**Use Case Descriptions**

### Use case-01: Login

The use case description for the login process is shown in Table 1.

Table 1: Use case description-01: Login

|  |  |
| --- | --- |
| **ID:** | 1 |
| **Name of Use Case:** | Login |
| **Actors:** | Admin, User, Service Provider, Database |
| **Description:** | The admin/user can log in to the application. |
| **Pre-Condition:** | Admin/User should’ve their account registered beforehand. |
| **Post-Condition:** | Admin/User has been signed into the application, successfully. |
| **Events:** | 1. Admin/user opens the app. 2. Admin/user clicks on the login / sign in button. 3. Admin/user fills the required fields (username and password) 4. After the authentication has been done, user will be redirected to dashboard |
| **Alternatives Flow:** | In case of forget password, user will be provided with the reset link by the app. |
| **Exceptions:** | None |

### Use case-02: Sign Up

The use case description for the sign up process is shown in Table 2.

Table 2: Use case description-02: Sign Up

|  |  |
| --- | --- |
| **ID:** | 2 |
| **Name of Use Case:** | Sign Up |
| **Actors:** | User, Database |
| **Description:** | User/ service provider can create their respective account for application. |
| **Pre-Condition:** | No account of that credentials already exist. |
| **Post-Condition:** | User account has been created and they are logged in |
| **Events:** | 1. user opens the app. 2. user clicks on the sign up button. 3. user fills the required fields of the form 4. After the sign up procedure has been done, user will be redirected to dashboard |
| **Alternatives Flow:** | User already has an account. |
| **Exceptions:** | None |

### Use case-03: Logout

The description of the use case of the logout process is shown in Table 3.

Table 3: Use case description-03: Logout

|  |  |
| --- | --- |
| **ID:** | 3 |
| **Name of Use Case:** | Logout |
| **Actors:** | Admin, User |
| **Description:** | The user/admin can logout from the application. |
| **Pre-Condition:** | The user/admin should be signed in his/her account. |
| **Post-Condition:** | user/admin has been successfully logged out from the application. |
| **Events:** | 1. The user presses the logout button. 2. The user logs out and the login page appears. |
| **Alternatives Flow:** | The user has already logged out. |
| **Exceptions:** | None |

### Use case-04: Model Re-Training

The use case description for retraining the model is shown in Table 4.

Table 4: Use case description-04: Re-Training the model

|  |  |
| --- | --- |
| **ID:** | 4 |
| **Name of Use Case:** | Re-training the model |
| **Actors:** | Database, Google Map API |
| **Description:** | Application shall get the model from the database and then retrain it. |
| **Pre-Condition:** | Database must exist with the model. |
| **Post-Condition:** | Model has been retrained. |
| **Events:** | 1. Timeout has been occurred for retraining of the model 2. Database will feed in the updated dataset with latest reviews and comments to model 3. Application has been updated accordingly after the retraining |
| **Alternatives Flow:** | Timeout has not been occurred for retraining phase |
| **Exceptions:** | None |

### Use case-05: User Portal

The use case description for the User Portal is shown in Table 5.

Table 5: Use case description-05: User Portal

|  |  |
| --- | --- |
| **ID:** | 5 |
| **Name of Use Case:** | User Portal |
| **Actors:** | User, Database, Google Map API, Trained Model |
| **Description:** | User can manage his profile , view profiles of the places , add places to his/her favorites section, give feedback in the form of reviews and comments, get recommendations |
| **Pre-Condition:** | User has been logged in. |
| **Post-Condition:** | User has carried out its desire goal |
| **Events:** | 1. User opens the app. 2. User logs in 3. User chooses either of the task he wants to perform 4. User performs the task by following the valid set of operations |
| **Alternatives Flow:** | In case of forget password, user will be provided with the reset link by the app. |
| **Exceptions:** | User Portal has been deleted due to some reason by admin or by himself accidently. |

### Use case-06: Admin Portal

The use case description for the Admin Portal is shown in Table 6.

