



**ESCORT ON SPOT
(P.M.S)**

PROJECT REPORT ON**ESCORT ON SPOT.COM****SUBMITTED TO****Gujarat Technical University**

OF
DIPLOMA IN COMPUTER ENGINEERING
IN YEAR 2020-2021
BY

Sr No.	Enrollment No.	Name
1.	186400307009	Shashank M. Chaudhary.
2.	186400307029	Dhruv K. Joshi
3.	186400307052	Nehal O. Bachani



R.C Technical Institute

Sola, Ahmedabad - 60

CERTIFICATE

This is to certify that this work of **PROJECT-2** Subject & **3360707** Subject Code of **6th** Sem with title: **ESCORT ON SPOT (PMS)** represents the work of the following students for the fulfillment of the Certificate of Diploma in Computer Engineering at R.C Technical Institute Sola, Ahmedabad - 60, Gujarat, during the academic year **2020-2021** and the work is completed and found satisfactory.

Sr No.	Enrollment No.	Name
1.	186400307009	Shashank M. Chaudhary.
2.	186400307029	Dhruv K. Joshi
3.	186400307052	Nehal O. Bachani

Submitted to: **Hardik N. Talsania**

Lecturer, Computer Engg. Dept. RCTI

Head of Computer Dept.
RCTI

ACKNOWLEDGEMENT

We would like to express our Special thanks of gratitude to our faculty and guide **HARDIK N. TALSANIA** and head of the department **SANJAY B. DHAMI** Who gave us the golden opportunity to do this wonderful project on this topic **ESCORT ON SPOT (PMS)**, which also helped us in doing a lot of research and came to know about so many new things we are thankful to them. Secondly, we would also like to thank our friends who helped us in finishing this project within the limited time. And a very special thanks to all our faculties who has given us the guidance to complete the project. We are making this project to increase our knowledge and to learn some new things.

THANKS AGAIN TO ALL WHO HELPED US

REGARDS

Chaudhary Shashank M.(186400307009)

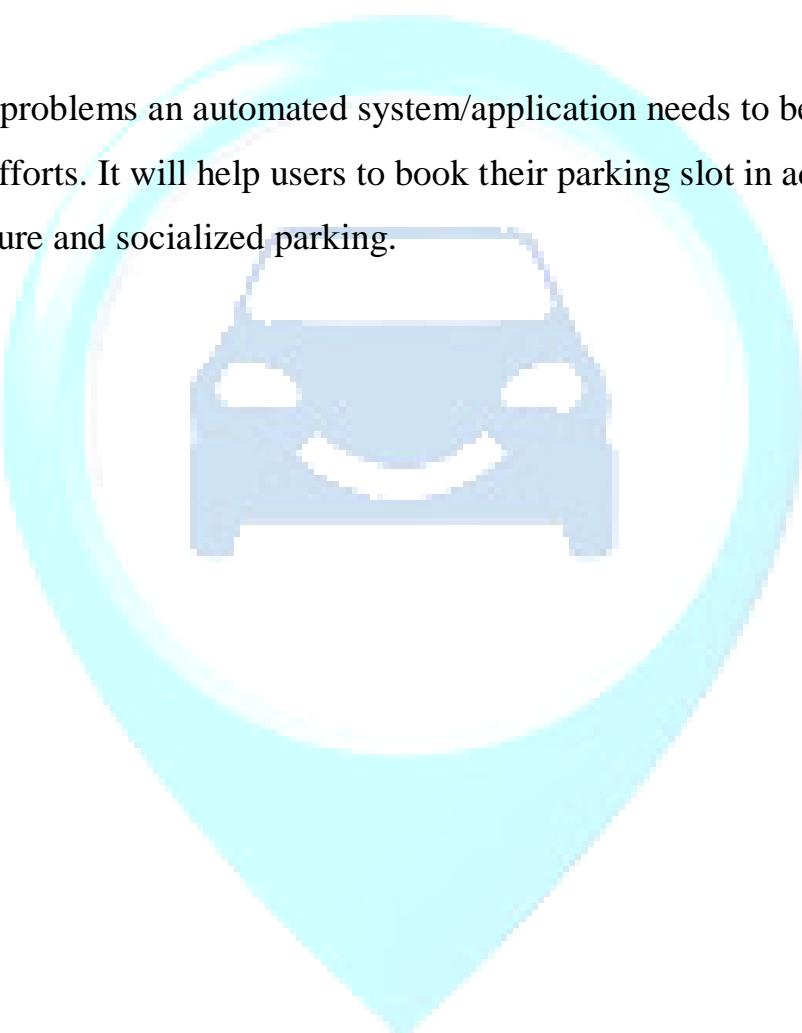
Joshi Dhruv K.(186400307029)

Bachani Nehal O.(186400307052)

ABSTRACT

This is an Android-Based project for Parking Management System. ***“33% of the urban traffic in India is actively seeking for a parking space”***. Our project exactly focuses on this problem and concentrates on making the current situation simple and efficient.

To overcome these problems an automated system/application needs to be developed to reduce the human efforts. It will help users to book their parking slot in advance, it is time saver and it is a secure and socialized parking.



List of Figures

SR.NO.	FIGURES NAME	PAGE NO.
1.	ITERATIVE WATERFALL MODEL	20
2.	GANTT CHART	22
3.	ADMIN ACTIVITY DIAGRAM	37
4.	USER ACTIVITY DIAGRAM	38
5.	ADMIN USE-CASE DIAGRAM	39
6.	USER USE-CASE DIAGRAM	40
7.	ADMIN SEQUENCE DIAGRAM	41
8.	USER SEQUENCE DIAGRAM	42
9.	E-R DIAGRAM	47
10.	DATA FLOW DIAGRAM (0 LEVEL)	48
11.	ADMIN DATA FLOW DIAGRAM (1 LEVEL)	48
12.	USER DATA FLOW DIAGRAM (1 LEVEL)	48
13.	ADMIN FLOWCHART	49
14.	USER FLOWCHART	50
15.	WORKING DIAGRAM OF SYSTEM	51
16.	MAIN PAGE DESIGN	54
17.	INTRODUCTION SPLASH SCREEN DESIGN	55
18.	HOME PAGE DESIGN	56
19.	VEHICLE TYPE SELECTION PAGE DESIGN	57
20.	INFORMATION TAKING PAGE DESIGN	58
21.	INSTRUCTION DIALOG DESIGN	59

22.	ALERT DIALOG DESIGN	60
23.	QR CODE GENERATOR PAGE DESIGN	61
24.	APPROVAL STATUS PAGE DESIGN	62
25.	APPROVAL STATUS ACCEPTED PAGE DESIGN	62
26.	APPROVAL STATUS REJECTED PAGE DESIGN	63
27.	APPROVAL STATUS PENDING PAGE DESIGN	63
28.	APPROVAL STATUS CHECKED-OUT PAGE DESIGN	64
29.	APPROVAL STATUS NOT FOUND PAGE DESIGN	64
30.	MAIN HOME PAGE DESIGN (Qr Scanner App)	65
31.	SCANNING PAGE DESIGN	66
32.	VALID QR CODE RESPONSE DESIGN	67
33.	INVALID QR CODE RESPONSE DESIGN	68
34.	MAIN HOME PAGE DESIGN (Admin Web-Site)	69
35.	ADMIN LOGIN PAGE DESIGN	70
36.	DASHBOARD PAGE DESIGN	71
37.	ADD ADMIN PAGE DESIGN	72
38.	ADMIN LIST PAGE DESIGN	73
39.	PENDING REQUESTS PAGE DESIGN	74
40.	PARKED VEHICLE PAGE DESIGN	75
41.	DASHBOARD PAGE DESIGN (2 nd Time)	76
42.	HISTORY PAGE DESIGN	77
43.	FEEDBACK & COMPLAINT PAGE DESIGN	78
44.	FEEDBACK RESPONSE MAIL	79
45.	BLACK BOX TESTING	91

LIST OF TABLES

SR.NO.	FIGURES NAME	PAGE NO.
1.	Roles and Responsibilities	21
2.	Admin Master	44
3.	Accept Master	44
4.	Pending Requests	45
5.	History Master	45
6.	Parkings Master	46
7.	Feedback Master	46
8.	Admin Test-Cases	92
9.	User Test-Cases	94
10.	QR Code Application Test-Cases	96
11.	Arduino Circuit Test-Cases	97

INDEX

1.0	Introduction	11
1.1	Project Summary & Profile	12
1.2	Purpose	13
1.3	Scope & Objectives	14
1.4	Technologies	15
2.0	Project Management	18
2.1	Project Planning	19
2.1.1	Project Development Approach & Planning	19
2.1.2	Roles and Responsibilities	21
2.2	Project Scheduling	22
2.3	Risk Management	23
3.0	System Requirement Study	26
3.1	Existing System	27
3.2	Proposed System	28
3.2.1	Modules and features of the new system	28
3.2.2	User Characteristics	32
3.2.3	Hardware and Software Requirements	33
4.0	System Analysis	34
4.1	Feasibility Study	35
4.2	System Activity Diagram	37
4.3	Use Case Diagram	39
4.4	Sequence Diagram	41
5.0	System Design	43
5.1	Database Design / Data Structure Design	44
5.1.1	Data Dictionary	44
5.1.2	E.R Diagram	47
5.1.3	Data flow Diagram	48
5.2	Input / Output and Interface Design	49
5.2.1	Flow Chart	49
5.2.2	Working Diagram of the System.	51

6.0	Implementation	52
6.1	Implementation Environment	53
6.2	Screen-Shots of Different Modules	54
6.3	Sample Coding	80
7.0	Testing	88
7.1	Testing Planning and Strategy	89
7.2	Testing Methods	91
7.3	Testing Cases	92
8.0	Future Planning	98
9.0	Conclusion	99
10.0	REFERENCE & BIBLIOGRAPHY	100

1.0 Introduction

1.1 Project Summary & Profile

1.2 Purpose

1.3 Scope & Objectives

1.4 Technologies



1.1 Project Summary and Profile (Details of Project)

The main functionality of our system is that the user can book the advance parking slot for their vehicle in our Parking and if all the slots of our parking is full then the user will also get the list of the nearest parkings.

Users will get options of 3 types of vehicles for the slot booking and choose their appropriate vehicle type before further process. The 3 types are given below:

1. 2 Wheeler (Bike, Scooty, etc)
2. 3 Wheeler (Rickshaw, tempo, etc)
3. 4 Wheeler (Car)

Users need to add their Vehicle No and Vehicle RC Book with their request for booking their parking slot. Once the user places the request it will go to the admin and the admin approves the request if the given Vehicle Number matches with the vehicle number in RC Book.

After reaching the parking destination user have to scan their QR code in the QR Code scanner device and the barrier will get opens up.

1.2 Purpose

The main purpose of the system is to reduce the urban traffic in our country. By using these system one can easily book their parking slot in advance without being waiting in a queue for parking their vehicle and searching the parking slots for their vehicles as "**33% of the urban traffic in India is actively seeking for a parking space**".

Nowadays, All the details and slot information of the parking are stored on paper and the user also gets the paper voucher for their parking booking. So, it may possible that papers are lost or destroyed because of some reasons.

So, our system provides all the details on the admin device and users can also book their parking slot sitting at home and get all the information regarding their slot booking on their device. And this process is secure.

All the paper-works are converted to online and it is very easy to maintain.

1.3 Scopes and Objectives

Our Project Scope is limited to android phones. Only android users can access our Android Application.

And all the Android users will get the following features listed below:-

1. Login and Registration

- Admin can login into their website and manage all the users' slots.
- Admin can approve or reject the requests sent by the users for booking.

2. Booking Slots

- Users can choose their vehicle type and book their parking slot.
- User need to enter their vehicle number and attach their RC Book before sending the request for the slot booking.
- User can generate their QR Code.
- After generating QR Code, users can take Screen-Shot of the QR Code and scan it after reaching the parking destination.

3. Approval Status

- User can see the Approval Status of their Vehicle.
- Here User can see if the Admin has "Approved" their Request or "Rejected" their request or their request is "Under Scrutiny" or their Vehicle has already "Checked-Out".

4. Alternate Parking

- User will get the lists of the nearest parkings with location on Google Map if our parking is full.

5. Complain box and Feedback

- Users can Complain about their issues to the admin.
- Users can also give some feedback or suggestions about the system.

1.4 Technologies

Users :-

Front end:- Java Programming(Android)

Back end:- MySQL

Platform :- Windows

Tools:- Android Studio

Java

- Java is a general purpose computer-programming language that is concurrent, class-based, object-oriented, and specially designed to have as few implementation dependencies as possible.
- It is intended to let application developers “write once, run anywhere” meaning that compiled java code can run on all platforms that support Java without the need of recompilation.
- Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of computer architecture.

Android

- Android is a mobile operating system based on the Linux kernel developed by Google.
- Android is an open platform and it's open source.
- Android is designed primarily for touchscreen mobile devices such as smartphones and tablet computers, with a specialized user interface for television, car, and wristwatches.
- Android is the most widely used operating system on a mobile phone today.

MySQL

- MySQL is the world's most popular open source database.
- Oracle drives MySQL innovation, delivering new capabilities to power next generation web, cloud, mobile and embedded applications.

 **Admin :-**

Front end:- HTML,CSS,PHP

Back end:- MySQL

Platform :- Windows

Tools :- Sublime Text, XAMPP

HTML and CSS

- HTML is for adding meaning to raw content by marking it up.
- CSS is for formatting that marked up content.
- HTML and CSS is mainly used for designing the website.

PHP

- PHP stands for HyperText Preprocessor.
- PHP is a server-side scripting language. I.e PHP is executed on server and output is returned to the client in plain HTML format.
- It's basic features is that it is free open-source, platform independent, compatible with all servers, supports a wide range of databases and it can easily be integrated with HTML and CSS.

MySQL

- MySQL is the world's most popular open source database.
- Oracle drives MySQL innovation, delivering new capabilities to power next generation web, cloud, mobile and embedded applications.

 **Model :-**

Front end:- Arduino
Back end:- MySQL
Platform :- Windows
Tools :- Arduino IDE

Arduino

- Arduino is an open-source electronics platform based on easy-to-use hardware and software.
- Arduino boards are able to read input in form of sensors or from android applications and returns output in form of activating motors, publishing something online etc.
- Users can tell the board what to do by setting a set of instructions to the microcontroller of the board.

MySQL

- MySQL is the world's most popular open source database.
- Oracle drives MySQL innovation, delivering new capabilities to power next generation web, cloud, mobile and embedded applications.

2.0 PROJECT MANAGEMENT

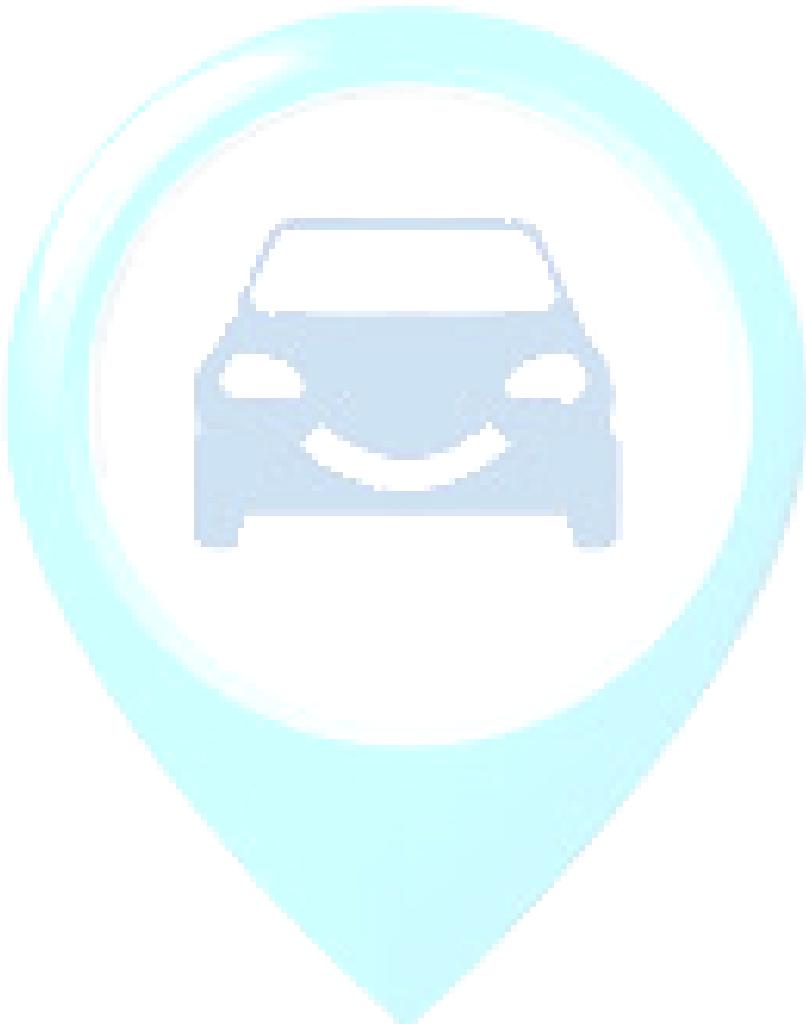
2.1 PROJECT PLANNING

2.1.1 Project Development Approach and Planning

2.1.2 Roles and Responsibilities

2.2 PROJECT SCHEDULING

2.3 RISK MANAGEMENT



2.1 Project Planning

Project planning is one of the major tasks that are performed during the development of the project. Using project planning, the task of finding the size of the project is done and with that total amount of time and cost required for the project is calculated.

The approach to developing the software system should follow some systematic way i.e. Software Development Life Cycle. Using the upper-level analysis and the environment of the project, which lifecycle model would fit properly for this project was judged. After deciding the proper software development lifecycle model, the development of this project according to the model was done.

2.1.1 Project Development Approach and Planning

- How to choose the right approach for a project is a large topic. The methodology you choose can depend on many things, including the structure and location of the project team, the technologies being used on the project, and the degree to which collaboration is a part of the company's culture.
- The project is done based on the decided development life cycle model. We decide on the Iterative Waterfall Model for our application.

Iterative Waterfall Model

In a practical software development project, the “Classical Waterfall Model” is hard to use. So, the Iterative waterfall model can be thought of as incorporating the necessary changes to the classical waterfall model to make it usable in practical software development projects. It is almost the same as the classical waterfall model except some changes are made to increase the efficiency of software development.

“The iterative waterfall model provides feedback paths from every phase to its preceding phases, which is the main difference from the classical waterfall model.”

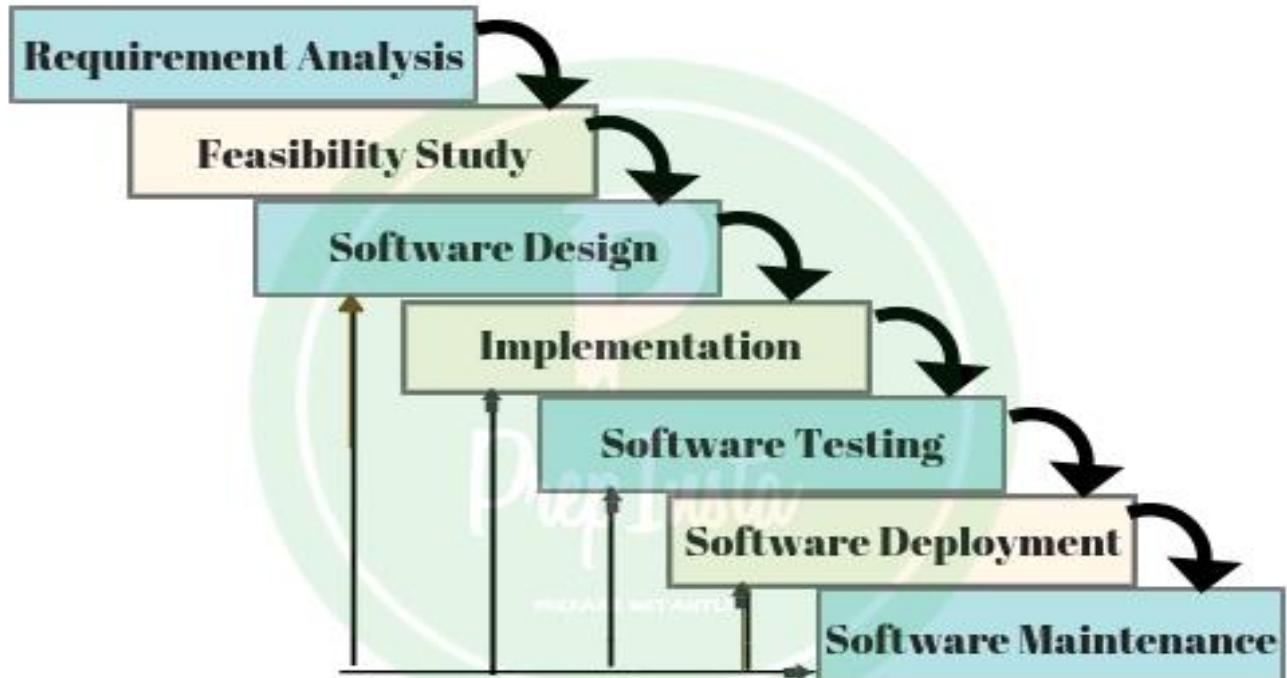


Fig 2.1 Iterative Waterfall Model

When errors are detected at some later phase, these feedback paths allow correcting errors committed by programmers during some phases. The feedback paths allow the phase to be reworked in which errors are committed and these changes are reflected in the later phases. But there is no feedback path to the stage – feasibility study, because once a project has been taken, it does not give up the project easily. It is good to detect errors in the same phase in which they are committed. It reduces the effort and time required to correct the errors.

Phase Containment of Errors: The principle of detecting errors as close to their points of commitment as possible is known as Phase containment of errors.

