



**RV Educational Institutions®**  
**RV College of Engineering®**

Autonomous  
Institution Affiliated  
to Visvesvaraya  
Technological  
University, Belagavi

Approved by AICTE,  
New Delhi, Accredited  
By NAAC, Bengaluru  
And NBA, New Delhi

*Go, change the world*

## **DEPARTMENT OF MASTER OF COMPUTER APPLICATIONS**



**Minor Project Report**

**18MCA46**

**on**

**MOBILE APPLICATION FOR  
SMART PARKING SYSTEM**

*Submitted in partial fulfillment of the requirements for the award of degree  
of*

**MASTER OF COMPUTER APPLICATIONS**

**By**

**CHETAN PAVATE (USN:1RZ18MCA18)**

**NILAY KUMAR (USN:1RZ18MCA18)**

**Under the Guidance**

**of**

**SAVITA SHEELAVANT**

**Assistant Professor**

**2019-2020**

# **RV COLLEGE OF ENGINEERING®**

(Autonomous Institution Affiliated to Visvesvaraya Technological University, Belagavi)

## **DEPARTMENT OF MASTER OF COMPUTER APPLICATIONS**

**Bengaluru– 560059**



### **CERTIFICATE**

Certified that the project work titled **“MOBILE APPLICATION FOR SMART PARKING SYSTEM”** carried out by **CHETAN PAVATE (USN:1RZ18MCA08)**, **NILAY KUMAR (USN:1RZ18MCA18)**, a bonafide student of **RV College of Engineering** in partial fulfilment for the award of **Master of Computer Applications, RV College of Engineering®, Bengaluru affiliated to Visvesvaraya Technological University, Belagavi** during the year **2019-20**. It is certified that all corrections/suggestions indicated for internal assessment have been **gineering®, Bengaluru** submitteincorporated in the report deposited in the departmental library. The project report has been approved as it satisfies the academic requirement in respect of project work prescribed for the said degree.

**Prof. Savita Sheelavant**  
Assistant Professor  
Department of MCA  
RVCE, Bengaluru –59

**Dr. Andhe Dharani**  
Professor & Director  
Department of MCA  
RVCE, Bengaluru–59

# **RV COLLEGE OF ENGINEERING®**

(Autonomous Institution Affiliated to Visvesvaraya Technological University, Belagavi)

## **DEPARTMENT OF MASTER OF COMPUTER APPLICATIONS**

**Bengaluru– 560059**

### **DECLARATION**

**Chetan Pavate & Nilay Kumar**, student of Fourth semester in Department of Master of Computer Applications , RV College of Engineering, Bengaluru declare that the project titled “**MOBILE APPLICATION FOR SMART PARKING SYSTEM**” has been carried out by me. It has been submitted in partial fulfilment of the course requirements for the award of degree in **Master of Computer Applications** of RV College of Engineering, Bengaluru affiliated to Visvesvaraya Technological University, Belagavi during the academic year **2019-20**. The matter embodied in this report has not been submitted to any other university or institution for the award of any other degree or diploma.

**Date of Submission:**

**Signature of the Student**

**Student Name: Chetan Pavate**

USN: 1RZ18MCA08

Department of MCA  
RV College of Engineering®,  
Bengaluru-560059

**Student Name: Nilay Kumar**

USN: 1RZ18MCA18

Department of MCA  
RV College of Engineering®,  
Bengaluru-560059

## **ACKNOWLEDGEMENT**

“Gratitude is the inward of kindness received”. We have taken efforts in our project **“MOBILE APPLICATION FOR SMART PARKING SYSTEM”**. However, it would not have been possible without the kind support and help of many individuals. We would like to extend our sincere thanks to all of them.

To begin with, we would like to express our gratitude towards our parents for their kind co-operation and encouragement which helped us in completion of this project.

We would also like to thank **Dr. K N Subramanya, Principal, RV College of Engineering<sup>®</sup>** for giving this opportunity to explore our talents; necessary resources and working environment in the college.

We are highly indebted to **Dr. Andhe Dharani, Proffessor and Director, Department of MCA, RV College of Engineering<sup>®</sup>** for her guidance and constant support.

We would also like to express my special gratitude and thanks to our guide **Prof. Savita Sheelavant, Assistant Professor, Department of MCA, RV College of Engineering<sup>®</sup>** for giving her guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

***With Heartfelt Gratitude to One and All***

**Chetan Pavate**

Department of MCA

RV College of Engineering<sup>®</sup>

Bengaluru-59

**Nilay Kumar**

Department of MCA

RV College of Engineering<sup>®</sup>

Bengaluru-59

## ABSTRACT

Mobile Application for Smart Parking System is a real-time detection of available parking slots. The parking system is a big issue in urban cities. Yearly, motorists spend an average time of 17 hours and approximately about Rs.7000 for places to park. In traditional system users manually need to find an empty parking slot and if not found, then the user must check for a slot in another parking lot. The proposed mobile application for smart parking system will overcome the challenges and the difficulties faced in the current conventional parking system. It saves the time and effort that is to put into this task and also saves the fuel wastage. Mobile application for Smart parking system will be a revolutionary change in the parking system which is a dire requirement in urban societies with increasing number of automobiles. It could bring order in the current parking system to make it simple and easy.

Mobile application for smart parking system minimum requirements are processor will be core i5 or higher , Ram will be 8GB , storage will be 30 gb , GPU GTX 980 4GB or higher , LCD monitor is also needed and Android mobile phone and in software we need the operating system like windows 10 , Ubuntu 16 or higher , in database we used sqlite3 and for front end we use Android studio.

Mobile application for Smart parking system overcomes the difficulties and challenges that every motorist faces and lets them find a parking space and reserve it within a snap of a finger. The system also stores all the activities of parking lot in a database which removes the need of maintaining manual logs of a parking lot.

## Contents

<b>Chapter</b>	<b>Title</b>	<b>Page. No</b>
1.	Introduction	7
2.	Project description	11
3.	Software Requirement Specification	13
4.	System Design Specification	17
5.	Implementation	22
6.	Testing	37
7.	Conclusion	40
8.	References	41

# Chapter 1

## Introduction

Searching for a vacant parking space in a metropolitan area is the daily concern for most drivers, and it is time-consuming. It commonly results more traffic congestion and air pollution by constantly cruising in certain area only for an available parking space. For instance , a recent survey [1] shows that during rush hours in most big cities, the traffic generated by cars searching for parking spaces takes up to 40% of the total traffic. To alleviate such traffic congestion and improve the convenience for drivers, many smart parking systems aiming to satisfy the involved parties(e.g. parking service providers and drivers) have been deployed.

### 1.1Preamble

Smart Parking is one of the most adopted and fast growing Smart City Solution across the world. Airports, Universities, shopping centers etc. are just a few entities that have begun to realize the significant benefits of automated parking technology.

Smart Parking involves an interface and applications that allow users to monitor available and unavailable parking slots.

It also reduces the human efforts by showing the parking slots for each and every vehicle. And cost is estimated according to the time of parked vehicle.

In today parking lots there are no standard system to check for parking spaces. The system heavily relies on human interaction with the physical space and entity. This leads to wastage of human manpower and also parking spaces at times. These parking lots are dependent on Human-to-Human Interaction (HHI) which is not efficient. Previously, various techniques have been proposed to overcome such problems. Smart parking application is design to provide Android interface for booking slots remotely. The wireless sensor network along with arduino used a data from each node to show if the parking space is available and the data was available only at the entry point and at the administration end.

## **1.2 Literature Review**

Too many cars, too much traffic and there is no enough parking area. This is the situation which is seen in most of the metropolitan cities today. The aim of this project is to automate the car and car parking as well. A miniature model of an automated car parking system that can regulate and manage number of cars that can be parked in given space at any given time based on the availability of parking slot. Automated parking is a method of parking and existing cars using sensing device. The entering and leaving to the lot is command [1].

The aim of our project is to develop and implement an automatic parking system that will increase convenience and security of the public parking lot as well as collecting parking fee without hassles of using magnetic card. The parking system will able to have less interaction of humans and use no magnetic card and its devices [2].

The application will be more convenient compared to the traditional approach because a motorist here is entitled with several finger tip features in his mobile phone from remote place. Hence there is no need for motorist to be physically present to book a parking slot [3].

In additions to that, it has parking guidance system that can show and guide user towards a parking space. Overall, the systems run with pre-programmed interface to make minimum human involvement in parking system processed by an android application [4].

Our system uses the reservation approach to provide slot service access in first come first serve bases. The first to register a particular slot will have the reservation over that slot for his given time period [5].