Table 6: Use case description-6: Admin Portal

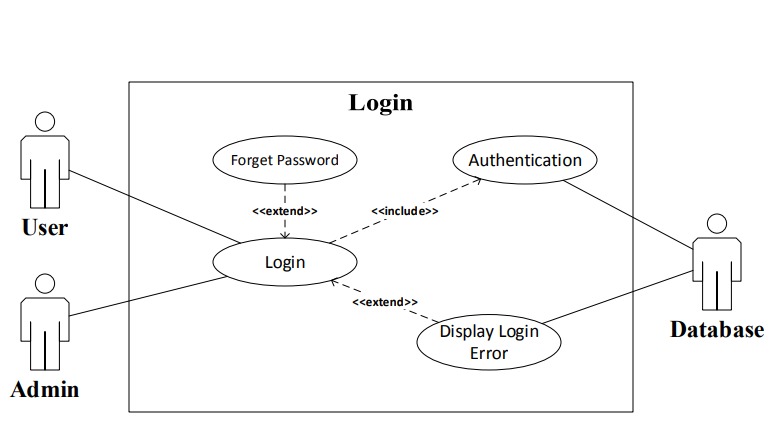
|  |  |
| --- | --- |
| **ID:** | 6 |
| **Name of Use Case:** | Admin Portal |
| **Actors:** | Admin, Database |
| **Description:** | Admin can approve the request for service provider accounts and can manage app users either by deleting or updating. |
| **Pre-Condition:** | Admin has logged in or admin is a valid admin |
| **Post-Condition:** | Admin has carried out its desire goal |
| **Events:** | 1. Admin opens the app. 2. Admin logs in 3. Admin chooses either of the task he wants to perform 4. Admin performs the task by following the valid set of operations |
| **Alternatives Flow:** | None |
| **Exceptions:** | Admin is not a valid admin |

## 

## Use case design

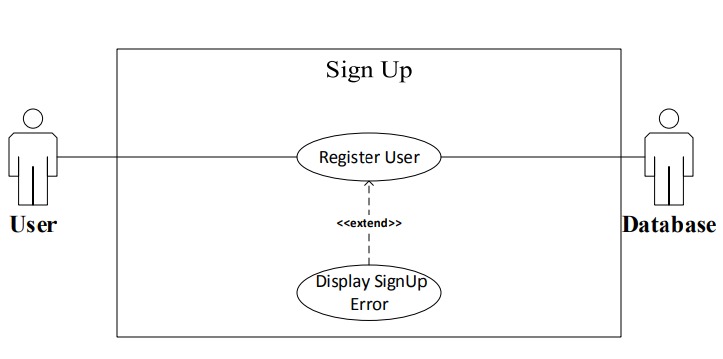
### Use case-01: User and Admin Login

The use case for the login process for the user and admin is shown in Figure 1



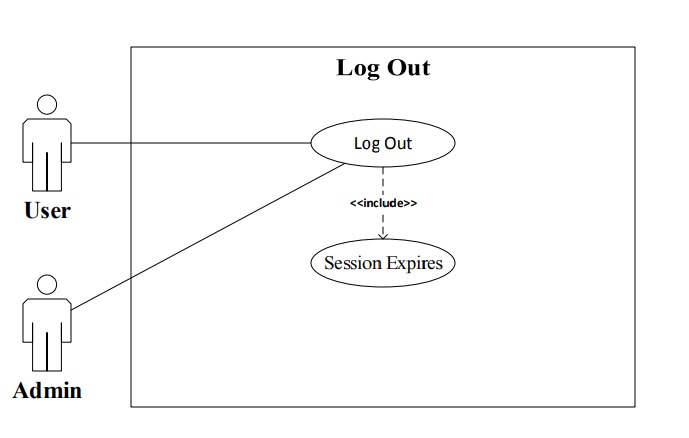
### Use case-02: User Sign Up

The use case for the signup process for the user is shown in Figure 2.