Advantages of Iterative Waterfall Model

- **Feedback Path:** In the classical waterfall model, there are no feedback paths, so there is no mechanism for error correction. But in iterative waterfall model feedback path from one phase to its preceding phase allows correcting the errors that are committed and these changes are reflected in the later phases.
 - **Simple:** The Iterative waterfall model is very simple to understand and use. That's why it is one of the most widely used software development models.

Drawbacks of Iterative Waterfall Model

- **Difficult to incorporate change requests:** The major drawback of the iterative waterfall model is that all the requirements must be clearly stated before starting the development phase. Customers may change requirements after some time, but the iterative waterfall model does not leave any scope to incorporate change requests that are made after the development phase starts.
- **Incremental delivery not supported:** In the iterative waterfall model, the full software is completely developed and tested before delivery to the customer. There is no scope for any intermediate delivery. So, customers must wait a long for getting the software.
- **Overlapping of phases not supported:** Iterative waterfall model assumes that one phase can start after completion of the previous phase, but in real projects, phases may overlap to reduce the effort and time needed to complete the project.
- **Risk handling not supported:** Projects may suffer from various types of risks. But the Iterative waterfall model has no mechanism for risk handling.
- **Limited customer interactions:** Customer interaction occurs at the start of the project at the time of requirement gathering and at project completion at the time of software delivery. These fewer interactions with the customers may lead to many problems as the finally developed software may differ from the customers' actual requirements.

2.1.2 Roles and Responsibilities (*of all the Members*)

Our system was decomposed into different modules and we are responsible persons for analysis, design, and implementation, documentation along with testing.

Table 2.1 Roles and Responsibilities

ACTIVITIES	ROLES AND RESPONSIBILITIES
Requirement Gathering	Dhruv Joshi, Nehal Bachani
Analysis	Dhruv Joshi, Nehal Bachani
Design	Nehal Bachani, Shashank Chaudhary
Data Dictionary	Shashank Chaudhary, Dhruv Joshi
Documentation	Nehal Bachani, Shashank Chaudhary, Dhruv Joshi
Implementation and Coding	Shashank Chaudhary

2.2 Project Scheduling

Project scheduling consists of identifying the tasks needed to complete the project, determine the dependency among different tasks, plan the starting and ending dates for various tasks and determine the chain of tasks that determine the duration of projecting scheduling. We decide the order in which to do the tasks.

ID	Task Name	Start	Finish	Jul 2020	Aug 2020	Sep 2020	Oct 2020
1	Project Definition	02/07/2020	20/07/2020				
2	Requirement Gathering & Analysis	20/07/2020	04/08/2020				
3	Development Approach	08/08/2020	05/09/2020				
4	System Analysis	06/09/2020	10/10/2020				
5	Project Documentation	02/07/2020	15/10/2020				

Fig 2.2(1) Gantt Chart

ID	Task Name	Start	Finish	Dec 2020	Jan 2021	Feb 2021	Mar 2021	Apr 2021	May 2021
1	Design	19/12/2020	16/01/2021						
2	Implementation & Coding	16/01/2021	20/03/2021						
3	Testing	20/03/2021	03/04/2021						
4	Correction	17/04/2021	30/04/2021						
5	Project Documentation	19/12/2020	01/05/2021						

Fig 2.2(2) Gantt Chart

2.3 Risk Management

Risk Identification

Risk is an inevitable concept of the softer project and it mainly concerns future happenings. We want to produce a well working system; therefore, we must consider all the possible defects and unexpected conditions.

Following are the possible risks for our project:

- Human risk: Potential losses due to a human error, done willingly or unconsciously.
- IT/System risk: Potential losses due to system failures and programming errors.
- Processes risk: Potential losses due to improper information processing, leaking or hacking of information and inaccuracy of data processing.

Risk Analysis

Risk analysis is a review of the risks associated with an event or action. It is applied to projects, information technology, security issues and any action where risks may be analyzed on a quantitative and qualitative basis.

Risk Planning

A risk is a list of all risks that threaten the project, along with a plan to mitigate some or all those risks. If there were no uncertainty, then every project plan would be accurate, and every project would go off without a hitch. Usually at the most inconvenient times. The risk plan is an insurance policy against uncertainty.

Higher the mitigation response is the action plan to eliminate, reduce or minimize the probability of a risk event occurring and or the impact of the project risk event should it occur.

Here are several reasons why risks may occur within a typical project:

- Changing circumstances, updates, and new developments.
- Detailed elaboration and further refinement of the overall project planning process.
- Changes to the constraints (like scope, time and budget).
- Changes to the initial project documentation (like Project Charter) resulted from further discussions and negotiations with the stakeholders.

All risk management processes follow the same basic steps, although sometimes different jargon is used to describe these steps. Together these 5 risk management process steps combine to deliver a simple and effective risk management process.

Step 1: Identify the Risk. You and your team uncover, recognize and describe risks that might affect your project or its outcomes. There are several techniques you can use to find project risks. During this step, you start to prepare your Project Risk Register.

Step 2: Analyse the risk. Once risks are identified you determine the likelihood and consequence of each risk. You develop an understanding of the nature of the risk and its potential to affect project goals and objectives. This information is also input to your Project Risk Register.

Step 3: Evaluate or Rank the Risk. You evaluate or rank the risk by determining the risk magnitude, which is the combination of likelihood and consequence. You make decisions about whether the risk is acceptable or whether it is serious enough to warrant treatment. These risk rankings are also added to your Project Risk Register.

Step 4: Treat the Risk. This is also referred to as Risk Response Planning. During this step, you assess your highest ranked risks and set out a plan to treat or modify these risks to achieve acceptable risk levels. How can you minimize the probability of the negative risks as well as enhancing the opportunities? You create risk mitigation strategies, preventive plans and contingency plans in this step. And you add the risk treatment measures for the highest ranking or most serious risks to your Project Risk Register.

Step 5: Monitor and Review the risk. This is the step where you take your Project Risk Register and use it to monitor, track and review risks.

3.0 SYSTEM REQUIREMENT STUDY

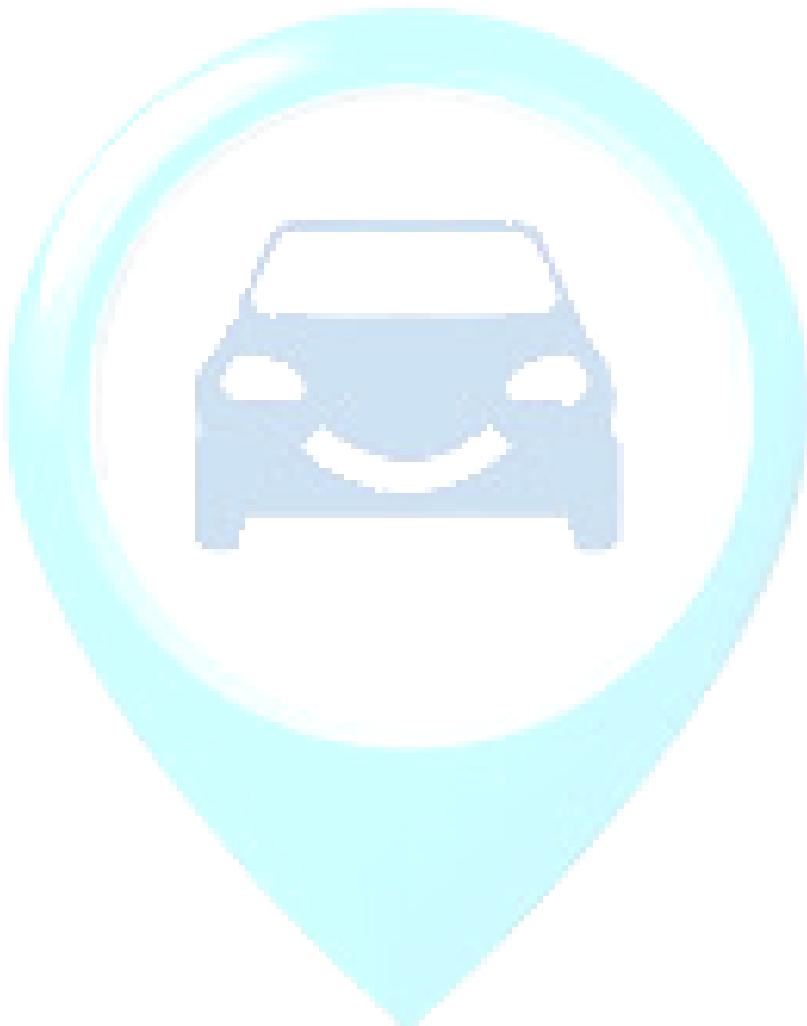
3.1 EXISTING SYSTEM / SCENARIO

3.2 PROPOSED SYSTEM (OUR PLANNED SYSTEM)

3.2.1 Modules and Features in the new System

3.2.2 User Characteristics (*Types of users dealing with the system*)

3.2.3 Hardware and Software Requirements (*minimum required to run your system*)



3.1 EXISTING SYSTEM/ SCENARIO (*Problems and how they can be solved*)

Existing System

- Currently, The system is time-consuming.
- The system is totally manual.
- The manual system is slow than the computerized system.
- It is a very complex system.
- It requires more clerical work and uses of manpower.
- The current system is not user-friendly.
- In this, paper losing or the burning paper is possible.
- It is difficult to store the data.

Scenario

- The new system will be time-saver.
- The new system is a totally computerized system.
- The system should be flexible.
- It saves time and cost.
- It is user-friendly.
- In this, the data require less space to store than the manual system.

3.2 Proposed System (*Our Planned System*)

The main goal of our application is to automate the process of data storing, slot handling, parking management, etc. In each parking whoever uses this application. The main advantage of this application is to provide computerized parking management in each parking in this digital generation and to save the time of the user and even it reduces the traffic in the urban areas at some levels.

3.2.1 Modules and features in the New System.

ADMIN SRS

R1: Login

- Admin needs to login into the system if he/she wants to use the system as Admin.

R.1.1: Login as Admin.

Input: Website address or “Admin Panel” Button on app.

Output: Admin Login Panel will open.

R.1.2: Click on Login

Input: “Login” Button.

Output: Will be asked to the admin for username and password.

R.1.3: Fill up the form and click

Input: Type username and password.

Output: Login Successfully.

R2: Services

- Admin will get the list of the functionalities to select the required service.

R.2.1: Select Service.

Input: Selected Service.

Output: Admin's selected service panel will open.

R.2.2: Pending Requests.

Input: Click “Approve/Reject” Button.

Output: Data will be approved if it will be appropriate or else it will be rejected.

R.2.3: Approved Request.

Input: “Parked Vehicle” Button.

Output: List of the approved(Parked) vehicles will be shown.

R.2.4: Rejected Request.

Input: “History” Button.

Output: List of the rejected and accepted(Currently leaved) vehicles will be shown.

R.2.5: Add Admin.

Input: “Add Admin” Button.

Output: Admin can add another Admin's and assign User-Id and Password to them.

R.2.6: Admin List.

Input: “Admin List” Button.

Output: List of the Admins with their E-mail Id and Phone Numbers.

R3: Complain/Feedback.

- Admin will get all the user's complaints/Feedback.

R.3.1: Complaint/Feedback Box.

Input: “Feedback” Button.

Output: Feedbacks/Complains of all the users will be shown along with their Email Id's..

 **User's SRS:-****R1: Select Role**

- User has to select its specific role or panel before moving further.

R.1.1: Choose your Role

Input: "User Panel" Button.

Output: User panel will open.

R2: Select Vehicle Type

- User has to select it's vehicle type before moving further.

R.2.1: Choose your Vehicle Type

Input: "User's Vehicle type" Button.

Output: User's Vehicle's Type panel will open.

R3: Fill Vehicle's Detail

- User has to fill-up their vehicle's details to use the system and proceed further.

R.3.1: "Proceed Next" Button.

Input: "Proceed Next" Button.

Output: Will be asked to fill details.

R.3.2: Fill up the form and click.

Input: Type vehicle number and upload an image of RC Book.

Output: Slot Availability.

R.3.3: Slot Availability.

Input: User's Request.

Output: Slot is available or not.

R4: Instructions

- User will show the instructions before moving further.

R.4.1: Instruction's Page.

Input: User will read instructions.

Output: Click "Next" Button.

R.4.2: "Next" Button.

Input: Click "Next" Button.

Output: Next page/panel.

R5: Generate QR Code.

- User has to generate and save it's QR code as proof or for future reference.

R.5.1: Generate QR Code.

Input: "Generate" Button.

Output: User's QR code will be generated.

R6: Complaint/Feedback.**R.6.1: Feedback Page.**

Input: "Feedback" Button.

Output: User will get Feedback Form.

R.6.2: Do Complaint/Feedback.

Input: Type complain/Feedback.

Output: Admin will get user's complaints/Feedback and it will be solved by admin.

R7: About Us.**R.7.1: About us Page.**

Input: "About Us" Button.

Output: User will get E-mail/Contact info to reach admin.

3.2.2 User Characteristics

- **Admin**

Admin is a type of user of the System. His main characteristic is that he/she will approve/reject pending requests and solve the complaints of the customer (if any). He/she can make changes to the system whenever they want. He / She can view currently parked vehicles, rejected vehicles etc. He / She can add another Admins to manage the System and assign User-Id and Password to them. He / She can also view the list of the Admins with their E-mail ID and Mobile Numbers.

- **User**

Users are those who are going to use the system for their comfort. They can book their slot for their vehicle sitting at home in advance. They can also complain to the admin if they have any issues related with booking or any other issues via Feedback/Complaint form or from contacting them.

3.2.3 Hardware and Software Requirements



➤ **Hardware Requirements:**

- Processor: Intel Core i3 7th Gen.
- Hard Disk Space: 10GB.
- RAM: Min. 4GB.

➤ **Software Requirements:**

- Operating System: Any Windows OS.
- User Interface: PHP, Android.
- Programming Language: PHP.
- IDE/Workbench: Sublime Text3, XAMPP.
- Database: MySQL.



➤ **Hardware Requirements:**

- Mobile Storage Space: 20MB-100MB.
- RAM: 4GB.
- Mobile Compatibility: Android.

➤ **Software Requirements:**

- Operating System: Any Windows OS.
- Mobile Version: Android.
- Android Version: Minimum. 4.3.2.
- Android Version (Name): Ice-Cream Sandwich.



➤ **Hardware Requirements:**

- Arduino UNO.
- Servo Motor SG90.
- ESP8266 Wifi Module.
- Breadboard.
- Jumper Wire.

➤ **Software Requirements:**

- Operating System: Any Windows OS.
- RAM: 4GB.
- Programming Language: Arduino.
- IDE/Workbench: Arduino IDE.
- Database: MySQL.

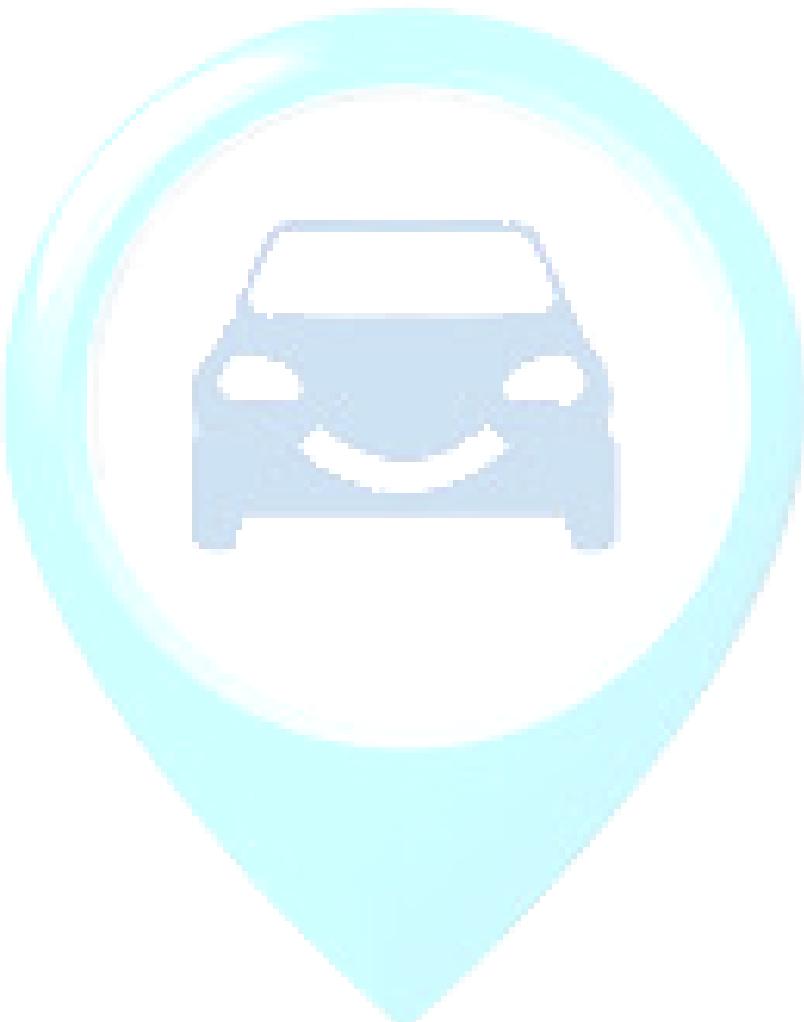
4.0 System Analysis

4.1 Feasibility Study

4.2 System Activity Diagram

4.3 Use Case Diagram

4.4 Sequence Diagram



4.1 FEASIBILITY STUDY

Feasibility is defined as the practical extent to which a project can be performed successfully.

To evaluate feasibility, a feasibility study is performed, which determines whether the solution considered to accomplish the requirements is practical and workable in the software. Information such as resource availability, cost estimation for software development, benefits of the software to the organization after it is developed and cost to be incurred on its maintenance are considered during the feasibility study. The objective of the feasibility study is to establish the reasons for deploying the software that is acceptable to users, adaptable to change and conformable to established standards. Various other objectives and feasibility study are listed below:-

- To analyse whether the software will meet organizational requirements.
- To determine whether the software can be implemented using the current technology and within the specified budget and schedule.
- To determine whether the software can be integrated with other existing software.

TECHNICAL FEASIBILITY:

Technical feasibility assesses the current resources (such as hardware and software) and technology, which are required to accomplish user requirements in the software within the allocated time and budget. For this, the software development team ascertains whether the current resources and technology can be upgraded or added in the software to accomplish specific user requirements. Technical feasibility also performs the following tasks.

- Analyses the technical skills and capabilities of the software development team members.
- Determines whether the relevant technology is stable and established.
- Ascertains that the technology chosen for software development has many users so that they can be consulted when problems arise, or improvements are required.

OPERATIONAL FEASIBILITY:

Operational feasibility assesses the extent to which the required software performs a series of steps to solve business problems and user requirements. This feasibility is dependent on human resources (software development team) and involves visualizing whether the software will operate after it is developed and be operative once it is installed. Operational feasibility also performs the following tasks.

- Determines whether the problems anticipated in user requirements are of high priority.
- Determines whether the solution suggested by the software development team is acceptable.
- Analyses whether users will adapt to a new software.

- Determines whether the organization is satisfied by the alternative solutions proposed by the software development team.

ECONOMICAL FEASIBILITY

Economic feasibility determines whether the required software can generate financial gains for an organization. It involves the cost incurred on the software development team, estimated cost of hardware and software, cost of performing feasibility study, and so on. For this, it is essential to consider expenses made on purchases (such as hardware purchase) and activities required to carry out software development. In addition, it is necessary to consider the benefits that can be achieved by developing the software. Software is said to be economically feasible if it focuses on the issues listed below.

- Cost incurred on software development to produce long-term gains for an organization.
- Cost required to conduct full software investigation (such as requirements elicitation and requirements analysis)
- Cost of hardware, software, development team, and training.

