**Project Report**

**On**

**Finding An Apartment**

**Mobile Application**

# **Project Co-Ordinator Project Guide**

# **Pargol Poshtareh Richard Grand'Maison**

**Sakkaravarthi ramanathan**

# **Team Members**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Name** | **Student ID** |
| 01 | Venkat Reddy Mallesh | 1993122 |
| 02 | Dinesh Avula | 1992713 |
| 03 | Saiteja Yatam | 1993100 |
| 04 | Srikanth Reddy Marakkagari | 1993120 |
| 05 | Vijay Kumar Reddy Jetty | 1992711 |
| 06 | Navya Manda | 1894204 |
| 07 | Kalyan Teja Danda | 1993115 |

**Project Description**

# The idea is to develop an app to ease the process of finding an apartment for people living anywhere in Québec. The app allows the user to efficiently visualize apartments for rent on a map. The user can then click any apartment to display additional information such as price, address, description, contact information and more. The user can also add a filter so that only apartments in a given price range are displayed. The user can also add apartments to a list to review them later. On the other hand, a renter must create an account if they wish to put an apartment for rent. When logged in, the user has access to an extra feature which allows them to add an apartment on the map with the information mentioned above. Logged in users can also update their profiles for contact information.

# **Specific Requirements**

## **Functional Requirements**

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | **Requirement Statement** | **High/Medium/Low** | **Comments** |
| FR001 | Tenant/Landlord Signup | High | Tenant/Landlord Signup support login Email |
| FR002 | Login | High | It is mandatory to register |
| FR003 | Reset Password/Forget Password | High | Tenant Can reset the password |
| FR004 | Profile | Medium | Tenant can able to update/edit modify user details like name email, phone, address details in the profile |
| FR005 | Add Apartments | High | Landlord can add new apartment profiles |
| FR006 | Edit Apartment | Low | Landlord can update/modify the details of the apartment |
| FR007 | Delete Apartment | Low | Landlord can delete the profile of the apartment |
| FR008 | Inbox | Low | landlord can see the requests made by the users and can contact them |
| FR009 | User Dashboard | Should | Tenant can check the posts made by the Renters for the flat |
| FR010 | Apartment Profile | High | Tenant can check the profile of the desired apartment |
| FR011 | Search Criteria | High | Tenant can search flats based on the user preferences |
| FR012 | Search Location | Medium | Tenant can search the flats using their desired locations |
| FR013 | Search using Price | Medium | Tenant can search the flats using the price range filter |
| FR014 | Search using Apartment type | Medium | Tenant can search the flats based on the type of apartment they are looking for |
| FR015 | Admin View Apartment | Medium | Admin can view the details of the Apartment posted by the Renters |
| FR016 | Admi Verify Apartment Details | High | Admin will verify the details of the apartment that it is real of fake |
| FR017 | Admi Delete Apartment | Medium | Admin can delete the post made by the Renters if the details of the apartments are fake/wrong |
| FR018 | Guest User | Low | Guest user can access application without login |
| FR019 | Favourite list | medium | Tenant can add favourite flats to add list for future purpose. |

## **Non-Functional Requirements**

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement Id | Requirement Statement | High/Medium/Low | Comments |
| NRF 1 | Performance | High | when the user logs in how the application executes various tasks and how instantaneously the next page opens. |
| NRF 2 | Capacity | Low | Capacity is delivering enough functionality required by the end users. |
| NRF 3 | Portability | High | This application runs on Android Devices. |
| NRF 4 | Security | High | Only allows the authorized users to get access to the app either admin or user. |
| NRF 5 | Usability | High | This app design is simple to understand and easy to use even for the person who does not have good knowledge of phones. |
| NRF 6 | Efficiency | High | All the activities work smoothly one after another. |
| NRF 7 | Consistency | High | Like-items should always be displayed and act the same way across the entire application (and even between applications). |
| NRF 8 | Scalability | Medium | It is a non-functional property of a system that describes the ability to appropriately handle increasing (and decreasing) workloads. |

