

**University Institute of Information Technology,**

**PMAS-Arid Agriculture University,**

**Rawalpindi Pakistan**

**Ez Parking**

***By***

**Bisma Arshad 17-Arid-1706**

**Maryam Ashraf 17-Arid-1720**

**Rimsha Masood 17-Arid-1743**

***Supervisor*Mr. Saqib Majeed**

***Bachelor of Science in Software Engineering (2017-2021)***

**The candidate confirms that the work submitted is their own and appropriate  
 credit has been given where reference has been made to the work of others**.

**DECLARATION**

We hereby declare that this software, neither whole nor as a part has been copied out from any source. It is further declared that we have developed this software documentation and accompanied report entirely on the basis of our personal efforts. If any part of this project is proved to be copied out from any source or found to be reproduction of some other. We will stand by the consequences. No Portion of the work presented has been submitted of any application for any other degree or qualification of this or any other university or institute of learning.

Bisma Arshad Maryam Ashraf Rimsha Masood

**CERTIFICATE OF APPROVAL**

It is to certify that the final year project of BS (SE) “EzParking” was developed by **“Bisma Arshad (17-Arid-17106)”, “Maryam Ashraf (17-Arid-1720)” and “Rimsha Masood (17-Arid-1743)”** under the supervision of “Mr. Saqib Majeed” and that in their opinion; it is fully adequate, in scope and quality for the degree of Bachelors of Science in Software Engineering.

---------------------------------------

**Supervisor**

---------------------------------------

**(Mr. Imran Khurram)**

**External Examiner**

---------------------------------------

**Administrator UIIT**

**Executive Summary**

We will develop an advanced and efficient dedicated parking system using latest technologies; these technologies will be android and desktop. This system will be introduced as a solution of these issues as well as a substitution to the manual car parking systems at commercial spaces. Hardware will also be involved in it. The users in this project will be car driver and parking lot owner. This app will allow the car drivers to register themselves and to reserve parking for any day at any time if available. Wi-Fi will be required to run the app. Ensured of security will be one of our main concerns for which Artificial Intelligence will be used. Time and money will be saved due to this system.

**Acknowledgement**

All praise is to Almighty Allah who bestowed upon us a minute portion of His boundless knowledge by virtue of which we were able to accomplish this challenging task.

We are greatly indebted to our project supervisor “Mr. Saqib Majeed” for personal supervision, advice, valuable guidance and completion of this project. We are deeply indebted to him for encouragement and continual help during this work.

And we are also thankful to our parents and family who have been a constant source of encouragement for us and brought us the values of honesty & hard work.