4.2 SYSTEM ACTIVITY DIAGRAM

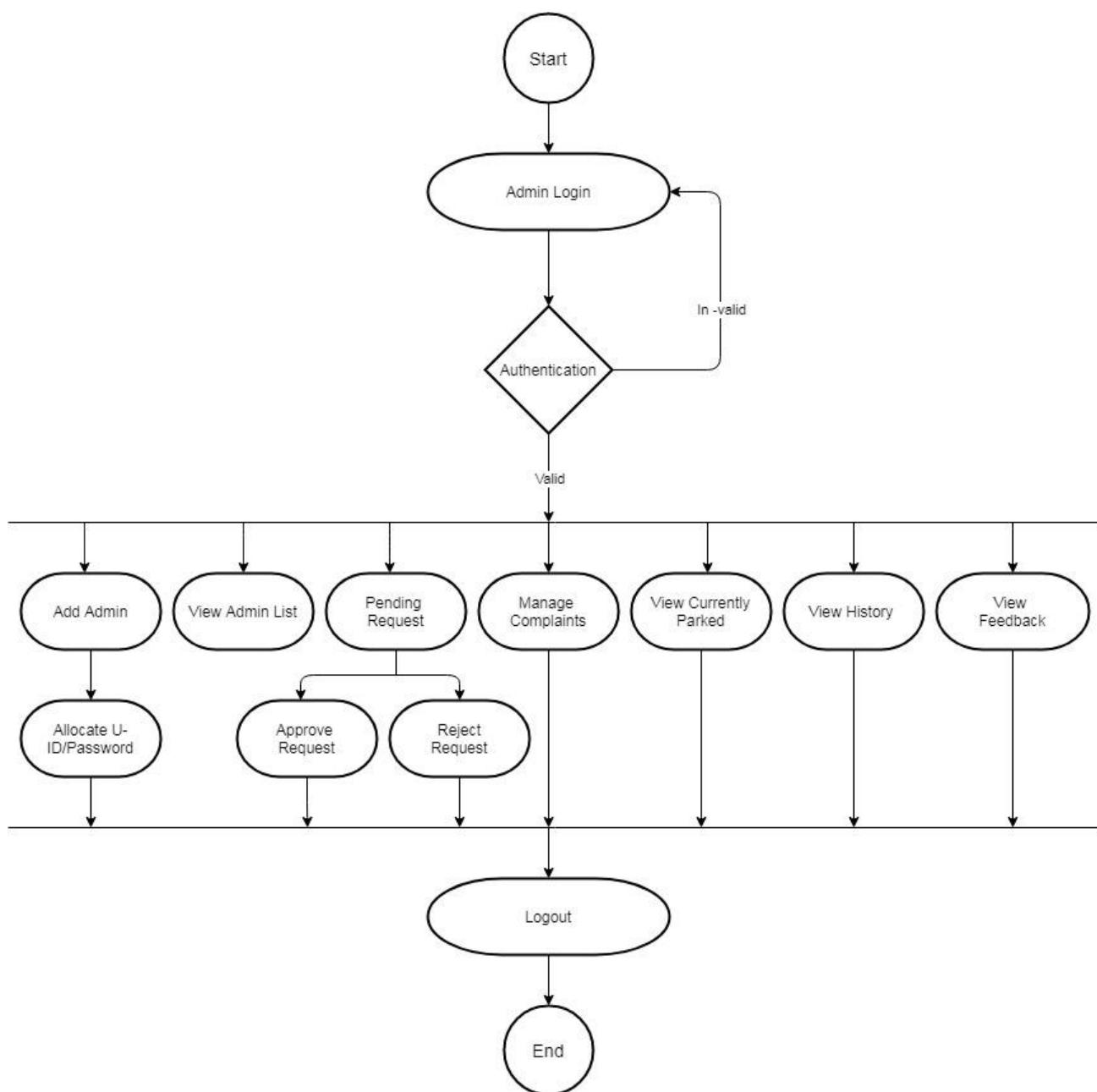


Fig. 4.1 Admin Activity Diagram

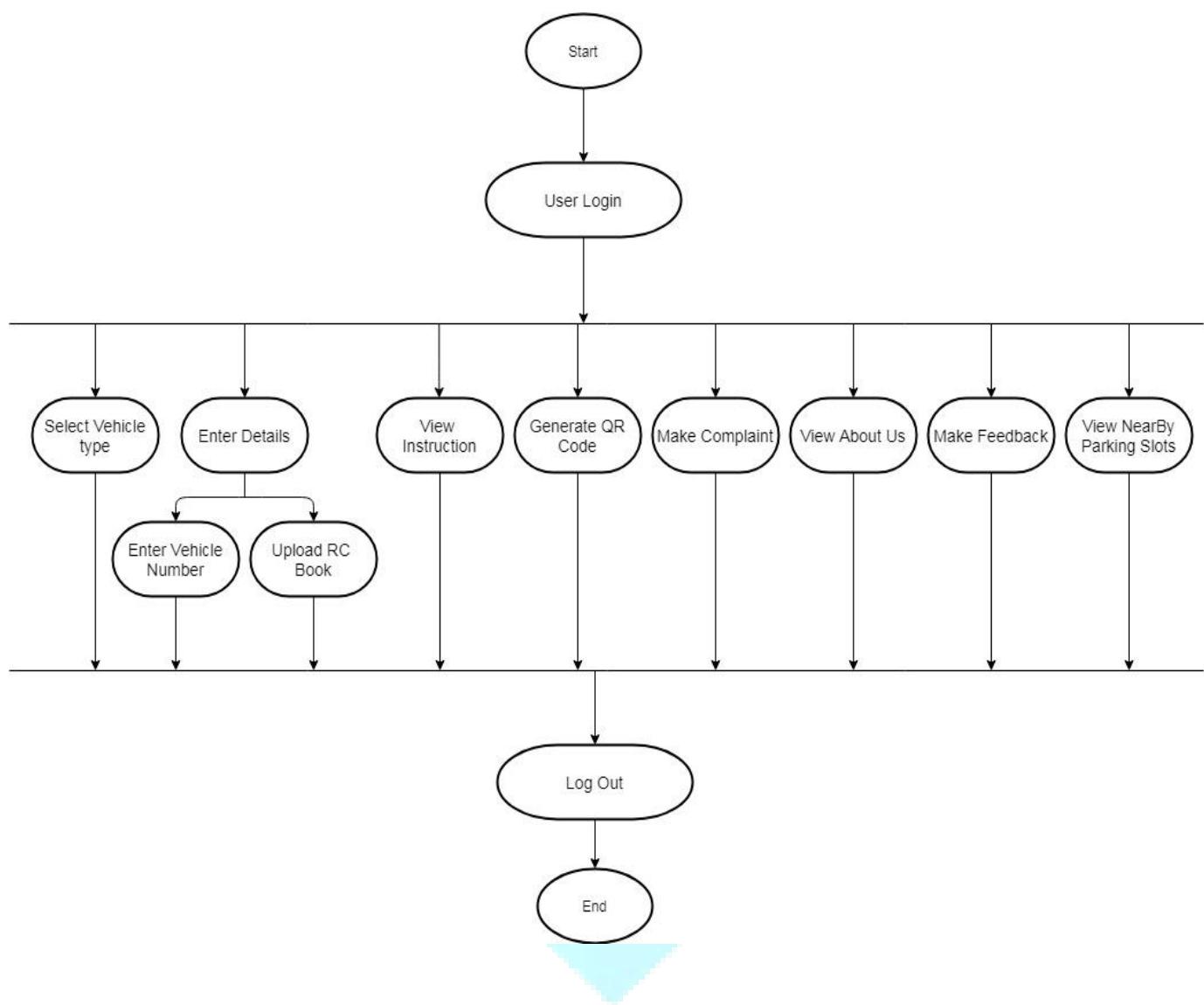


Fig. 4.2 User Activity Diagram

4.3 SYSTEM USE-CASE DIAGRAM

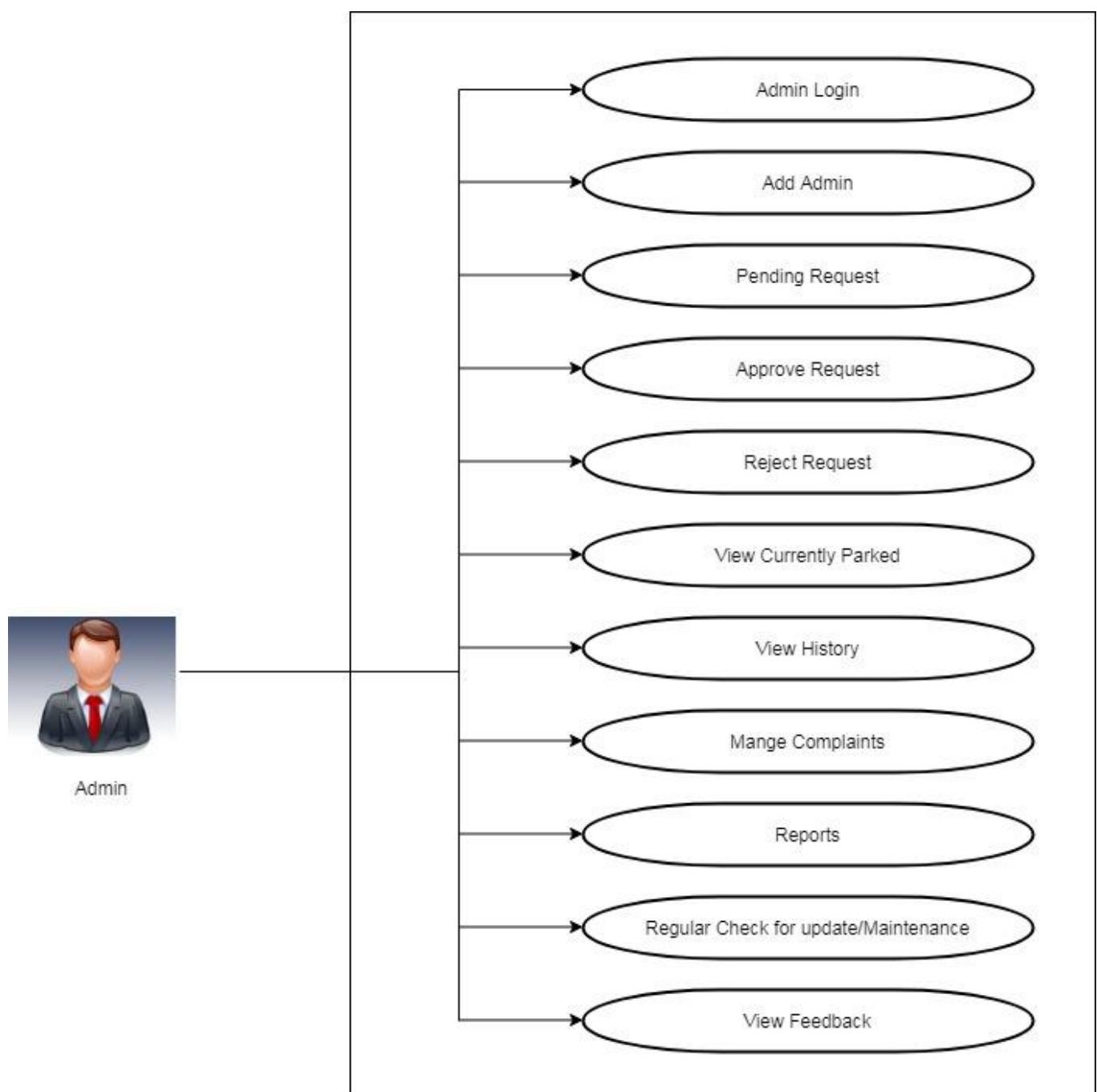
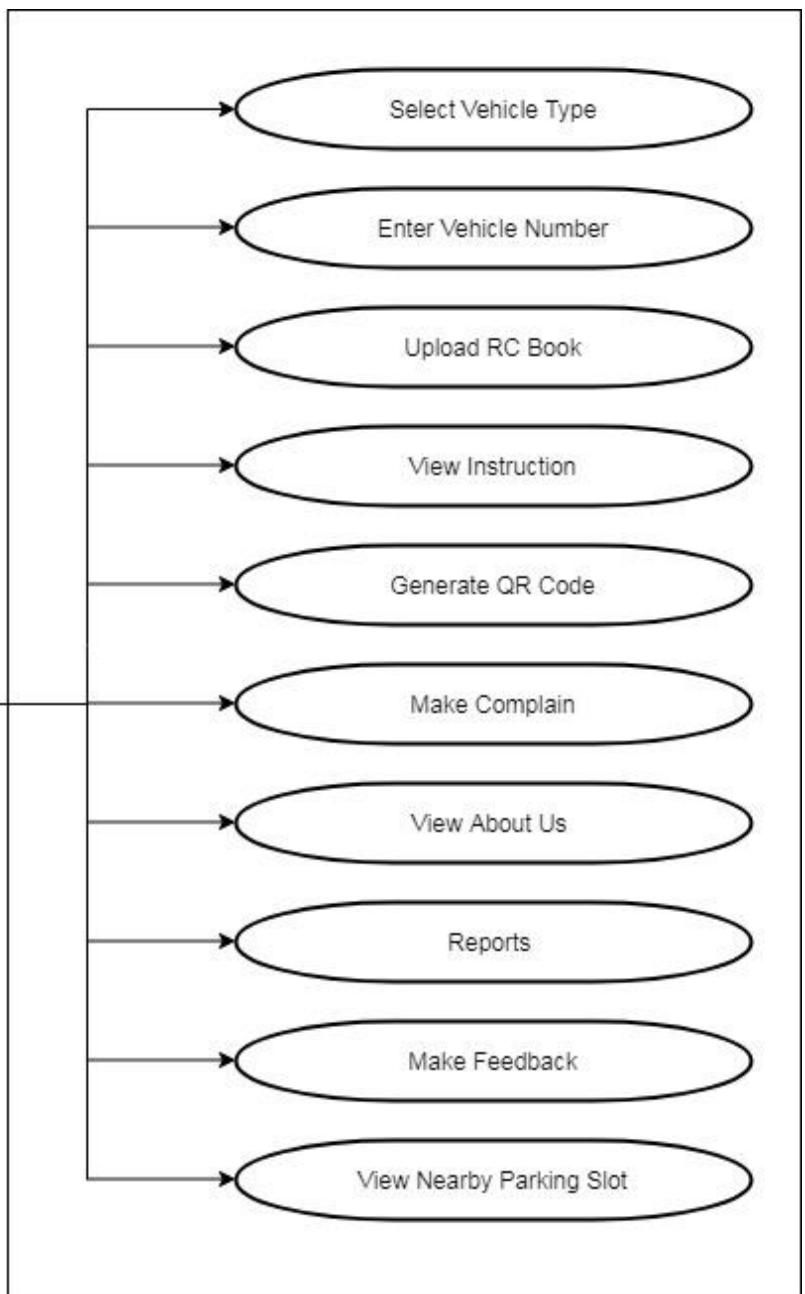


Fig. 4.3 Admin Use-Case Diagram



User

**Fig. 4.4 User Use-Case Diagram**

4.4 SYSTEM SEQUENCE DIAGRAM

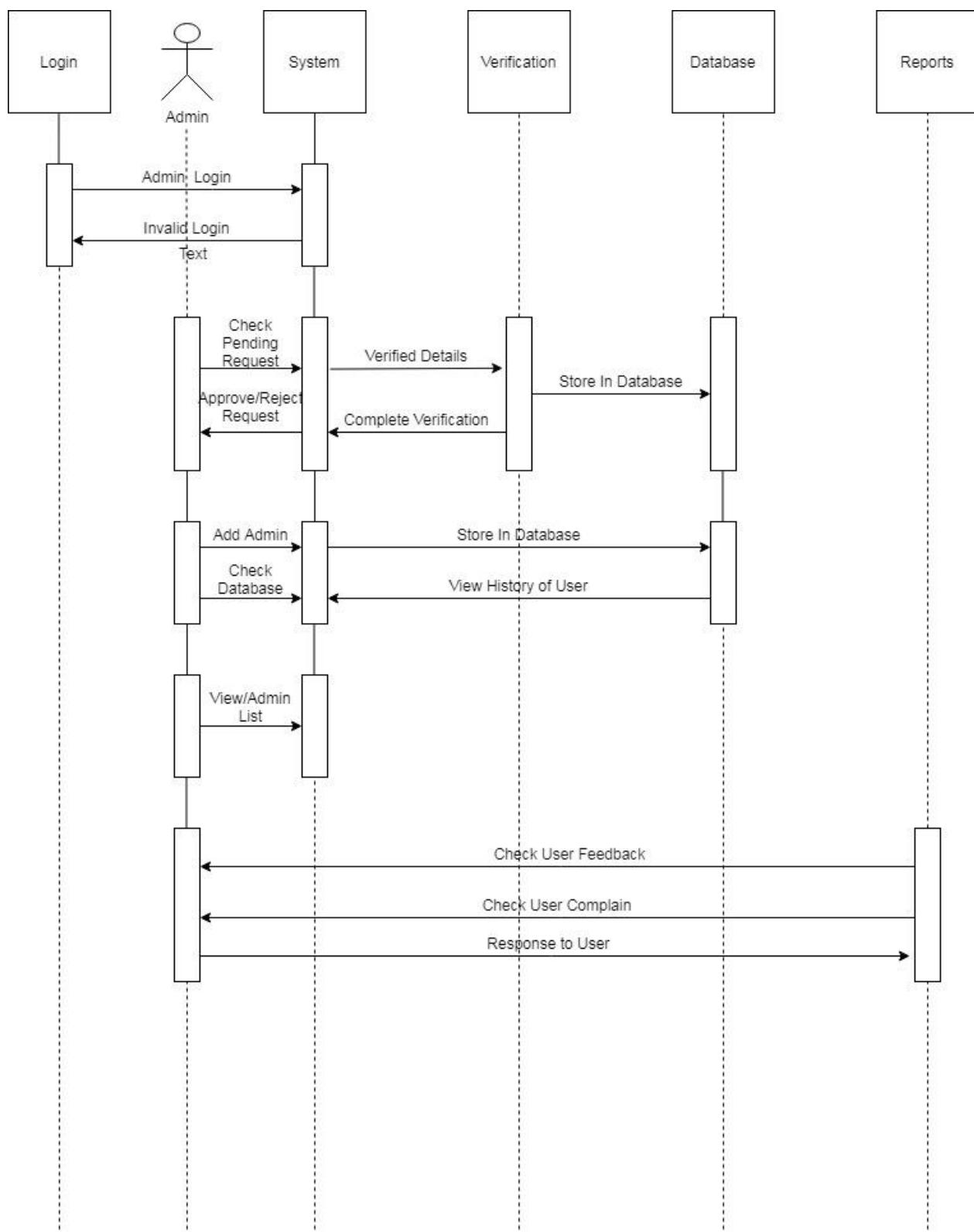
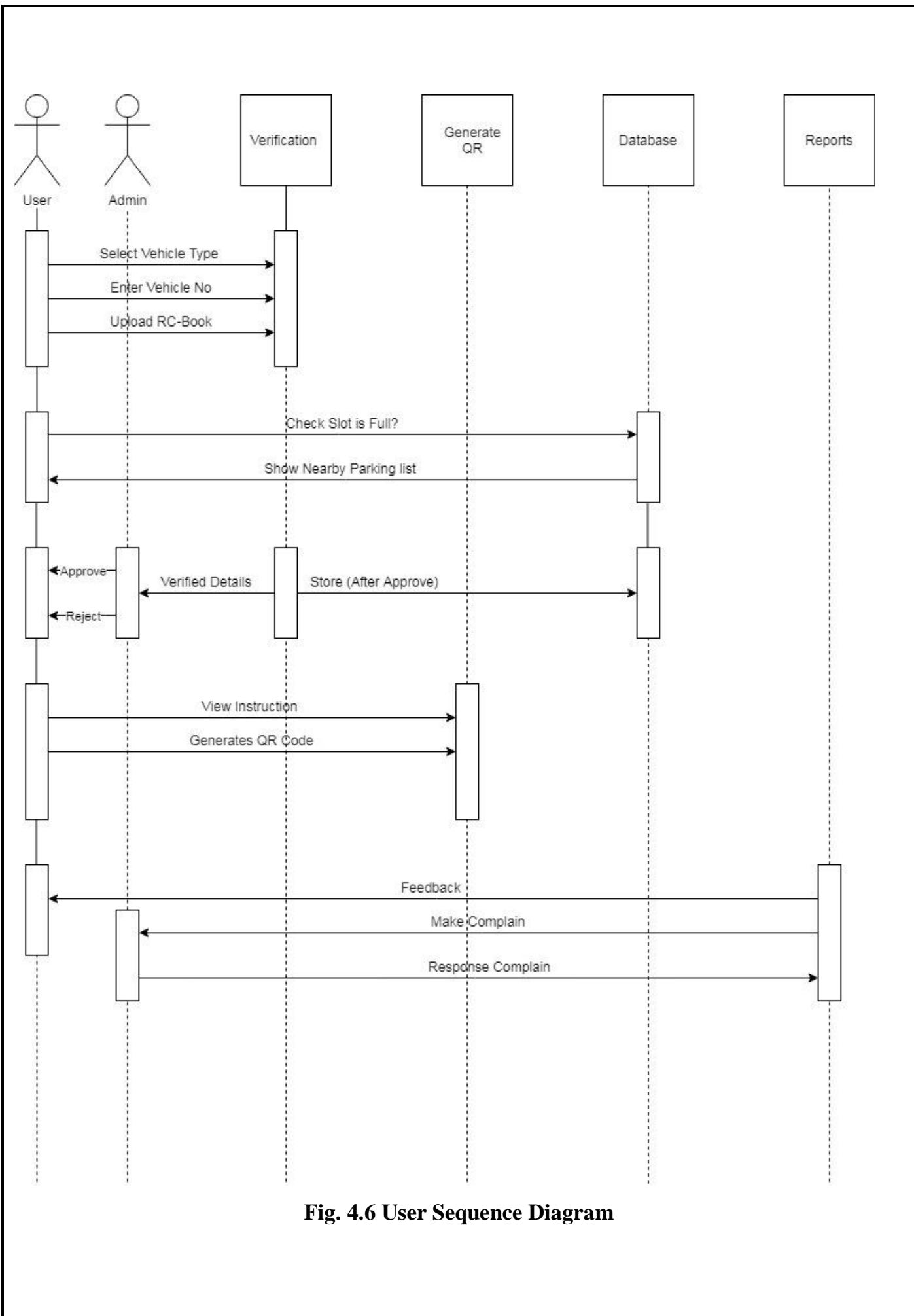


