SUNY Oswego - CSC 380

Prof Alex Pantaleev

Software Engineering Project

Group Members: Yingying Xia, Kim Brian Fadul, Matt Reeves

**Project Description**

In this restaurant searching application (name pending), users would be able to search up restaurant by names, or by types. This restaurant searching application is aimed to provide users a platform to find good restaurants and to post ratings for restaurants. Instead of believing in restaurants’ own beautiful advertisements, users would be able to find real restaurant critiques and reviews from other users who have already experienced or consumed in these restaurants. This will ensure that they, as users, will be able to make the best decision possible upon choosing restaurants. This application will also track similar restaurants with the same type so that if the users fancy themselves some Asian cuisine they don't just have to settle for one restaurant.

Food is a significant part of human lives, and eating out is also an important part of people’s social experiences. Being able to find decent restaurants can be a pain, thus the aim of our restaurant searching application is to lighten this load by having a restaurant rating system, which are mainly composed by user critiques and carefully maintained by administrators. This ensures a lower likelihood of a restaurant getting false reviews, and also saves consumers from spending money and time on terrible dining experiences.

**Project Requirements**

Part A: The Systems Requirement

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **System Requirement** | **Priority Weight**  **(in a scale of 1 to 5 stars)** | **Estimated Completion Time** |
| SREQ1 | The system should be able to save information and retrieve information in the database. | ★★★★★ | 15 days  (SREQ1, SREQ2, SREQ 2.1, SREQ3) |
| SREQ2 | The system should allow existing users to login. | ★★★★★ | 15 days  (SREQ1, SREQ2, SREQ 2.1, SREQ3) |
| SREQ2.1 | The system should allow new users to sign up. | ★★★★★ | 15 days  (SREQ1, SREQ2, SREQ 2.1, SREQ3) |
| SREQ3 | The system should keep every user’s login information confidential. | ★★★★★ | 15 days  (SREQ1, SREQ2, SREQ 2.1, SREQ3) |
| SREQ4 | The system should be able to send an automated confirmation E-mail upon completion of creating a new account. (Not implemented) | ★★☆☆☆ | 5 days |
| SREQ5 | The system should get restaurant data based on users’ search and list results based on total average ratings (from highest to lowest). | ★★★★★ | 20 days |
| SREQ6 | The system should have a help page, about page, FAQ page. | ★☆☆☆☆ | 3 days |
| SREQ7 | The system should have a way of tracking user preferences.  (admin delete restaurant, delete bad reviews, add restaurant, find password, etc) | ★☆☆☆☆ | 3 days |
| SREQ8 | The system should have a welcome page that allows users to login/sign up. | ★☆☆☆☆ | 3 days |
| SREQ9 | System should allow user to view restaurant information(Name,Address,Operation hours,Average rating, restaurant types, and user reviews) | ★★★★☆ | 20 days  (SREQ9, SREQ10) |
| SREQ10 | In addition to SREQ9, each restaurant should displays its operation hours. | ★☆☆☆☆ | 20 days  (SREQ9, SREQ10) |
| SREQ11 | System should provide users an option of getting password in case they forget their password. | ★☆☆☆☆ | 3 days |
| SREQ12 | System should allow user to view user accounts(others+self) | ★★★★☆ | 15 days |
| SREQ13 | System should allow users to add reviews to a particular restaurant | ★★★★★ | 15 days |

Part B: User stories

|  |  |  |
| --- | --- | --- |
| **ID** | **User Stories** | **Priority Weight**  **(in a scale of 1 to 5 stars)** |
| UST1 | As an administrator I should be able to access the user account data. | ★★★★★ |
| UST2 | As an administrator I should be able to delete a user account. | ★☆☆☆☆ |
| UST3 | As a potential user I should be able to make an account. | ★★★★★ |
| UST4 | As a user I should be able to access my account. | ★★★★★ |
| UST5 | As a user I should be able to search restaurants by names. | ★★★★★ |
| UST6 | As a user I should be able to search restaurants by types. | ★★★★★ |
| UST7 | As a user I should be able to post ratings for restaurants. | ★★★★★ |
| UST8 | As a user I should be able to see my own rating history. | ★★★☆☆ |
| UST9 | As a user my interaction with the application should be easy and appealing. | ★★☆☆☆ |
| UST10 | As an administrator I should be able to delete restaurant, add restaurant, delete unwanted and non-sense ratings. | ★★☆☆☆ |
| UST11 | As a user I should be able to see restaurants’ address and ratings. | ★★★★☆ |
| UST12 | As a user I should be able to see each restaurant’s operation hour. | ★★★☆☆ |
| UST13 | As a user I should be able to save restaurants to my favorite list (Not implemented). | ★☆☆☆☆ |
| UST14 | As a user i should be able to have an option of either making my profile private or public (Not implemented). | ★☆☆☆☆ |
| UST15 | As a user i should be able to see profiles and rating history of other users, if allowed. | ★☆☆☆☆ |
| UST16 | As a user I should be able to find account password. | ★☆☆☆☆ |
| UST17 | As a user I should be able to post reviews for restaurants. | ★★★★★ |

Use Cases

|  |  |  |
| --- | --- | --- |
| **Actor** | **Actor’s Goal** | **Use Case Name** |
| User | To be able to sign up as a new user. | signUp (UC-1) |
| User | To be able to log in as a previous user. | logIn (UC-2) |
| Admin | To be able to log in. | logIn (UC-2) |
| Admin | To remove user accounts | AdminProfilePage (UC-3) |
| Admin | To remove/add restaurant info data | adminDeleteRes(UC-4)  addRestaurant(UC-13) |
| Admin | To remove bad review | adminDeleteReview(UC15) |
| User | To search restaurants by types, and have the option to list either from high to low ratings, or from low to high ratings. | MainSearch(UC-5) |
| User | To search restaurants by name, and have the preference either from low to high ratings, or from high to low ratings. | MainSearch(UC-5) |
| Admin | To search restaurants by types, and have the option to list either from high to low ratings, or from low to high ratings. | MainSearch(UC-5) |
| Admin | To search restaurants by name, and have the preference either from low to high ratings, or from high to low ratings. | MainSearch(UC-5) |
| User | To inspect past rating history in profile page | ProfilePage(UC-12) |
| System | To control and authenticate user signup and user logins | signUp (UC-1)  logIn(UC-2) |
| System | To send confirmation email to users upon signup completion | sendConfirmation(UC-6) |
| System | To display restaurant data result based on user search and preference | MainSearch(UC-7) |
| System | To display individual restaurant page info | RestaurantProfile(UC-11) |
| System | To have a help, FAQ, feedback page. | FAQ(UC-8) |
| System | To have a welcome page | welcomePage(UC-9) |
| User | To get forgotten passwords | ForgetPassword(UC-10) |
| Admin | To get forgotten passwords | ForgetPassword(UC-10) |
| User/Admin | To post reviews for a particular restaurant. | addNewReview(UC-14) |