## **Other Requirements**

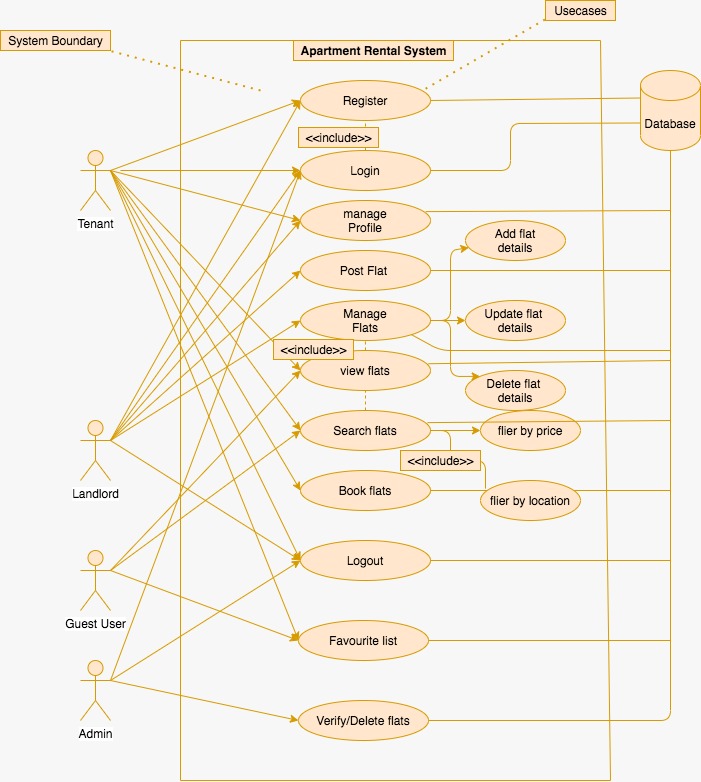
## **Hardware Requirements**

* Android Mobile
* Connector.

## **Software Requirements**

Android Oreo

**Use case Diagram**



**Fully Dressed Use case scenarios**

|  |  |  |
| --- | --- | --- |
| Use Case ID | UC1 | |
| Use Case Name | Registration page | |
| Actor | Tenant | |
| Description | This use case describes how the User enters all the information for the first-time registration to the app. | |
| Triggering event | User selects the **register** link | |
| Preconditions | User must be a new user before registration. | |
| Flow of events | Actor | Flow of events |
| 1. Tenant enters the first name  2. Tenant enters the last name  3. Tenant enters the phone number  4. Tenant enters the email ID.  5. Tenant enters the Username  6. Tenant enters the password.  7.profile pic upload  8. Clicks **Submit** | 1.1. System check the all the information enters by the Tenant will meet the requirements  2.1. System will send all the information to the database  3.1. system saves all the information in the database |
| Post conditions | The Actor see the menu page after successful login. | |
| Exception - Conditions | Database connection Issues | |

|  |  |  |
| --- | --- | --- |
| Use Case ID | UC2 | |
| Use Case Name | Tenant Login | |
| Actor | Tenant | |
| Description | This use case describes how the user logs in to the app with his mail id and password. | |
| Triggering event | Actor selects the **Login** link | |
| Preconditions | The Actor must have registered mail id before login to the app. | |
| Flow of events | Actor | System |
| 1. Tenant enters his registered mail id or username.  2. Tenant enters his password.  3. Click Login | 1 All the information (mail id and password) sends to the database.  2 Data base authenticates the information and acknowledges to the system as a valid user.  3 System displays the menu page to the user. |
| Post conditions | The Actor see the menu page after successful login. | |
| Exception - Conditions | Database connection Issues | |

