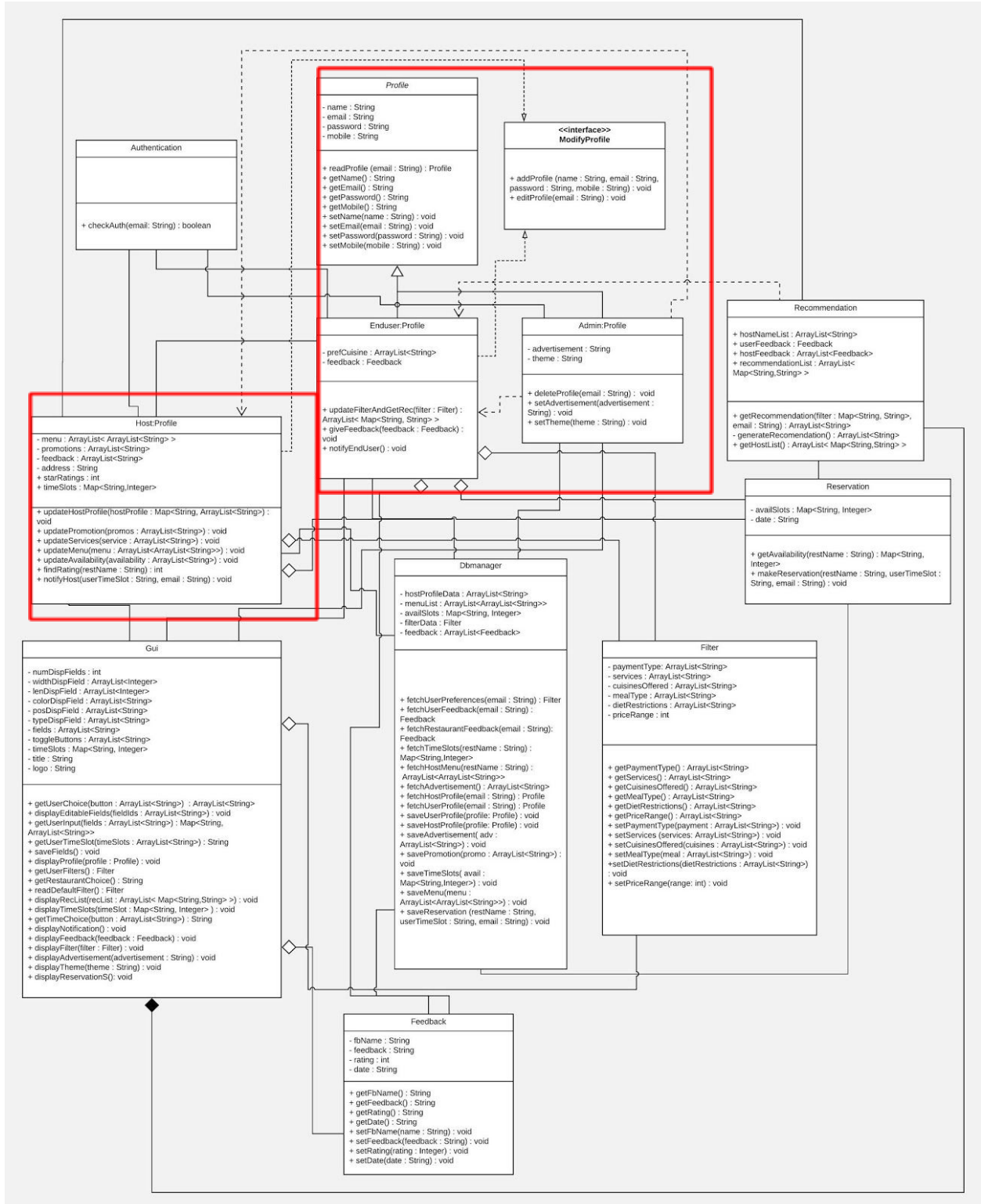
Previous Class Diagram:



Refactored Class Diagram:



Completed Class Diagram:



Summary:

1. Incorporating review comments:

Based on the feedback received for project part 2, we went back and made changes in all the UML diagrams (use case document, use case diagram, activity diagram, sequence diagram, and class diagram) to incorporate the review comments. This made us understand our implementation better, in terms of the OOP principles being used

2. Initial setup of project classes and functionality:

Based on the class diagram, we created all the classes that will define the core functionality of the entire system in a Java project in Eclipse. The classes contain the attributes, constructor(s), getter and setter methods, and methods that make up the functionality of the said class. These methods are prototypes, that will be implemented as we progress in the project.