|  |  |
| --- | --- |
| **Use Case UC-1** | **SignUp** |
| Related Requirements: | SREQ1, SREQ2, SREQ2.1, SREQ4 |
| Initiating Actor: | Potential User. |
| Actor’s Goal: | To Sign Up, set username, password, first name, last name, security question and security class. |
| Participating Actors: | User, UserProfile (class). |
| Preconditions: | The User wants to make a profile, the user does not know what username exists in the user database. |
| Postcondition: | Either the user receives an error because the username already exists or signup is complete. |
| Flow of Events for Success: | → The actor provokes the signUp method in the Userprofile class.  ← The system recognizes this and asks for username, password, first name, last name, email, security question and answer.  → The actor receives “Sign Up successful”. |
| Flow of Other Scenario: | → The actor provokes the signUp method in the Userprofile class.  ← The system recognizes this and asks for username, password, first name, last name, email.  → The User receives an error due to conflicting username. |

|  |  |
| --- | --- |
| **Use Case UC-2** | **LogIn** |
| Related Requirements: | SREQ1, SREQ2, SREQ3, SREQ4 |
| Initiating Actor: | User, Admin |
| Actor’s Goal: | Log In to the system |
| Participating Actors: | User, Admin, AdminProfile (class), UserProfile (class). |
| Preconditions: | The user/admin wants to log in. The profiles are in the system. |
| Postcondition: | Either the user receives a success message or an error message |
| Flow of Events for Success: | → The actor provokes the logIn method in the Userprofile/Adminprofile class.  ← The system recognizes this and asks for username, password.  ← The system compares profile information and makes sure the data matches  → The actor receives “Log in successful”. |
| Flow of Other Scenario: | → The actor provokes the logIn method in the UserProfile/AdminProfile class.  ← The system recognizes this and asks for username, password.  ← The system compares profile information and makes sure data matches.  → The User receives an error due to conflicting username or password. |

|  |
| --- |
| **UC-3 removeUser (Not used in the project)**  Related Requirement: SREQ-1  Initiator Actor: Admin  Actor’s Goal: to remove user accounts if necessary  Participating actor: Admin, System  Precondition: The system authenticate Admin to login                       The user accounts/password/emails are stored in the database.                        The admins get the right to modify the database once successfully authenticated.  Postcondition: The deleted user account no longer exist in the database, thus no longer can be logged in.  Flow of Events for main success Scenario:  **->Admin** authenticates and gets the permission to modify the database. *Login UC1*  **<-System** get the “signal” to change the database, deleting the targeted user account from the database. |

|  |
| --- |
| **UC-4 adminDeleteRes**  Related Requirement: SREQ 1, SREQ 2 and SREQ3  Initiator Actor: System  Actor’s Goal: to remove restaurant's if necessary  Participating actor: System  Precondition: The system authenticate Admin to login                       The restaurant infos are stored in the database.                        The admins get the right to modify the database once successfully authenticated.  Postcondition: The restaurant no longer exist in the database, thus no longer can be searched and displayed when user search for results.  Flow of Events for main success Senario:  **->Admin** authenticates and gets the permission to modify the database. *Login UC1*  **<-System** get the “signal” to change the database, deleting the targeted restaurant information from the database. |

|  |
| --- |
| **UC-5 MainSearch**  Related Requirement: SREQ 5, 7  Initiator Actor: Admin/Users  Actor’s Goal: to search for restaurants based on preference ( type/name/ranking)  Participating actor: Admin, System, Users  Precondition: The system authenticate Admin/users to login                       The restaurant infos are stored in the database.                        The admins/users search for restaurants based on restaurant names or types, and                   having the option to either display ranking from high to low, or low to high. The system generates the result based on either names/types, and the preference of ranking from admin/users.  Postcondition: The system displays the search result  Flow of Events for main success Senario:  **->System** authenticates **Admins/Users**. *Login UC1*  *->***System** get admin/user search preferences and generates the results  **<-System** get the “signal” to display desired results. |

|  |
| --- |
| **UC6 sendConfirmation**  Related Requirement: SREQ 4  Initiator Actor: Admin/Users  Actor’s Goal: to get a confirmation email from the system once signUp for new accounts completed  Participating actor: Admin, System, Users  Precondition: The admin/user accounts are stored in the database/system.                       The system get the signal to check if the prompted user information already exist.  Postcondition: The system sends a confirmation to the user email. The user can now login using the newly register information.  Flow of Events for main success Senario:  **->System** get the sigal to signup. Signup *UC1*  *->***Sytstem** get **admin/user** prompted information  **->System** creates a new account and add on to the database  <- If successful, the **System** sends the **admin/user** a confirmation email.  , |

|  |
| --- |
| **UC7 DisplayResRank (\*\*modiefied\*\*Is integrated/combined with UC5 MainSearch)**  Related Requirement: SREQ 9 ,10  Initiator Actor: Admin/Users  Actor’s Goal: to display restaurant based on preference ( which is based on type/name/ranking)  Participating actor: Admin, System, Users  Precondition: The system authenticate Admin/users to login                       The restaurant infos are stored in the database.                        The admins/users search for restaurants based on restaurant names or types, and having the option to either display ranking from high to low, or low to high  Postcondition: The system returns the result based on either names/types, and the preference of ranking from admins/users.  Flow of Events for main success Senario:  **->System** authenticates **Admins/Users**. *Login UC1*  *->***Sytstem** get admin/user preferences  **<-System** get the “signal” to display desired results. |