Fig. 4.5 Admin Sequence Diagram

**Fig. 4.6 User Sequence Diagram**

5.0 System Design

5.1 DATABASE DESIGN/DATA STRUCTURE DESIGN

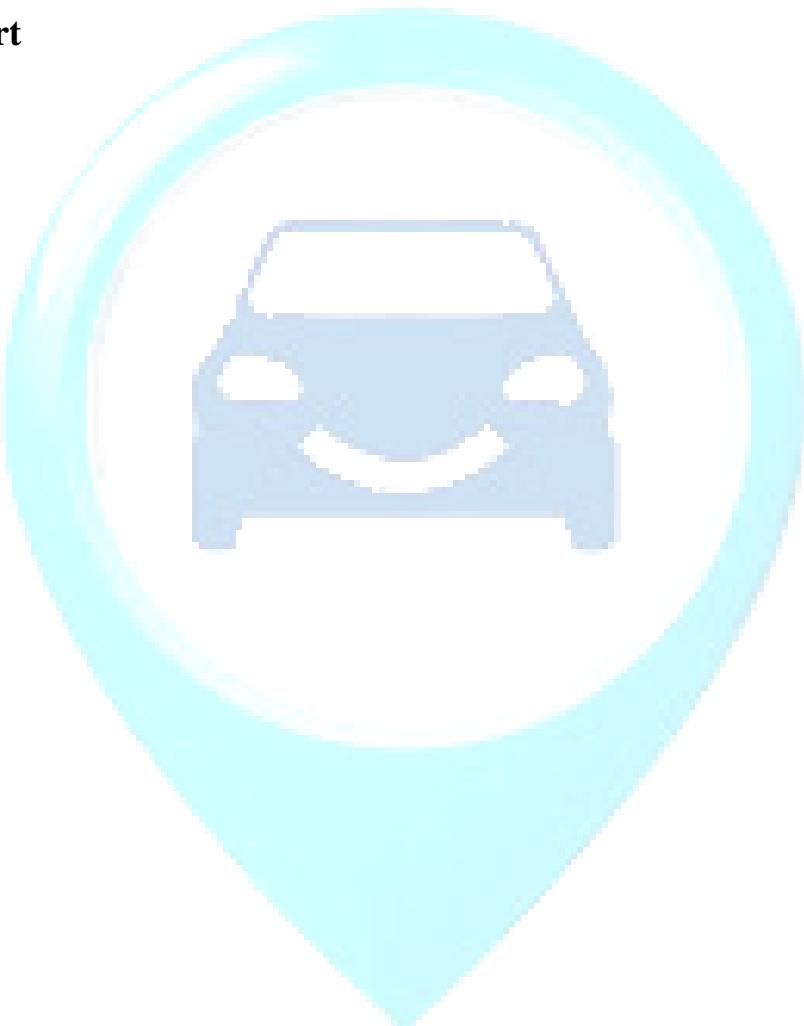
5.1.1 Data Dictionary

5.1.2 ER Diagram

5.1.3 Data Flow Diagram

5.2 INPUT/OUTPUT AND INTERFACE DESIGN

5.2.1 Flowchart



5.1 DATABASE DESIGN/DATA STRUCTURE DESIGN

5.1.1 Data Dictionary

<u>Sr.No</u>	<u>Name</u>	<u>Data type</u>	<u>Description</u>
1.	AID	Number(5)	Primary Key and Auto Increment
2.	User-Name	Varchar(30)	Unique Key
3.	Password	Varchar(30)	Password of the Admin
4.	E-mail-ID	Varchar(30)	E-mail Id of the Admin
5.	Mobile-Number	Number(10)	Contact Number of the Admin

Table 5.1 Admin Master

<u>Sr.No.</u>	<u>Name</u>	<u>Data type</u>	<u>Description</u>
1.	AID	Number(5)	Primary Key and Auto Increment
2.	Parking_Id	Varchar(5)	Stores Parking ID.
3.	Vehicle_Number	Varchar(15)	Unique Key
4.	Vehicle_Type	Varchar(15)	Stores type of Vehicle
5.	Image_Path	Varchar(100)	Stores image path of RC Book
6.	Date	Date	Stores Date of Record Insertion.
7.	Time	Time	Stores Time of Record Insertion.
8.	Status	Varchar(50)	Stores Status of Check-In

Table 5.2 Accept Master

<u>Sr.No.</u>	<u>Name</u>	<u>Data type</u>	<u>Description</u>
1.	PID	Number(5)	Primary Key and Auto Increment
2.	Parking_Id	Varchar(5)	Stores Parking ID.
3.	Vehicle_Number	Varchar(15)	Unique Key
4.	Vehicle_Type	Varchar(15)	Stores type of Vehicle
5.	Image_Path	Varchar(100)	Stores image path of RC Book
6.	Date	Date	Stores Date of Record Insertion.
7.	Time	Time	Stores Time of Record Insertion.

Table 5.3 Pending Requests

<u>Sr.No.</u>	<u>Name</u>	<u>Data type</u>	<u>Description</u>
1.	HID	Number(5)	Primary Key and Auto Increment
2.	Parking_Id	Varchar(5)	Stores Parking ID.
3.	Vehicle_Number	Varchar(15)	Unique Key
4.	Vehicle_Type	Varchar(15)	Stores type of Vehicle
5.	Image_Path	Varchar(100)	Stores image path of RC Book
6.	Date	Date	Stores Date of Record Insertion.
7.	Time	Time	Stores Time of Record Insertion.
8.	Admin_Status	Varchar(9)	Stores Status of Admin Approval

Table 5.4 History Master

<u>Sr.No.</u>	<u>Name</u>	<u>Data type</u>	<u>Description</u>
1.	Parking_Id	Varchar(5)	Primary Key and Auto Increment
2.	Parking_Name	Varchar(50)	Unique Key

Table 5.5 Parkings Master

<u>Sr.No.</u>	<u>Name</u>	<u>Data type</u>	<u>Description</u>
1.	FID	Number(5)	Primary Key and Auto Increment
2.	Name	Varchar(50)	Stores name of the user
3.	E-mail-ID	Varchar(30)	Stores E-mail ID of the user
4.	Feedback_type	Varchar(15)	Stores type of Feedback
5.	Message	Varchar(200)	Stores Message of the user
6.	Date	Date	Stores Date of Record Insertion.
7.	Time	Time	Stores Time of Record Insertion.

Table 5.6 Feedback Master

5.1.2 ER Diagram

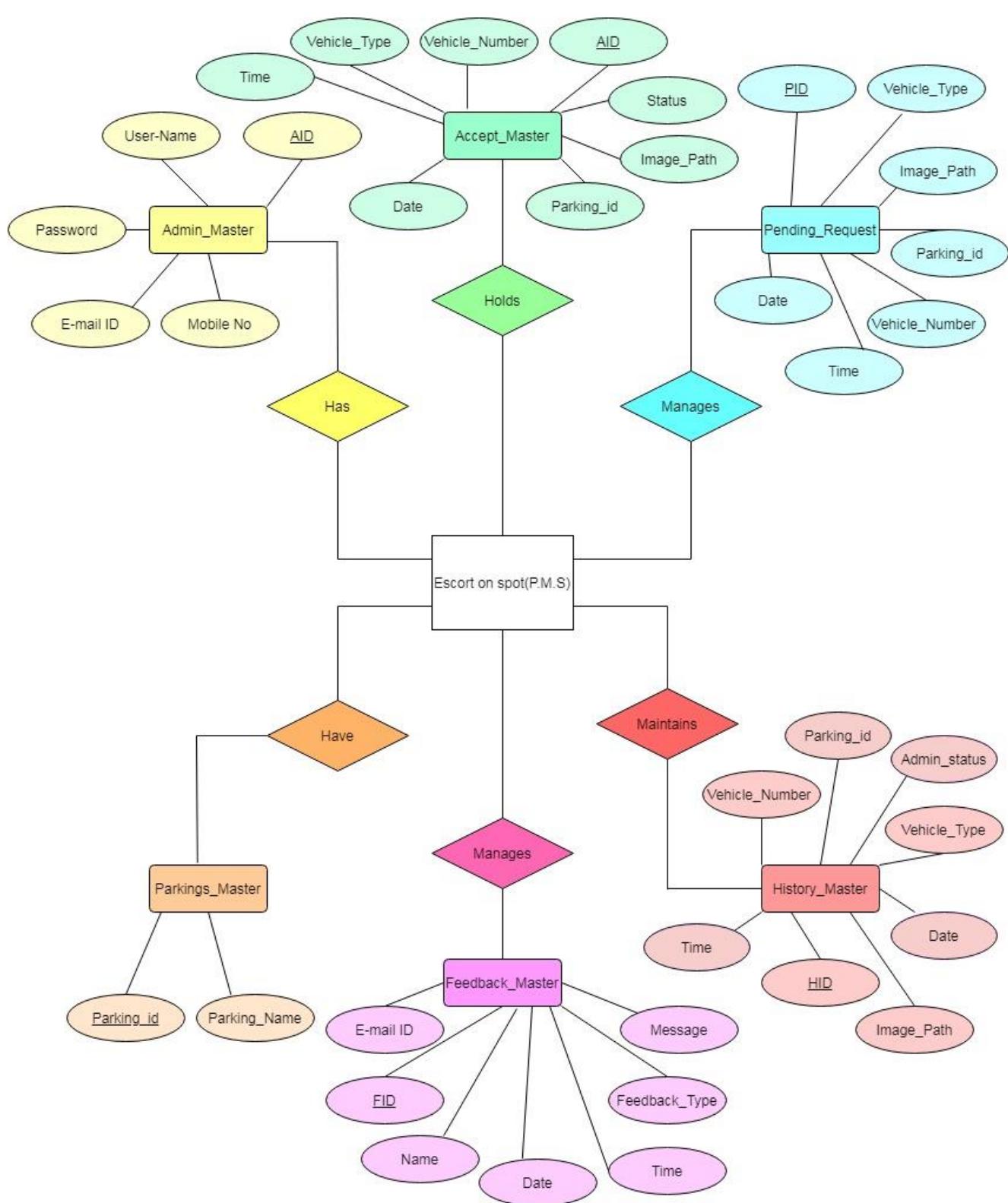


Fig. 5.1 ER Diagram.

5.1.3 Data Flow Diagram

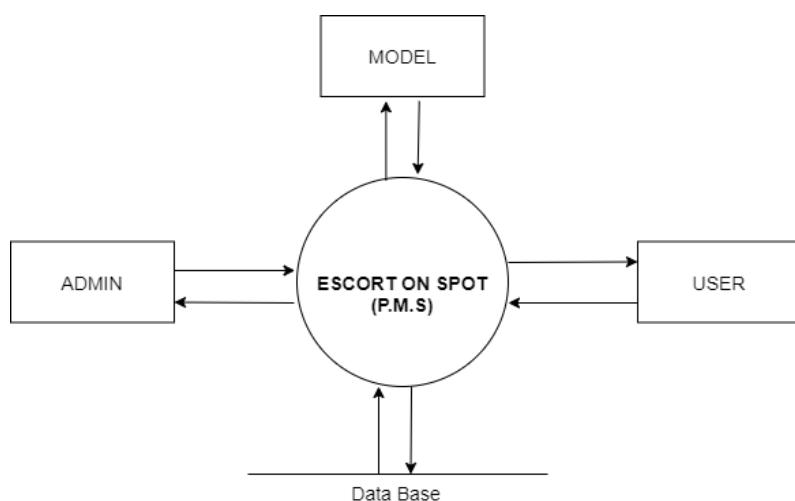


Fig. 5.2 DFD[Level-0] Diagram.

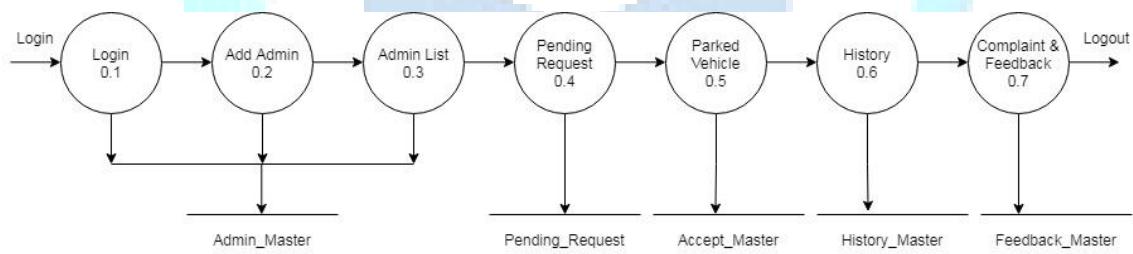


Fig. 5.3 Admin DFD[Level-1] Diagram.

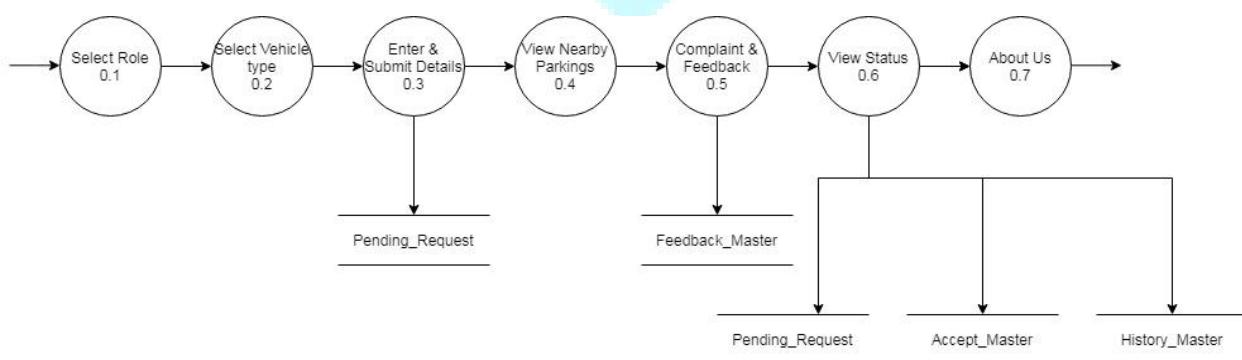


Fig. 5.4 User DFD[Level-1] Diagram.