|  |  |  |
| --- | --- | --- |
| Use Case ID | UC3 | |
| Use Case Name | Tenant Reset password | |
| Actor | Tenant, | |
| Description | This use case describes how the Tenant reset his password using his mail id | |
| Triggering event | Tenant selects the **Forgot Password** link | |
| Preconditions | The Tenant must have registered mail id before login to the app. | |
| Flow of events | Actor | System |
| 1. Tenant enters his registered mail id or username.  2. Click submit | 1 All the information (mail id and password) sends to the database.  2 Data base authenticates the information and acknowledges to the system as a valid user.  3 if the user exists them system sends him a mail regarding password resetting |
| Post conditions | The Actor see the menu page after successful login. | |
| Exception - Conditions | Database connection Issues | |

|  |  |  |
| --- | --- | --- |
| Use Case ID | UC4 | |
| Use Case Name | Manage Profile page | |
| Actor | Tenant | |
| Description | Tenant able to see all the details of his/her profile in application. | |
| Triggering event | Tenant clicks on the edit profile button in profile screen | |
| Preconditions | Tenant must enter their profile details | |
| Flow of events | Actor | Flow of events |
| 1. Tenant clicks profile button.  2. Tenant change username and click save button.  3. Tenant change password and click the save button.  3. Tenant change password and click the save button.  5. Tenant change the Phone number and click the save button. | 1.1. Display the profile of the user  2.1. Update the database with the changes if any and save the information in the database |
| Post conditions | Tenant will logout from application after checking his/her profile. | |
| Exception - Conditions | While updating profile, the details should be correct. | |

|  |  |  |
| --- | --- | --- |
| Use Case ID | UC5 | |
| Use Case Name | Add Apartment | |
| Actor | Landlord | |
| Description | Landlord able to add the details of the apartment of his/her profile in application. | |
| Triggering event | Landlord clicks on the **Add Apartment** button in home screen | |
| Preconditions | Landlord must be logged in | |
| Flow of events | Actor | Flow of events |
| 1. Landlord adds the photo.  2. Landlord lists the price of the apartment  3. Landlord adds the address  3. Landlord will give the description of the flat  5. Landlord click on the save button and add the apartment in his/her profile | 1.1. System store the information of the apartment in the database |
| Post conditions | Tenant will logout from application after checking his/her profile. | |
| Exception - Conditions | While updating profile, the details should be correct. | |

|  |  |  |
| --- | --- | --- |
| Use Case ID | UC6 | |
| Use Case Name | Edit Apartment Profile | |
| Actor | Landlord | |
| Description | Landlord can update the details of the apartment of his/her profile in application. | |
| Triggering event | Landlord clicks on the **Edit Apartment** button in home screen | |
| Preconditions | Landlord must be logged in | |
| Flow of events | Actor | Flow of events |
| 1. Landlord Edit the photo.  2. Landlord edits the price of the apartment  3. Landlord edits the address  3. Landlord edits the description of the flat  5. Landlord click on the save button and update the apartment in his/her profile | 1.1. System will update the stored information of the apartment in the database |
| Post conditions | Tenant will logout from application after checking his/her profile. | |
| Exception - Conditions | While updating profile, the details should be correct. | |

|  |  |  |
| --- | --- | --- |
| Use Case ID | UC7 | |
| Use Case Name | Delete Apartment Profile | |
| Actor | Landlord | |
| Description | Landlord able to delete the details of the apartment of his/her profile in application. | |
| Triggering event | Landlord clicks on the **Delete Apartment** button in home screen | |
| Preconditions | Landlord must be logged in | |
| Flow of events | Actor | Flow of events |
| 1 Landlord click on the delete button and update the apartment in his/her profile | 1.1. System will delete the entire record of its existence from the stored information in the database |
| Post conditions | Tenant will logout from application after checking his/her profile. | |
| Exception - Conditions | While updating profile, the details should be correct. | |

|  |  |  |
| --- | --- | --- |
| Use Case ID | UC8 | |
| Use Case Name | Inbox | |
| Actor | Landlord | |
| Description | Landlord able to read the notification received form the Tenant | |
| Triggering event | Landlord clicks on the **Inbox** button in home screen | |
| Preconditions | Landlord must be logged in | |
| Flow of events | Actor | Flow of events |
| 1. Landlord check the requests made by the users and will responds to that accordingly | 1.1. System will send the requests made by the Tenant and their contact details to the particular Landlord using their id |
| Post conditions | Tenant will logout from application after checking his/her profile. | |
| Exception - Conditions | While updating profile, the details should be correct. | |