|  |
| --- |
| **UC-8** **FAQ**  Related Requirement: SREQ-6  Initiator Actor: User/Admin  Actor’s Goal: To access a help, FAQ, feedback page.  Participating actor: Admin,User,System  Precondition: The System needs to authenticate that it’s a true profile  Postcondition: A page gets opened that has frequently asked questions and answers  Flow of Events for main success Senario:  ->User/Admin requests for help  <-system returns a help page |

|  |
| --- |
| **UC-9 welcomePage**  Related Requirements: SREQ-7  Initiator Actor: System  Actor’s Goal: To have a hard coded welcome page for the profiles.  Participating actor: System  Precondition: before login so there is no precondition other than just starting the program  Postcondition: Display a page that has signin and signup, forgetPassword capabilities. The startup page is what the user/admin use to gain access to the application but there is no interaction until the user/admin initiates it.  Flow of Events for main success Senario:  <-System shows a welcome page |

|  |
| --- |
| **UC-10 ForgetPassword**  Related Requirements: SREQ-11  Initiator Actor: System  Actor’s Goal: to find the password of the account  Participating actor: User/Admin  Precondition: A validation must be made to authenticate that it is the actual user via email or question.  PostCondition: The System will go into the user database and get the users/admins password  Flow of Events for main success Senario:  ->User/Admin requests to find account password  <-System prompts user for username and email  ->User/Admin passes verification  <-System returns the security Question  ->User passes the verification of the security question  <-System returns his/her password database from the profile database |

|  |
| --- |
| **UC-11 RestaurantProfile**  Related Requirement: SREQ 9,10  Indicated Actor: User/Admin  Actor’s goal: to view specific information about an interested restaurant  Participating actor: User/Admin/System  Precondition: user/admin are authenticated.                       The restaurant infos are stored in the database.                       User/admin clicked restaurant data information are stored in the database.  Postcondition: System displays restaurant information(Restaurant name, Restaurant address, restaurant contact, restaurant operation hours, restaurant review, etc on GUI.  Flow of Events for main success Senario:  ->User/Admin clicked into a restaurant from the restaurant result list.  <-system gets the query and displays restaurant info in a separate activity page, with all information. |

|  |
| --- |
| **UC-12 ProfilePage**  Related Requirement: SREQ 12  Indicated Actor: User/Admin  Actor’s goal: to view specific information about a user  Participating actor: User/Admin/System  Precondition: User/admin are authenticated.                        The profile infors and the user/admin clicked user profile info are stored in the user database.  Postcondition: System displays profile information (Profile name, profile type, and history review history) on GUI.  Flow of Events for main success Senario:  ->User/Admin clicked into a username from review posts on a restaurant page  <-System gets the query and displays the user profile info on a separate activity page |

|  |
| --- |
| **UC-13  addRestaurant**  Related Requirement: SREQ 1, SREQ 2 and SREQ3  Initiator Actor: System  Actor’s Goal: to add restaurant's if necessary  Participating actor: System  Precondition: The system authenticate Admin to login                       The restaurant infos are stored in the database.                        The admins get the right to modify the database once successfully authenticated.  Postcondition: The restaurant database is updated with newly added restaurant information.  Flow of Events for main success Senario:  **->Admin** authenticates and gets the permission to modify the database. *Login UC1*  **<-System** get the “signal” to change the database, adding the targeted restaurant information from the database. |

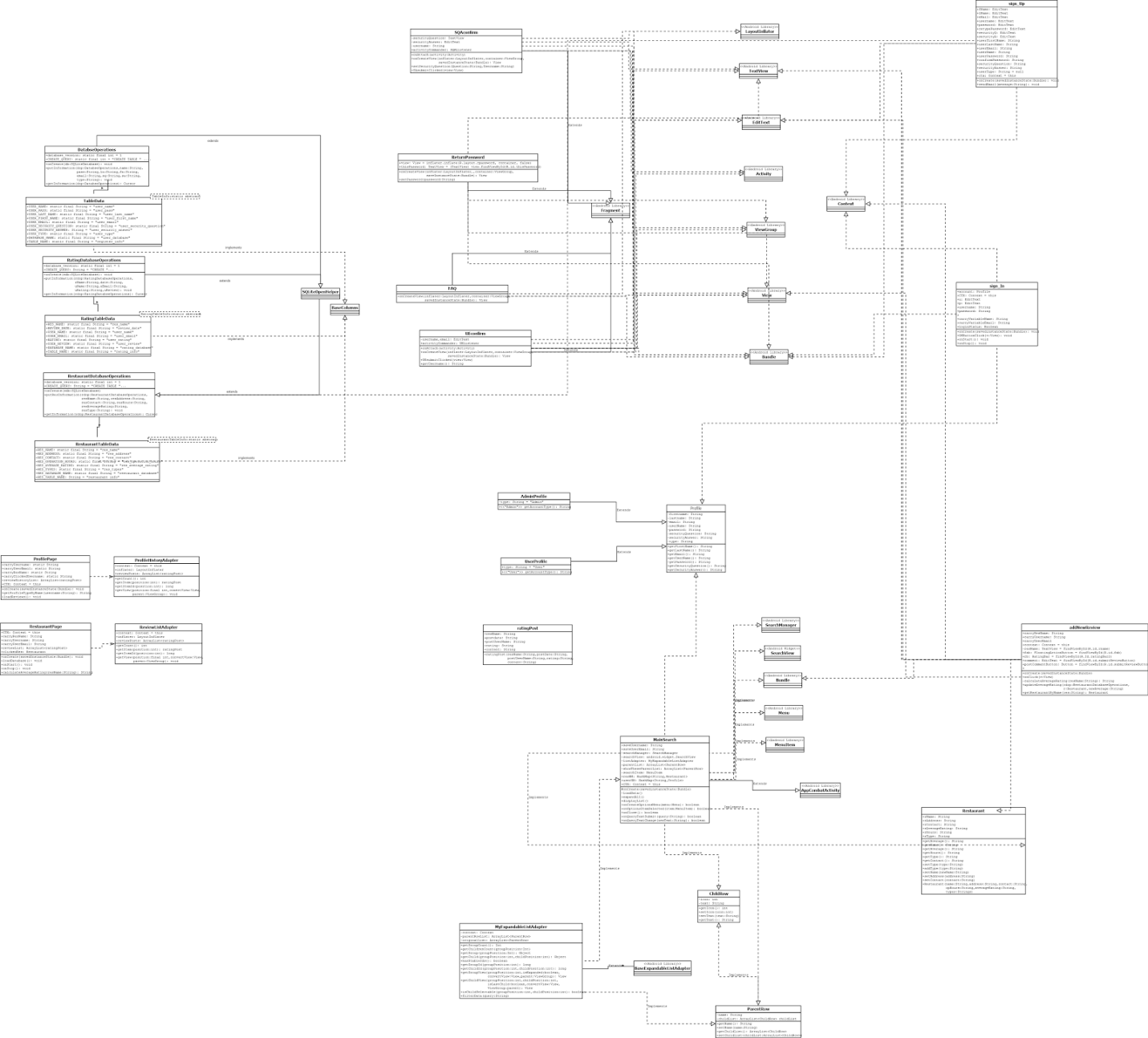
|  |
| --- |
| **UC-14 addNewReview**  Related Requirement: SREQ1, SREQ2, and SREQ13  Initiator Actor: Users/admins  Actors’s goal: to add reviews for restaurants  Participating actor: System,Users  Precondition: The restaurant and review infos are stored in the database.                       The users get the right to post new ratings for restaurants.  Postcondition: The new review is added to the review database.                         The restaurant average rating is being recalculated and modified in the resturant database.  Flow of Events for main success Senario:  ->Users/admins initiates by clicking adding new reviews on a particular restaurant profile page.  <-System get the “signal” to add new reviews, adding new reviews into the review database.  <-System updates the new average rating for this particular restaurant. |