5.2 INPUT/OUTPUT AND INTERFACE DESIGN

5.2.1 Flowchart

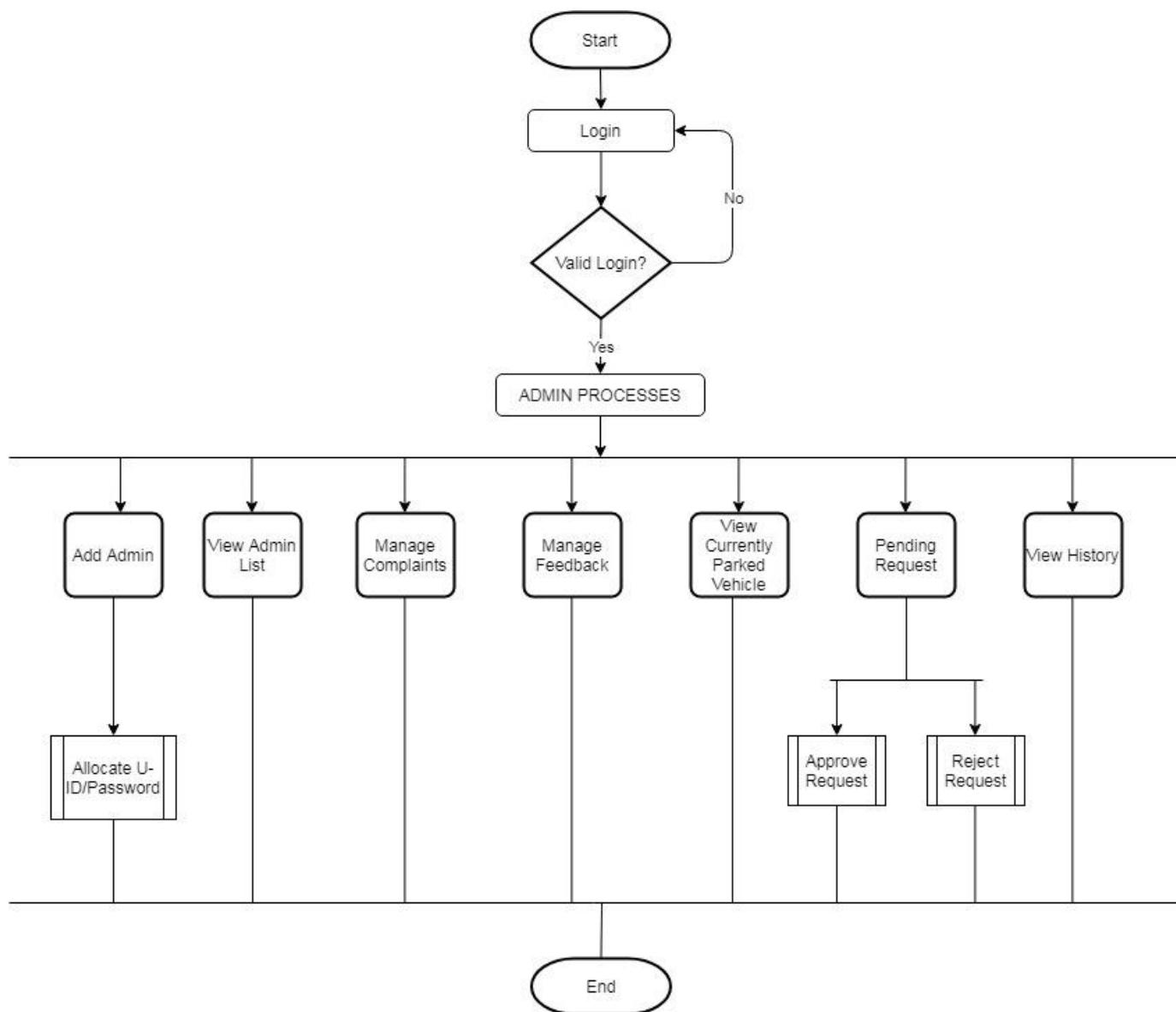


Fig. 5.5 Admin Flow-Chart Diagram

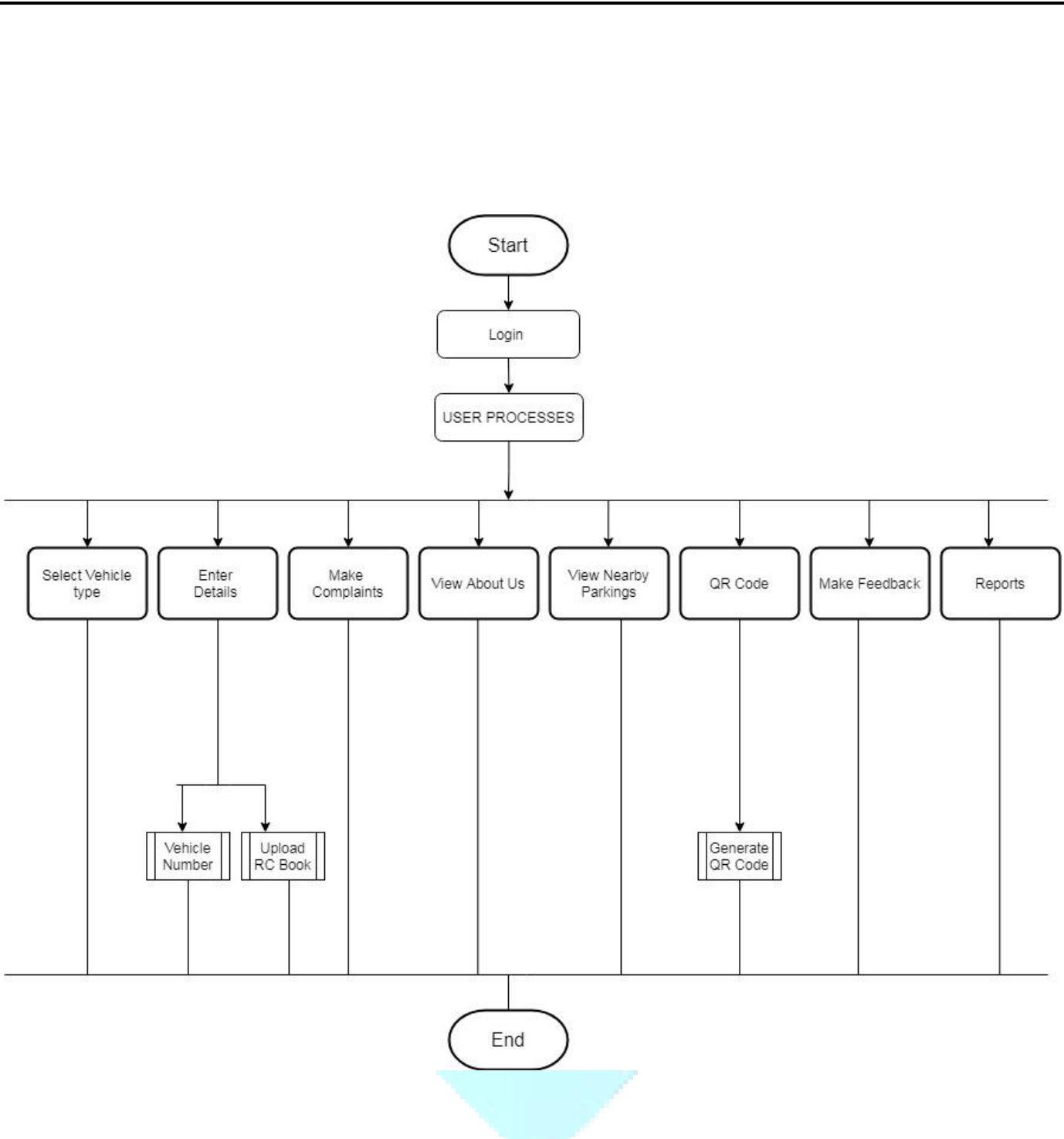


Fig. 5.6 User Flow-Chart Diagram

5.2.2 Working Diagram of the System.

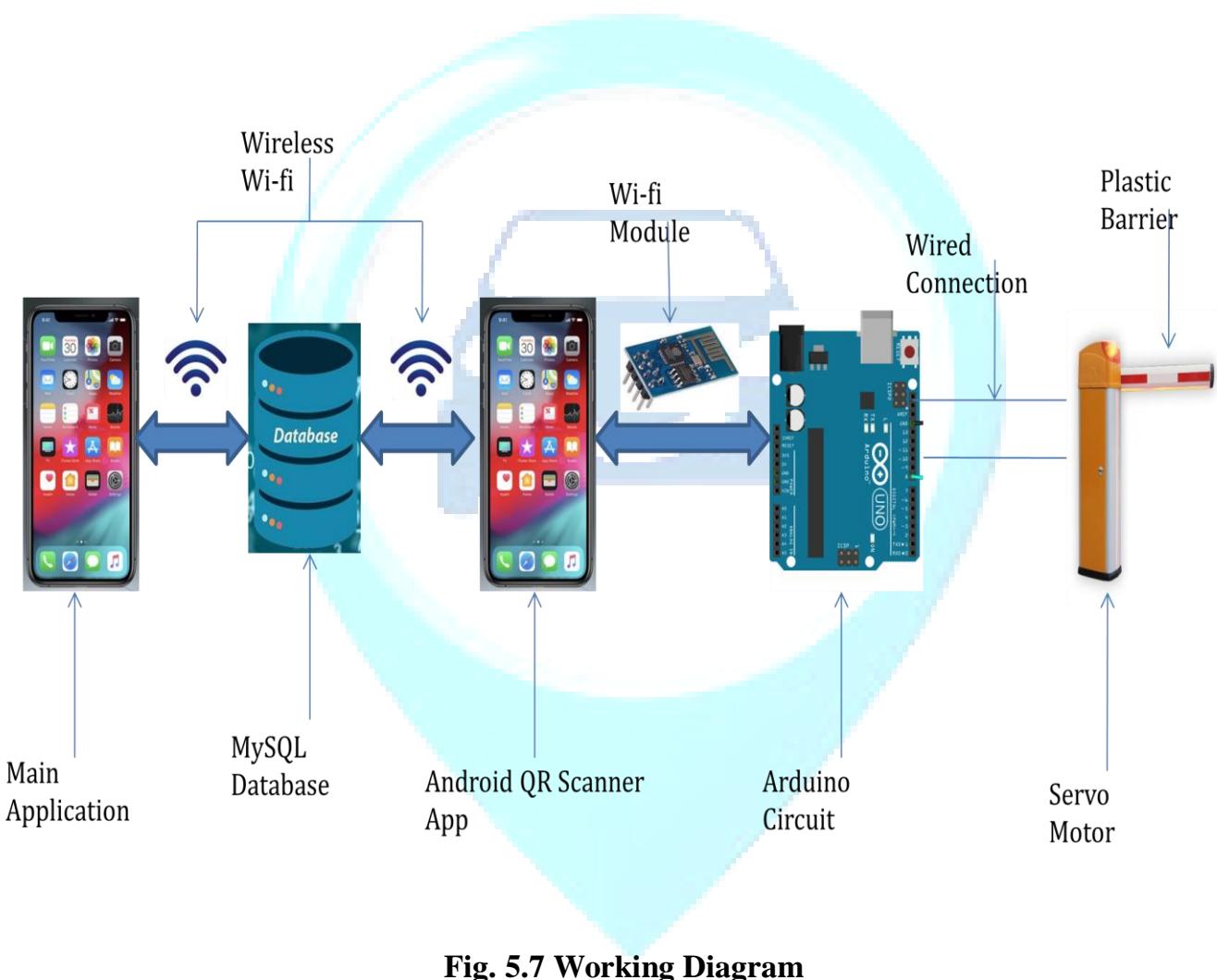


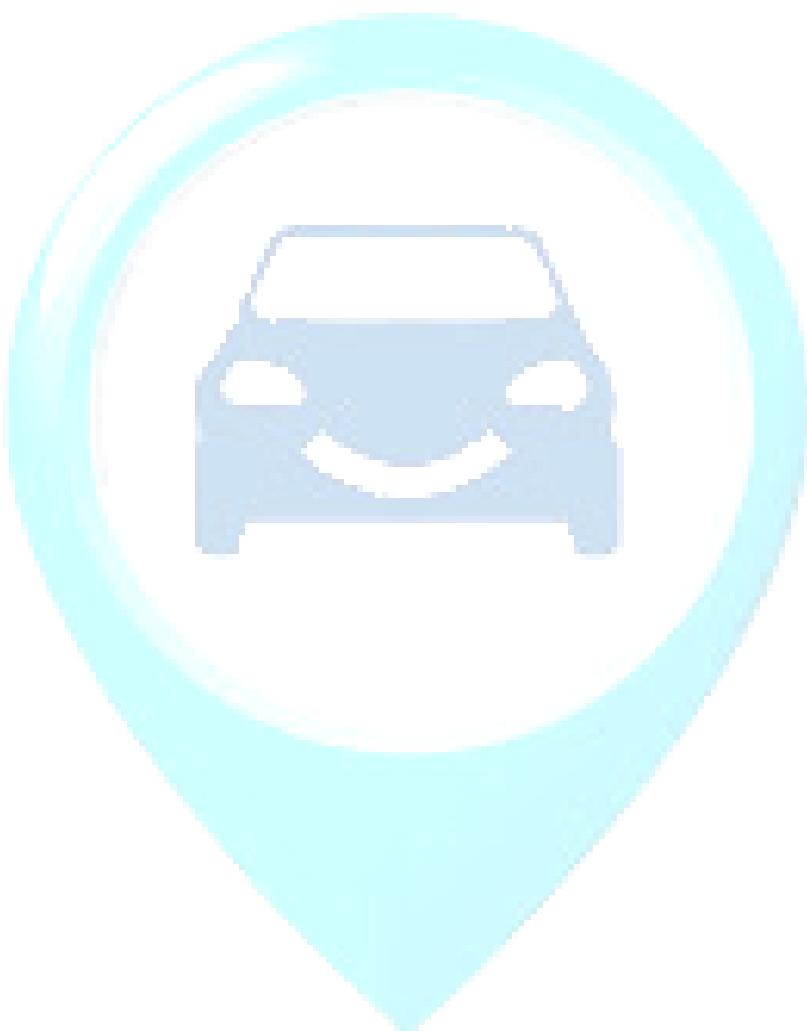
Fig. 5.7 Working Diagram

6.0 IMPLEMENTATION

6.1 IMPLEMENTATION ENVIRONMENT

6.2 SCREENSHOTS OF DIFFERENT MODULES

6.3 SAMPLE CODING



6.1 IMPLEMENTATION ENVIRONMENT

Basically, we worked on the **ANDROID STUDIO**, **XAMPP**, **SUBLIME TEXT**, **ARDUINO IDE** which led us to the completion of our Project.

- **GUI vs Non-GUI :-**

- This application has both **GUI** and **Non-GUI** based Implementation Environment.
- As we all know that every system has GUI to interact with user and our system also have **Non-GUI** environment.

- **Single vs Multiuser :-**

- Our Project has Multi-user Environment.
- One is Admin and others are the one who uses the application or you can say clients/customers.

6.2 SCREENSHOTS OF DIFFERENT MODULES

Main Application Screen-Shots :-

- Main Page



Fig. 6.1 Main Page Design

- Introduction Splash Screen

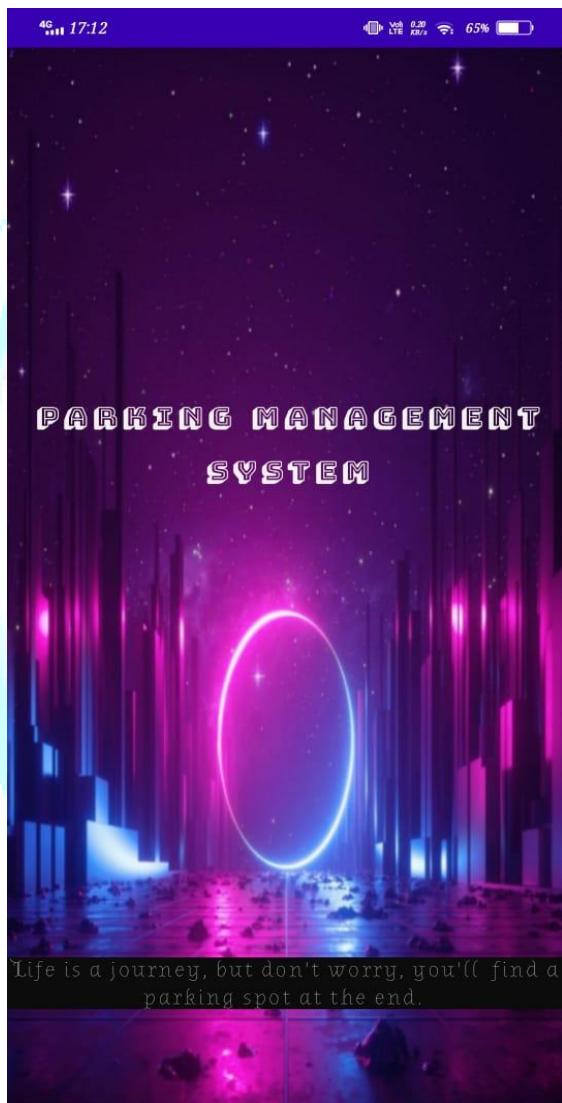


Fig. 6.2 Introduction Splash Screen Design

- Home Page

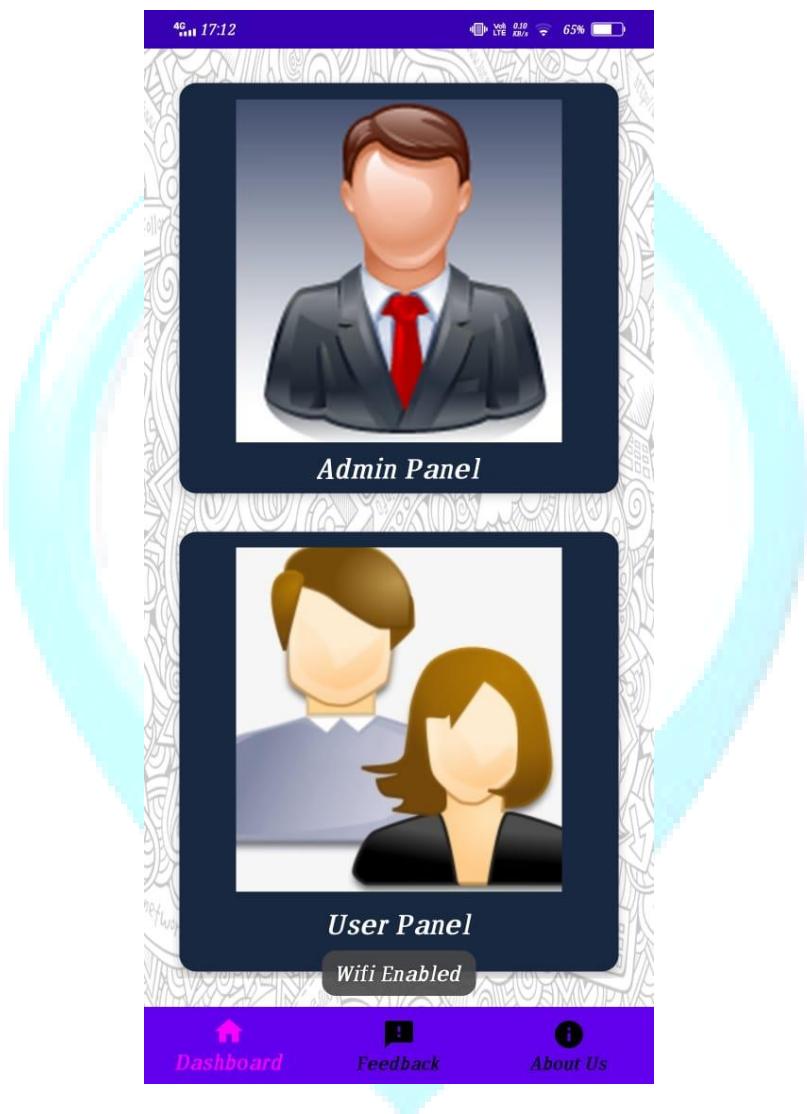


Fig. 6.3 Home Page Design

- Vehicle Type Selection Page

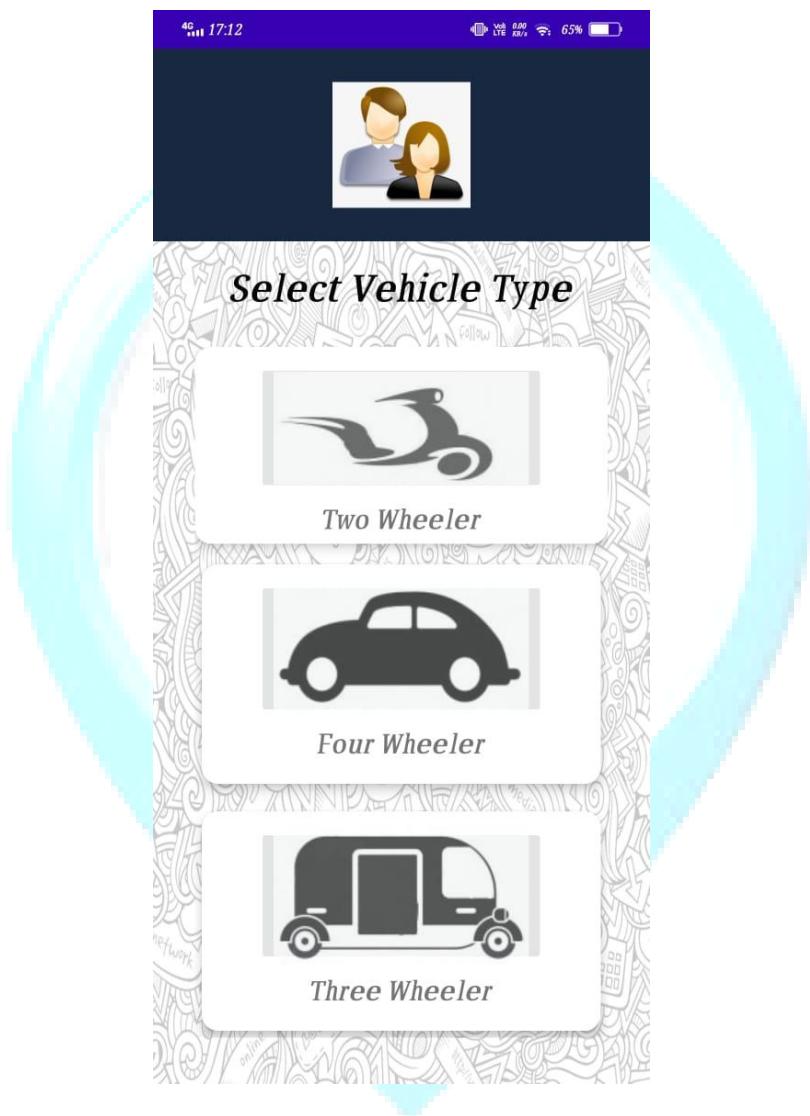


Fig. 6.4 Vehicle Type Selection Page Design

- Information Taking Page



Fig. 6.5 Information Taking Page Design

- Instruction Dialog (If Slot is Available)

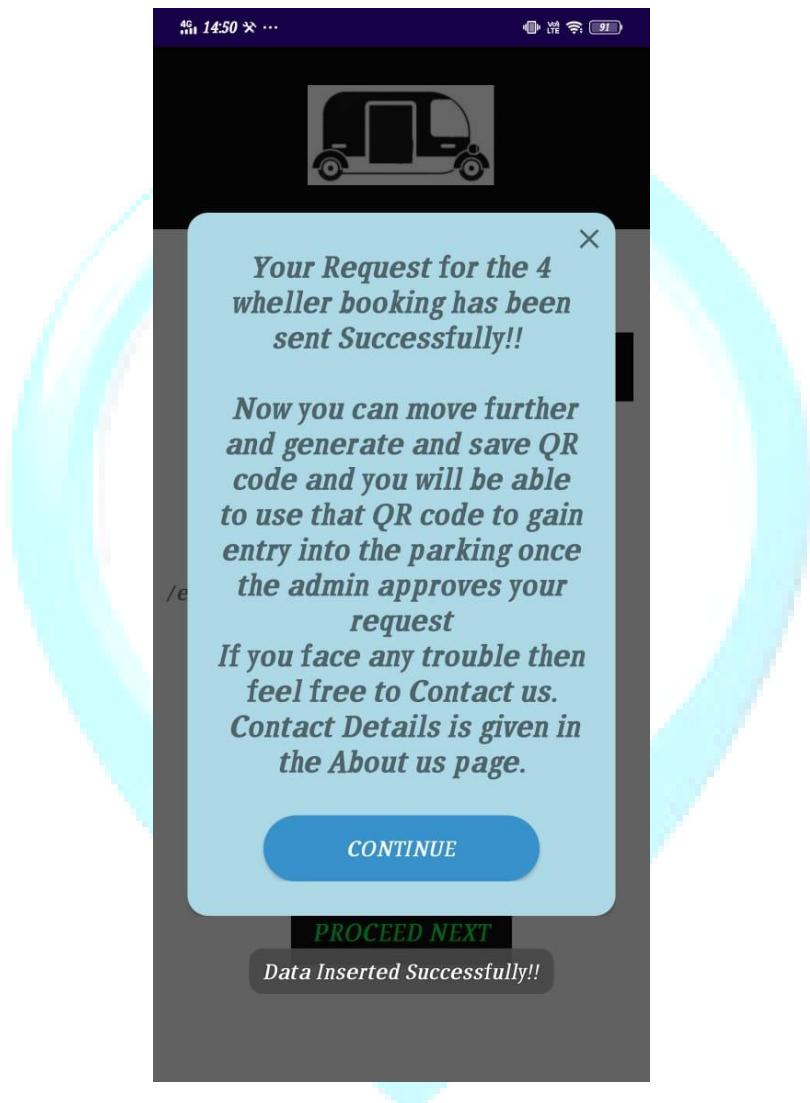


Fig. 6.6 Instruction Dialog Design

- Alert Dialog (If Slot is Unavailable)



Fig. 6.7 Alert Dialog Design

- QR Code Generator Page (If Slot is Available)