Bisma Arshad Maryam Ashraf Rimsha Masood

**Table of Contents**

**Introduction 1**

[1.1 Brief](#_Toc268523777) 1

[1.2 Relevance to Course Modules](#_Toc268523779) 2

[1.3 Project Background](#_Toc268523780) 2

[1.4 Literature Review](#_Toc268523780) 3

[1.5 Methodology and Software Life Cycle](#_Toc268523782) 4

**Problem Definition** 6

[2.1 Purpose](#_Toc268523795) 6

[2.2 Product Functions](#_Toc268523796) 6

[2.3 Proposed Architecture](#_Toc268523804) 7

[2.4 Project Deliverables](#_Toc268523806) 8

[2.5 Operating Environment](#_Toc268523807) 8

[2.6 Assumptions and Dependencies](#_Toc268523808) 9

**Requirement Analysis** 10

[3.1 Functional Requirments](#_Toc268523823) 10

[3.2 Non – Functional Requirments](#_Toc268523825) 10

[3.3 Use case Model](#_Toc268523823) 11

[3.3.1 Use Case Diagarm](#_Toc268523787) 11

[3.3.2 Actors Discription](#_Toc268523787) 12

[3.3.3 Use Case Discription](#_Toc268523787) 12

**The Design** 15

[4.1 System Architecture](#_Toc268523830) 15

[4.2 System Design](#_Toc268523830) 15

[4.2.1 UML Structural Diagrams](#_Toc268523787) 16

[4.2.1.1 Component Diagram](#_Toc268523787) 16

[4.2.1.2 Package Diagram](#_Toc268523830) 17

[4.2.1.3 Deployment Diagram](#_Toc268523830) 18

[4.2.2 UML Behavioral Diagrams](#_Toc268523787) 19

[4.2.2.1 Activity Diagram](#_Toc268523830) 19

[4.2.2.2 State Machine Diagram](#_Toc268523830) 20

[4.2.3 UML Interaction Diagrams](#_Toc268523787) 21

[4.2.3.1 Sequence Diagrams](#_Toc268523830) 21

[4.3 Node Structure](#_Toc268523830) 24

[4.4 Class Diagram](#_Toc268523830) 25

[4.5 Communication Design Protocol](#_Toc268523830) 26

**Implementation** 31

[5.1 Communication Protocol Implementation](#_Toc268523830) 31

[5.2 Component Diagram](#_Toc268523830) 32

[5.3 User Interface](#_Toc268523830) 35

**Testing and Evaluation** 40

[6.1 Verification](#_Toc268523830) 40

[6.1.1 Functional Testing](#_Toc268523787) 40

[6.1.2 Static Testing](#_Toc268523787) 44

[6.2 Validation](#_Toc268523830) 44

[6.3 Usability Testing](#_Toc268523830) 45

[6.4 Unit Testing](#_Toc268523830) 45

* 1. [Integration Testing](#_Toc268523830) 45

6.6 [System Testing](#_Toc268523830) 45

6.7 [Acceptance testing](#_Toc268523830) 45

6.8 [Stress Testing](#_Toc268523830) 46

6.9 [Hardware Configuration Testing](#_Toc268523830) 46

6.10 [Evaluation](#_Toc268523830) 46

6.11 [Deployment](#_Toc268523830) 46

6.12 [Maintenance](#_Toc268523830) 46

**Future Work** 47

**References** 48

**List of Figures**

Fig 3.1 Use Case Diagram 11

Fig 4.1 System Architecture 15

Fig 4.2 Component Diagram 16

Fig 4.3 Package Diagram 17

Fig 4.4 Deployment Diagram 18

Fig 4.5 Activity Diagram 19

Fig 4.6 State Machine Diagram 20

Fig 4.9 Sequence Diagram 21

Fig 4.14 Node Structure 24

Fig 4.15 Class Diagram 25

Fig 4.16 Communication Diagram 26

# Chapter 1: Introduction

In this chapter we will discuss over view of our whole project, its brief introduction, and how it is relevant to the courses which we have studied during our degree. We will also discuss project literature review and its analysis and methodology that we will use in project.

# Brief

Increase in population is leading to increase in necessities of life. Over the decades Pakistan has been developed significantly, presently we are in this state that we have a part of well reached roads, commercial building and expanding number of automobiles. For parking these cars in parking space, we use a manual procedure of parking where most of the cases are unplanned and need of discipline due to this, individuals can stop their cars anyplace they need to, which makes a mess as individuals don't take after the specific prompt most of the time. As a result of this, a tremendous traffic jam takes place. One of main need of life is transportation so increase in number of vehicles requires a smart automated parking system to reduce mess and crowd. Many parking systems have been introduced in past but there were many flaws in them. To address these flaws, we will develop an advanced and efficient dedicated parking system using latest technologies; these technologies will be web and android. This system will be introduced as a solution of these issues as well as a substitution to the manual car parking systems at commercial spaces. Hardware will also be involved in it. The users in this project will be car driver and parking lot owner. This app will allow the car drivers to register themselves and to reserve parking for any day at any time if available. Wi-Fi will be required to run the app and for connectivity of hardware and application. Ensured of security will be one of our main concern for which Artificial Intelligence will be used. Time and money will be saved due to this system.

As we all know that the world is advancing day by day new technologies come and go, many new methods are been introduced almost daily, therefore, the demand of the new systems have been increased in every organization old system have been replaced by new systems, the question is why? It’s because every organization demands excellent quality work more profit to be earned in short span of time, to gain as much market trust as they can, and not to forget as quick they can so they can compete with other organizations. New systems should be user friendly, efficient; they should provide security and fast. As these four are mentioned, my idea will be based upon these. This Application allows the user to register and to reserve parking slot for any day at any time if available. Wi-Fi Technology is used for the connectivity of hardware and application. For security purpose, image processing and passcodes are used. Database is managed by the lot owner through desktop application. Hardware is also involved in our project. The future of the smart parking market is expected to be significantly influenced by the arrival of automated vehicles (AVs). The goal of intelligent parking systems is to know where parking is available and to let the driver know as well, making it easier for cars to find their way into parking spots.

