

**Apartment For Rental Application**

By: Vidya Sagar Mulkanoori (1895253)

Alekya Valisetty (1892744)

Anilkumar Vilasagar (1898475)

Akvinder Kaur (1895975)

Sagar Dawan (1895739)

Sukhdeepsinghbrar (1795558)

**Software Requirements Specification**

**Document**

**Version: 1** **Date: 16/05/2020**

Table of Contents

1. Introduction 2
2. System Analysis 3
3. Specific Requirements 3

3.1 Functional Requirements 3

3.2 Non Functional Requirements 4

3.3 Other Requirements 5

3.3.1 Software Requirements 5

3.3.2 Hardware Requirements 5

4. Analysis Models 6

4.1 Use case Diagram 6

4.2 Class Diagram 7

4.3 ER Diagram 7

5. Database 8

5.1 Database Schema Diagram 8

5.2 Data Dictionaries 8

5.3 Database Scripts 9

6. Screens 11

1. **Introduction**

Renting a flat is a type of business for renting an apartment or land, buildings and offices. Real estate agencies duties include, to give the property on rent or else to sell the property. Many people search for apartments for many purposes like residence, offices, and etc. Every individual wants his flat to be in the best location with the best facilities. People need to meet the agent in person, for checking the apartment details and also need to visit the location. It takes a long time to look for the desired location and desired type of apartment. Thus, The Gaspésie Systems Group wants you to develop an app to ease the process of finding an apartment for people living anywhere in Québec. The app allows the user to visualize efficient 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. This app can help you to get the best apartment by just sitting at home or anywhere. People can book their favorite property online just after a few clicks. In this system the Renter can add the apartment for renting purposes and users can book a property for rent. This system has two modules namely, Renter and User. Renter can add the property details for buildings, retail sites, flats, houses, bungalows. Users need to register and then login just by using credentials. 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. Users can view the properties for rent or for buying purposes. He/ she can mark the favorite apartments and can also book appointments for visiting the viewed favorite places.

1. **System Analysis**

**1. Title:** Analysis and Design of an Apartment Rental Mobile Application

**2. The Problem Statement:** The Gaspésie Systems Group wants you to develop an app to ease the process of finding an apartment for people living anywhere in Québec.

(a) Overall goals of the app allows the user to visualize efficient 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.

(b) Scope of the project IN-Scope: This will include only users and their requests regarding apartments. The Apartment Rental Mobile Application includes to find an apartment for people living anywhere in Quebec.

**3. Specific Requirements**

**3.1. Functional Requirements**

…………To be updated…….

**3.2. Non-functional requirements**

Requirements on usability, reliability, performance, supportability, security, recovery, interface, implementation, operation, and legal. It describes aspects of the system that are concerned with how the system provides the functional requirements. They are:

* The system will be a Mobile-based application.
* Menus should be organized in a hierarchical manner (usability)
* Easy to see and use navigation
* Maintain readable content
* The application should be reliable to perform the business,i.e. when a user performs some important action it should be acknowledged with confirmation.
* The system provides a help and support menu in all interfaces for the user to interact with the system.
* The system provides security to the customer by encrypting all their personal data.(Security)
* All the application data should be secured and be encrypted with minimum.
* The system response time for every instruction conducted within time of 60 seconds.
* Ability to maintain mass number of customers on the server at once without crashing
* Speedy performance / transmission of data
* Have a quick recovery time if anything were to go wrong
* Display accurately and efficiently on all devices (responsive view)
* In case of a system crash, application recovers the user data by maintaining backup of customer records in multiple databases.
* Applications will be backed up daily. (Back up)
* The system is available for 24 hours, 7 days a week.

**3.3. Other Requirements**

**3.3.1. Software Requirements**

…………To be updated…….

