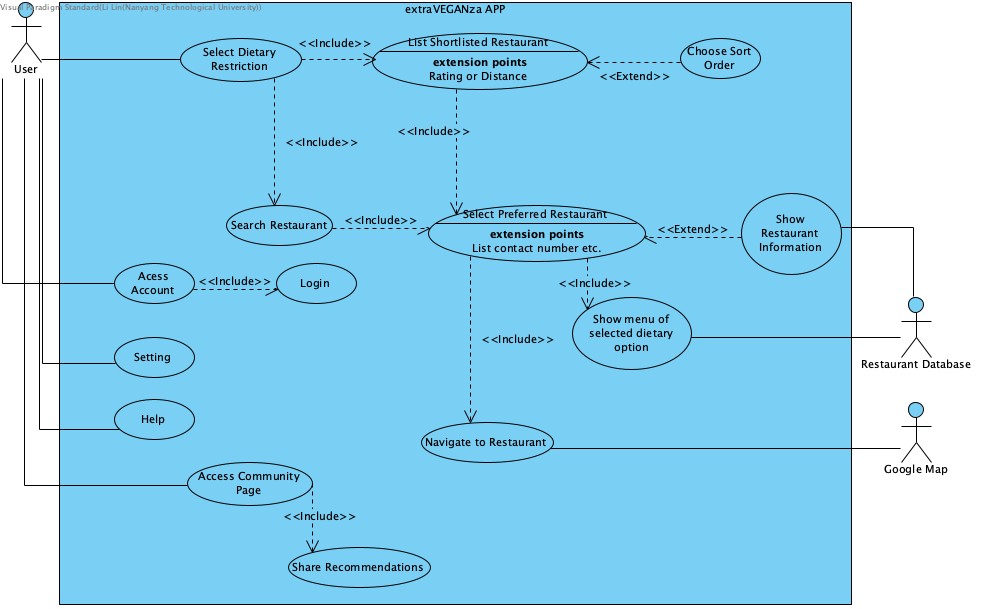
**3.3 Visualize and refine requirements with Use Case models**

**3.3.1 From the set of functional requirements, identify the preliminary Use Cases. Depict them on a Use Case diagram using the UML modeling tool.**

****

* Settings → various settings
* Help **→** Show how to use the app, report, etc.
* Login → Further login
* Account

**3.3.2 For each Use Case, start writing the use case description about how the user interacts with the system to carry out the system functionality. As a rule-of-thumb, each Use Case should have a maximum of 6~7 steps in its flow of events.**

**A small Use Case indicates that the functionality has been sliced too finely; a large Use Case can be further broken down.**

1. Select Dietary Restrictions
   1. Description: This use case allows users to select their own dietary restrictions.
   2. Anticipating Actors:
      1. User (Initiating actor)
   3. Flow of events:
      1. The user opens extraVEGANza APP.
      2. The user is on the main page of extraVEGANza APP.
      3. The user clicks on the preferred dietary option.
      4. The user views the specific information on selected option.
      5. The user confirms the selected option.
2. List Shortlisted Restaurants
   1. Description: This use case allows users to obtain a list of restaurants based on a predetermined criteria.
   2. Anticipating Actors:
      1. User
   3. Flow of events:
      1. The APP traverses the database. ??
      2. If a restaurant provides the selected diet, the APP shows that restaurant on the page.
3. Choose Sort Order
   1. Description: This use case allows users to choose the sorting order of the listing of restaurants.
   2. Anticipating Actors:
      1. User
   3. Flow of events:
      1. The user chooses sorting order, either by distance or by rating.
      2. The APP reorders the listing restaurants according to the user’s request.
4. Search Restaurant
   1. Description: This use case allows users to search a restaurant from the restaurant database.
   2. Anticipating Actors:
      1. User
   3. Flow of events:
      1. The user clicks on the search box.
      2. The user enters the name of the restaurant.
      3. The APP searches the name of the restaurant in the database.
      4. The APP shows the result.
5. Select Preferred Restaurants
   1. Description: This use case allows users to select a restaurant from the listing of restaurants.
   2. Anticipating Actors:
      1. User
   3. Flow of events:
      1. The user clicks on the preferred restaurant.
      2. The user confirms the restaurant.
   4. Alternative Flows:
      1. If the user likes the restaurant, she can bookmark it by clicking the “❤️”.
6. Show Restaurant Information
   1. Description: This use case allows users to view the information of a restaurant after being selected.
   2. Anticipating Actors:
      1. User
      2. Restaurant Database
   3. Flow of events:
      1. APP retrieves the restaurant’s information, including location, contact number and operation hours.
7. Show Restaurant Menu
   1. Description: This use case allows users to view a restaurant’s menu with only selected dietary options.
   2. Anticipating Actors:
      1. User
      2. Restaurant Database
   3. Flow of events:
      1. APP shows the restaurant’s menu with selected dietary options.
      2. APP shows other users’ ratings and recommendations.
      3. The user clicks on the food option.
      4. APP shows the food’s information, including ingredients, calories and allergies.
8. Navigate to Restaurant
   1. Description: This use case allows users to navigate to the restaurant via Google Map.
   2. Anticipating Actors:
      1. User
      2. Google Map
   3. Flow of events:
      1. The user chooses the restaurant she wants to go.
      2. The user chooses the way of reaching the selected restaurant.
      3. Google Map gives direction.
9. Access Community Page
   1. Description: This use case allows users to access the application’s community page.
   2. Anticipating Actors:
      1. User (Initiating actor)
   3. Flow of events:
      1. The user enters her username and password.
      2. The APP verifies the username and password with the data stored in the database.
      3. If the username and password are verified, the APP shows the community page.
      4. The user views other users’ recommendations
10. Share Recommendations
    1. Description: This use case allows users to share their recommendation of food / restaurants to the community page.
    2. Anticipating Actors:
       1. User
    3. Flow of events:
       1. The user selects food or restaurant she wants to recommend.
       2. The user writes her recommendation in the text box shown.
       3. The user sends out the recommendation.
11. Login
    1. Description: This use case allows users to login into / signup for their extraVEGANza application account.
    2. Anticipating Actors:
       1. User
    3. Flow of events:
       1. The user will be prompted to enter their username and password for their account.
       2. If authentication of the account is successful, the user will be able to login to their account.
       3. The user will be guided back to the home page.
       4. (alternate flow) If the authentication of the account is unsuccessful, the user will be prompted for their username and password again (repeat from step 1).
12. Access Settings
    1. Description: This use case allows users to manage the settings of the application.
    2. Anticipating Actors:
       1. User
    3. Flow of events:
       1. The user is shown a list of settings that can be managed.
       2. The user chooses which settings to be changed, and changes it according to the user’s preferences.
13. Access Help
    1. Description: This use case allows users to access the help page of the application.
    2. Anticipating Actors:
       1. User
    3. Flow of events:
       1. The user is shown a help page for the application.
14. Access Account
    1. Description: This use case allows users to manage their account’s settings.
    2. Anticipating Actors:
       1. User
    3. Flow of events:
       1. The user is shown a list of account’s settings that can be managed.
       2. The user chooses which settings to be changed, and changes it according to the user’s preferences.

**3.3.3 Iterate over your Use Case model to identify included Use Cases, extended Use Cases, and generalization relationships, if any.**