# Relevance to Course Modules

Almost everything which we are using in our project “EzParking” is totally relevant to our course materials.

* The android app which we are developing in our project is relevant to the subject “Modern Programming Language".
* The data of the users which will be stored in Database and status will be shown on screen, is relevant to the subject Database Management System.

# Project Background

Nowadays, in this busy world it’s really hard for a person to find a spot for parking. The current parking system doesn’t give the user a specified parking slot inside the area. Parking in general in a long and time-consuming process and we hope to provide a solution to alleviate this problem.

The main aim that we have is to create a completely automated car parking system with minimal human interference. In the context of the present age, it is said that we do not have time. In our daily life, usually in the office, on the street, in shopping complexes, everywhere is the only problem, and that is the parking problem. Due to illegal parking, we have to face extreme misery every day. And we all know how much time is wasted due to this disaster. And that is why we have to plan this system. There has been a lot of research on this before. But we hope our ideas will be different and more perfect than others. Because, we have seen in existing systems sometimes accidents can occur in parking situations by cars going at highspeed caused by frustrated drivers unable to find a parking space for a long period of time. In our project we propose a smart and automated car parking model that will help the user in booking their parking spaces beforehand and the vehicle will be able to park automatically once in the parking zone. Hence, we aim to provide a completely safe and automated experience that is robust and can be implemented in real time and hopefully be implemented as the general norm for parking systems in the future.

# Literature Review

Most of the prevailing car parks don’t have a significant system. The matter that always occurred at the parking lot was wasting of time in checking out the available parking spaces. Traditional or manual car parking system was everywhere in our country but that system was full of problems. We have seen huge traffic jam in front of the parking side in many shopping malls and hospitals. The parking guard stopped the entire vehicle and gave a payment slip, due to which traffic jam occurred. Many existing systems have been introduced in the past regarding parking systems. Car parking system was being used in many congested area or location where there were many meeting points of people like where there are more than one shopping complex near to each other or where there is megamall or stadium. Security problem was another problem in manual car parking, people mostly entered in parking slot and there snatching, robbery mostly happened. Here we are discussed different systems like Park Mobile - Find Parking and Park NYC.

* Park NYC

The Park NYC app is customizable to set the availability of parking based on certain preferences such as user parking spaces or cheaper prices and Park NYC app used to be so convenient, didn't have to leave your car just to discover that the park meter machine isn't working. Park NYC App does not permit you to extend parking time even when you have not gone beyond allotted time as stated on signs

* Park Mobile – Find Parking

Park Mobile using token system currently offers its mobile payments program. Parking zone based on your GPS location and one has to press start to start your parking session.

You can also manually enter a parking zone (the parking zone number on the meter or area sign) or choose a zone from your favorites. There is also an option to pay the parking fee in the application itself and in park mobile no automatically parking spaces shown and no regular parameter for vehicle detection technology.

# Analysis from Literature Review

The goal is to make a cleaner and less complex system of parking. So as per the idea we enabled futuristic system that will reduce work of man power and proceed towards automation. Here the app is designed in a way which everyone can use as a pass and book slots anytime within a given time scape. The app allows flexible booking and cancelling in the customer side. The app for the parking owners is managed in a way that helps them tracks the slot of customers and let their check in and check out times automated using vehicle number plate scanning mechanism. This mechanism allows only the booked slot customers in, which makes unwanted people to visit and crowd the parking place and to avoid culprit action.

# Methodology and Software Lifecycle for This Project

We use Agile methodology. The proposed system is going to follow the agile development method because this method assists in responding to the unpredictability of constructing software. This method offers a light framework and focus on rapid delivery of the software. Moreover, agile method facilitates us to create and respond to change in an uncertain and turbulent situation. Our focus is to collaborate with stakeholders during project to ensure the product quality which is impossible without using agile methodology.