**3.3.2. Hardware Requirements**

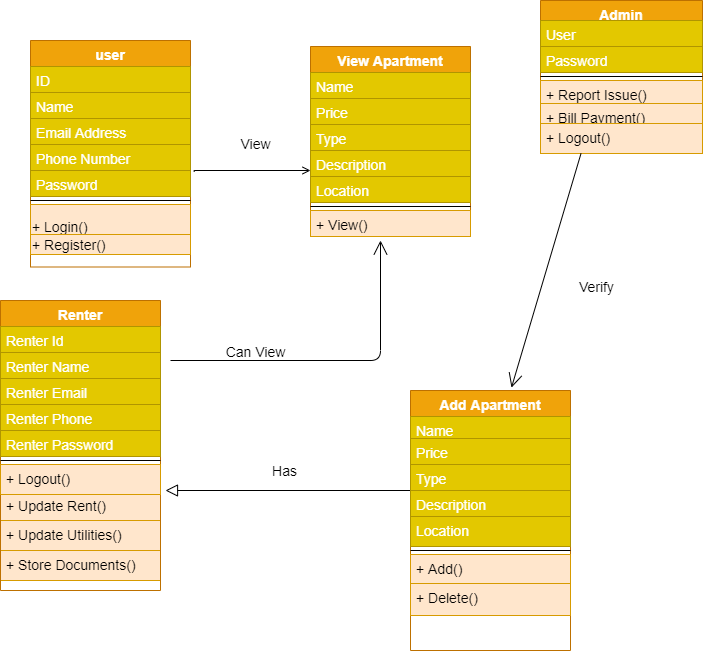
…………To be updated…….

**4. ANALYSIS MODELS**

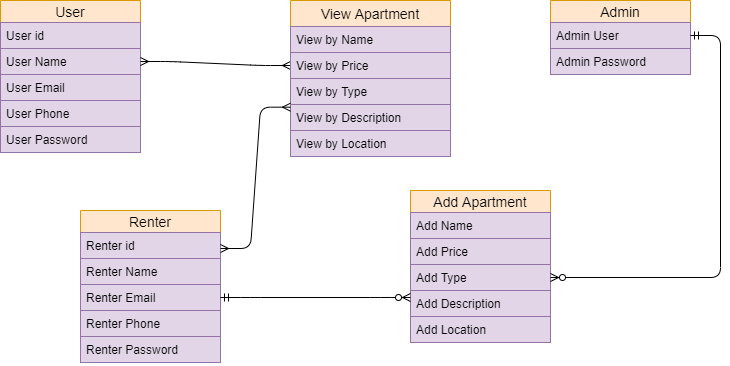
**4.1 Use Case Diagram**

…………To be updated…….

**4.2 Class Diagram**

****

**4.3 ER Diagram**



1. **Database**

A database is an organized collection of data, generally stored and accessed electronically from a computer system. These model data as rows and columns in a series of tables, and the vast majority use SQL for writing and querying data.

**5.1 Database Schema Diagram**

…………To be updated…….

**5.2 Data Dictionaries**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Admin** | | | | | |
|  | **Column Name** | **Data Type** | **Length** | **Nullable** | **Description** |
| 1 | Admin id | INT | 10 | N | Contains Admin id |
| 2 | username | VARCHAR | 50 | N | Username for admin login |
| 3 | password | VARCHAR | 50 | N | Password for admin login |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Renter/User Registration** | | | | | |
|  | **Column Name** | **Data Type** | **Length** | **Nullable** | **Description** |
| 1 | user\_id | INT | 10 | N | Contains renter/user id |
| 2 | Firstname | VARCHAR | 50 | N | First name for user login |
| 3 | Last Name | VARCHAR | 50 | N | Last name for user login |
| 4 | Phone | VARCHAR | 50 | N | Phone number |
| 5 | Email | VARCHAR | 50 | N | E-mail of renter/user |
| 6 | Password | VARCHAR | 50 | N | Password for renter/user login |
| 7 | Module | VARCHAR | 10 | N | Contains type of user/renter |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Add Apartment/Property** | | | | | |
|  | **Column Name** | **Data Type** | **Length** | **Nullable** | **Description** |
| 1 | A\_id | INT | 10 | N | Contains apartment id |
| 2 | P\_name | VARCHAR | 50 | N | Apartment name |
| 3 | P\_type | VARCHAR | 50 | N | Apartment type(2bhk,3bhk,etc) |
| 4 | Location | VARCHAR | 50 | N | Location of apartment |
| 5 | Price | DOUBLE | 10 | N | Apartment price |
| 6 | Description | VARCHAR | 500 | N | Description of apartment |
| 7 | A\_Date | Date |  | N | Available Date |
| 8 | Status | VARCHAR | 10 | N | Apartment status |
| 9 | Lat | VARCHAR | 50 | N | Latitude of apartment |
| 10 | Log | VARCHAR | 50 | N | Longitude of apartment |

