****



**Apartment For Rental Application**

**420-ENW-MT: PROJECT REPORT**

**( INTERNSHIP)**

By: Vidya Sagar Mulkanoori (1895253)

Alekya Valisetty (1892744)

Anilkumar Vilasagar (1898475)

Akvinder Kaur (1895975)

Sagar Dawan (1895739)

Sukhdeepsinghbrar (1795558)

**SUBMITTED TO:**

**Sakkaravarthi Ramanathan**

**Gilles-Philippe Grégoire**

**Sakshi Sharma**

**SilviyaPaskaleva**

**Harshkumar Dave**

**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 6

4. Analysis Models 6

4.1 Use case Diagram 6

4.2 Class Diagram 7

4.3 ER Diagram 8

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**

The system will be password-protected. Apartment Rental Application will be a multi-user system where every user must log in. This application needs to perform the following functions:

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | Requirement Statement | MUST/COULD/  SHOULD/WOULD | Comments |
| FR1 | The application must have login screen | MUST | App must have the login page |
| FR2 | The application must have register screen and users must register by entering details | MUST | User shall have to register before using the application |
| FR3 | Users must login application with valid credentials | MUST | The system shall have to take the valid details to login |
| FR4 | Users could manage their profile | COULD | The system shall edit the profile to store in the database |
| FR5 | Users could manage Forgot password | COULD | The system shall manage to recreate the password to store in the database |
| FR6 | Renter can add apartments with details which include (price, location, rooms, etc) | MUST | The system shall add the data according to the renter requirement |
| FR7 | Renter could edit apartments | COULD | The system shall edit details to store in database |
| FR8 | Renter could delete apartments | COULD | The system shall delete data and store in the database |
| FR9 | Renter could see user requests | COULD | The system shall allow renter to see user requests |
| FR10 | Users could view apartments | COULD | The system shall allow user to view apartments |
| FR11 | Users could select and view desired apartment details | COULD | The system shall allow users to select and view desired apartment from the database |
| FR12 | Users could search based on price criteria | COULD | The system shall allow user to search according to the price they want to. |

**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:

* NF1-The application must be a Mobile-based application.
* NF2-Menus should be organized in a hierarchical manner (usability)
* NF3-The application should be user friendly
* NF4-Application must be with readable content
* NF5-The application should be reliable to perform the business, i.e. when a user performs some important action it should be acknowledged with confirmation.
* NF6-The application must be providing a help and support menu in all interfaces for the user to interact with the system.
* NF7-All the application data should be secured and be encrypted.
* NF8-The application should have response time for every instruction conducted within time of 60 seconds.
* NF9- Application should be able to maintain mass number of customers on the server at once without crashing.
* NF10- Application must be able to transform data quickly (Speedy performance).
* NF11-Application should be backed up daily. (Back up)

**3.3. Other Requirements**

**3.3.1. Software Requirements**

For developing the application the following are the software requirements

1. Android Development Tool version 3.6.3

2. Android SDK for Android ADT (Recent version).

**Technologies and Languages used to Develop**

1. Android

2. Java

3. Front end: XML

4. Back end Database: MySQL

**Debugger and Emulator**

1. Android Dalvik Debug Monitor service

2. Android Emulator (Android Virtual Device)

**3.3.2. Hardware Requirements**

For developing the application the following are the Hardware Requirements:

■ Processor: Pentium IV or higher

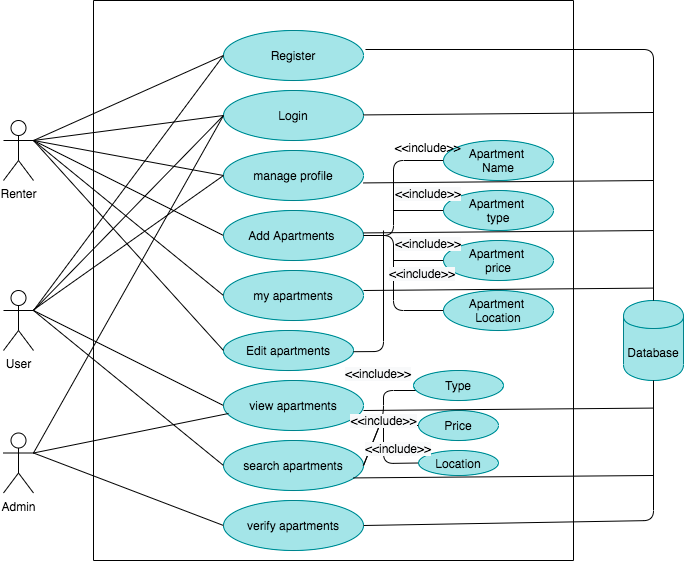
■ RAM: 256 MB

■ Space on Hard Disk: minimum 512MB

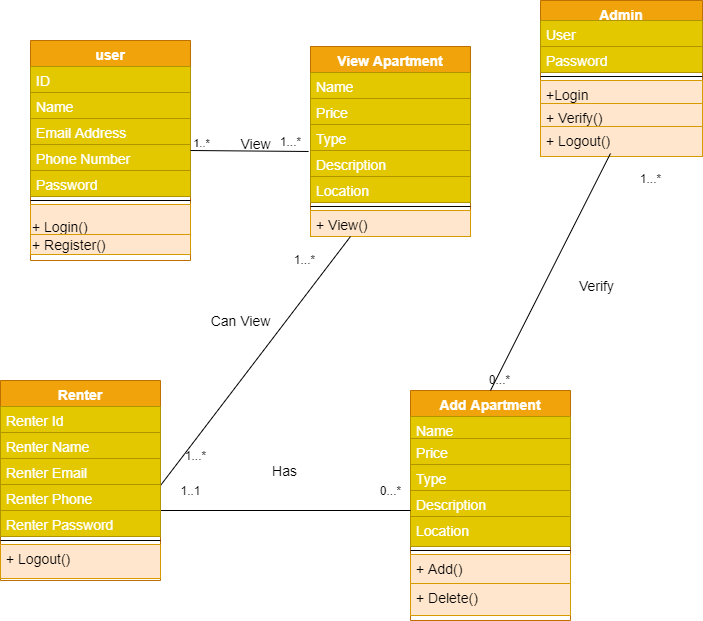
**4. ANALYSIS MODELS**

**4.1 Use Case Diagram**

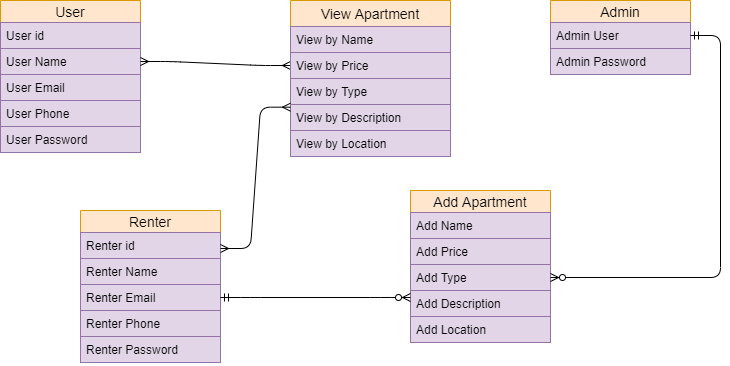
A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved.