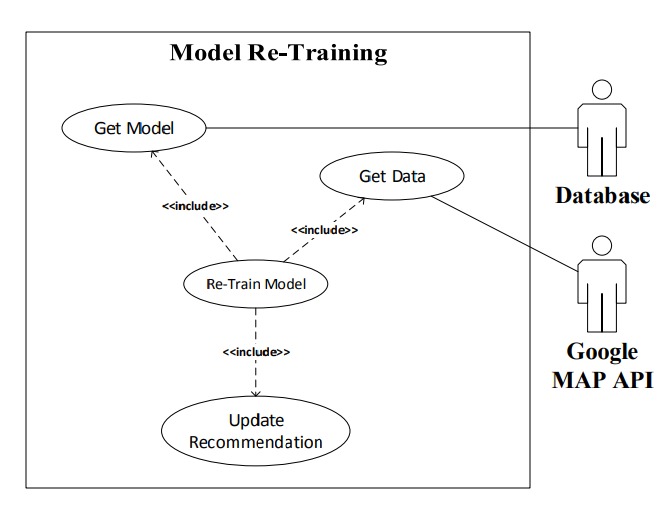
### Use case-03: User and Admin Logout

The use case for the logout process for the user and admin is shown in Figure 3.



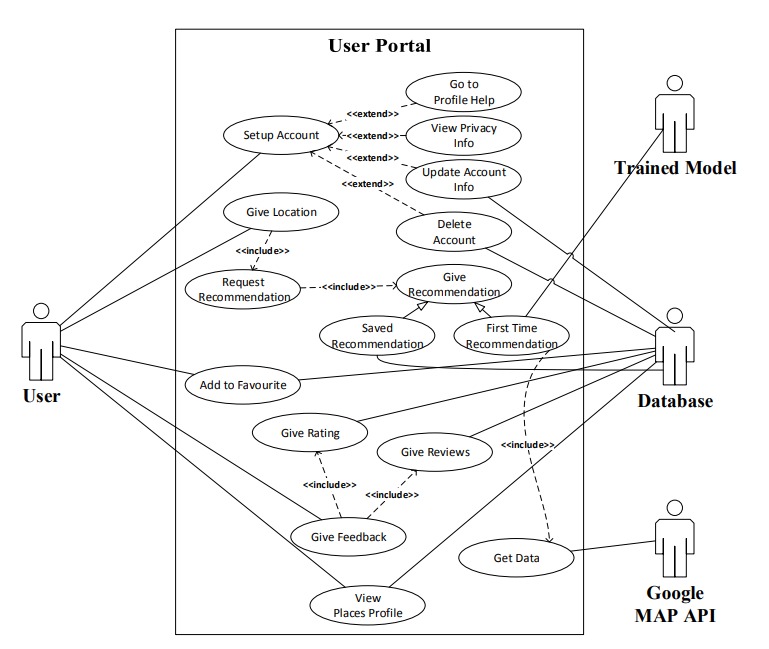
### Use case-04: Model Retraining

The use case for the model training is shown in Figure 4.



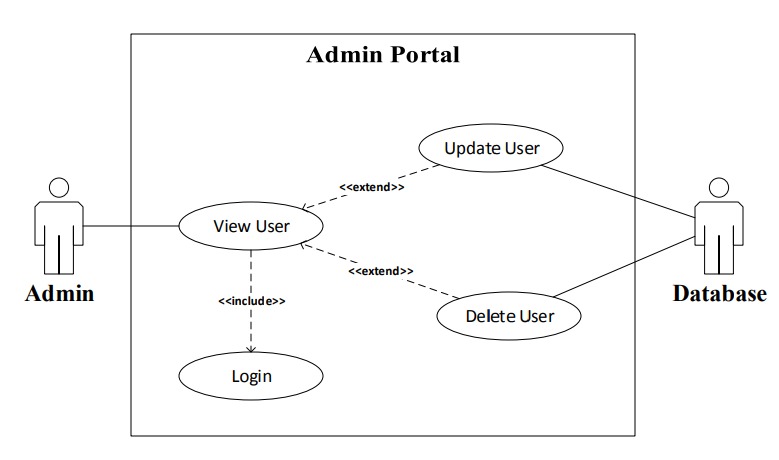
### Use case-05: User Portal

The use case for the user portal shown in Figure 5.



### Use case-06: Admin Portal

The use case for the admin portal is shown in Figure 6.



### Use case-08: Recommendation System (Complete System)

The use case in 8 depicts the overview of the complete recommendation system by combining all modules of the system in one diagram for a better understanding of the viewer.