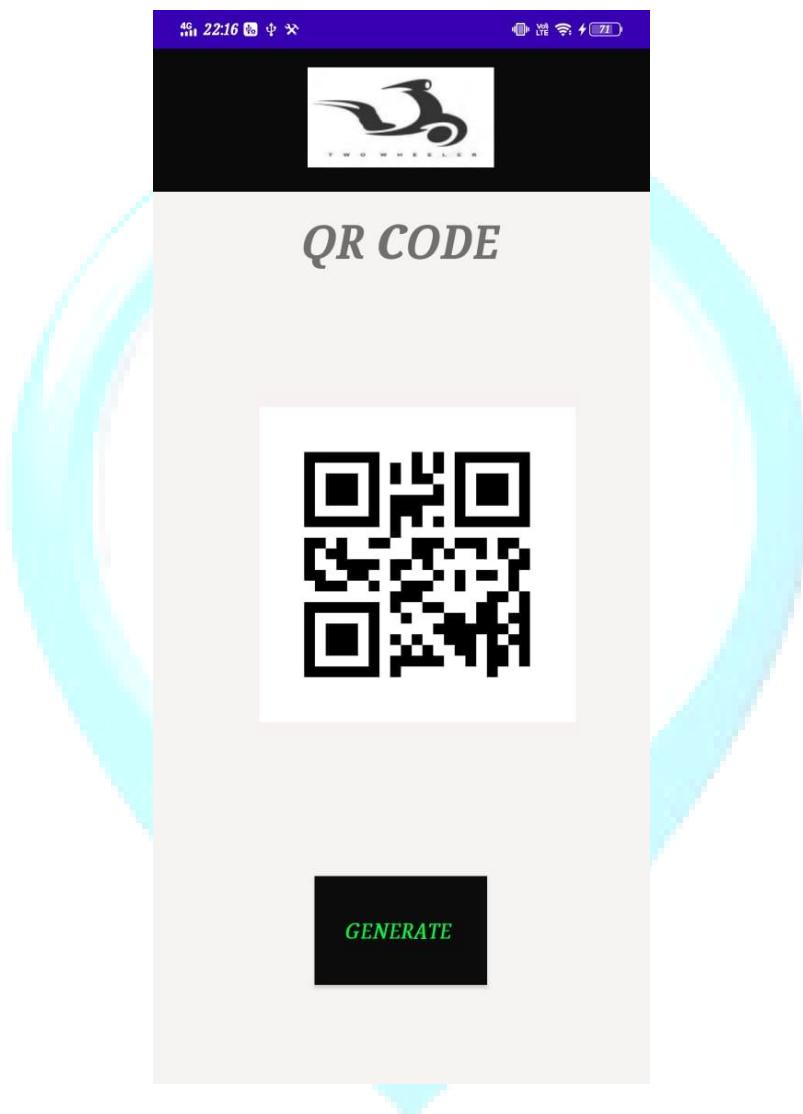


Fig. 6.8 QR Code Generator Page Design

- Approval Status

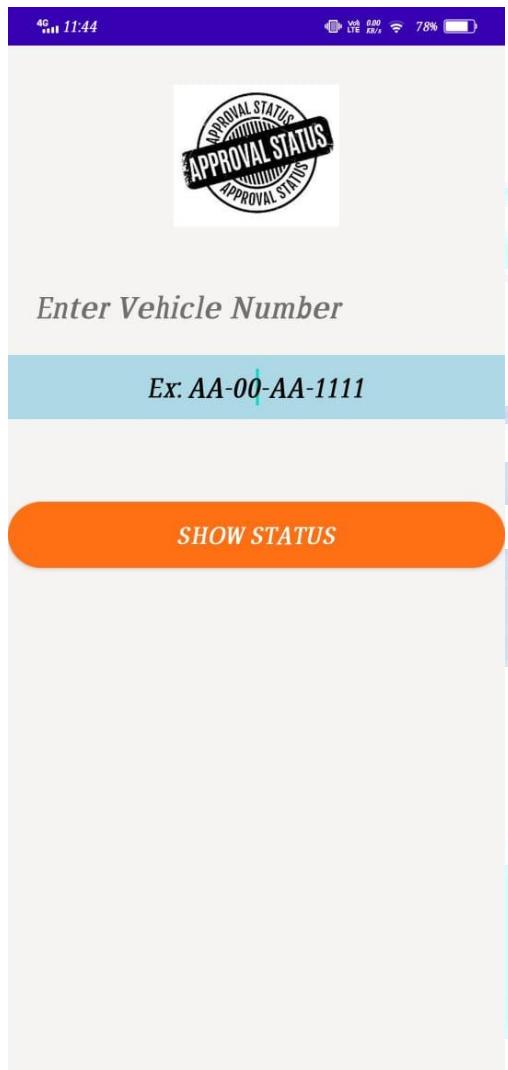


Fig. 6.9 Approval Status Page Design

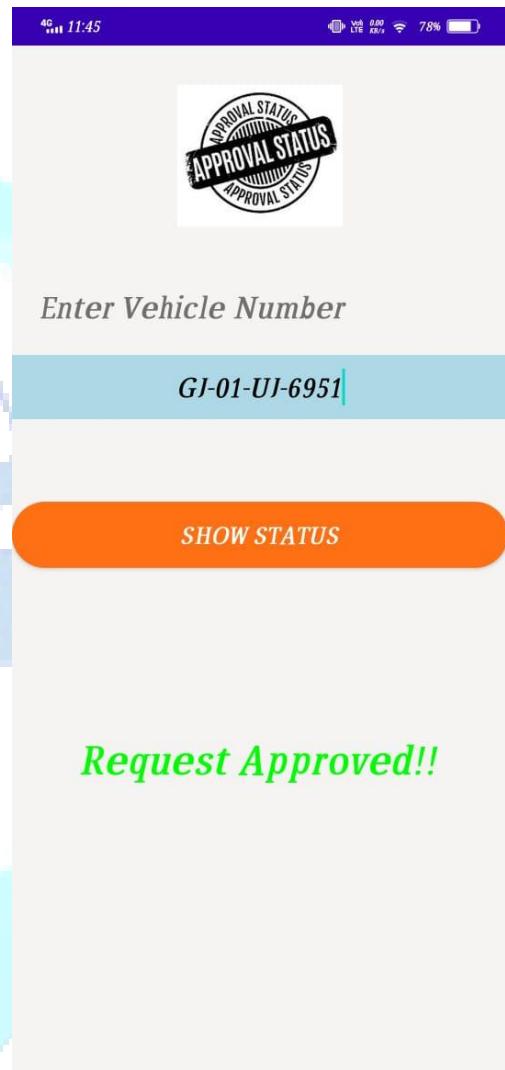
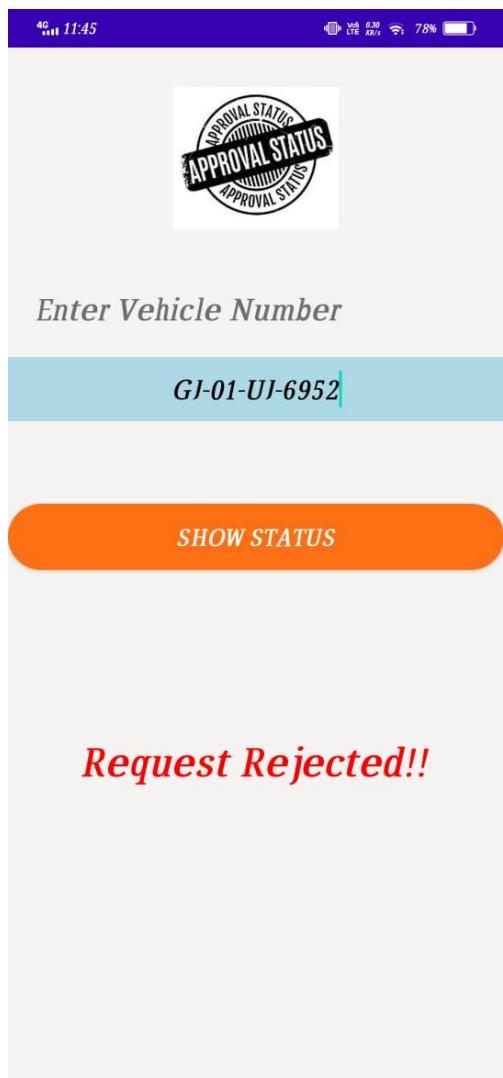
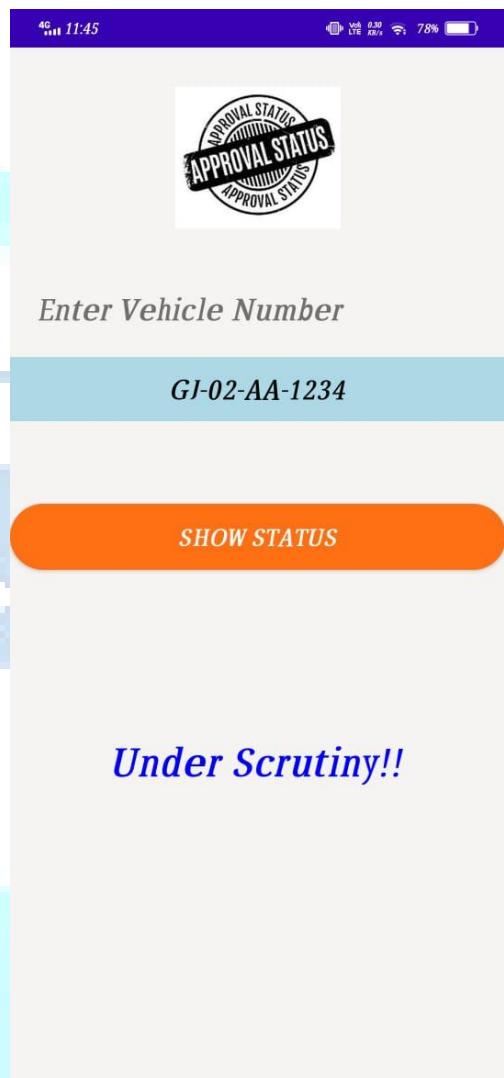


Fig. 6.10 Approval Status Accepted Page Design



Request Rejected!!

Fig. 6.11 Approval Status Rejected Page Design



Under Scrutiny!!