3. Understanding how to build Spring framework for the final application:

We followed the procedure to build the Spring framework for a custom Java-based application. To test the basic functionality of the framework, a sample model class along with a controller class was written and mvn Spring-Boot was configured to run on local host on a custom port (e.g: 8080) and the get request from a browser were mapped to a specific request mappings in the controller class. We haven't perfected this and are still working on it.

Breakdown:

The team worked on first understanding and correcting all the points which were a part of Part2 feedback, this included rework on the sequence diagrams, activity diagrams and the class diagram. Each member of the team contributed in making sure we meet all the guidelines and correct all our misunderstanding and mistakes.

Once, we had a better understanding of our requirement and their mapping to the UML diagrams, each of us worked on refactoring the class diagram. In particular :

Rishi : Worked on defining the methods which will be used to meet functional requirements of adding a new end user/host/admin and making a reservation.

Pavan : Worked on defining the DataBase and how the Dbmanager class will support access to the MySQL database for fetching and saving data.

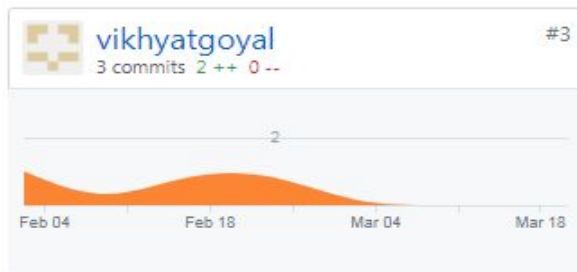
Vikhyat : Worked on defining the recommendation and GUI class methods. Each use case was looked in from GUI perspective and methods were defined accordingly.

GitHub Graph:

Feb 4, 2018 – Mar 22, 2018

Contributions: Commits ▼

Contributions to master, excluding merge commits



Next Iteration:

1. Complete integrating the Spring framework in the project.
 2. Initial implementation of methodologies to save and retrieve the data from the database.
 3. Initial implementation of the Recommendation class algorithm to generate the recommendation list based on user preferences, user feedback and restaurant reviews.
- Following is the points-based algorithm for generating recommendation:

Recommendation:

✓ (1) User Preferences
(2) User Feedback
(3) Host Feedback

$$\begin{aligned}
 & \bullet \text{cuisine} * (3) = 24 + \frac{24}{\# \text{cuisines displayed}} * \# \text{cuisines offered} \\
 & \bullet \text{price range} * (3) = 20 + 20 \times \text{range-matched (1 or 0)} \\
 & \bullet \text{payment type } \Delta (1) = 8 + \left(\frac{8}{\# \text{payments available}} \right) * \# \text{payments offered} * \# \text{user-payment pref} \\
 & \bullet \text{services } \square (2) = 16 + \left(\frac{16}{\# \text{services available}} \right) * \# \text{user pref} * \# \text{services offered} \\
 & \quad \bullet \text{wifi} = 4 \\
 & \quad \bullet \text{Home Delivery} = 4 \\
 & \quad \bullet \text{Take out} = 4 \\
 & \quad \bullet \text{Parking} = 4 \\
 & \bullet \text{Meal Type } \Delta (1) = 8 + \left(\frac{8}{\# \text{meal options}} \right) * \# \text{user pref} * \# \text{meal-types offered} \\
 & \bullet \text{Dietary Restriction } \times (3) = 24 + \left(\frac{24}{\# \text{diet resty}} \right) * \# \text{user pref} * \# \text{host diet rest offered}
 \end{aligned}$$

100/13 = 2.8

Cur. User Specific:

Name	Rating	Feedback
Name + cuisine + dish	out of 5	"Some food Ambiance"
:	:	:
:	:	:

Host Feedback:

Rating	Cuisine + Dish
:	:

4. Identify the Design Patterns that can be used in the project.
5. Modifying the class diagram to show at-least 3 Design Patterns implemented.

Estimated Remaining Effort:

1. Implementing the core recommendation algorithm for the project.
2. Complete integrating the Spring framework in the project.
3. Implement the functionality of all the methods in all the classes and establish relationships between the classes.
4. Implement the GUI as represented in the UI mockup diagrams and the class diagram.
5. Implement the Design Patterns in code.
6. Implementation of methodologies to save and retrieve data from the database.
7. Designing and running unit tests to test the functionality of all the modules.