**5.3. Database Script**

drop database if exists Rent\_Apartment;

create database Rent\_Apartment;

use Rent\_Apartment;

**/\* Admin table \*/**

create table if not exists admin(

a\_id int(10) not null,

user\_name varchar(50) not null,

user\_password varchar(50) not null,

primary key(a\_id)

);

**/\* Renter/User table \*/**

create table if not exists user\_registration(

user\_id int(10) not null,

first\_name varchar(50) not null,

Last\_name varchar(50) not null,

Email varchar(50) not null,

phone\_num int(20) not null,

user\_password varchar(50) not null,

user\_type varchar(10) not null,

primary key (user\_id),

);

**/\* Add apartment table \*/**

create table if not exists add\_apartment(

apart\_id int(10) not null,

P\_name varchar(50) not null,

P\_type varchar(50) not null,

Location varchar(50) not null,

Price int(20) not null,

description varchar(500) not null,

a\_date Date not null,

status varchar(50) not null,

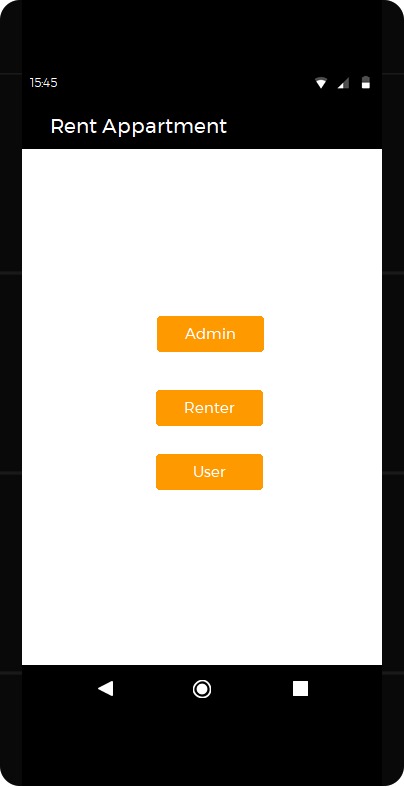