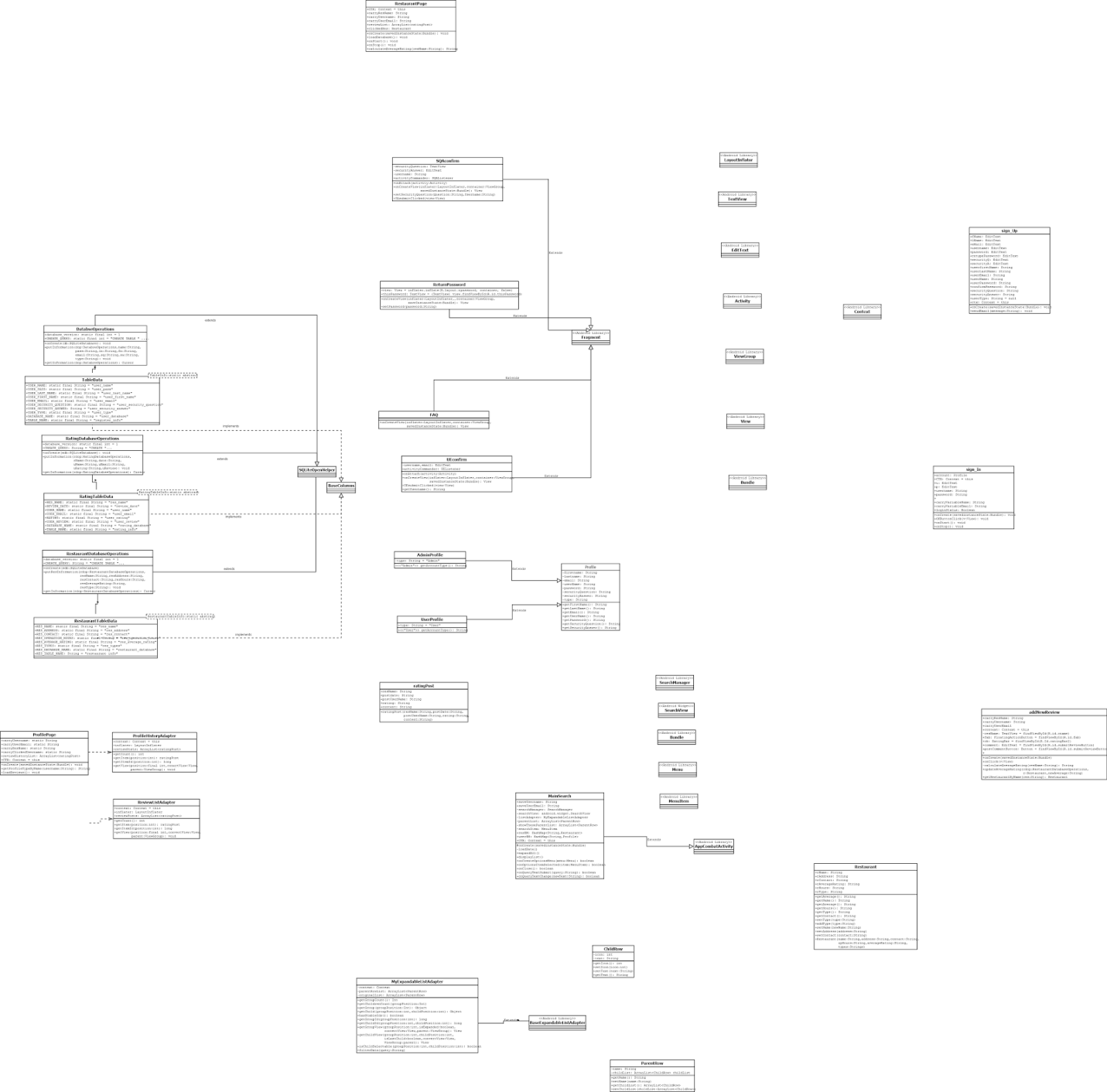
|  |
| --- |
| **UC15 - adminDeleteReview**  Related Requirement: SREQ1, SREQ5, and SREQ7  Initiator Actor: Admins  Actors’s goal: to delete bad reviews for restaurants  Participating actor: System, Admind  Precondition: The restaurant and review infos are stored in the database.                       The admin get the right to delete reviews.  Postcondition: The new review is deleted from the review database.                         The restaurant average rating is being recalculated and modified in the restaurant database.  Flow of Events for main success Scenario:  ->Admins initiates by clicking adding new reviews from the Setting page.  <-System get the “signal” to remove new reviews, thus removing selected reviews from the review database.  <-System updates the new average rating for this particular restaurant. |