Fig. 6.12 Approval Status Pending Page Design

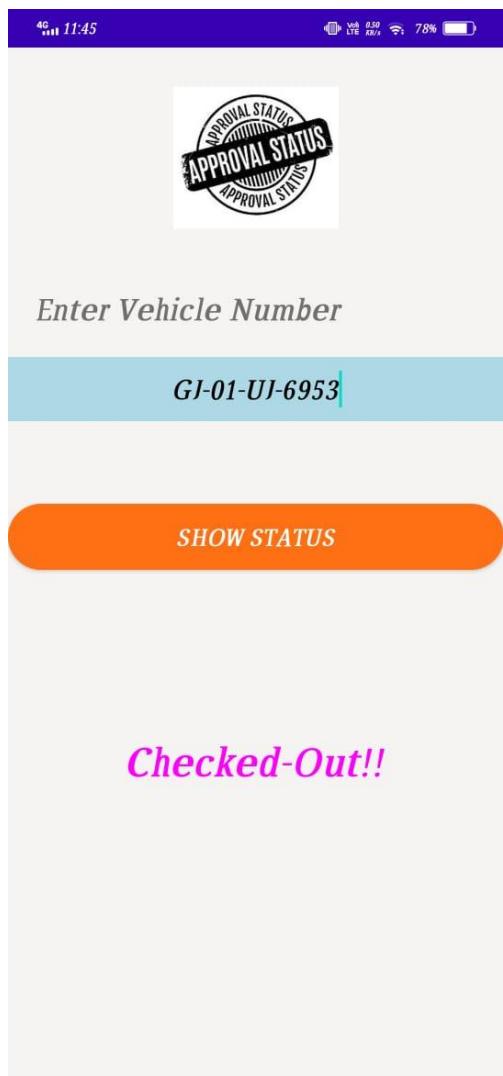


Fig. 6.13 Approval Status Checked Out Design

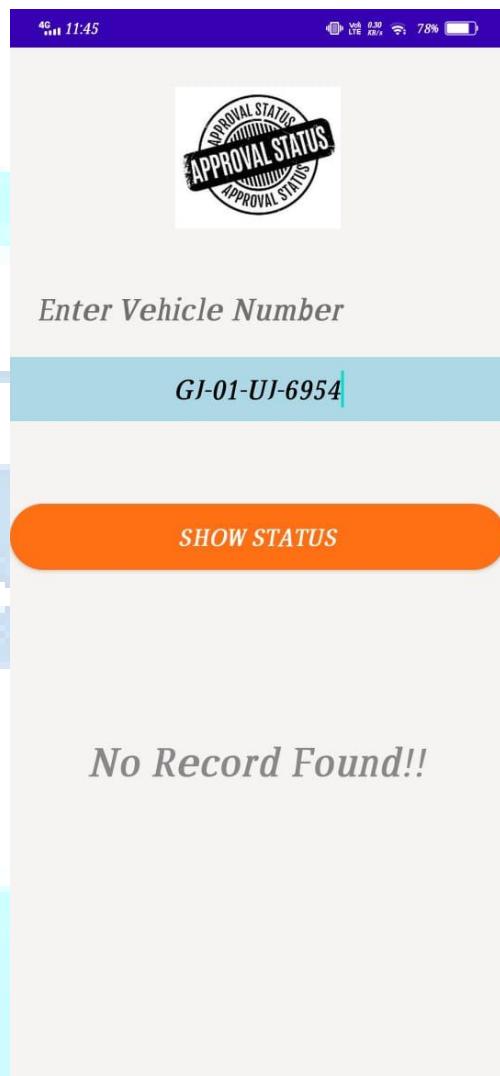


Fig. 6.14 Approval Status Not Found Page Design

QR Scanning Application Screen-Shots :-

- Main Home Page



Fig. 6.15 Main Home Page Design

- Scanning Page

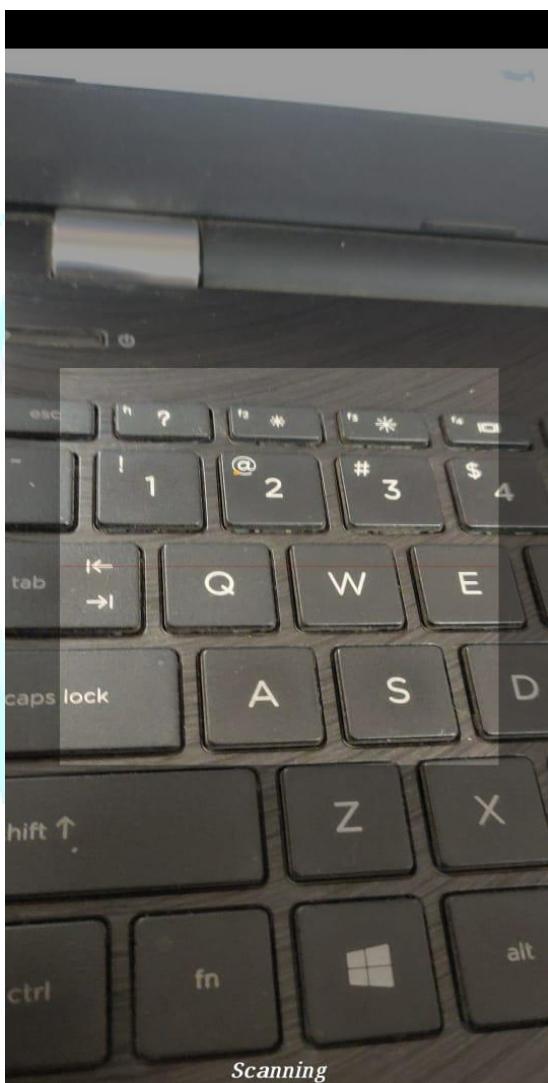


Fig. 6.16 Scanning Page Design

- Valid Vehicle Dialog

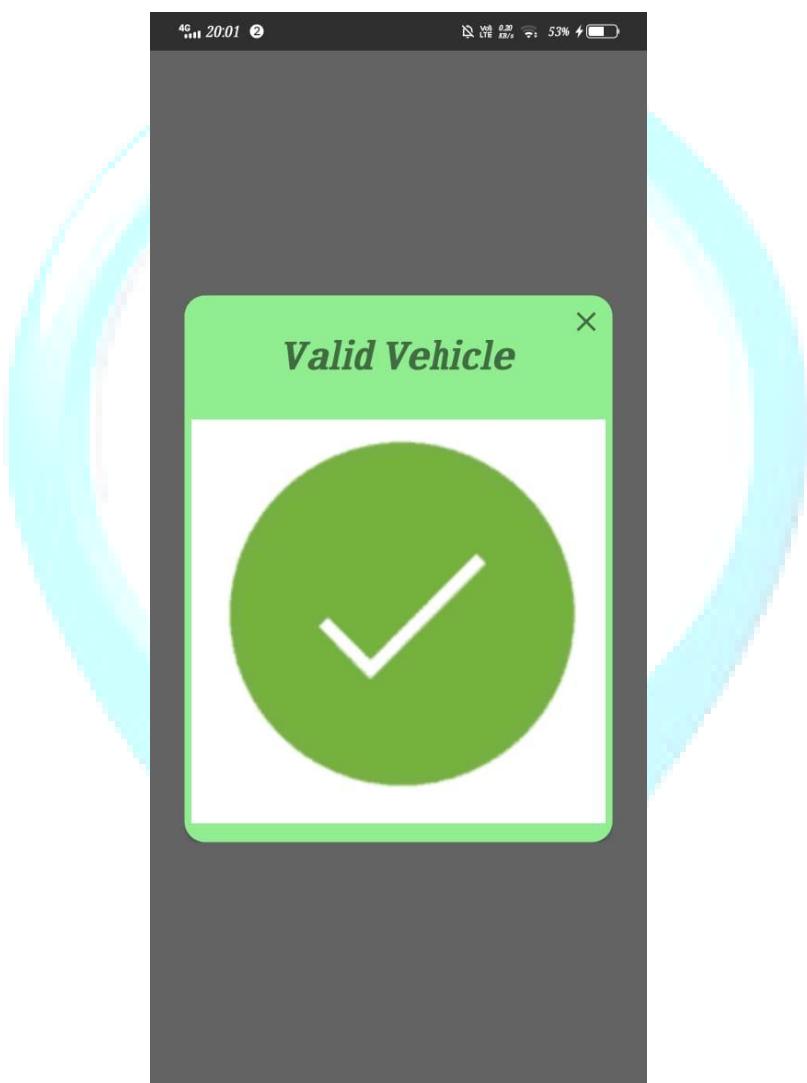


Fig. 6.17 Valid Vehicle Dialog Design

- Invalid Vehicle Dialog

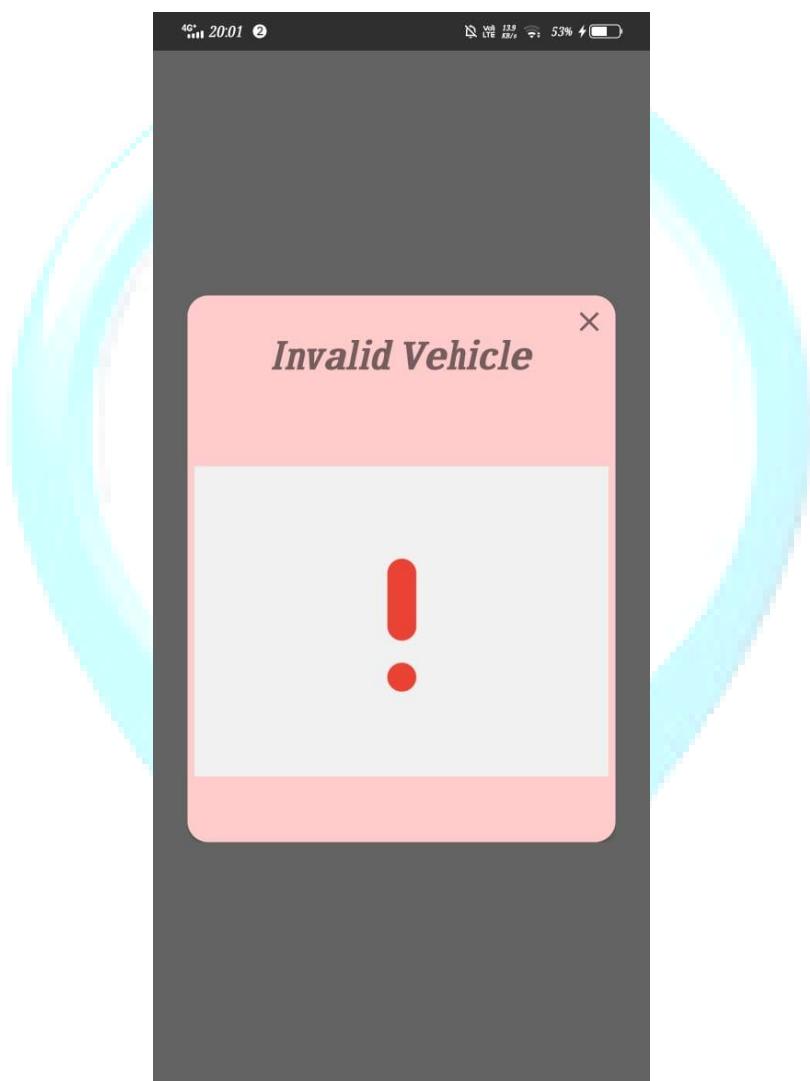


Fig. 6.18 Invalid Vehicle Dialog Design

Admin Panel Web-site Screen-Shots :-

- Main Home Page

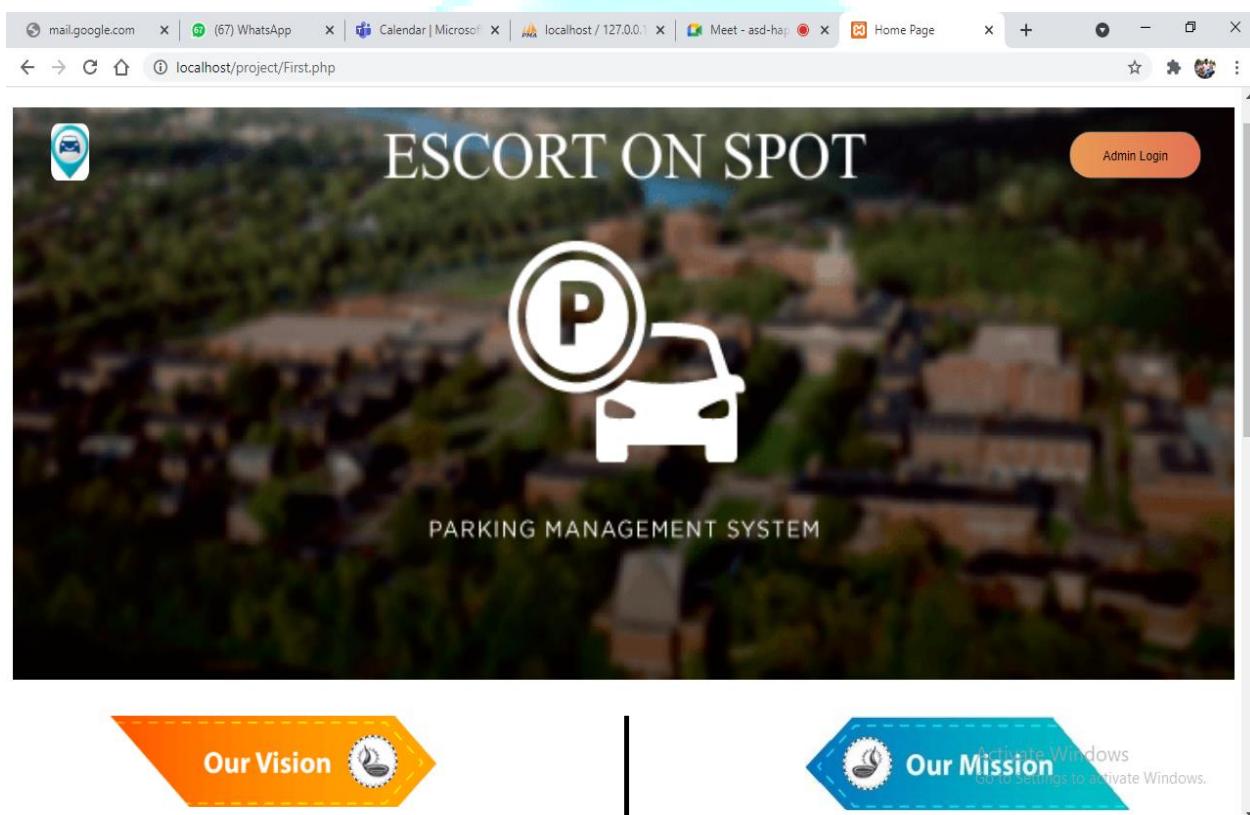


Fig. 6.19 Main Home Page Design

- Admin Login Page

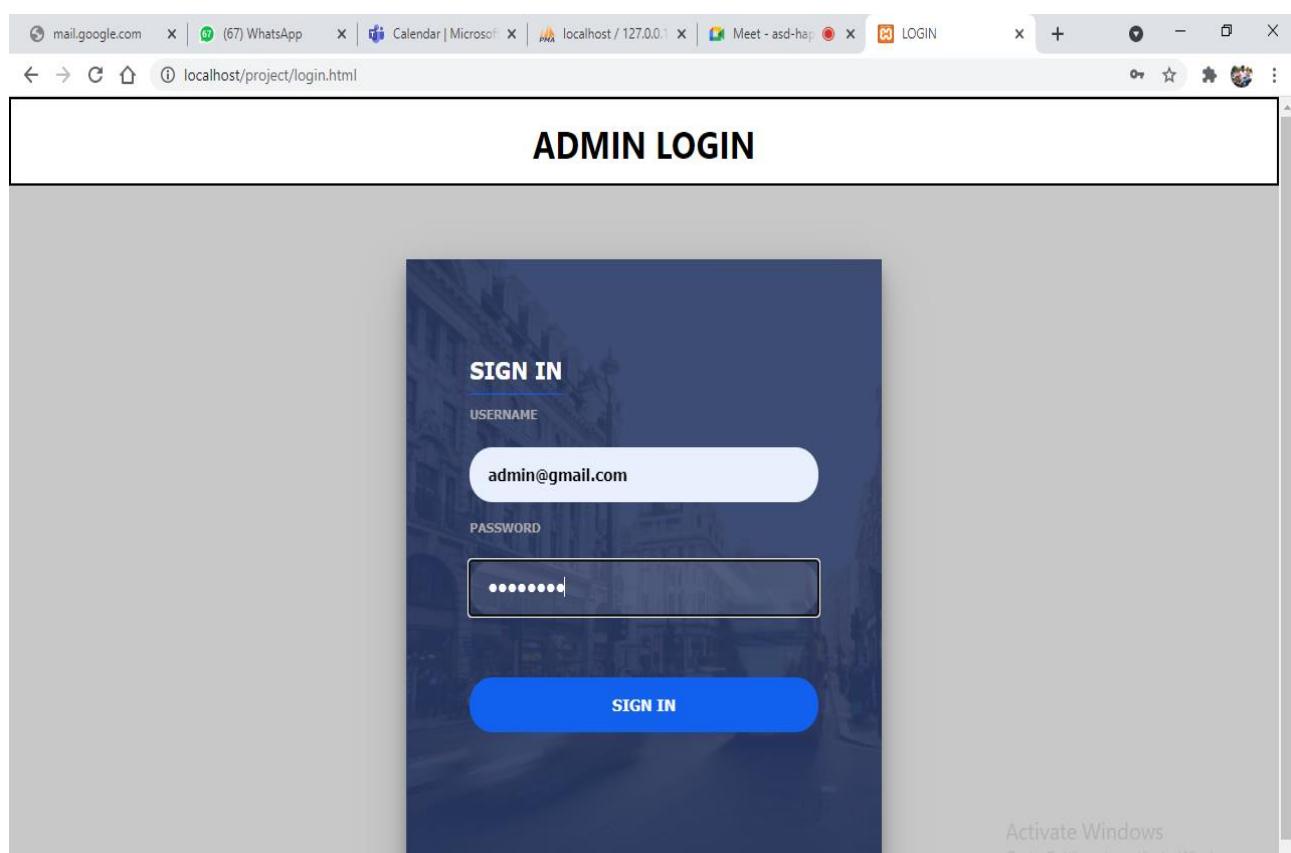


Fig. 6.20 Admin Login Page Design

- Dashboard Page

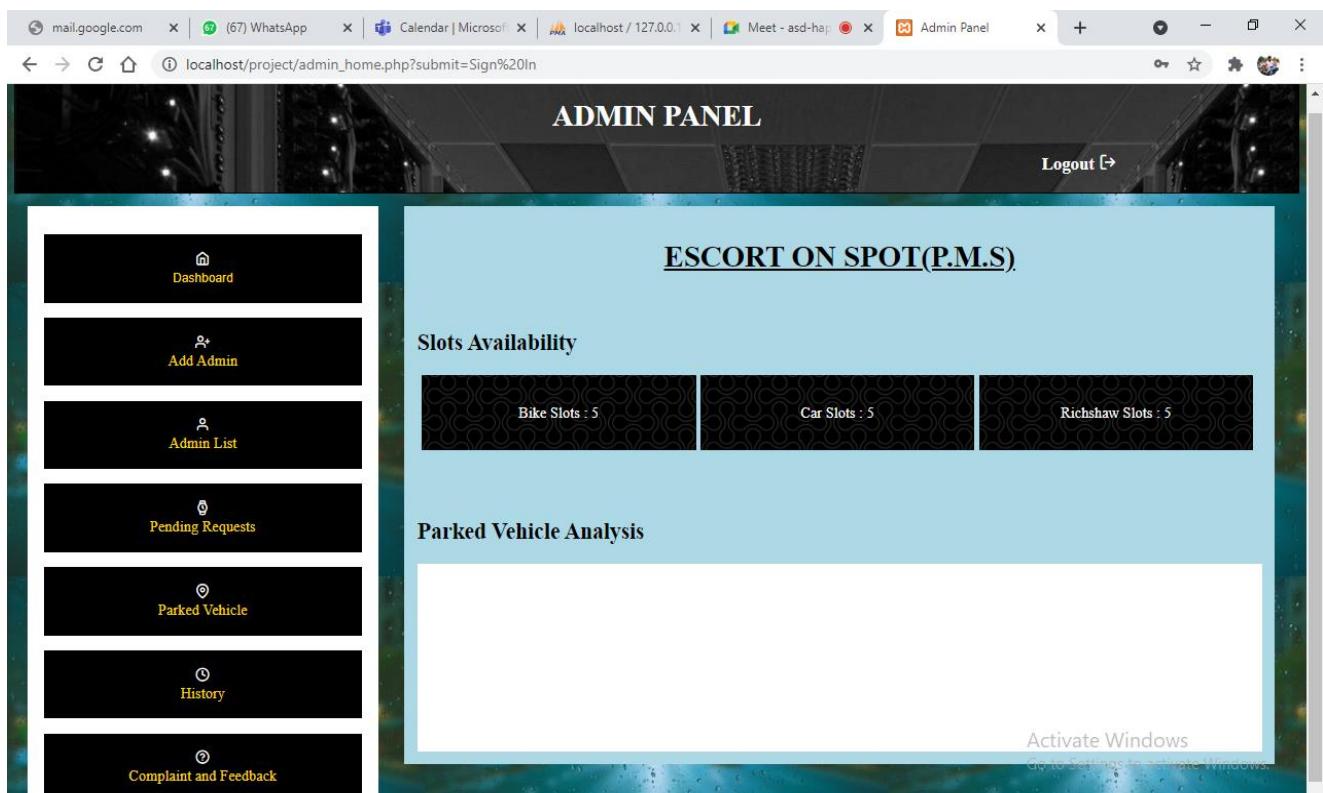


Fig. 6.21 Dashboard Page Design

- Add Admin Page

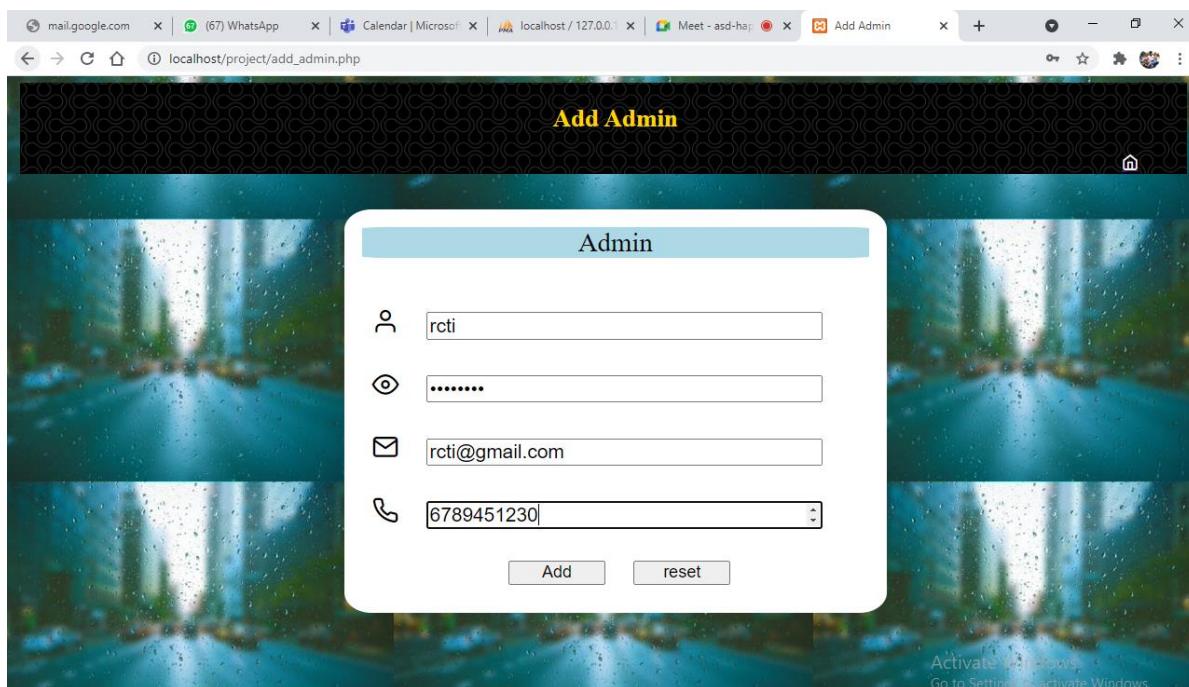


Fig. 6.22 Add Admin Page Design

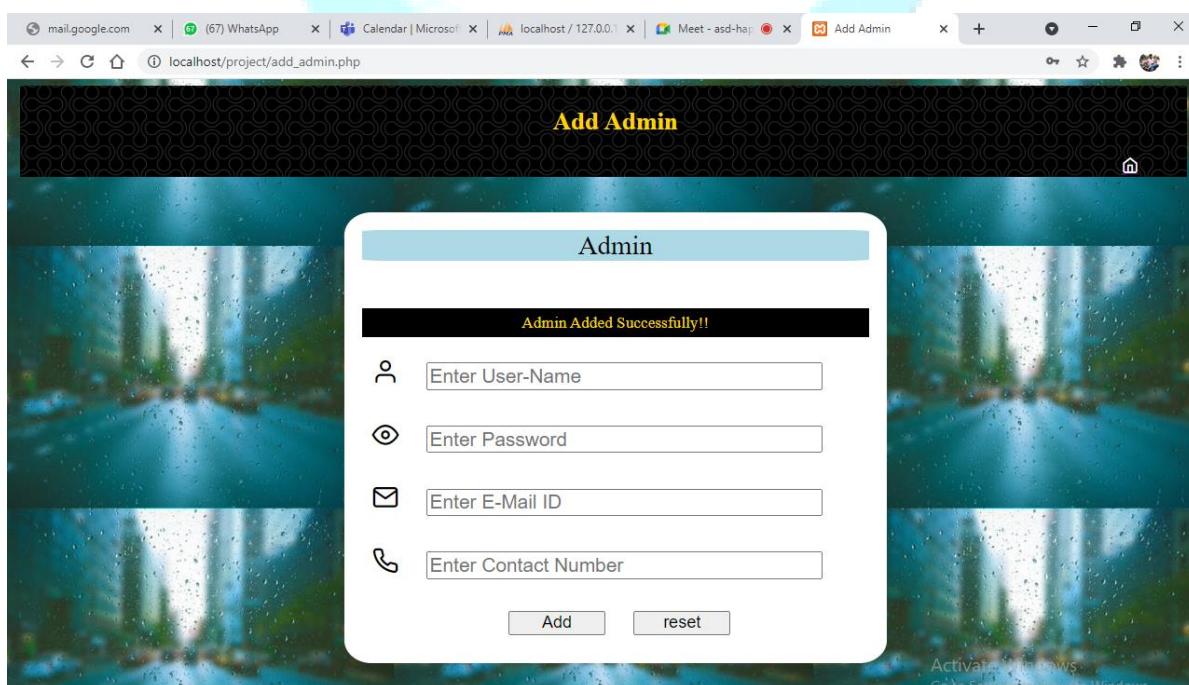


Fig. 6.22 Add Admin Page Design

- Admin List Page

The screenshot shows a web browser window with multiple tabs open at the top. The active tab is titled "Admin Panel" and displays the "Admin List" page. The page has a dark header with the title "Admin List". Below the header is a table with four columns: "Username", "Email", "Mobile Number", and "Action". The table contains four rows of data:

Username	Email	Mobile Number	Action
student	student@gmail.com	987654321	Delete Admin
admin@gmail.com	shashankchaudhary231@gmail.com	6353140359	Delete Admin
rcti	rcti@gmail.com	6789451230	Delete Admin

Fig. 6.23 Admin List Page Design

- Pending Requests Page

The screenshot shows a web browser window with multiple tabs open at the top. The active tab is titled "localhost/project/show_pending_requests.php". The main content area is titled "Pending Requests". At the top, there are three red-bordered boxes displaying "Bike Slots : 5", "Car Slots : 5", and "Richshaw Slots : 5". Below this is a table with the following data:

Parking Id	Vehicle Number	Vehicle_Type	Image Path	Date	Time	Approval	
P01	GJ-02-UU-2245	Bike	image	2021-04-18	15:07:12	Approve	Reject
P01	GJ-01-UJ-6951	Car	image	2021-04-18	15:07:45	Approve	Reject
P01	GJ-02-AA-1234	Car	image	2021-04-18	15:17:10	Approve	Reject
P01	GJ-02-AA-1235	Car	image	2021-04-18	15:17:19	Approve	Reject
P01	GJ-02-AA-1236	Car	image	2021-04-18	15:17:25	Approve	Reject
P01	GJ-02-AA-1237	Car	image	2021-04-18	15:17:32	Approve	Reject
P01	GJ-02-AA-1230	Richshaw	image	2021-04-18	15:18:39	Approve	Reject
P01	GJ-02-AA-1223	Richshaw	image	2021-04-18	15:18:47	Approve	Reject
P01	GJ-02-AA-1224	Richshaw	image	2021-04-18	15:18:53	Approve	Reject

Fig. 6.24 Pending Requests Page Design

- Parked Vehicle Page

The screenshot shows a web browser window with multiple tabs open at the top. The active tab is titled 'localhost/project/show_approved_requests.php'. The main content area has a dark header bar with the title 'Parked Vehicle' in yellow. Below this is a table with a light blue border and white background. The table has columns for Parking Id, Vehicle Number, Vehicle_Type, Image Path, Date, Time, and Status. There are three rows of data:

Parking Id	Vehicle Number	Vehicle_Type	Image Path	Date	Time	Status
P01	GJ-02-UU-2245	Bike	image	2021-05-02	17:39:11	Check-in Pending
P01	GJ-01-UJ-6951	Car	image	2021-05-02	17:39:15	Check-in Pending
P01	GJ-02-AA-1230	Rickshaw	image	2021-05-02	17:39:17	Check-in Pending

Below the table is a large, blurry image of a parking lot with several vehicles. A watermark in the bottom right corner of the image reads 'Activate Windows! Go to Settings to activate Windows.' and '17.20'.

Fig. 6.25 Parked Vehicle Page Design

- Dashboard Page (2nd time after approving vehicles)

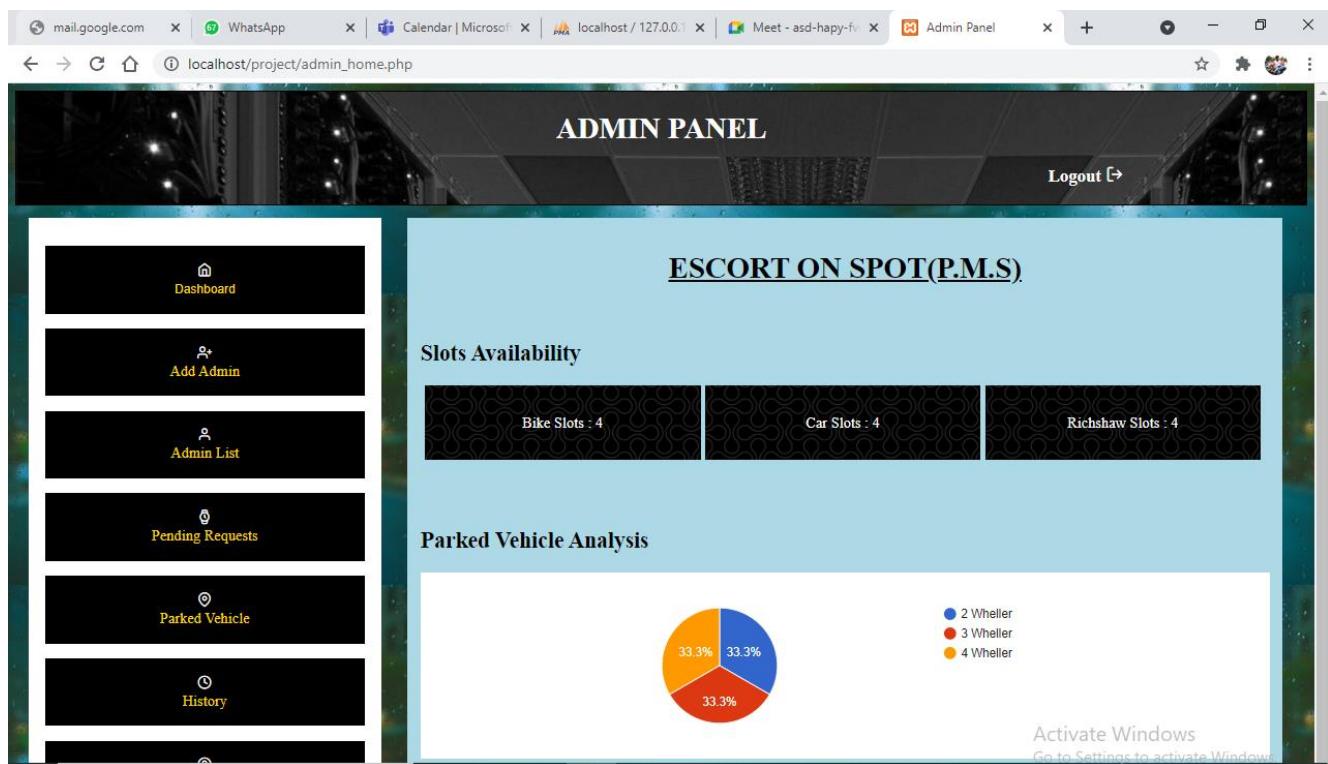


Fig. 6.26 Dashboard Page Design

- History Page

The screenshot shows a web browser window with multiple tabs open at the top. The active tab is titled "localhost/project/show_history.php". The main content area displays a table with the following columns: Parking Id, Vehicle Number, Vehicle_Type, Image Path, Date, Time, and Admin Status. The table contains six rows of data, all corresponding to parking ID P01. The vehicle type is consistently listed as "Bike". The image path column contains the word "image" repeated six times. The dates range from 2021-03-27 to 2021-04-18. The times range from 14:45:40 to 15:15:48. The admin status column shows alternating colors: red for Rejected and green for Accepted. At the bottom right of the table, there is a watermark that reads "Activate Windows Go to Settings to activate Windows".