lat varchar(50) not null,

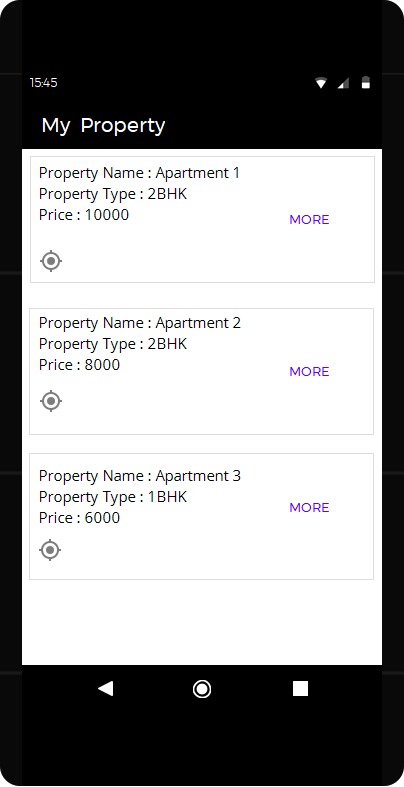
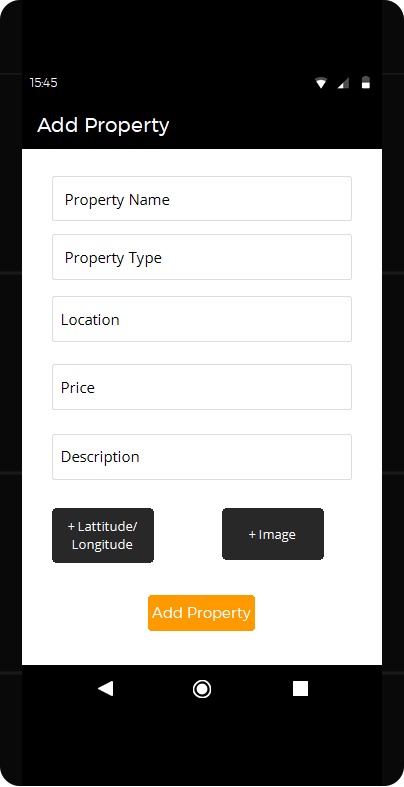
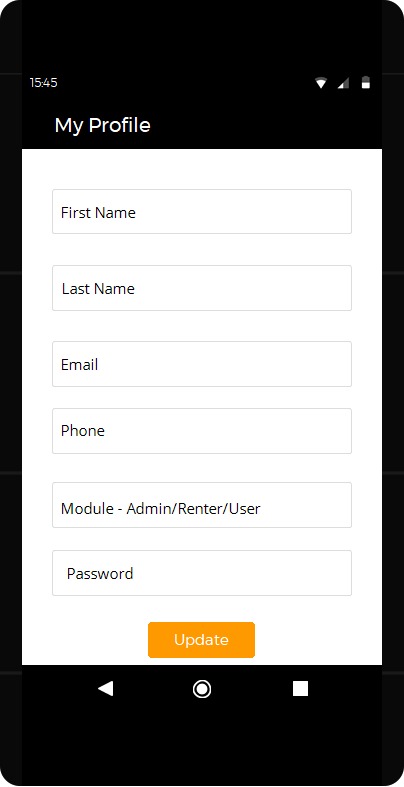
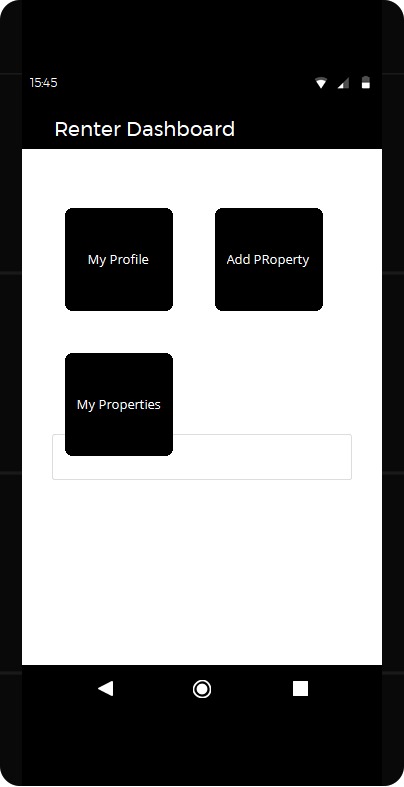
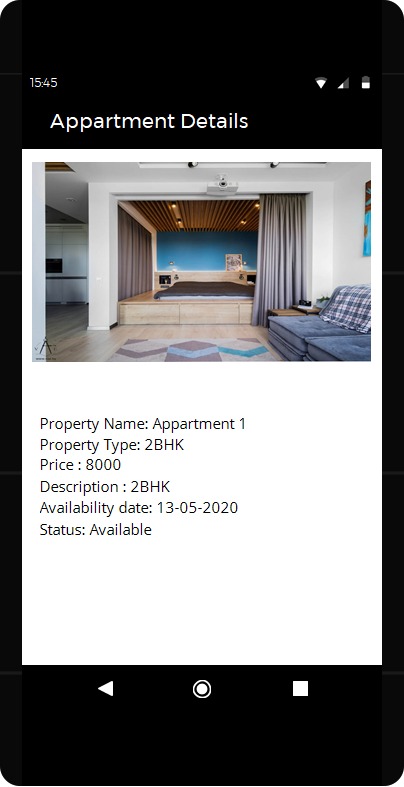
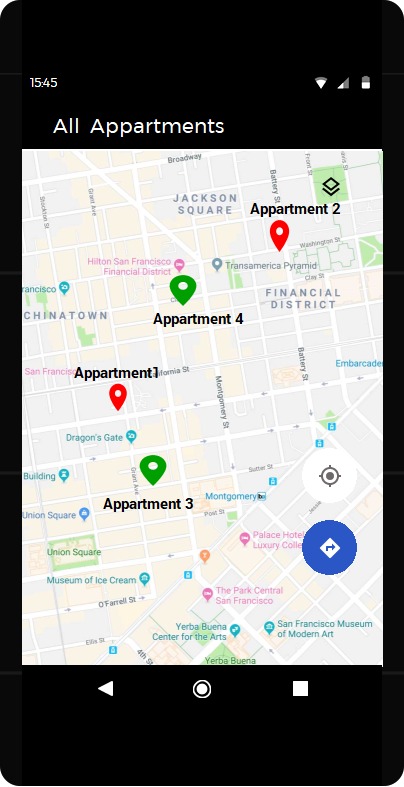
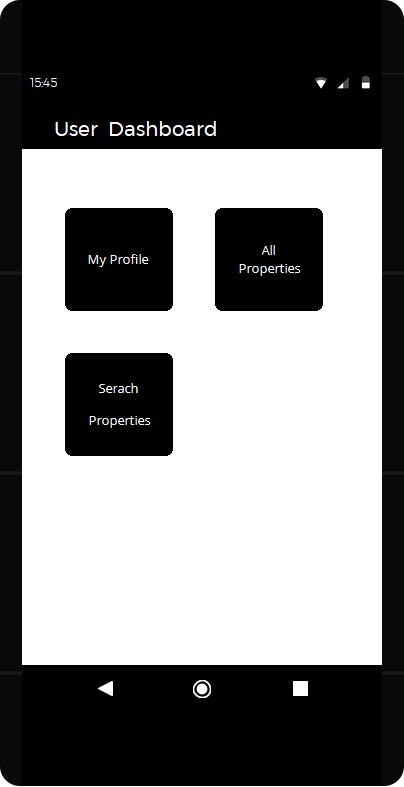
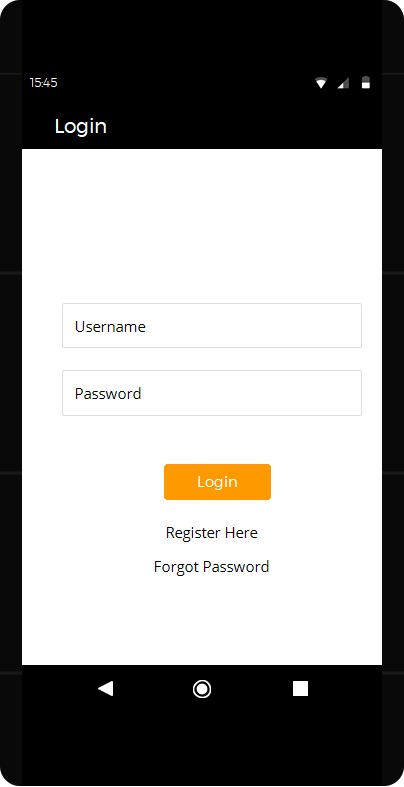
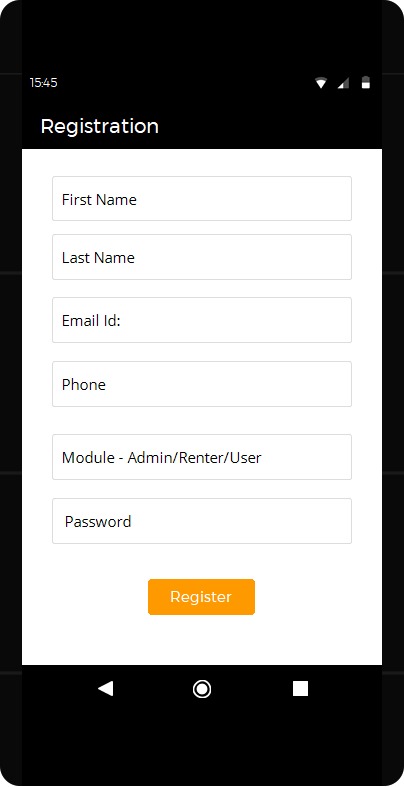
log varchar(50) not null,

primary key (apart\_id),

);

**6. Screens**

****

****