|  |  |  |
| --- | --- | --- |
| Use Case ID | UC9 | |
| Use Case Name | Tenant Dashboard | |
| Actor | Tenant | |
| Description | Tenant are able to see the posts made by the Renter | |
| Triggering event | After logging in the user redirected to the Home page | |
| Preconditions | Tenant must be logged in | |
| Flow of events | Actor | Flow of events |
| 1. Tenant check the posts made by the Landlord and contact him directly through the application | 1.1. system will send all the posts made by the Landlords for all Tenant so that they can check every apartment profile |
| Post conditions | Tenant will logout from application after checking his/her profile. | |
| Exception - Conditions | While updating profile, the details should be correct. | |

|  |  |  |
| --- | --- | --- |
| Use Case ID | UC10 | |
| Use Case Name | Apartment Profile | |
| Actor | Tenant | |
| Description | Tenants are able to see the profile of the particular apartment | |
| Triggering event | Tenant clicks on the **Details** button on the home Screen | |
| Preconditions | Tenant must be logged in | |
| Flow of events | Actor | Flow of events |
| 1. Tenant checks the profile of the apartment which consists of details like price location and description | 1.1. System will fetch the information of the apartments by using their unique id |
| Post conditions | Tenant will logout from application after checking his/her profile. | |
| Exception - Conditions | While updating profile, the details should be correct. | |

|  |  |  |
| --- | --- | --- |
| Use Case ID | UC11 | |
| Use Case Name | Search Apartment | |
| Actor | Tenant | |
| Description | Tenants are able to search the profile of the particular apartment | |
| Triggering event | Tenant clicks on the **Search** button on the home Screen | |
| Preconditions | Tenant must be logged in | |
| Flow of events | Actor | Flow of events |
| 1. Tenant can search the apartment that interests him using search intent | 1.1. System will fetch the information of the apartments using the search key at ease |
| Post conditions | Tenant will logout from application after checking his/her profile. | |
| Exception - Conditions | While updating profile, the details should be correct. | |

|  |  |  |
| --- | --- | --- |
| Use Case ID | UC12 | |
| Use Case Name | Search Apartment using Location | |
| Actor | Tenant | |
| Description | Tenants are able to search the profile of the particular apartment using location | |
| Triggering event | Tenant clicks on the **Search** button on the home Screen | |
| Preconditions | Tenant must be logged in | |
| Flow of events | Actor | Flow of events |
| 1. Tenant can search the apartment in a particular location based on his interest | 1.1. System will fetch the information of the apartments using the location filter that he is looking for |
| Post conditions | Tenant will logout from application after checking his/her profile. | |
| Exception - Conditions | While updating profile, the details should be correct. | |

|  |  |  |
| --- | --- | --- |
| Use Case ID | UC13 | |
| Use Case Name | Search Apartment using Price | |
| Actor | Tenant | |
| Description | Tenants are able to search the profile of the particular apartment using price | |
| Triggering event | Tenant clicks on the **Search** button on the home Screen | |
| Preconditions | Tenant must be logged in | |
| Flow of events | Actor | Flow of events |
| 1. Tenant can search the apartment that he can afford using price intent in search filter | 1.1. System will fetch the information of the apartments in a particular price range that Tenant is looking for |
| Post conditions | Tenant will logout from application after checking his/her profile. | |
| Exception - Conditions | While updating profile, the details should be correct. | |

|  |  |  |
| --- | --- | --- |
| Use Case ID | UC14 | |
| Use Case Name | Search Apartment using Apartment type | |
| Actor | Tenant | |
| Description | Tenants are able to search the profile of the particular apartment using apartment type | |
| Triggering event | Tenant clicks on the **Search** button on the home Screen | |
| Preconditions | Tenant must be logged in | |
| Flow of events | Actor | Flow of events |
| 1. Tenant can search the apartment based on the number of the rooms that he requires | 1.1. System will fetch the information of the apartments that fulfils the search criteria |
| Post conditions | Tenant will logout from application after checking his/her profile. | |
| Exception - Conditions | While updating profile, the details should be correct. | |