**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\_idint(10) not null,

user\_namevarchar(50) not null,

user\_passwordvarchar(50) not null,

primary key(a\_id)

);

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

create table if not exists user\_registration(

user\_idint(10) not null,

first\_namevarchar(50) not null,

Last\_name varchar(50) not null,

Email varchar(50) not null,

phone\_numint(20) not null,

user\_passwordvarchar(50) not null,

user\_typevarchar(10) not null,

primary key (user\_id),

);

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

create table if not exists add\_apartment(

apart\_idint(10) not null,

P\_namevarchar(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,

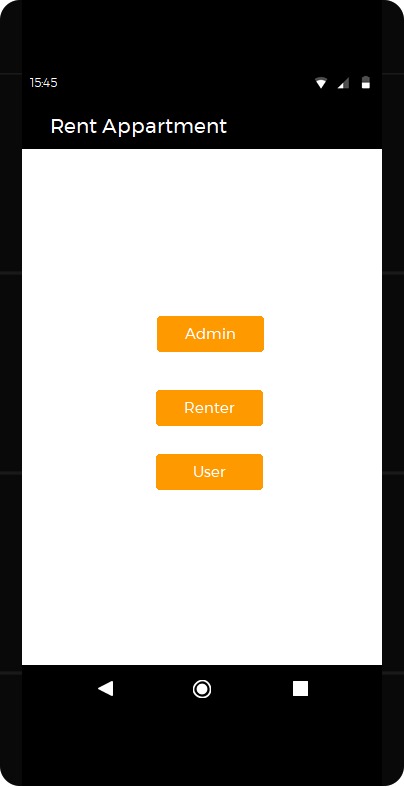
log varchar(50) not null,

primary key (apart\_id),

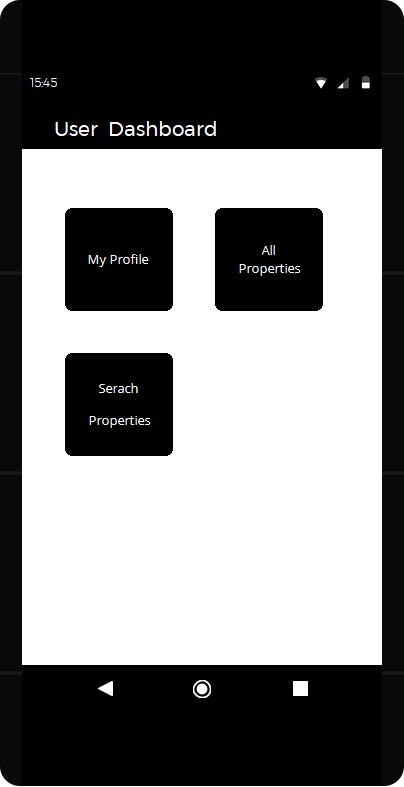
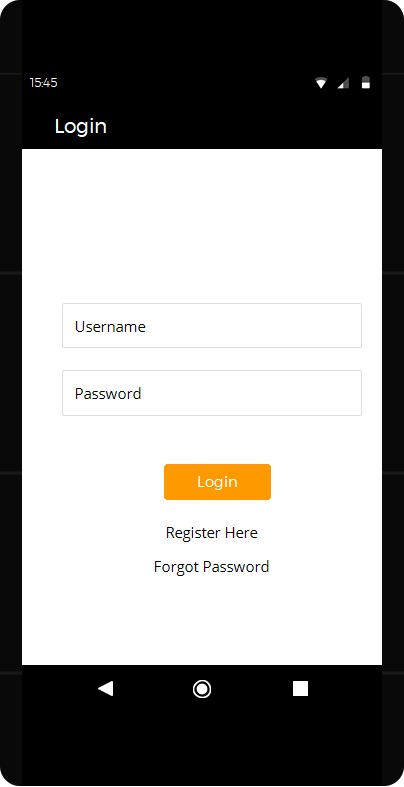
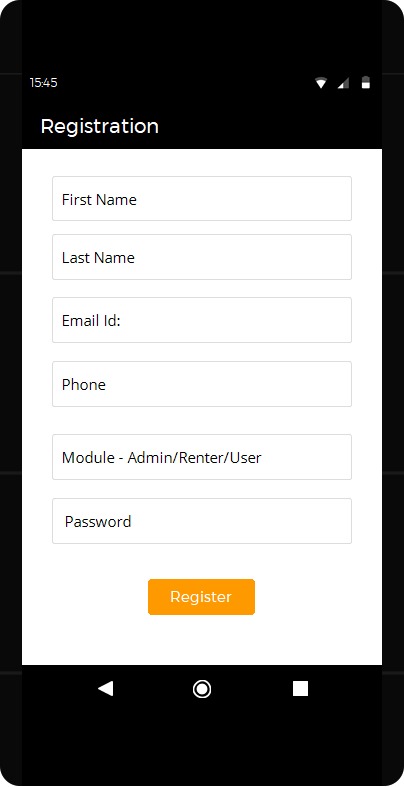
);

1. **Screens**

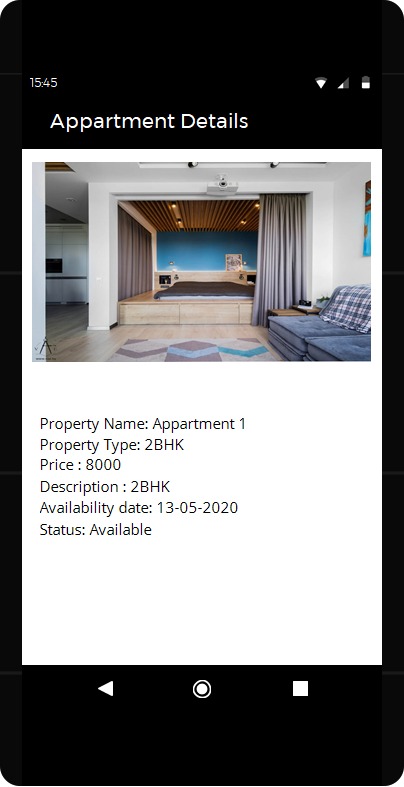
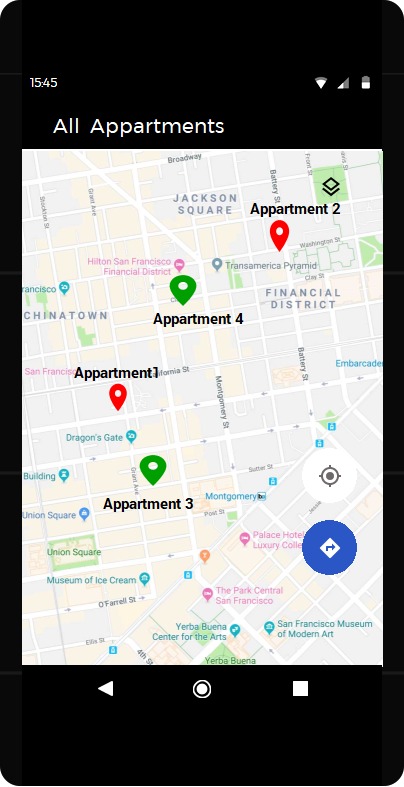
The Rent Apartment application is designed with a simple user interface with the purpose of providing great usability for users. After downloading and installing the application the person has the option to register as renter or user. As shown in the figure below.

****

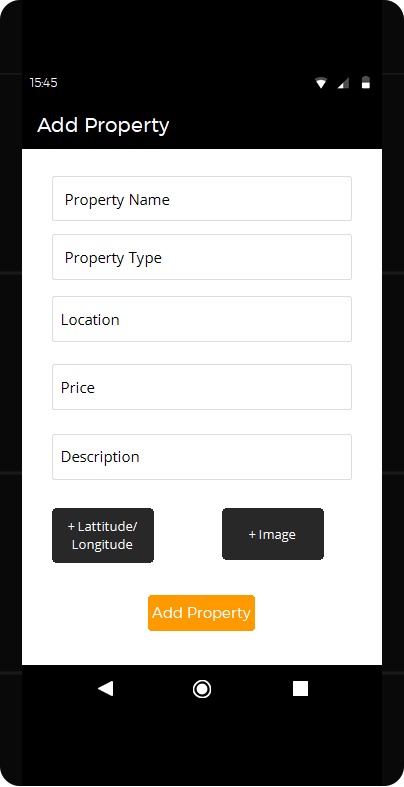
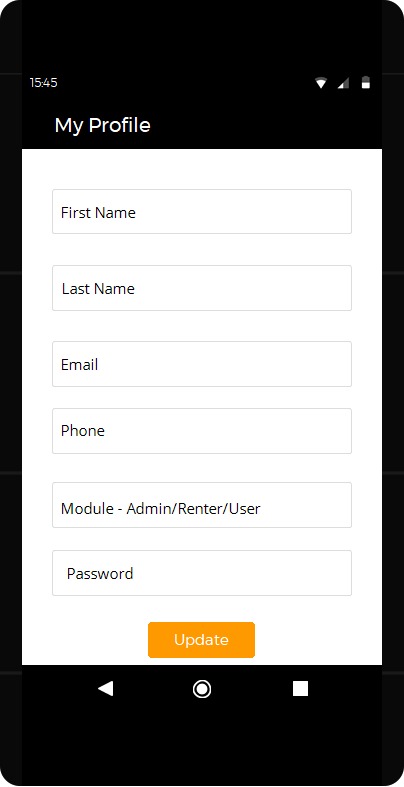
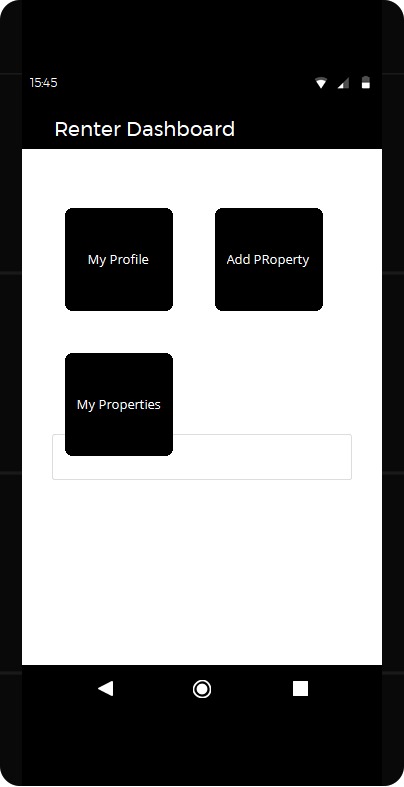
After the person select the option than they has to register with the application using his identification information (first name, last name, email id, phone, module where you want to register as the user or the renter, and the password) after they register they have to login and user also has the dashboard where they can edit profile and check all properties and also search the properties as shown below.

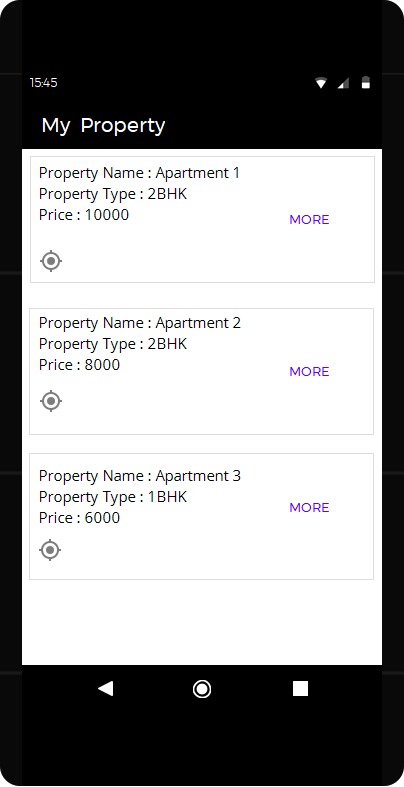
****

Once the user login successfully he/she can search for the properties which are available in the area and can view the apartment and also the description, location, price, availability date as shown below

****

There is also other feature where renter can check his/her profile or edit his/her profile, also can add new property and also can view his/her property where he/she posted previously as shown in the figure below

****

****