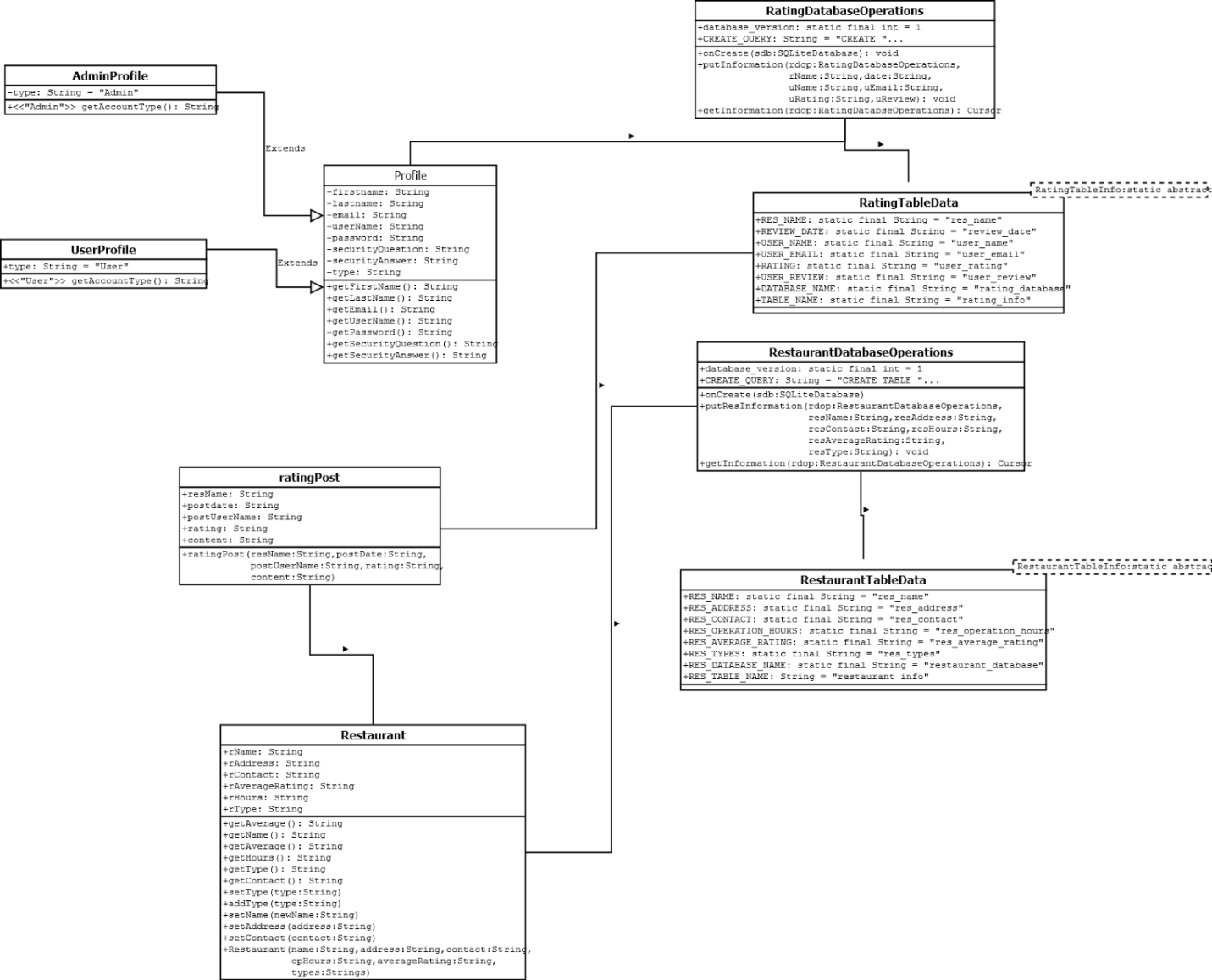
**Traceability Matrix**

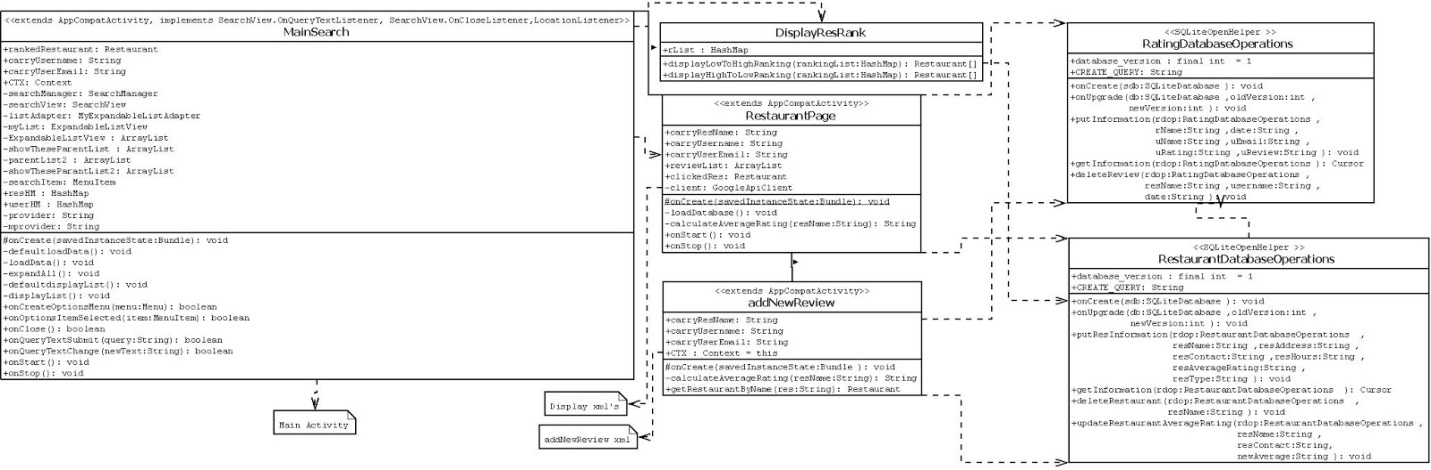
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Requirements** | **PW** | **UC1** | **UC2** | **UC3** | **UC4** | **UC5** | **UC6** | **UC7** | **UC8** | **UC9** | **UC10** | **UC11** | **UC12** | **UC13** | **UC14** | **UC15** |
| **SREQ 1** | 5 |  |  | ✔ | ✔ |  |  |  |  |  | ✔ |  |  | ✔ |  | ✔ |
| **SREQ 2** | 5 | ✔ | ✔ |  | ✔ |  |  |  |  |  |  |  |  | ✔ |  |  |
| **SREQ 3** | 5 | ✔ | ✔ |  | ✔ |  |  |  |  |  |  |  |  | ✔ |  |  |
| **SREQ4** | 2 |  |  |  |  |  | ✔ |  |  |  |  |  |  |  |  |  |
| **SREQ5** | 5 |  |  |  |  | ✔ |  |  |  |  |  |  |  |  |  | ✔ |
| **SREQ6** | 1 |  |  |  |  |  |  |  | ✔ |  |  |  |  |  |  |  |
| **SREQ7** | 1 |  |  |  |  | ✔ |  |  |  |  |  |  |  |  |  | ✔ |
| **SREQ8** | 1 |  |  |  |  |  |  |  |  | ✔ |  |  |  |  |  |  |
| **SREQ9** | 4 |  |  |  |  | ✔ |  | ✔ |  |  |  | ✔ |  |  |  |  |
| **SREQ10** | 3 |  |  |  |  | ✔ |  | ✔ |  |  |  | ✔ |  |  |  |  |
| **SREQ11** | 1 |  |  |  |  |  |  |  |  |  | ✔ |  |  |  |  |  |
| **SREQ12** | 5 |  |  |  |  |  |  |  |  |  |  |  | ✔ |  |  |  |
| **SREQ13** | 5 |  |  |  |  |  |  |  |  |  |  |  |  |  | ✔ |  |

Class Diagram



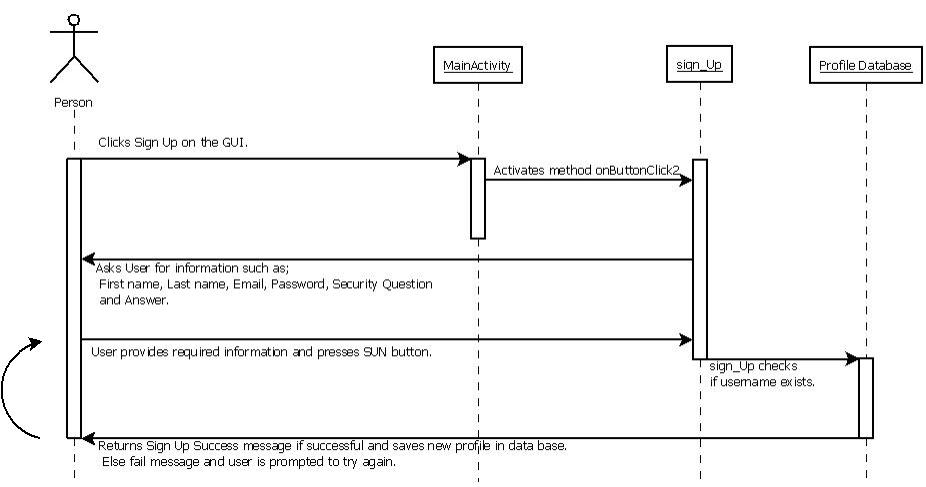




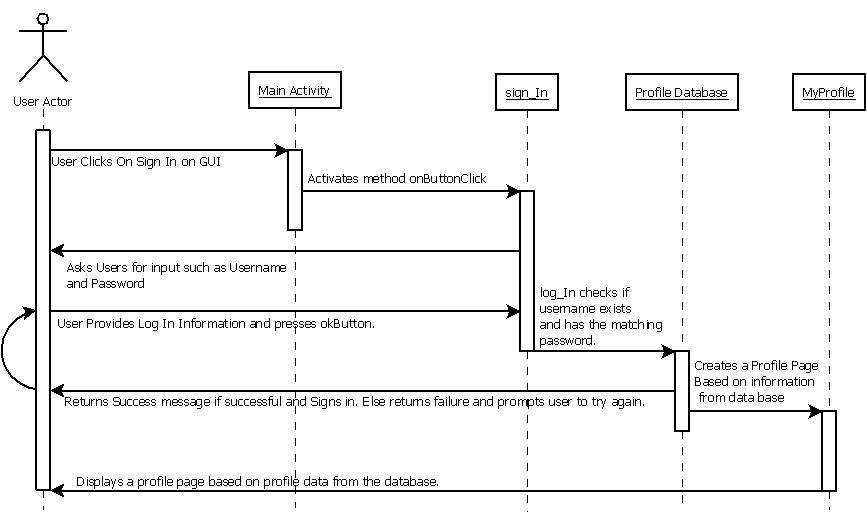


Sequence Diagrams

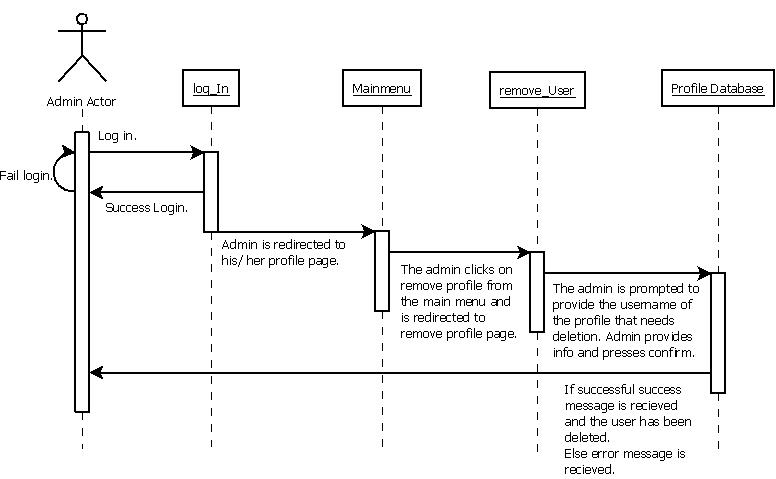
UC1- Sign Up



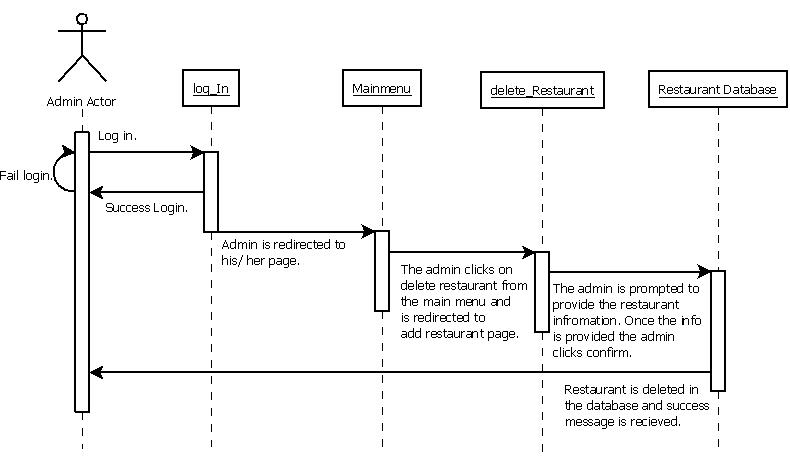
UC2- Log In



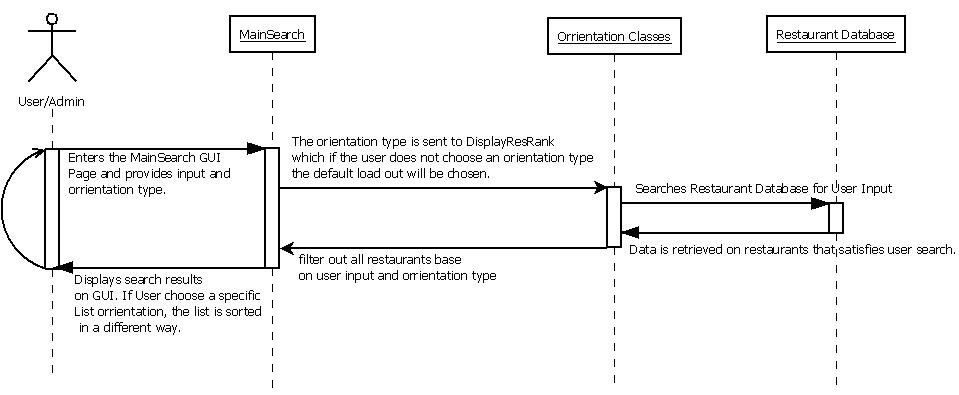