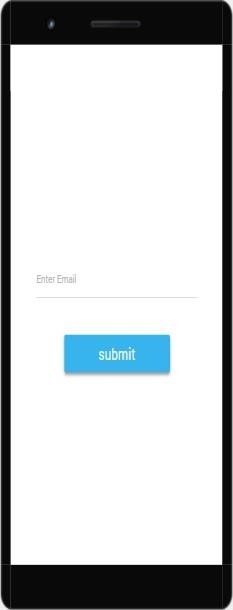
|  |  |  |
| --- | --- | --- |
| Use Case ID | UC15 | |
| Use Case Name | Admin Login | |
| Actor | Admin, | |
| Description | This use case describes how the Admin logs in to the app with his mail id and password. | |
| Triggering event | Actor selects the **Admin** link | |
| Preconditions | The admin must have registered mail id before login to the app. | |
| Flow of events | Actor | System |
| 1. Admin enters his registered mail id or username.  2. Admin enters his password.  3. Click Login | 1 All the information (mail id and password) sends to the database.  2 Data base authenticates the information and acknowledges to the system as a valid user.  3 System displays the menu page to the user. |
| Post conditions | The Actor see the menu page after successful login. | |
| Exception - Conditions | Database connection Issues | |

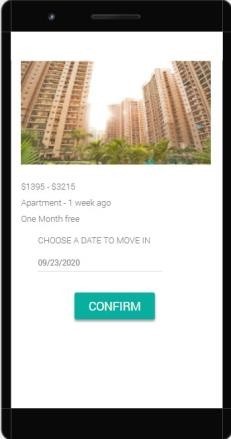
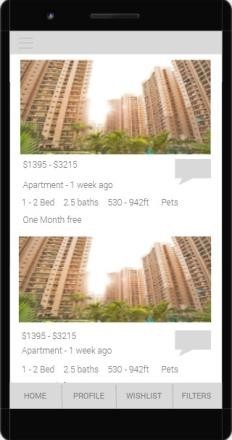
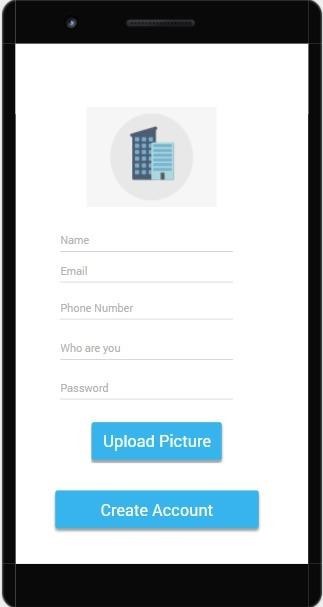
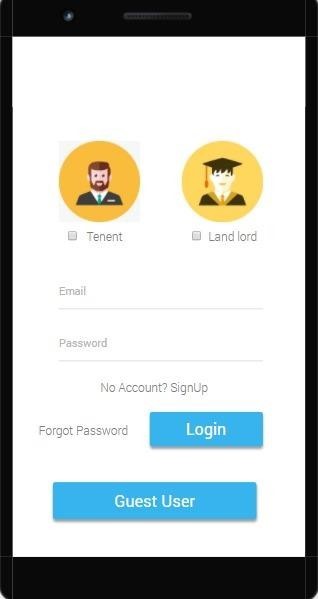
|  |  |  |
| --- | --- | --- |
| Use Case ID | UC16 | |
| Use Case Name | Admin view apartment | |
| Actor | admin | |
| Description | Admin can able to view the details of the apartment | |
| Triggering event | After logging in admin redirects to his homepage | |
| Preconditions | Admin must be logged in | |
| Flow of events | Actor | Flow of events |
| 1. Admin can see the details of the apartment uploaded by the Renters | 1.1. System will fetch the information of the apartments profile from the database |
| Post conditions | User will logout from application after checking his/her profile. | |
| Exception - Conditions | While updating profile, the details should be correct. | |