Parking Id	Vehicle Number	Vehicle_Type	Image Path	Date	Time	Admin Status
P01	GJ-01-UJ-6952	Bike	image	2021-03-27	15:15:05	Rejected
P01	GJ-01-UJ-6951	Bike	image	2021-03-27	15:15:48	Accepted
P01	GJ-01-UJ-6953	Bike	image	2021-03-27	15:17:18	Accepted
P01	GJ-01-AS-3628	Bike	image	2021-04-03	14:45:40	Rejected
P01	GJ-01-UJ-6951	Bike	image	2021-04-03	14:46:11	Accepted
P01	gjnd	Bike	image	2021-04-18	15:04:55	Rejected

Fig. 6.27 History Page Design

- Feedback & Complaint Page

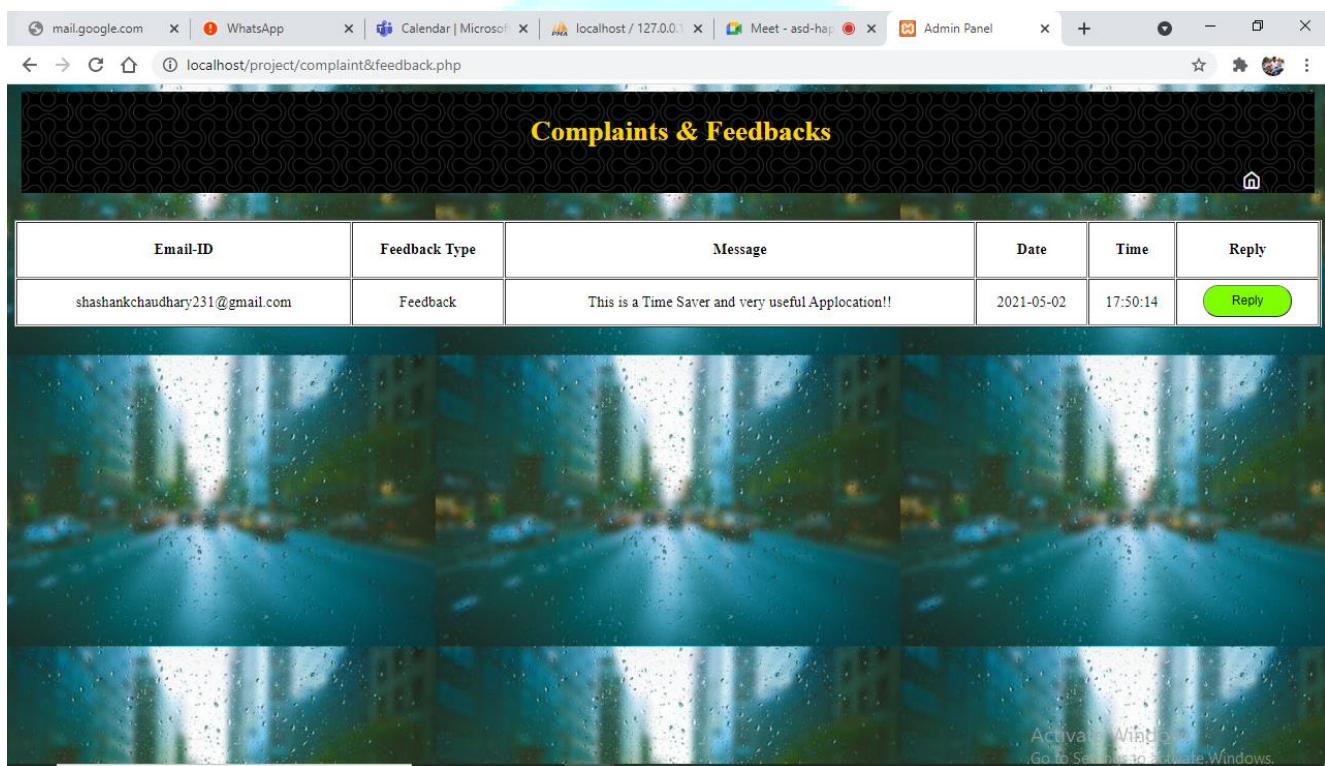


Fig. 6.28 Feedback & Complaint Page Design

- Feedback Response Mail

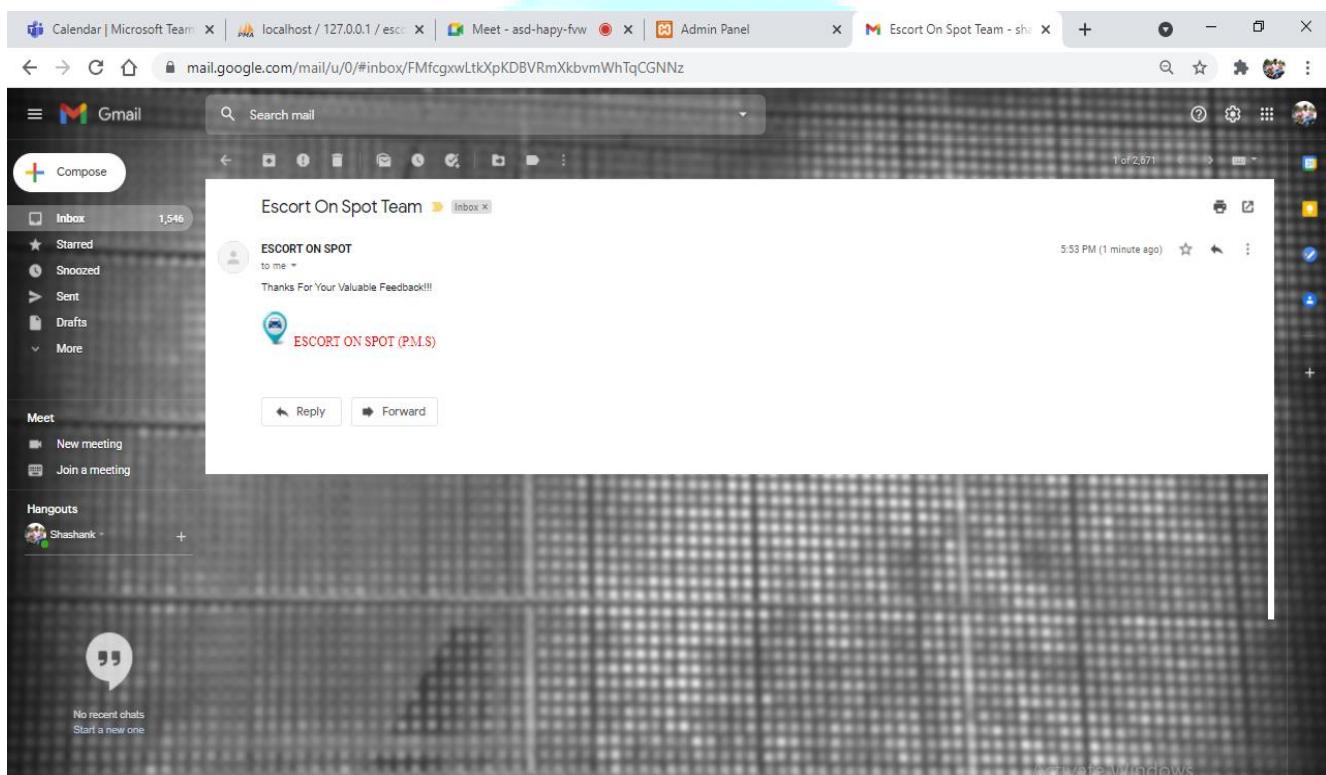


Fig. 6.29 Feedback Response Mail

6.3 SAMPLE CODING

- **XML CODE :-**

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/resize_1594294209571459772back"
    android:alpha="150"
    android:orientation="vertical"
    tools:context=".HomeActivity">

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:gravity="center_horizontal"
            android:orientation="vertical"
            android:textAlignment="center">

            <ImageButton
                android:id="@+id/imageButton"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_marginTop="15dp"
                android:contentDescription="Admin Panel"
                android:gravity="center_horizontal"
                android:padding="5dp"
```

```
    android:textAlignment="center"  
    android:tooltipText="Admin Panel"  
    app:srcCompat="@drawable/admin_logo" />
```

```
<TextView  
    android:id="@+id/textView4"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:padding="5dp"  
    android:text="Admin Panel"  
    android:textColor="#FAF5F5"  
    android:textSize="18sp" />
```

```
<ImageButton  
    android:id="@+id/imageButton2"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_marginTop="15dp"  
    android:gravity="center_horizontal"  
    android:textAlignment="center"  
    android:tooltipText="User Panel"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toBottomOf="@+id/imageButton"  
    app:srcCompat="@drawable/user_logo" />
```

```
<TextView  
    android:id="@+id/textView5"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:gravity="center_horizontal"  
    android:padding="10dp"  
    android:text="User Panel"  
    android:textAlignment="center"
```

```
        android:textColor="#F6F0F0"
        android:textSize="18sp"
        app:layout_constraintTop_toBottomOf="@+id/imageButton2" />
    </LinearLayout>
</ScrollView>

<RelativeLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"

    tools:context=".bottom_About_Us">

    <com.google.android.material.bottomnavigation.BottomNavigationView
        android:id="@+id/bottom_navigation"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        app:itemBackground="@color/colorPrimary"
        app:itemIconTint="@drawable/selector"
        app:itemTextColor="@drawable/selector"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:menu="@menu/menu_navigation"
        tools:ignore="MissingConstraints" />
</RelativeLayout>
</LinearLayout>
```

- Java Code :-

```
package com.example.myapplication;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.content.Context;
import android.content.Intent;
import android.net.ConnectivityManager;
import android.net.NetworkInfo;
import android.net.Uri;
import android.os.Bundle;
import android.view.MenuItem;
import android.view.View;
import android.view.animation.Animation;
import android.view.animation.AnimationUtils;
import android.widget.ImageButton;
import android.widget.Toast;

import com.google.android.material.bottomnavigation.BottomNavigationView;

public class HomeActivity extends AppCompatActivity {

    //Variables Initializing...
    private ImageButton imageView;
    private ImageButton imageView2;
    private BottomNavigationView bottomNavigationView;
    private long backpresses;
    Animation anim,anim1;
    private Toast backtoast;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_home);
```

```
imageButton = (ImageButton) findViewById(R.id.imageButton);
imageButton2 = (ImageButton) findViewById(R.id.imageButton2);
anim = AnimationUtils.loadAnimation(this,R.anim.bounce);
anim1 = AnimationUtils.loadAnimation(this,R.anim.bounce1);
imageButton.startAnimation(anim);
imageButton2.startAnimation(anim1);
checkconnection();

ConnectivityManager manager = (ConnectivityManager)
getApplicationContext().getSystemService(Context.CONNECTIVITY_SERVICE);
NetworkInfo activeNetwork = manager.getActiveNetworkInfo();

if(null!=activeNetwork) {
    if (activeNetwork.getType() == ConnectivityManager.TYPE_WIFI || activeNetwork.getType() ==
ConnectivityManager.TYPE_MOBILE ) {
        //Variables Assignment...

bottomNavigationView = (BottomNavigationView) findViewById(R.id.bottom_navigation);
bottomNavigationView.setSelectedItemId(R.id.dashboard);

//Admin Panel Click Listener...
imageButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        openAdminPanel();
    }
});

//User Panel Click Listener...
imageButton2.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        openUserPanel();
    }
});
```

```
//Bottom Navigation ItemSelected Listener...
bottomNavigationView.setOnNavigationItemSelectedListener(new
BottomNavigationView.OnNavigationItemSelectedListener() {

    @Override
    public boolean onNavigationItemSelected(@NonNull MenuItem menuItem) {
        switch (menuItem.getItemId()){
            case R.id.dashboard:
                startActivity(new Intent(getApplicationContext(), HomeActivity.class));
                overridePendingTransition(0,0);
                return true;

            case R.id.Feedback:
                startActivity(new Intent(getApplicationContext(), bottom_Feedback.class));
                overridePendingTransition(0,0);
                return true;

            case R.id.About_us:
                startActivity(new Intent(getApplicationContext(), bottom_About_Us.class));
                overridePendingTransition(0,0);
                return true;
        }
        return false;
    }

});
}

}

public void openAdminPanel(){
    try {
        Intent intent = new Intent(Intent.ACTION_VIEW,
        Uri.parse("http://192.168.0.104/project/login.html"));
        startActivity(intent);
    }
}
```

```
}

catch (Exception e){

    Toast.makeText(this,"Server is Unreachable! Try again Later",Toast.LENGTH_LONG).show();

}

}

public void openUserPanel(){

    Intent intent = new Intent(this, openUserPanel.class);

    startActivity(intent);

}

public void checkconnection(){

    ConnectivityManager manager = (ConnectivityManager)

getApplicationContext().getSystemService(Context.CONNECTIVITY_SERVICE);

    NetworkInfo activeNetwork = manager.getActiveNetworkInfo();

    if(null!=activeNetwork) {

        if (activeNetwork.getType() == ConnectivityManager.TYPE_WIFI) {

            Toast.makeText(this, "Wifi Enabled", Toast.LENGTH_SHORT).show();

        } else if (activeNetwork.getType() == ConnectivityManager.TYPE_MOBILE) {

            Toast.makeText(this, "Data Network Enabled", Toast.LENGTH_SHORT).show();

        }

    }

    else{

        Toast.makeText(this,"No Internet Connection", Toast.LENGTH_SHORT).show();

    }

}

@Override

public void onBackPressed() {

    if (backpresses + 2000 > System.currentTimeMillis()){

        super.onBackPressed();

        moveTaskToBack(true);

        android.os.Process.killProcess(android.os.Process.myPid());

        System.exit(1);

        backtoast.cancel();

    }

}
```

```
        return;  
    }else {  
        backtoast = Toast.makeText(getApplicationContext(),"Press Again to Exit",Toast.LENGTH_SHORT);  
        backtoast.show();  
    }  
    backpresses = System.currentTimeMillis();  
  
}  
}
```



7.0 TESTING

7.1 TESTING PLAN AND STRATEGY

7.2 TESTING METHODS

7.3 TESTING CASES



7.1 TESTING PLAN AND STRATEGY

This state a number of rules that can serve well as testing objectives:

- Testing is a process execution of a program with the intent of finding an error.
- A good test case is one that has a high probability of finding an as-yet undiscovered error.
- A successful test is one that uncovers an as-yet undiscovered error. These objectives imply a dramatic change in viewpoint for some hardware developers. They move counter to the commonly held view that a successful test is one in which no errors are found. Our objective is to design tests that systematically uncover different classes of errors and to do so with minimum amount of time & effort.

There are mainly five testing principles which are described below:

Principle #1: All tests should be traceable to customer requirements.

The objective of software testing is to uncover errors. It follows that the most sever defects are those that cause the program to fail to meet its requirements.

Principle #2: Test should be planned long before the actual testing begins.

Planning can begin as soon as the analysis model is complete. Detailed definition of test cases can begin as soon as the design model has been solidified.

Therefore, all tests can be planned and designed before any code has been generated.

Principle # 3: The Pareto principle applies to hardware testing.

Stated simply, the Pareto principle implies that 80 percent of all errors uncovered during testing will likely be traceable to 20 percent of all program components. The problem of course, is to isolate these suspect components and to thoroughly test them.

Principle # 4: Testing should begin “in the small” and progress toward testing “in the large.”

The first test planned and executed generally focuses on individual components. As testing progresses, focus shifts in an attempt to find errors in integrated clusters of components and ultimately in the entire system.

Principle # 5: Exhaustive testing is not possible.

The number of path permutations for even a moderately sized program is exceptionally large. For this reason, it is impossible to execute every combination of paths during testing. It is possible, however, to adequately cover program logic and to ensure that all conditions in the component-level design have been exercised.

Top down testing:

In top-down testing, testing starts with the most abstract components and works downwards.

Back-to-Back testing:

It is used when different versions of a system are available. They are tested together and their outputs are compared. Testing can be done to check the performance of the product.

System Testing:

System Testing typically involves running through every possible input to verify that it results in the right outputs using the software as an end-user would. It is needed to improve the performance of the system so user can have fast processing of all work.

7.2 TESTING METHODS

Black-Box Testing:

The technique of testing without having any knowledge of the interior workings of the application is called black-box testing. The tester is oblivious to the system architecture and does not have access to the source code. Typically, while performing a black-box test, a tester will interact with the system's user interface by providing inputs and examining outputs without knowing how and where the inputs are worked upon.

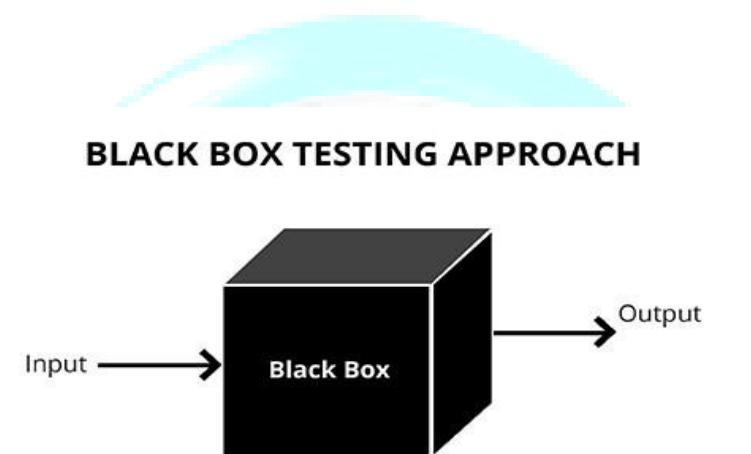


Fig. 7.1 Black-Box Testing

White-Box Testing:

White-box testing is the detailed investigation of internal logic and structure of the code. White-box testing is also called glass testing or open-box testing. In order to perform white-box testing on an application, a tester needs to know the internal workings of the code.

The tester needs to have a look inside the source code and find out which unit/chunk of the code is behaving inappropriately.

Grey-Box Testing:

Grey-box testing is a technique to test the application with having a limited knowledge of the internal workings of an application. In software testing, the phrase the more you know, the better carries a lot of weight while testing an application.

Mastering the domain of a system always gives the tester an edge over someone with limited domain knowledge. Unlike black-box testing, where the tester only tests the application's user interface; in grey-box testing, the tester has access to design documents and the database. Having this knowledge, a tester can prepare better test data and test scenarios while making a test plan.

7.3 TESTING CASES

Admin Test-Cases.

PURPOSE	REQUIRED INPUT	EXPECTED OUTPUT	Original Output
Login	Valid Username and Valid Password	User enters both things valid, then user will be redirected to the Dashboard.	
	Invalid Username and Password	If any of the field is incorrect, then error will be generated of the incorrect fields. And, until the user inserts the correct information, he/she will not be re-directed to the dashboard.	
Add Admin	Insert Admin's Details.	Admin has to provide all the required informations such as Username, password, E-mail id, contact number and the click on “Add” Button to add the Admin.	
	Incomplete Details	If any field is empty on clicking the “Add” button, It will show an error message saying “Please Fill out this fields”. And, until the Admin inserts all the required fields, he / she will not be able to add the admin to the system.	
	Approve Button	If Admin recognize as the given details provided by the user is correct then he / she will accept the request by pressing the “Approve” Button.	

Pending Requests		After that the request will move from the pending request to the parked vehicle and showing the status as “Check-in pending” until the vehicle is not checked-in.	
	Reject Button	If Admin recognize as the given details provided by the users is incorrect then he / she will reject the request by pressing the “Reject” Button. After that the request will move from the pending requests to the history as showing the status as “Rejected”.	
Complaint & Feedback	Reply	Admin can see all the Complaints and Feedbacks in this section and can reply to any complaints and feedback by pressing the “Reply” button.	

Note :-

-  Icon represents that our output and expected output are same and satisfying.
-  Icon represents that our output and expected output are not same or not satisfying.

⊕ User Test-Cases.

PURPOSE	REQUIRED INPUT	EXPECTED OUTPUT	Original Output
Select Role	Select Their Role	User will select their role that he / she is the Admin or User of the System.	
Select Vehicle Type	Select Their Vehicle Type	User will select their vehicle type from given options i.e. Two Wheeler, Three Wheeler, Four Wheeler and then move forward.	
Fill Details	Insert Details	User will have to fill their details like Vehicle number and have to upload their RC Book and click on “Proceed Next”.	
	Missing Details	If any details is missing and not filled by the user then it will show the Toast Message saying “Fill Required Fields”. And, until the user will not fill all the details he / she will not be able to move forward.	
Slot Availability	Slot Available	When the User presses the “Proceed Next” button then the system will check the availability of the slots of the specific vehicle type and if the slots are available then it will show the instruction dialog and can move further to generate QR Code.	
Slot Un-Available	Slot Unavailable	When the User Presses the “Proceed Next” Button then the system will check the availability of the slots of	

		the specific vehicle type and if the slot is unavailable then the user will be shown the dialog containing the list of the “Nearby Parkings” with their Locations on Google Map.	
QR Code	Generate QR Code	Now if the slot is available then after reading the instructions in the “Instructions Dialog” user will click on “Next” Button and generate QR Code by pressing the “Generate” Button and then can take the Screen-shot of the QR Code for Further Reference.	
Complaint & Feedback	Complaint / Feedback	User can make complaint or can give Feedback of the System by going on the complaint/Feedback page located on the Home page. Their User need to fill the Form by giving their correct informations and selecting the type of the message i.e. Complaint or Feedback. And lastly when clicking on “Submit” button the feedback or complaint will be registered successfully to the Admin.	

Note : -

-  Icon represents that our output and expected output are same and satisfying.
-  Icon represents that our output and expected output are not same or not satisfying.

QR Scan Application

PURPOSE	REQUIRED INPUT	EXPECTED OUTPUT	Original Output
Entry Gate QR Code Scanner	Approved QR Code	User will have to put the QR Code under the QR Code Scanner Application and if the Request of the user containing the QR Code has been approved by the admin then it will show that the QR Code is Valid and will send command to open the barrier and set the status of the user as "Checked-In" in the parked vehicle section on the Admin Panel.	
	Rejected QR Code	User will have to put the QR Code under the QR Code Scanner Application and if the Request of the user containing the QR Code has been rejected by the admin then it will show that the QR Code is Invalid and will send command to not open the barrier.	
Exit Gate QR Code Scanner	QR Code	User will have to put the QR Code under the QR Code Scanner Application and it will show that QR Code is Valid and will send command to open the barrier and move the user's record from parked vehicle to history and set Admin Status as "Accepted" on the Admin Side.	

Note :-

-  Icon represents that our output and expected output are same and satisfying.
-  Icon represents that our output and expected output are not same or not satisfying.

⊕ Arduino Circuit Test-Cases

PURPOSE	REQUIRED INPUT	EXPECTED OUTPUT	Original Output
Barrier Opening	Valid User (1)	When the User is Valid then the QR Code Scanner Application will send the command as “1” to the Arduino Circuit through the “ESP-8266 Bluetooth Module” and the Arduino Circuit will open the barrier so can user can go to the parking slot to park their vehicle.	
	Invalid User (0)	When the User is Invalid then the QR Code Scanner Application will send the command as “0” to the Arduino Circuit through the “ESP-8266 Bluetooth Module” and the Arduino Circuit will Turn on the Red light on the barrier and do not open the barrier so can user can not go to the parking slot to park their vehicle.	

Note : -

-  Icon represents that our output and expected output are same and satisfying.
-  Icon represents that our output and expected output are not same or not satisfying.

8.0 FUTURE PLANNINGS

❖ Short-term Future Plans

- To add time slots according to payment.
- To add layout for choosing a slot.
- To Provide proper QR code embedded ticket to the user.
- Separate User account for every user.
- To add google map view to reach our parking destination.
- To enhance and use advanced design of GUI for the application.

❖ Long-term Future Plans

- To make system “**Dynamic**”.
- To add image recognition system in place of admin to approve/reject user request.
- To make it available for those who want to maintain their parking all over the world.

Note :-

- Our Short-term future Goals are designed for the flexibility and comfort-ability for the users of the system.
- Our Long-term future Goals are designed for the flexibility and comfort-ability for the admin of the system.

9.0 CONCLUSION

Till now, we conclude that we had completed our documentation and also Implementation of our Application. We also conclude that till now the implementation we had done on our application was the best experience of all the things, because we had got somewhat advance kind of knowledge and we have learnt so many new concepts of Android and PHP while doing the project. We also conclude that we have learnt so many new concepts of testing methods and strategies and different types of waterfall models and many more things while doing the whole documentation..

We finally can conclude that according to the contents of the above documentation our Project has been **COMPLETED SUCCESSFULLY.**

10.0 REFERENCES AND BIBLIOGRAPHY

- www.stackoverflow.com
- www.draw.io
- www.wikipedia.org
- www.javapoint.com
- www.tutorialspoint.com
- www.github.com