UC3- Remove User



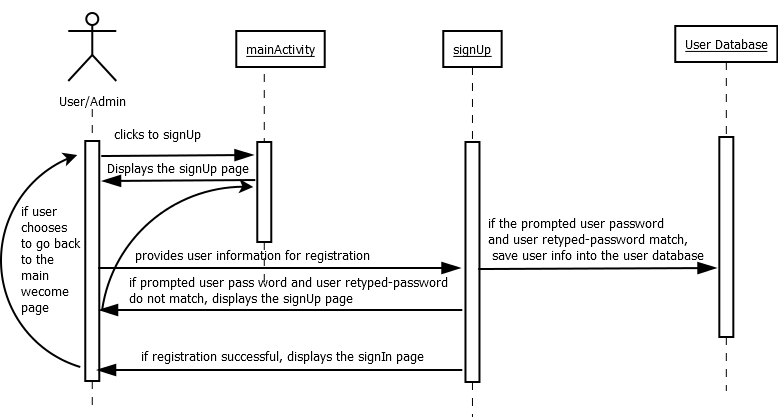
UC4- deleteRestaurant



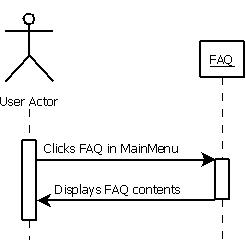
UC5- Search



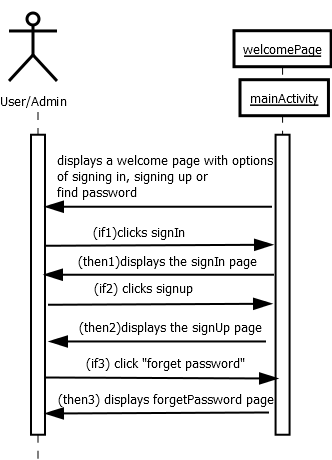
UC6-send Confirmation(not successfully implemented yet)