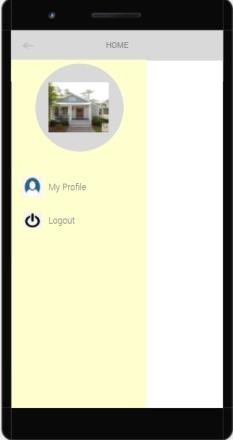
|  |  |  |
| --- | --- | --- |
| Use Case ID | UC17 | |
| Use Case Name | Admin Verify Apartment | |
| Actor | Admin | |
| Description | Admin will verify the details of the apartment | |
| Triggering event | After logging in admin redirects to his homepage | |
| Preconditions | Admin must be logged in | |
| Flow of events | Actor | Flow of events |
| 1. Admin will verify the details of the apartment whether it is fake or real | 1.1. System will fetch the information of the apartments profile from the database |
| Post conditions | User will logout from application after checking his/her profile. | |
| Exception - Conditions | While updating profile, the details should be correct. | |

|  |  |  |
| --- | --- | --- |
| Use Case ID | UC18 | |
| Use Case Name | Admin Delete Apartment | |
| Actor | admin | |
| Description | admin will Delete the profile of the apartment | |
| Triggering event | After logging in admin redirects to his homepage | |
| Preconditions | Admin must be logged in | |
| Flow of events | Actor | Flow of events |
| 1. Admin will verify the details of the apartment whether it is fake or real. And delete the profile of the apartment if it is fake | 1.1. System will fetch the information of the apartments profile from the database and give access to the admin to delete if the details given by the renter are fake |
| Post conditions | User will logout from application after checking his/her profile. | |
| Exception - Conditions | While updating profile, the details should be correct. | |

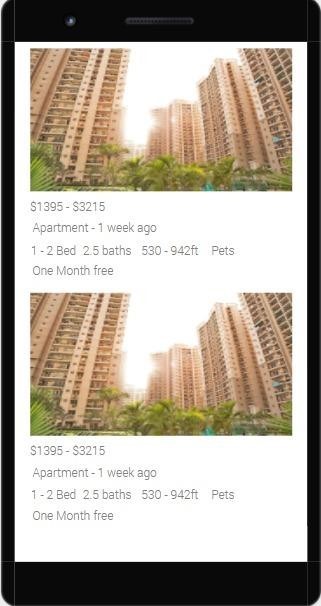
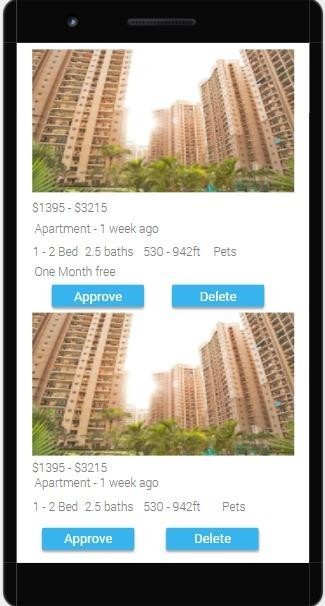
TENANT :