In order to book a slot through our application, motorist has to register himself in the system and prove that the motorist is genuine license holder for his vehicle also his other details are to be registered for bookkeeping [6].

Our project for parking space detection uses traditional database approach. It will display the number of current available parking lots in the parking area in a



transparent color, a booked slot in red color and available slot in green color . This proposed system, has been developed in software platform only [7].

In a traditional way of parking system , the logs of all the vehicles are maintained using ledgers and payments by motorists are made through cash and receipts in the form of paper which decreases the security and maintaining of documents. But with our system , payments are made through UPI using Google Pay UPI Gateway and logs of the motorists are maintained by admin digitally [8].

In addition, the accessibility of the data can vary for different users based on the type of their vehicle and date of booking. Admin is enabled to record all the details about motorists and can keep track of all the bookings of any motorist and their payment. Admin can also keep track of current bookings of the lot all of these are which maintained in the database [9].

The system we intent to develop makes it easy and saves lot of motorists' time that they would usually waste in finding a right parking slot . Also our system encourages cashless currency system which makes the transactions more secure [10].

### **1.3 Problem statement and Scope of the project**

#### **Problem Statement**

- To build an android interface for providing solutions to Mobile application for Smart Parking System that helps people to automatically find the availability of free slot based on type of vehicle for specified duration and estimate amount to be paid

#### **Scope of the project**

- Automatic identification of vacant slots for parking. The available slots for particular type of vehicle are automatically recognized and are displayed on interface available to be booked.
- Any available slot can be booked for specified number of time and estimate the parking cost for the same.

- Parking being Human-Human interaction system, saves lot of time and reduces human efforts since all the operations like checking availability, booking a slot, cancelling a slot and estimating the fare can all be performed remotely in an android interface.

### **Expected Outcome**

Booking parking slot without wasting any time.

Simple way of parking slot booking.

Android interface for booking slots remotely.

## **1.4 Methodology adopted in the project**

The slot allocation method follows a sequence as stated below:

Step1: Initially the slot selection is made by the user from his mobile phone. He checks for the availability of a parking slot that is nearest to his location. If it is available, he moves to the next stage or else go to the initial state.

Step2: Transfers request for parking slot from the mobile using Android application.

Step3: The Parking Control Unit (Admin) gets the slot number requested by the user.

Step4: If the payment is done successfully, then the requested slot is reserved in the parking area.

Step5: After reserving a particular slot by the user then the status of that respective slot will be marked as RED=RESERVED and the remaining will be GREEN=EMPTY.

Step6: From the specified time, the timer gets ON and measures the total time.

Step7: The timer will be automatically stopped after specified time is complete or motorist can drive away before his time is complete and make the slot available again.

## **1.5 Technical Features of the project**

### **Android Studio**

Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development.

- Instant App Run.

It is an advanced technology in which it cleverly understands the transmutations done in the applications and deliver it instantly without taking time to rebuild the apk and installations make. So, quick that you can see the changes in app immediately.

- Visual Layout Editor.

Layout editor helps to build the layout quickly by adding different attributes either by hard-code or drag and drop. The preview of the codes can be seen easily on the visual editor screen and changes can be made accordingly by resizing it dynamically.

- Help To Build Apps For All Devices.

Android studio builds applications for every screen size, for wear and gear devices etc. It also can stimulate the various type of features which a hardware has like GPS location tracker, multi-touch.

## **Chapter 2**

### **Project Description**

The concept for the parking system was and is driven by two factors: a need for parking spaces and a scarcity of available land. Smart parking application that provides real time information on parking availability. This Application allows users to book their spot just by making a few clicks. With the help of a mobile-based app that helps you in locating a free parking slot and then guiding you all the way to the empty slot found.

A parking lot is an area that is assigned for parking. Normally, the parking spaces are marked on the ground with white or yellow lines that form squares that each fit one car. Parking lots are common near shops, bars, restaurants and other facilities that require parking. There are parking lots that are open throughout the year, but there are also improvised parking lots that are specially assigned for an event. The smart parking system implemented mainly in the Europe, United States and Japan is developed with the incorporation of advanced technologies and researches from various academic disciplines. With its deployment in the car park, it is hoped that it would solve the aforementioned problems faced by the patrons within the car park.

#### **2.1 Benefits**

1. Optimized parking – Users find the best spot available, saving time, resources and effort. The parking lot fills up efficiently and space can be utilized properly by commercial and corporate entities.
2. Reduced traffic – Traffic flow increases as fewer cars are required to drive around in search of an open parking space.
3. Reduced pollution – Searching for parking burns around one million barrels of oil a day. An optimal parking solution will significantly decrease driving time, thus lowering the amount of daily vehicle emissions and ultimately reducing the global environmental footprint.
4. Enhanced User Experience – A smart parking solution will integrate the entire user experience into a unified action. Driver's payment, spot identification, location search and time notifications all seamlessly become part of the destination arrival process.

5. Decreased Management Costs – More automation and less manual activity saves on labour cost and resource exhaustion.

## **2.2 Tools**

1. Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems or as a subscription-based service in 2020. It is a replacement for the Eclipse Android Development Tools (ADT) as the primary IDE for native Android application development.

2. SQLite is a relational database management system (RDBMS) contained in a C library. In contrast to many other database management systems, SQLite is not a client–server database engine. Rather, it is embedded into the end program.

## **Chapter 3**

### **Software Requirement Specification**

As soon as the project idea is conformed, I have started working on the requirements for the implementation of the project. The idea is to develop a mobile application that can receive information about the occupancy status of the parking space from client (here the database) and post that information to the mobile application. Also passing information to database through mobile application and updating the changes in the system.

#### **3.1 Introduction**

The Software Requirements Specification lays the foundation for the systematic approach towards the development of this project. It provides a general overview of the system. It helps to translate the ideas in the minds of the clients. It includes understanding of the problem domain, identifying all external entities that will interact with the system, the users and defining their functions, the constraints that would exist and the overall functioning of the system.

The ability to automatically recognize parking slots based on the availability is important in Online Parking slot booking. This research proposes a system for remote booking of a slot to park at specified time using using android interface. The process is bifercated into two main actors : (1) Admin and (2) Motorist.

As for the analytical part the system gives a daily detailed log of the events occurred in the Parking lot whenever the admin wishes to see.

#### **Definitions, Acronyms and Abbreviations**

This is developed for any Campus Parking lot. So, there will be many actors such as Motorist , Admin etc.

**Table 3.1 Acronyms and Abbreviations**

XML	Extensible Markup Language
IDE	Integrated Development Environment
Motorist	The end user group
Admin	The user who have Admin access to the system.
SRS	Software Requirements Specification
DFD	Data Flow Diagram
UML	Unified Modelling Language

## 3.2 Hardware and Software Requirements

### Hardware Requirement

- Processor – CORE i5 (3.0GHZ) or higher
- RAM - 8 GB
- Storage - 30 GB
- GPU - GTX 980 4GB or higher
- LCD Monitor
- Android Phone

### Software Requirements

- Operating System – Windows 10, Ubuntu 16 or higher
- Database – sqlite3
- Front End – Android Studio (Java, XML)

### 3.3 Functional Requirement

#### SQLite

SQLite is an in-process library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine. It is a database, which is zero-configured, which means like other databases you do not need to configure it in your system.

#### Java

Java is a general-purpose, concurrent, object-oriented, class-based, and the runtime environment(JRE) which consists of **JVM** which is the cornerstone of the Java platform. It is an object-oriented language similar to C++, but with advanced and simplified features. Java is free to access and can run on all platforms.

#### XML

XML files are used to design the layouts for our application. XML stands for extensible markup language. XML tags identify the data and are used to store and organize data, rather than specifying how to display it like HTML tags, which are used to display the data.

### 3.4 External Interfaces Requirements

#### User Interface

The user interface for the application shall only be compatible to Android Smart Phone

#### Hardware Interface

- Android Smart Phone have enough storage capacity to install and run the application.

#### Software Interface

- OS: Android 4.4 kitkat and above
- TECHNOLOGIES USED: Java , XML
- DATABASE: sqlite3



### **3.5 Performance Requirements**

It is expected that the server & the database would perform functionally all the requirements that are specified by the client. The system is expected to be running 24 hours a day. Therefore, the performance cannot be compromised during the design phase.

- The performance of the system should be fast and accurate
- Mobile application for Smart Parking System shall handle expected and unexpected errors in ways that prevent loss in information and inaccurate results. Thus, it contains validations to ensure smooth functioning of the application.
- System should be able to handle multiple calculations, thus it should without any fault.
- The environment should not be too bright and dark

### **3.6 Design Constraints**

All the inputs should be checked for validation and messages should be given for the improper data. Invalid data are to be ignored and error messages should be given.

#### **Standard Compliance**

There shall be consistency in variable names within the system. The Android user interface shall have a consistent look and feel.

#### **Hardware Limitations**

The end user who is using android phone but lower version than Android 4.4 kitkat they can't use the application properly.

## Chapter 4

### System Design Specification

UML diagram are used to explain the design of the system. Once the requirement gathering is completed, system design is done using Unified Modeling Language(UML). UML plays an important role in designing object oriented software by using graphical notation to depict the design of the system.

#### 4.1 Architectural Design:

##### 4.1.1 Problem Specification:

To build an android interface for providing solutions to Mobile application for Smart Parking System that helps people to automatically find the availability of free slot based on type of vehicle for specified duration and estimate amount to be paid

##### 4.1.2 Block Diagram

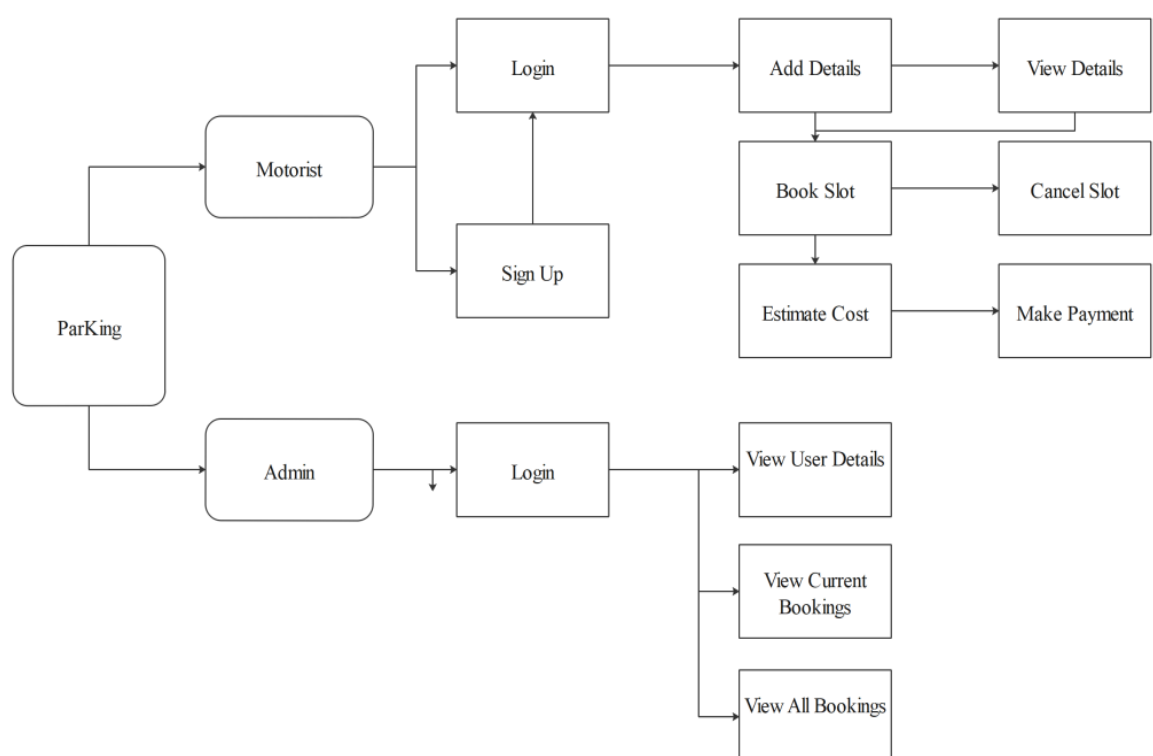


Figure 4.1: Block Diagram of Smart parking system

The Figure 4.1 explains the various tasks that can be perform by user in Mobile application for Smart Parking System.

## **Module Specification**

### **Module 1: Sign in**

In this module the motorist can create there account in smart parking system application by giving there username , password and conform password. After giving the information that they can click on sign in button for account creation in to the Smart parking system application. After creating there respective account they can go to the login page and fill there credentials to enter there respective account.

### **Module 2: Book a slot**

In this module the motorist or user can fill date , time , duration and type of wheeler for the booking of slot.

### **Module 3: Cost Calculation**

In this module after the motorist or user select a slot for booking. The Cost estimation is done for number of hours a parking slot is booked. After paying the cost for booking the slot will be successfully booked.

### **Module 4: Cancel a slot**

In this module motorist or user can cancel the slot which he/she booked earlier by providing the booking id of booking.

## **4.2 System Design:**

### **Object Modelling**

The object model represents the static and most stable phenomena in the modelled domain. Main concepts are classes and associations with attributes and operations. Aggregation and generalization (with multiple inheritance) are predefined relationships

## Class Diagram

Class diagram is a static diagram represents the static view of an application describes the attributes and operations of a class and also the constraints imposed on the system

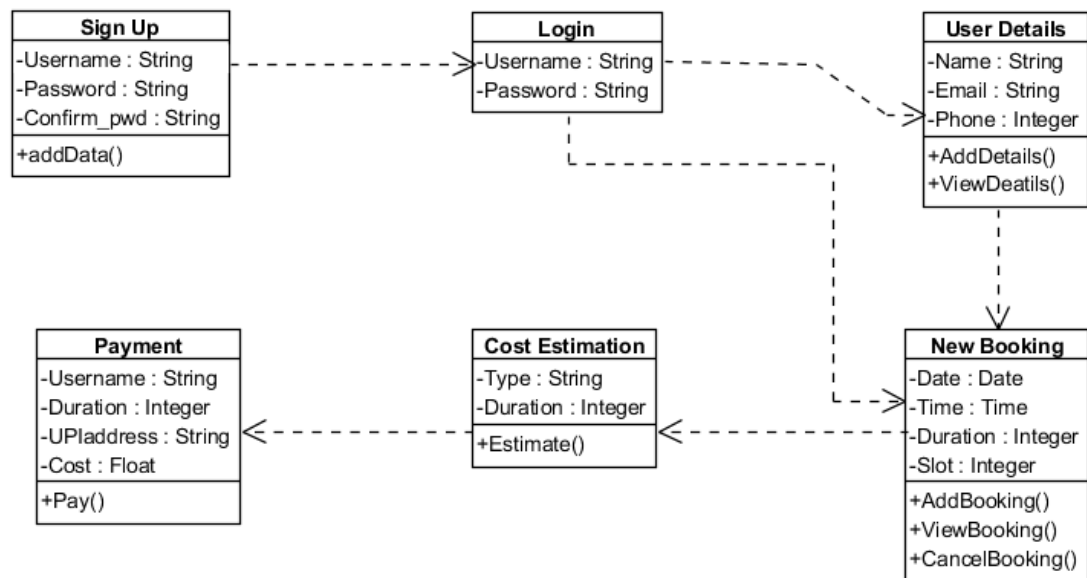


Figure 4.2: Class Diagram

The Figure 4.2 explains the static structure of classifiers in Mobile application for Smart Parking System.

## Dynamic Modelling

A dynamic model represents the behaviour of an object over time. It is used where the object's behaviour is best described as a set of states that occur in a defined sequence.

## Use case diagram

A use case diagram is a dynamic or behaviour diagram in UML. Use case diagrams model the functionality of a system using actors and use cases. Use cases are a set of actions, services, and functions that the system needs to perform.

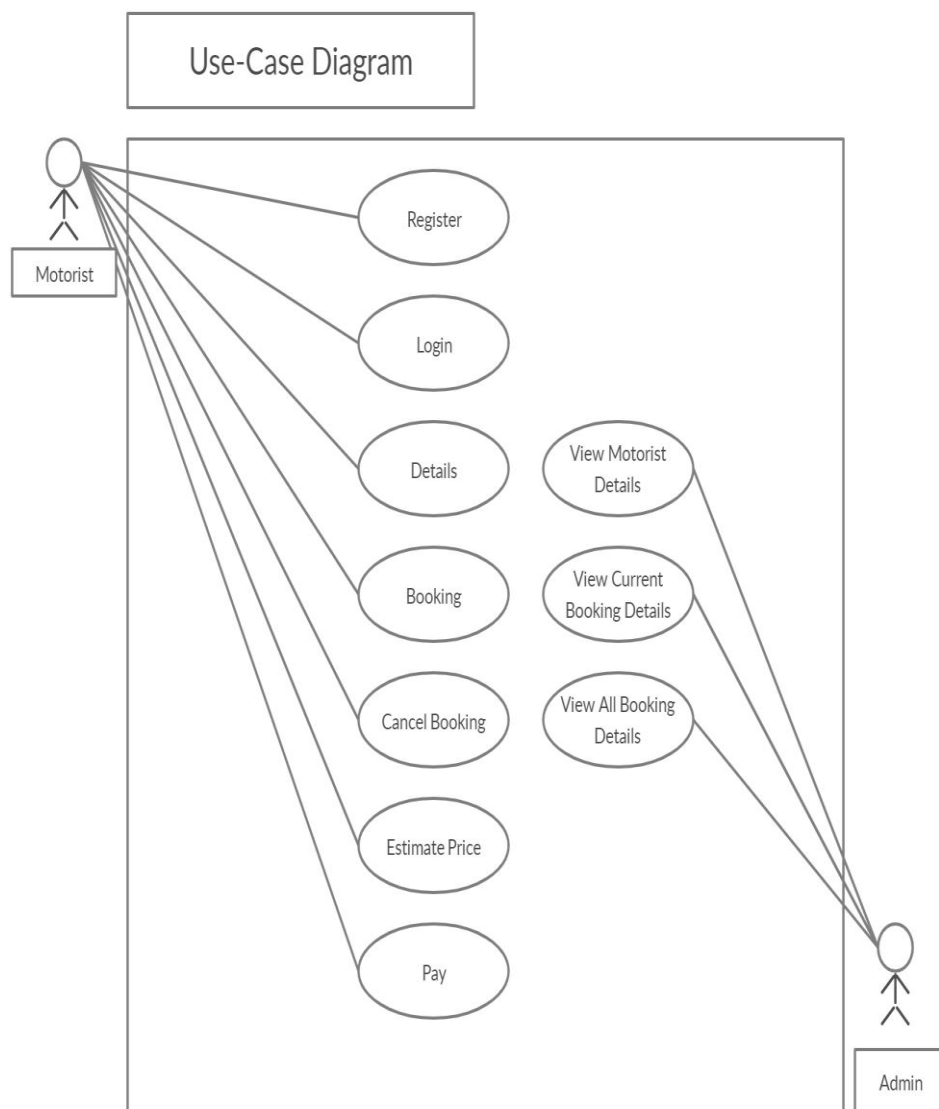


Figure 4.3: Use Case Diagram

The Figure 4.3 explains the context , requirements and validate system architecture in Mobile application for Smart Parking System.

## Database Design

An entity–relationship model describes interrelated things of interest in a specific domain of knowledge. A basic ER model is composed of entity types and specifies relationships that can exist between entities.

## ER Diagram

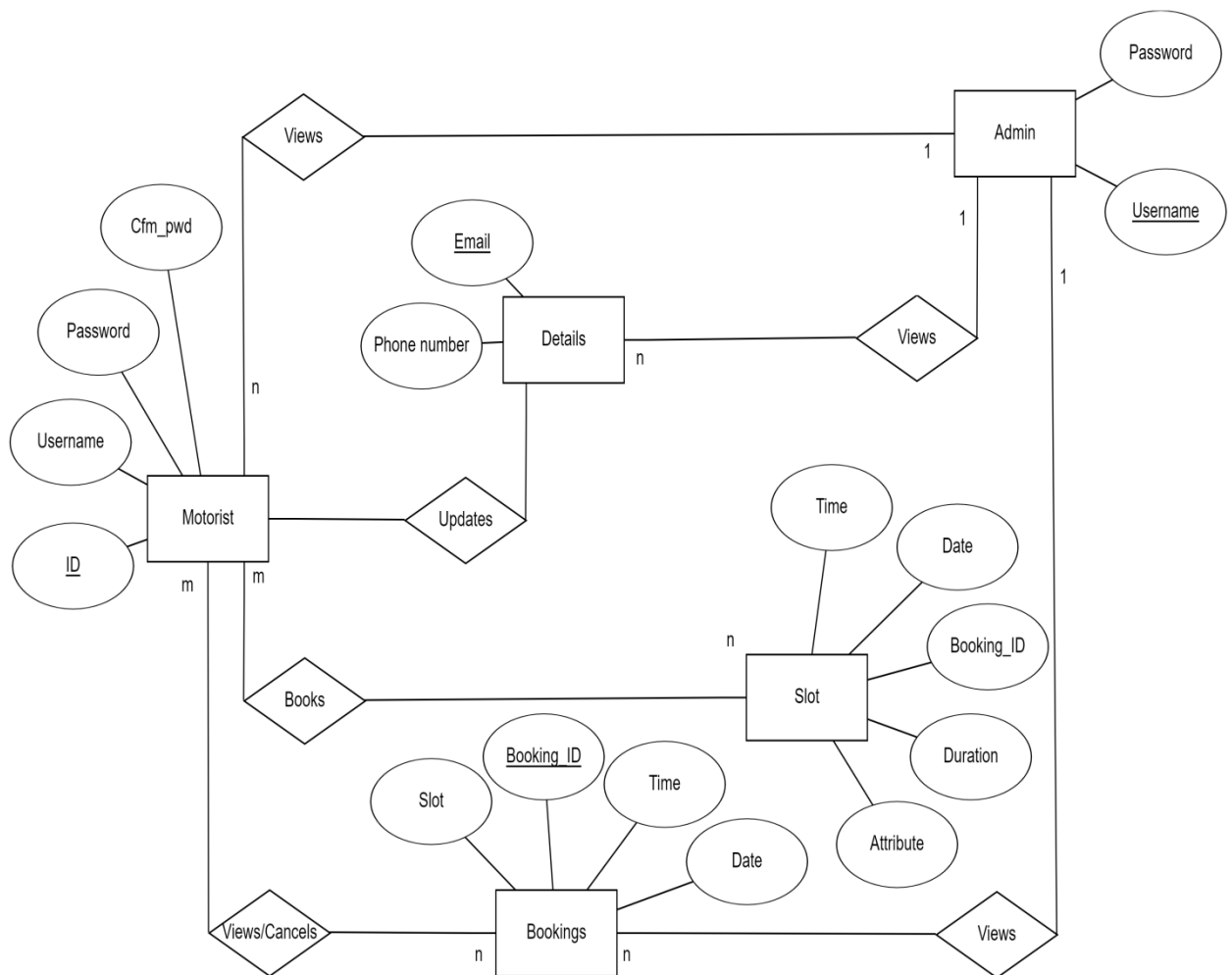


Figure 4.4: ER Diagram

The Figure 4.4 explains the entity types and specifies relationships that can exist between entities of Mobile application for Smart Parking System.

## Chapter 5

### Implementation

#### 5.1 Code for mobile application

##### AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.lucifer.parking">

    <uses-permission
        android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.READ_PHONE_STATE" />
    <uses-permission
        android:name="android.permission.READ_EXTERNAL_STORAGE" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme">
        <activity android:name=".Vehicle4" />
        <activity android:name=".vehicle" />
        <activity
            android:name=".MainActivity"
            android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".SignUPActivity" />
        <activity android:name=".ViewBooking" />
```

```
<activity
    android:name=".MyDetails"
    android:label="@string/title_activity_my_details"
    android:theme="@style/AppTheme.NoActionBar" />
<activity
    android:name=".Home"
    android:label="@string/title_activity_home"
    android:theme="@style/AppTheme.NoActionBar" />
<activity android:name=".Feedback" />
<activity android:name=".Cancel" />
<activity
    android:name=".Create"
    android:label="Create" />
<activity
    android:name=".MyDetails2"
    android:label="Create" />
<activity android:name=".CC" />
<activity android:name=".NewBooking" />
<activity android:name=".EnterInformation" />
<activity android:name=".twowheeler" />
<activity android:name=".fourwheeler" />
<activity
    android:name=".Help"
    android:label="@string/title_activity_help"
    android:theme="@style/AppTheme" />
<activity android:name=".Date" />
</application>

</manifest>
```



**MainActivity.java**

```
package com.lucifer.parking;
import android.app.Activity;
import android.app.Dialog;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class MainActivity extends Activity
{
    Button btnSignIn,btnSignUp;
    LoginDataBaseAdapter loginDataBaseAdapter;
    public static final String MyPREFERENCES = "MyPrefs" ;
    SharedPreferences sharedPreferences;
    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        sharedPreferences = getSharedPreferences(MyPREFERENCES,
Context.MODE_PRIVATE);

        // create a instance of SQLite Database
        loginDataBaseAdapter=new LoginDataBaseAdapter(this);
        loginDataBaseAdapter=loginDataBaseAdapter.open();

        // Get The Reference Of Buttons
        btnSignIn=(Button)findViewById(R.id.buttonSignIn);
        btnSignUp=(Button)findViewById(R.id.buttonSignUp);
```

```
// Set OnClick Listener on SignUp button
btnSignUp.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        // TODO Auto-generated method stub

        /// Create Intent for SignUpActivity abd Start The Activity
        Intent intentSignUp=new
Intent(getApplicationContext(),SignUpActivity.class);
        startActivity(intentSignUp);
    }
});

// Methos to handleClick Event of Sign In Button
public void signIn(View V)
{
    final Dialog dialog = new Dialog(MainActivity.this);
    dialog setContentView(R.layout.login);
    dialog.setTitle("Login");

    // get the Refferences of views
    final EditText
editTextUserName=(EditText)dialog.findViewById(R.id.editTextUserNameToLogin)
;
    final EditText
editTextPassword=(EditText)dialog.findViewById(R.id.editTextPasswordToLogin);

    Button btnSignIn=(Button)dialog.findViewById(R.id.buttonSignIn);

    btnSignIn.setOnClickListener(new View.OnClickListener() {

        public void onClick(View v) {
            String userName=editTextUserName.getText().toString();
            String password=editTextPassword.getText().toString();
```

```
// fetch the Password form database for respective user name
String storedPassword=loginDataBaseAdapter.getSinlgeEntry(userName);
// check if the Stored password matches with Password entered by user
if(password.equals(storedPassword))
{

    Toast.makeText(MainActivity.this, "Congrats: Login Successfull",
Toast.LENGTH_LONG).show();

    dialog.dismiss();
    SharedPreferences.Editor editor = sharedPreferences.edit();

    editor.putString("username", userName);
    editor.commit();
    Intent intentHome=new Intent(getApplicationContext(),CC.class);

    startActivity(intentHome);
}
else
{
    Toast.makeText(MainActivity.this, "User Name or Password does not
match", Toast.LENGTH_LONG).show();
}
}
});

dialog.show();
}

@Override
protected void onDestroy() {
    super.onDestroy();
    loginDataBaseAdapter.close();
}
```

```
}
```

**Activity\_main.xml**

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
```

```
xmlns:tools="http://schemas.android.com/tools"
```

```
android:layout_width="match_parent"
```

```
android:layout_height="match_parent"
```

```
android:orientation="vertical"
```

```
android:gravity="center_vertical"
```

```
    android:background="@drawable/qaz"
```

```
    android:weightSum="1">
```

```
    <Button
```

```
        android:id="@+id/B21"
```

```
        android:layout_width="361dp"
```

```
        android:layout_height="231dp"
```

```
        android:layout_gravity="center"
```

```
        android:background="@drawable/ss_logo" />
```

```
    <Button
```

```
        android:id="@+id/buttonSignIn"
```

```
        android:layout_width="fill_parent"
```

```
        android:layout_height="wrap_content"
```

```
        android:text="Sign In"
```

```
android:onClick="signIn"
```

```
android:textColor="@color/abc_search_url_text_pressed" />
```

```
<Button
```

```
android:id="@+id/buttonSignUp"
```

```
android:layout_width="fill_parent"
```

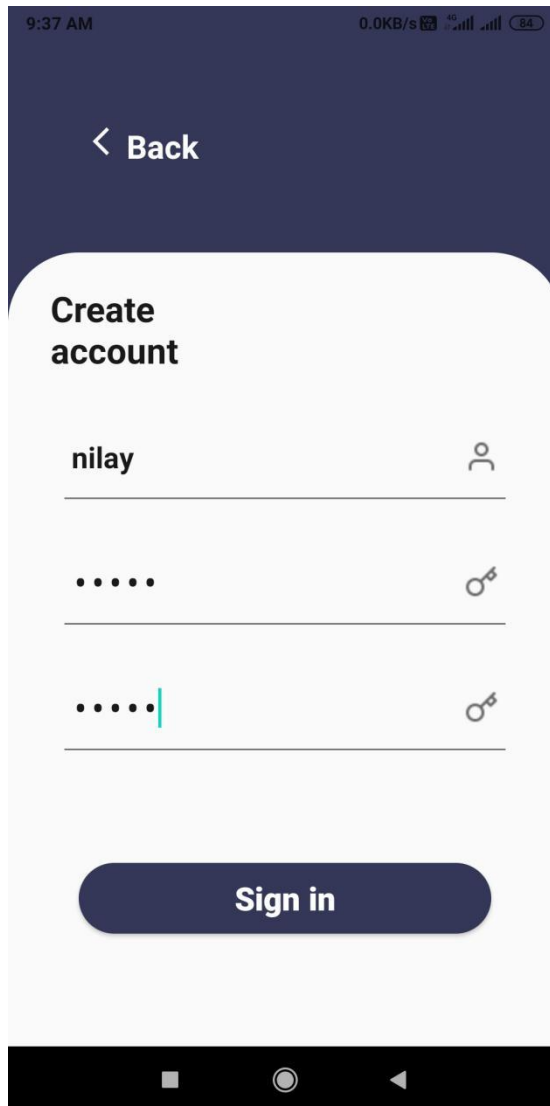
```
android:layout_height="wrap_content"
```

```
android:text="Sign Up"
```

```
android:textColor="@color/abc_search_url_text_pressed" />
```

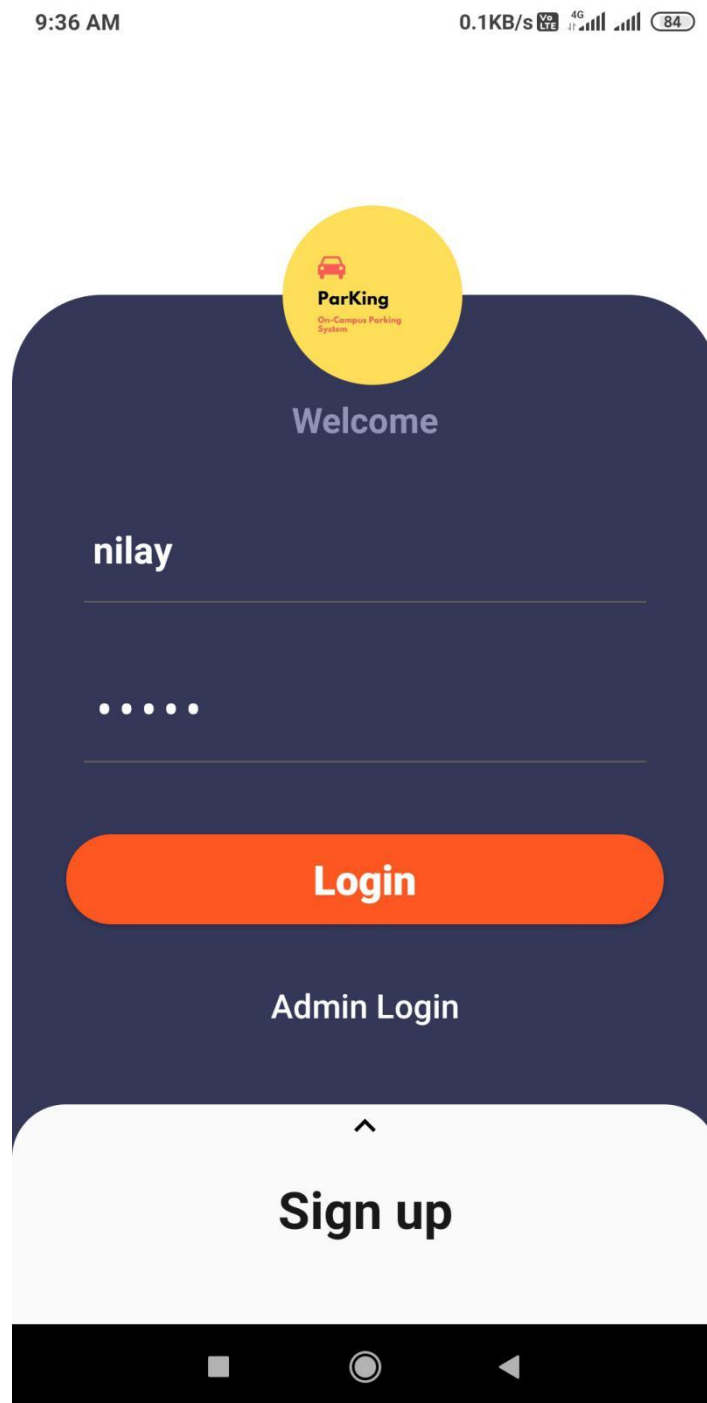
```
</LinearLayout>
```

## 5.2 Screenshots



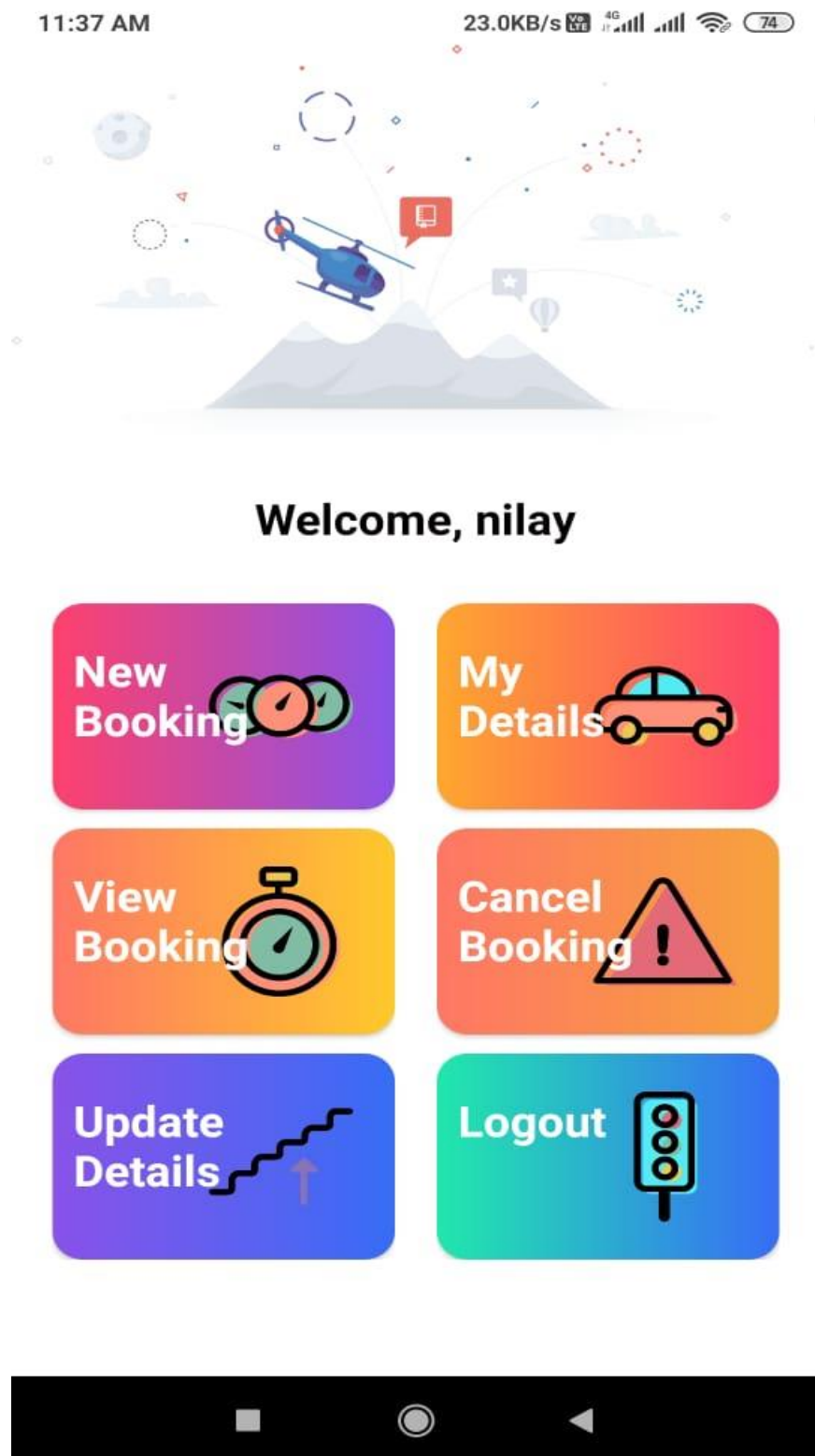
**Figure 5.1 User Sign up**

The Figure 5.1 shows that the motorist can create there account in smart parking system application by giving there username , password and conform password. After giving the information they can click on sign in button for account creation in to the Smart parking system application.



**Figure 5.2 user login**

The Figure 5.2 shows that after account creation the motorist or user can login to there respective account by giving there credentials like username and password. After filling the credentials they just need to click on login button.



**Figure 5.3 welcome page**

The Figure 5.3 shows welcome page which will appear when motorist or user login to there respective account. In this welcome page they get many options like New Booking , My Details , View Booking , Cancel Booking , Update Details and Logout.



9:40 AM 0.0KB/s 4G 84

< Back

**New Booking**

16/7/2020

12:17

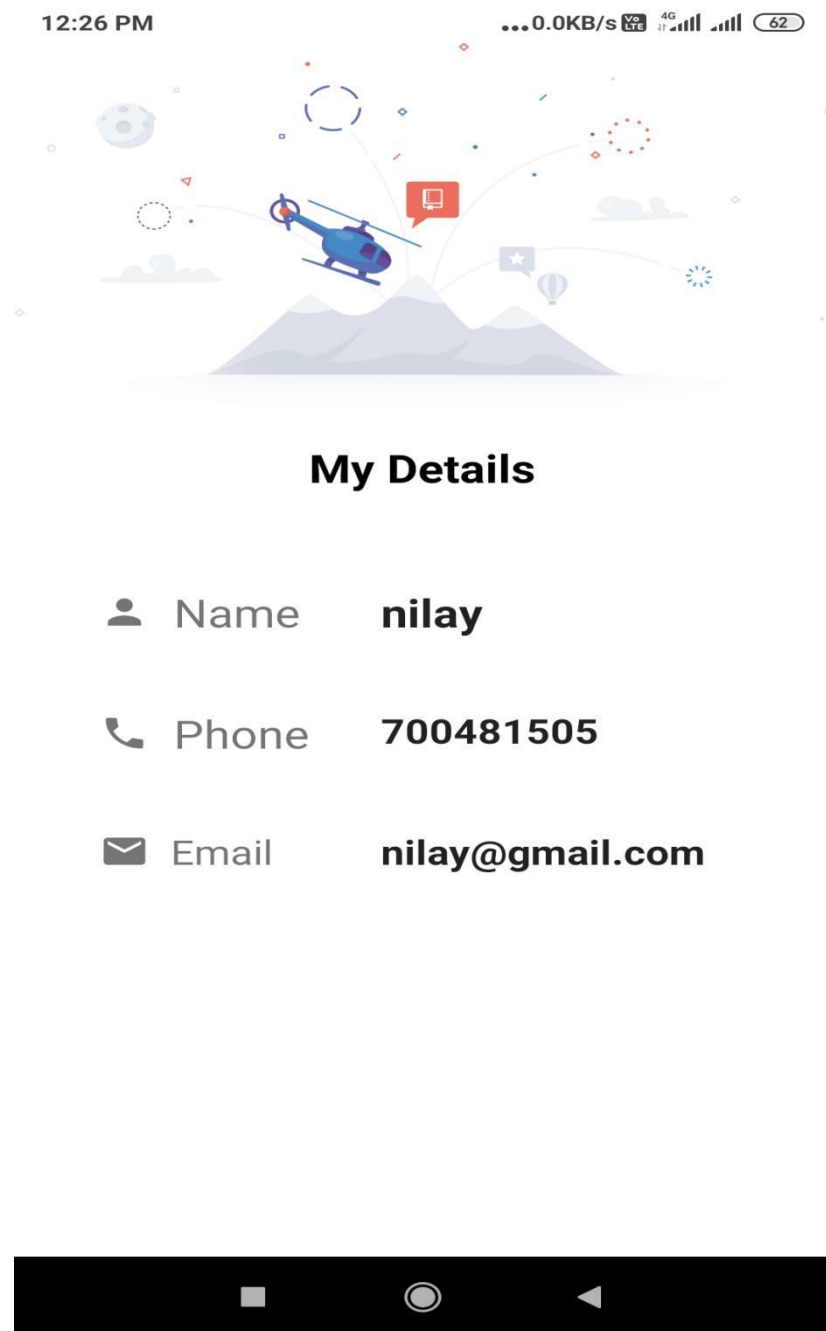
3

2 wheeler

**Search**

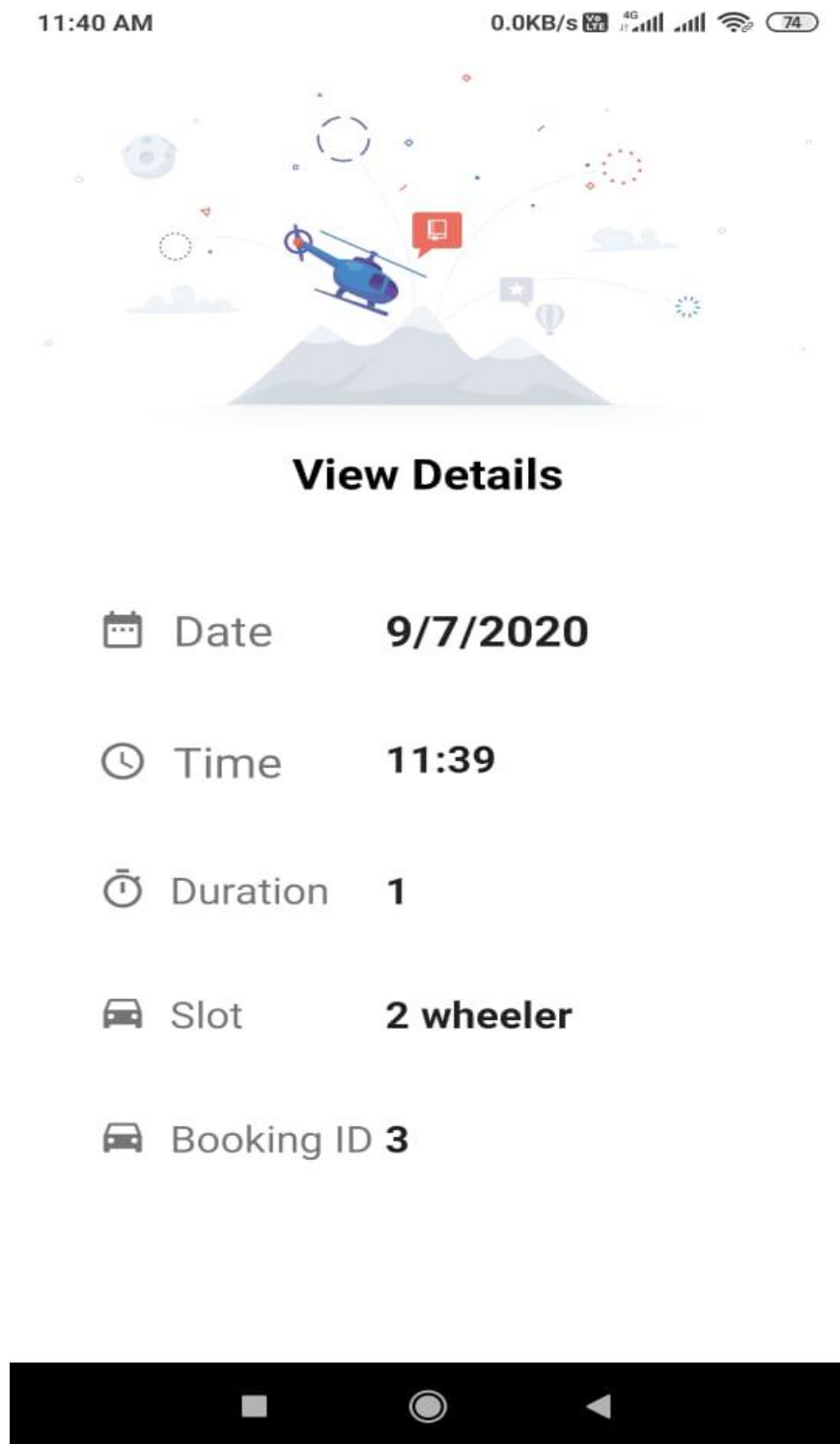
**Figure 5.4 new booking**

The Figure 5.4 shows New Booking option of the welcome page. In this the motorist or user can fill date , time , duration and type of wheeler for the booking of slot.



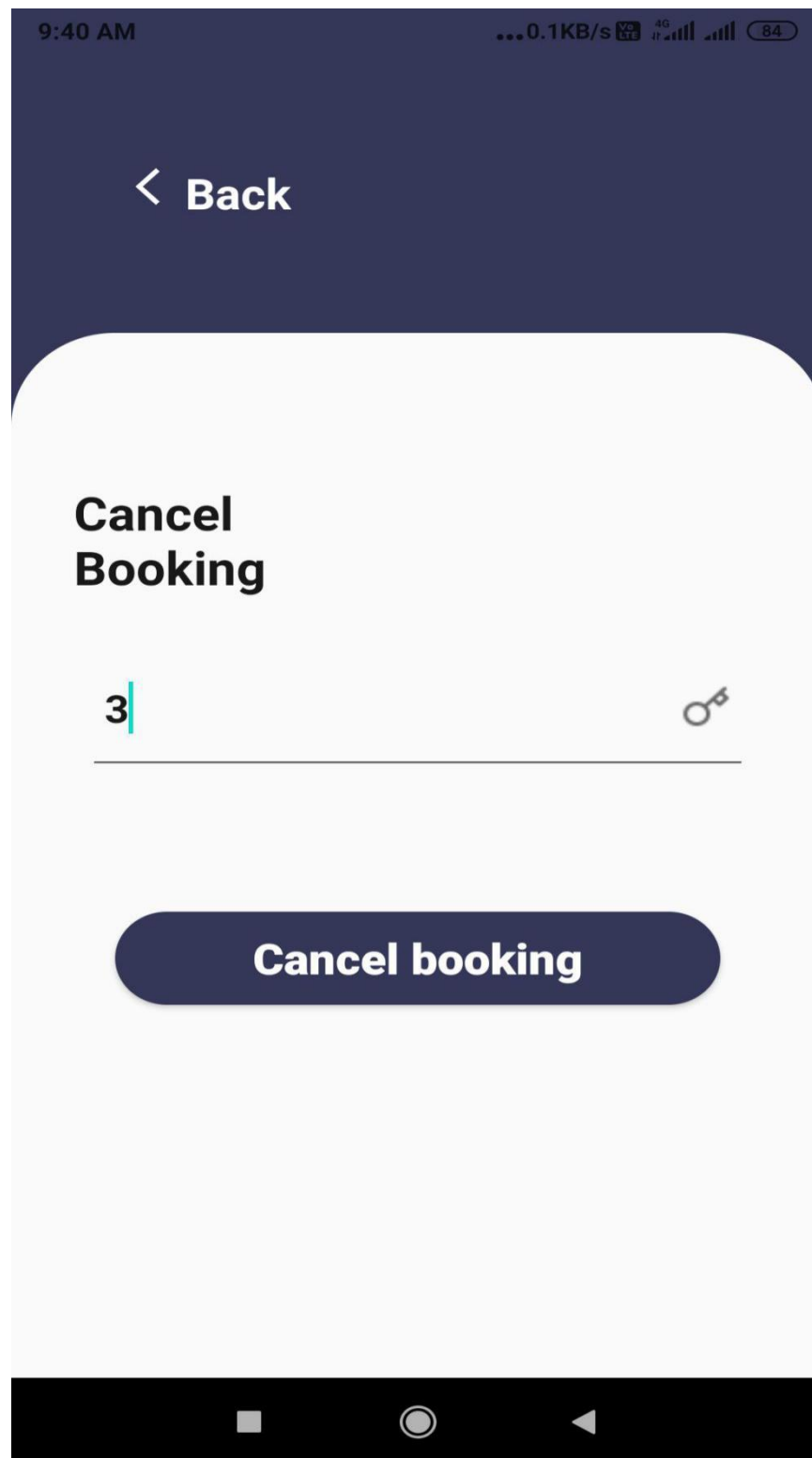
**Figure 5.5 My details**

The Figure 5.5 shows the My details option of the welcome page. In this motorist or user can see there details like name , phone no. and email id .



**Figure 5.6 view booking**

The Figure 5.6 shows the View booking option of the welcome page. In this motorist or user can see the booking details like date , time ,duration , type of wheeler and booking id.




**Figure 5.7 Cancel booking**


The Figure 5.7 shows the Cancel booking option of the welcome page. In this motorist or user can fill booking id to cancel there booking of slot. After filling the booking id they just need to click on the cancel booking button.


9:41 AM ...0.0KB/s 4G 84

< Back

### User Information

nilay 

nilay@gmail.com 

7004801505 

**Update Information**

**Figure 5.8 update details**

The Figure 5.8 shows the update details option of the welcome page. In this motorist or user can update there details like name , email id , phone no. After filling the details they just need to click on update information button.

## Chapter 6

### Testing

This Chapter covers the testing phase of the project. The two types of testing that has been covered in this chapter are Unit testing and Integration testing.

#### 6.1 Unit Testing

**Table 6.1: Test cases for Admin Login**

Test case ID	Description	Sample input	Expected output	Actual output	Result
TC 1	Admin login	Correct username and Correct password	Successful login	Successful login	Pass
TC 2	Admin login	Correct username and Incorrect password	Admin Credentials does not match	Admin Credentials does not match	Pass
TC 3	Admin login	Incorrect username and Correct password	Admin Credentials does not match	Admin Credentials does not match	Pass
TC 4	Admin login	Incorrect username and Incorrect password	Admin Credentials does not match	Admin Credentials does not match	Pass
TC 5	Admin login	Correct username and Vacant password	Admin Credentials does not match	Admin Credentials does not match	Pass
TC 6	Admin login	Vacant username and Correct password	Admin Credentials does not match	Admin Credentials does not match	Pass
TC 7	Admin login	Correct username and correct password	Successful login	Admin Credentials does not match	Fail

**Table 6.2: Test cases for User Login**

Test case ID	Description	Sample input	Expected output	Actual output	Result
TC 1	User login	Correct username and Correct	Successful login	Successful login	Pass

		password			
TC 2	User login	Correct username and Incorrect password	User Name or Password does not match	User Name or Password does not match	Pass
TC 3	User login	Incorrect username and Correct password	User Name or Password does not match	User Name or Password does not match	Pass
TC 4	User login	Incorrect username and Incorrect password	User Name or Password does not match	User Name or Password does not match	Pass
TC 5	User login	Correct username and Vacant password	User Name or Password does not match	User Name or Password does not match	Pass
TC 6	User login	Vacant username and Correct password	User Name or Password does not match	User Name or Password does not match	Pass
TC 7	User login	Correct username and Correct password	Successful login	User Name or Password does not match	Fail

**Table 6.3: Test cases for user registration**

Test case ID	Description	Sample input	Expected output	Actual output	Result
TC 1	User registration	Username , password and confirm password	Successful registered	Successful registered	Pass
TC 2	User registration	Vacant username , password and confirm password	Field vacant	Field vacant	Pass
TC 3	User registration	Username , Vacant password , confirm password	Field vacant	Field vacant	Pass
TC 4	User registration	Username , password and Vacant confirm password	Field vacant	Field vacant	Pass

## 6.2 Integration Testing

**Table 6.4: Test cases for selecting and paying the rent**

Test case ID	Description	Sample input	Expected output	Actual output	Result
TC 1	Booking a slot	Slot selection , rent paid	Slot Successfully booked	Slot successfully booked	Pass
TC 2	Booking a slot	No slot selection , no rent paid	Slot not booked	Slot not booked	Pass
TC 3	Booking a slot	Slot selection , no rent paid	Slot not booked	Slot not booked	Pass
TC 4	Booking a slot	No slot selection , rent paid	Slot not booked	Slot not booked	Pass
TC 5	Booking a slot with network issue	Slot selection , rent paid	Slot Successfully booked	Slot not booked	Fail
TC 6	Booking a slot with bank issue	Slot selection , rent paid	Slot Successfully booked	Slot not booked	Fail
TC 7	Booking a slot with payment application did not reflected the money deduction	Slot selection , rent paid	Slot Successfully booked	Slot not booked	Fail



## **Chapter 7**

### **Conclusion**

“Mobile application for Smart Parking System” has been used in every campus or parking area. Although the project is just a small part of the whole system, but an effort has been made to improve upon the existing GUI (Graphical User Interface) and to make it more attractive and user friendly and has been appreciated by the users who tested it. The Project is user friendly in the sense that there are buttons for navigation to each page and it has benefited the developer in the sense that various concepts of Core Java could be implemented as well as newer concepts of SQL SERVER on the project.

The Mobile application for Smart Parking system is user friendly, any user or motorist can easily find the status (vacant/occupied) of the parking space and also can see the parking details. Starting from coming up with project idea, understanding the requirements and choosing the best technologies for the implementation, all this gave me very good experience and exposure in development of an Android mobile application. Upon completing this project successfully, I got familiar with Android Studio and Database tools usage.

### **Future Enhancement**

This application is an initial step in reaching the effective solution for the daily concern. This project can be extended in multiple ways:

- The project is easily extensible and can be improved by further incremental releases of the same.
- We can extend this project by adding Sensor to it.
- More analysis can be done using the parking history data by which User or motorist can get recommendations or suggestions on parking spaces and their availability trends.
- New modules can be added easily.

## References

- [1] J. Anitha, Y. Thoyajakshi, A. Ramya, V. Sravani, Prashant Kumar “Intelligent Parking System Using Android Application” International Journal of Pure and Applied Mathematics , ISSN: 1311-8080 (printed version); ISSN: 1314-3395 (on-line version) , Volume 114 , No. 7 , Page 165-174 , April 14-15 , 2017
- [2] Kapil Gajbhiye , Nikhil Dhoke , Bhushan Hiwase “Online Vehicle Parking Android Application: Smartpark” International Journal of Innovative Research in Science, Engineering and Technology , ISSN(Online): 2319-8753 ISSN (Print): 2347-6710 , Vol. 7 , Issue 3 , March 2018
- [3] K.Indumathi , G.Mathi Bala , R.Megala “MOBILE APP FOR SMART VEHICLE PARKING SYSTEM” International Journal For Technological Research In Engineering , ISSN (Online): 2347 - 4718 , Volume 5, Issue 7 , March-2018
- [4] ElakyaR, Juhi Seth, Pola Ashritha, R Namith “Smart Parking System using IoT” International Journal of Engineering and Advanced Technology , ISSN: 2249 – 8958, Volume-9 Issue-1, October 2019
- [5] Mr K.Devendran ME , Su.Nivethaa “ANDROID APPLICATION FOR VEHICLE PARKING SYSTEM” International Journal of Scientific & Engineering Research , ISSN 2229-5518 , Volume 8, Issue 7, July-2017
- [6] Mehala Chandran , Nur Fadila Mahrom , Thennarasan Sabapathy , Muzammil Jusoh , Mohd Nasrun Osman , Mohd Najib Yasin , N.A.M Hambali , R.Jamaluddin , N.Ali , Yasmin Abdul Wahab “An IoT Based Smart Parking System” International Conference Computer Science and Engineering , ISSN 1742-6596 , Volume 1339 , 16 December 2019
- [7] J. Cynthia, C. Bharathi Priya, P. A. Gopinath “IOT based Smart Parking Management System” International Journal of Recent Technology and Engineering , ISSN: 2277-3878 , Volume-7 , Issue-4S , November 2018
- [8] Hina C. Parmar, Nisha N. Shirvi “Development of an Android Application for Smart Parking System” International Journal of Engineering Development and Research , ISSN: 2321-9939 , Volume 6 , Issue 2 , 2018

- [9] Tejal Lotlikar , Minla Chandrahasan , Ankita Mahadik , Madhusmita Oke , Anjali Yeole “Smart Parking Application” International Journal of Computer Applications , ISSN 0975 – 8887 , Volume 149 – No.9, September 2016
- [10] Supriya Gatalwar, Radhika Agnihotri, Nitesh Gujarathi, Atmesh Behere “ParkSmart: Android Application for Parking System” International Journal of Computer Science and Network , ISSN (Online) : 2277-5420 , Volume 5, Issue 1, February 2016
- [11] Aekarat Saeliw , Watcharasuda Hualkasin, Supattra Puttinaovarat “Smart Car Parking Mobile Application based on RFID and IoT” International Journal of Interactive Mobile Technologies , ISSN: 1865-7923 , Vol. 13, No. 5, 2019
- [12] Abhirup Khanna , Rishi Anand “IoT based Smart Parking System” International Conference on Internet of Things and Applications , ISBN: 978-1-5090-0044-9 , 22-24 Jan. 2016
- [13] Rick Boyer “Android 9 Development Cookbook” Third Edition, Packt Publishing Ltd , ISBN 978-1-78899-121-6 , October 2018
- [14] Anusha , Arshitha M S , Anushri , Geetanjali Bishtannavar “Review Paper on Smart Parking System” International Journal of Engineering Research & Technology , ISSN: 2278-0181 , Volume 7, Issue 08 , Special Issue - 2019
- [15] Jayakshei Dadaji Bachhav, Prof. Mechkul M. A. “Smart Car Parking System” International Research Journal of Engineering and Technology , ISSN: 2395 -0056 , Volume: 04 Issue: 06 , June -2017
- [16] Vishwanath Y , Aishwarya D Kuchalli , Debarupa Rakshit “Survey paper on Smart Parking System based on Internet of Things” International Journal of Recent Trends in Engineering & Research , ISSN: 2455-1457 , Volume 02 , Issue 03; March - 2016