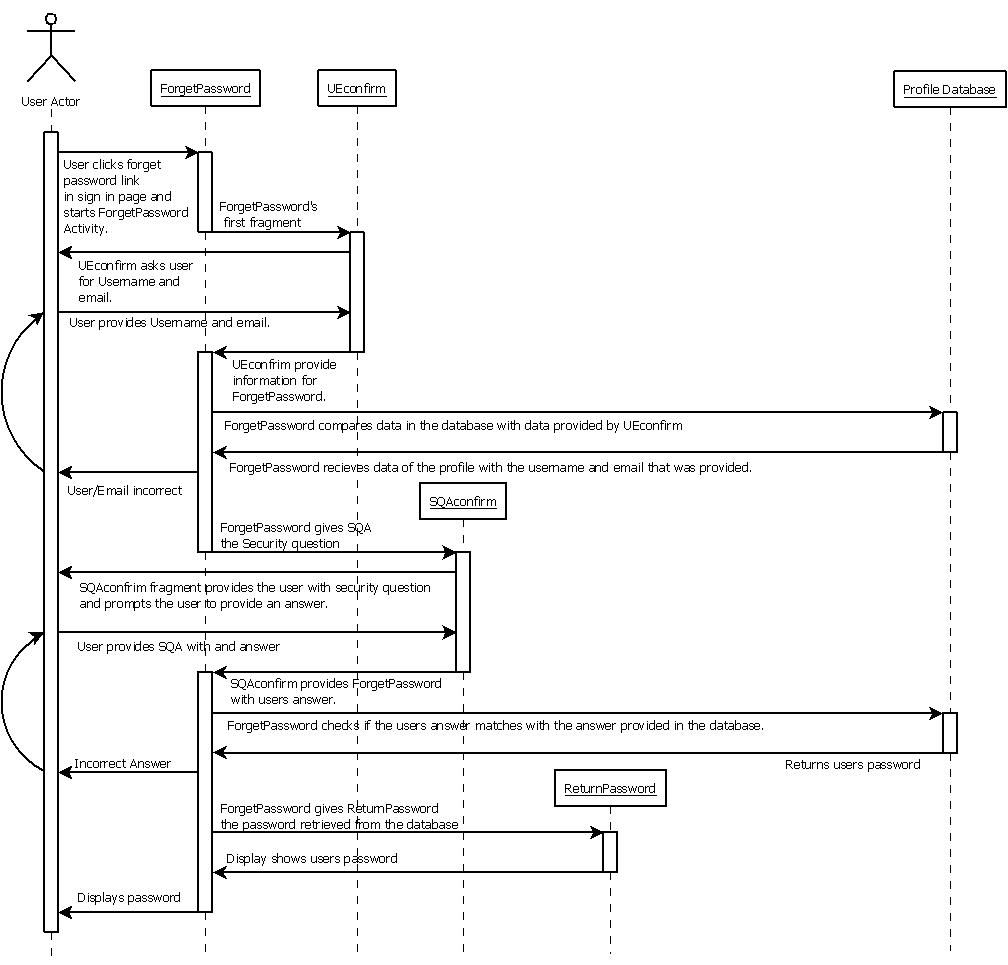
UC8- FAQ



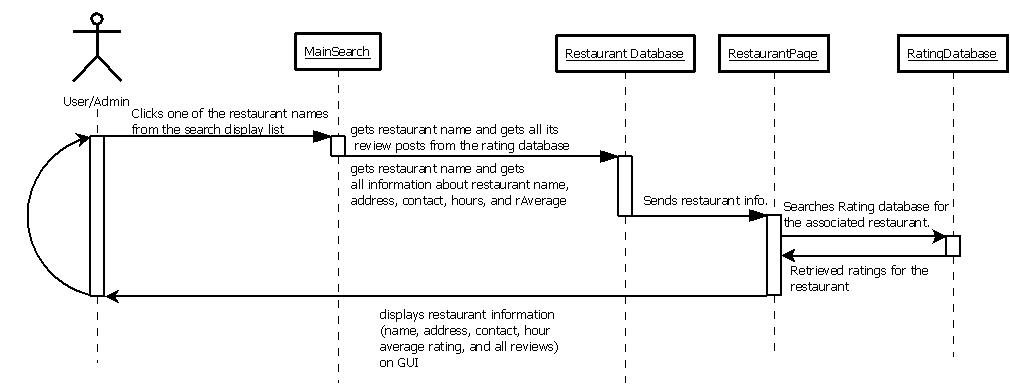
UC9-welcome Page



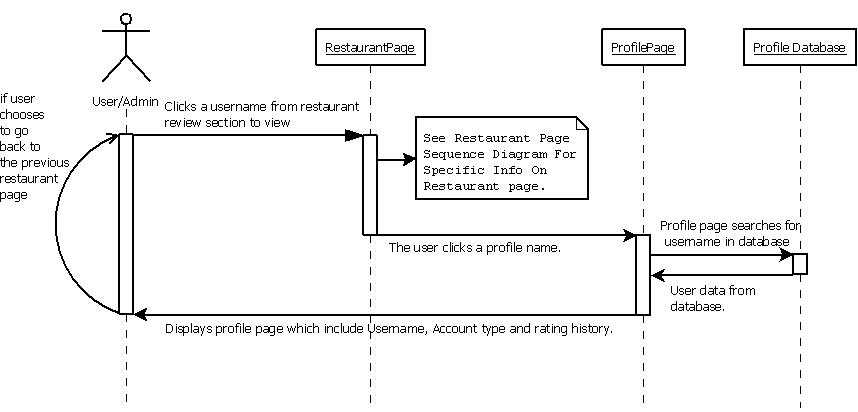
UC10- ForgetPassword



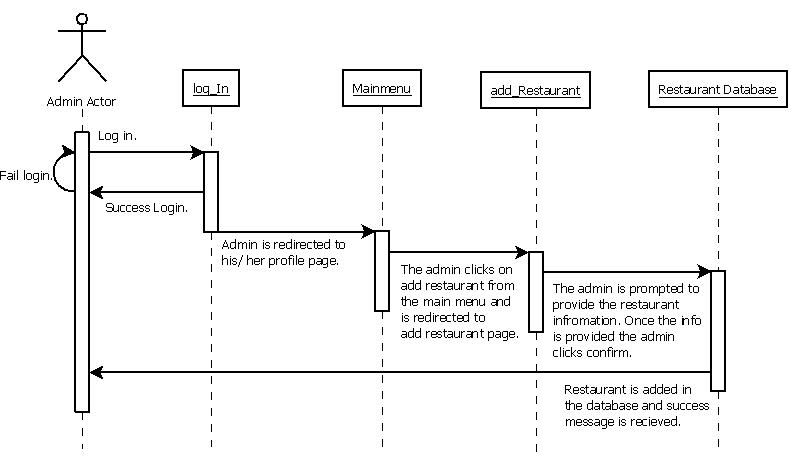
UC11-viewRestaurantPage



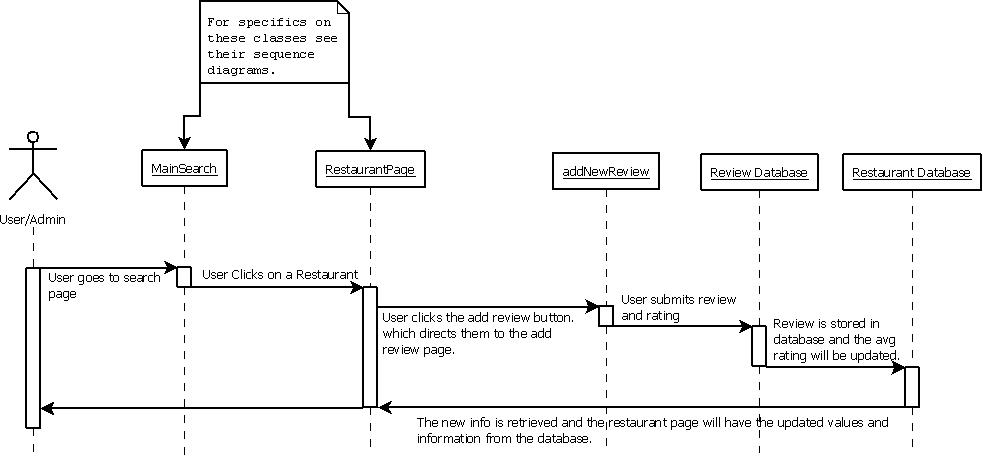
UC12-viewProfilePage



UC13- addRestaurant



UC14-addNewReview



UC 15- DeleteReview