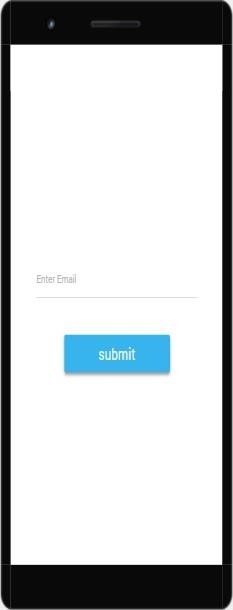


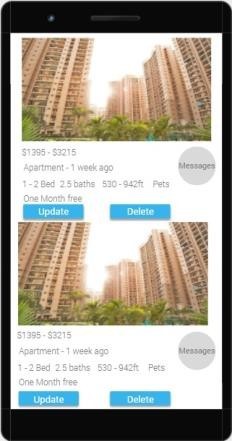
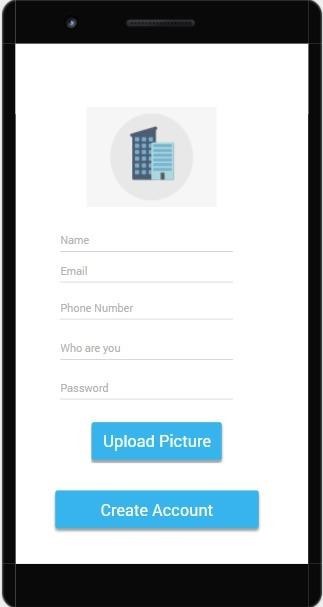
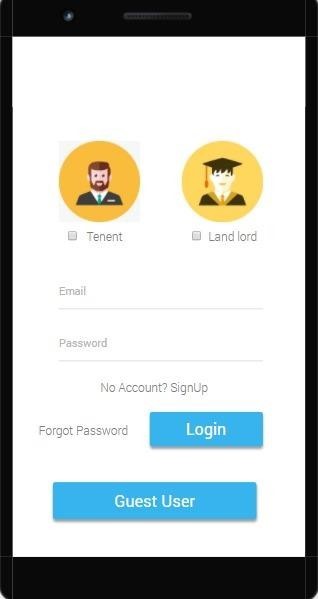


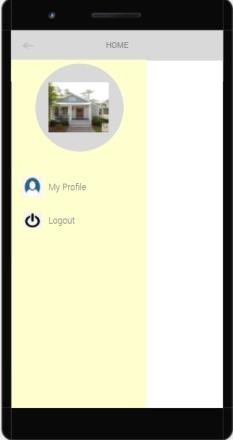
ADMIN



LANDLORD:

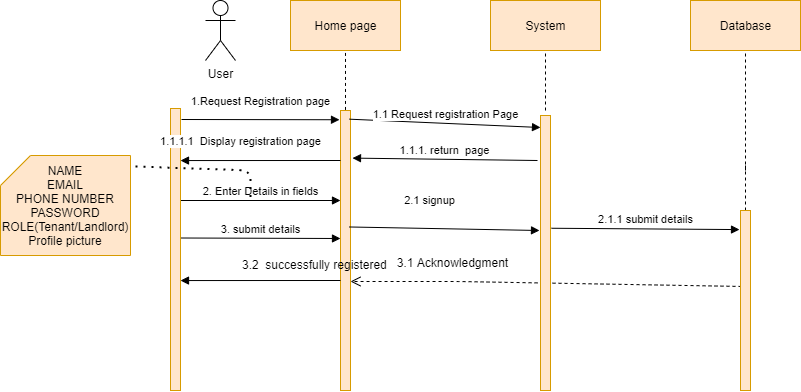




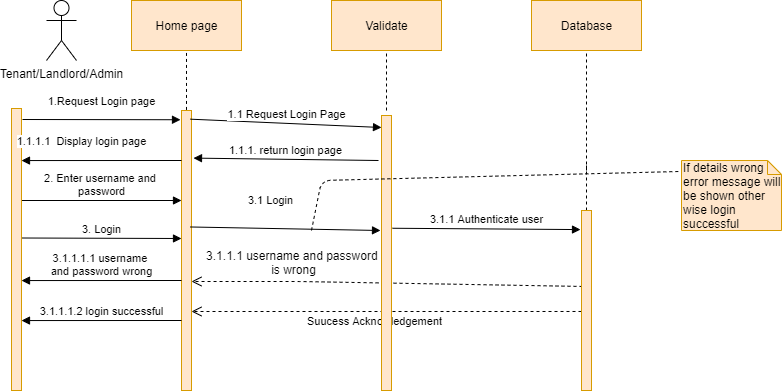


SEQUENCE DIAGRAM:-

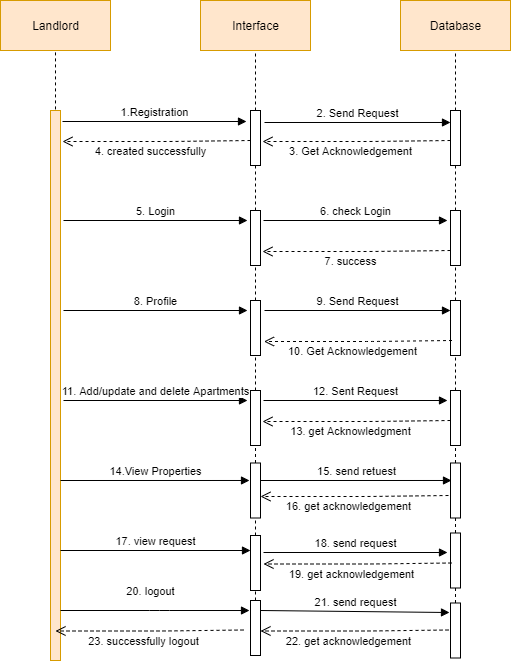
TENANT REGISTRATION



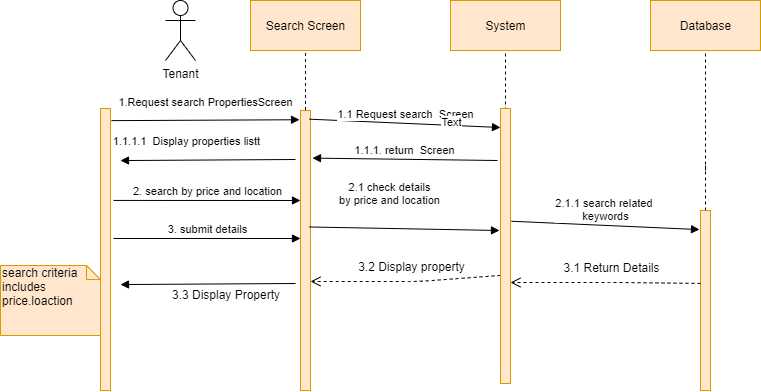
TENANT/LANDLORD/ADMIN LOGIN



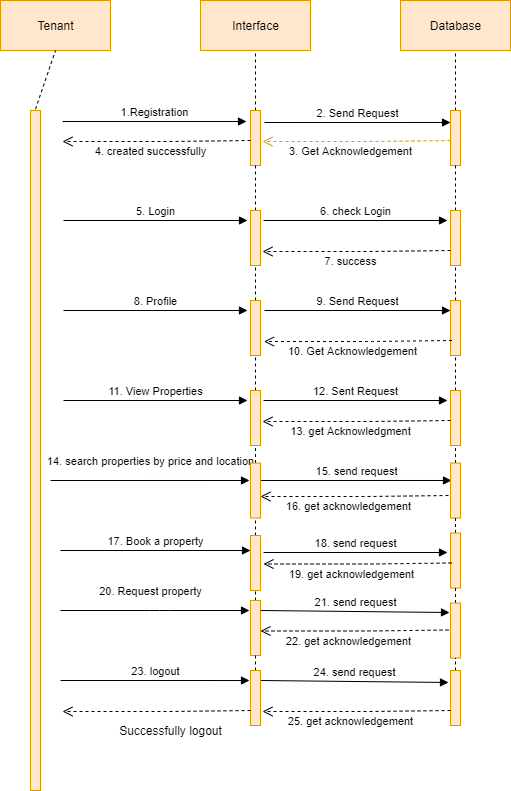
LANDLORD :



TENANT SEARCH:



TENANT COMPLETE FUNCTIONALITY:



DATABASE SCHEMA DIAGRAM:

