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 either with or without users’ current locations. 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 in the area 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. | ★★☆☆☆ | 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, and a feedback page. | ★☆☆☆☆ | 3 days |
| SREQ7 | The system should have a way of tracking user preferences. | ★☆☆☆☆ | 3 days |
| SREQ8 | The system should have a welcome page that displays app logo for a few seconds before allowing users to login/sign up. | ★☆☆☆☆ | 3 days |
| SREQ9 | Each restaurant should have a separate page that displays its address and user ratings. | ★★★★☆ | 20 days  (SREQ9, SREQ10) |
| SREQ10 | In addition to SREQ9, each restaurant should displays its menu and operation hours. | ★★☆☆☆ | 20 days  (SREQ9, SREQ10) |
| SREQ11 | System should provide users an option of setting up new password in case they forget their password. | ★☆☆☆☆ | 3 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 (fetch restaurants data nearby with or without user current location). | ★★★★★ |
| UST6 | As a user I should be able to search restaurants by types (fetch restaurants data nearby with or without user current location). | ★★★★★ |
| UST7 | As a user I should be able to post ratings for restaurants (text & images). | ★★★★★ |
| 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 user unwanted and non-sense ratings. | ★★☆☆☆ |
| UST11 | As a user I should be able to see restaurants’ address (displaying on the map) and ratings. | ★★★★☆ |
| UST12 | As a user I should be able to see each restaurant’s operation hour and menu. | ★★★☆☆ |
| UST13 | As a user I should be able to save restaurants to my favorite list. | ★☆☆☆☆ |
| UST14 | As a user i should be able to have an option of either making my profile private or public. | ★☆☆☆☆ |
| 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 change account password. | ★☆☆☆☆ |

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 | removeUser (UC-3) |
| Admin | To add and remove restaurant info data | saveRestaurant(UC-4) |
| User | To search restaurants by types, and have the option to list either from high to low ratings, or from low to high ratings. | search(UC-5) |
| User | To search restaurants by name, and have the preference either from low to high ratings, or from high to low ratings. | search(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. | search(UC-5) |
| Admin | To search restaurants by name, and have the preference either from low to high ratings, or from high to low ratings. | search(UC-5) |
| User | To inspect past rating history | inspectRatingHistory(UC-11) |
| 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 | DisplayResRank(UC-7) |
| System | To have a help, FAQ, feedback page. | helpPage(UC-8) |
| System | To have a welcome page | welcomePage(UC-9) |
| User | To change passwords | changePassword(UC-10) |
| Admin | To change passwords | changePassword(UC-10) |

|  |  |
| --- | --- |
| **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. |
| 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.  → 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 or email. |

|  |  |
| --- | --- |
| **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**  Related Requirement: SREQ-1  Initiator Actor: Admin  Actor’s Goal: to remove user accounts if necessary  Participanting 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 Senario:  **->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 saveRestaurant**  Related Requirement: SREQ 2 and SREQ3  Initiator Actor: Admin  Actor’s Goal: to remove restaurant's) if necessary  Participating actor: Admin, 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 search**  Related Requirement: SREQ 5, 7, 9 ,10  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**  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** **Help**  Related Requirement: SREQ-6  Initiator Actor: User  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 |

|  |
| --- |
| **UC-9 welcomePage**  Related Requirements: SREQ-7  Indicator 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 login/createUser capabilities. The startup page is what the user/admin use to gain access to the website but there is no interaction until the user/admin initiates it. |

|  |
| --- |
| **UC-10 changePassword**  Related Requirements: SREQ-11  Indicated Actor: System  Actor’s Goal: to change 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 csv and change the users/admins password |

**Traceability Matrix**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Requirements** | **PW** | **UC1** | **UC2** | **UC3** | **UC4** | **UC5** | **UC6** | **UC7** | **UC8** | **UC9** | **UC10** |
| **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 |  |  |  |  |  |  |  |  |  | ✔ |