* + 1. Rationale behind Selected Methodology
* Creativity and innovation
* Lower costs
* Improved quality
* Customer satisfaction
* Focus on users
* Early and predictable delivery
* Predictable costs and schedule
* Allows for change

**Chapter 2: Problem Definition**

# Purpose:

The main aim that we have is to create a completely automated car parking system with minimal human interference. In the context of the present age, it is said that we do not have time. In our daily life, usually in the office, on the street, in shopping complexes, everywhere is the only problem, and that is the parking problem. Due to illegal parking, we have to face extreme misery every day. And we all know how much time is wasted due to this disaster. And that is why we have to plan this system. There has been a lot of research on this before. But we hope our ideas will be different and more perfect than others. Because, we have seen in existing systems sometimes accidents can occur in parking situations by cars going at high-speed o caused by frustrated drivers unable to find a parking space for a long period of time. In our project we propose a smart and automated car parking model that will help the user in booking their parking spaces beforehand and the vehicle will be able to park automatically once in the parking zone. Hence, we aim to provide a completely safe and automated experience that is robust and can be implemented in real time and hopefully be implemented as the general norm for parking systems in the future.

# Product Functions:

The product functions of EZParking are, Register**:** In registration the user can add their personal information and the information can be verified and store in to the database and the user account created **Login:** The user provide their account credential after that they can use the app features. **Manage Database:** The admin can manage database i.e., update user, add user, view user, search users, delete users etc. **Slot booking**: The user can send request for booking and then the system will display the time slots and user will select the time slot. **Add User:** Admin add users and enters the user’s name and its details after that it stored in to database. **Cancel Booking:** if user want to cancel the booking, then payment will not refundable. **Manage user info:** The admin can manage user info the admin can delete, add, update user name, details and its other information. **Scan No Plate:** Camera will scan the car number plate if number plate match with user data, then barrier will allow to enter user.

* register
* login
* Manage Database
* Slot Booking
* Add users
* Cancel booking
* Manage User Info
* Scan Number Plate

**2.3. Proposed Architecture:**

3Tier Architecture has been used in our project.

We have three layers in 3Tier Architecture:

1. Presentation Tier
2. Application Tier
3. Data Tier

* **Presentation Tier:**

We will develop an Android App in which we will create interfaces by using Java Language. These interfaces will display all the information of User.

* **Application Tier:**

In this layer we will apply logics in our Android Application by using Java Language.

This tier- also called Middle tier, Logic tier and Business tier, pulled from presentation tier.

* **Data Tier:**

Data is this tier is kept independent of application servers and Business logic.

In this layer we will use MongoDB which will manage to store the data of users.

**2.4. Project Deliverables:**

Following are the deliverables and development requirements:

**2.4.1 Projects Deliverables:**

In this project the deliverables will consist of the input as equipment (i.e., hardware components and software components) then there will be a process (i.e., development phases in which development will be completed) applied on the input and then as a result there will be an output (Project being completed “EZParking”). In this case product deliverables are the completed parts or modules of the project. Input will be hardware and software-based components. The project is divided into in different modules and each module is major milestone in the project.

**2.4.2 Development Requirements:**

Development requirements are the requirements needed for the development purposes without which the development is not possible. It can be hardware, software or any kind of requirements. These include the software and hardware equipment’s, time and date constraints, budget, planning, following the SDLC etc. development requirements are met accordingly to make sure that the end result does not differ from what is expected and that it can perform its functionality accurately and perfectly without any glitches.

**2.5. Operating Environment:**

Operating environment for the EzParking is as listed below:

**Operating System**: Window 10 and Android Operating system

**Database**: Firebase

**Platform:** Android studio in which android app will be developed using java language. Front end will be designed in android and backend will be designed in Java.

Android App will be communicated with Arduino.

**2.6. Assumptions and Dependencies:**

**Assumptions:**

The application developed from this technique is more efficient than others. This software gives prefect result in real time for finding parking lots. The customer will not fail to park at his or her assigned parking spot. Parking garages are open 24 hours a day. A reservation can be made for any time of the day.

**Dependencies:**

This software is dependent on internet connectivity.

Application is also dependent on proper placement of hardware.

# Chapter 3: Requirement Analysis

In this chapter we will define all the requirements of proposed system that include functional and non-functional requirements. We will also discuss about use cases of the system and see how our system will respond to various use cases.

# Functional Requirements

For our system to work and facilitate the user number of functional requirements have been are needed. These functional requirements also be presumed as interface requirements as they are all but interface.

* The system will allow the user to register himself so that he can make further future reservation.
* The system will authenticate the user through login functionality.
* The system will display the available time slots
* The system will allow the user to book the display time slot.
* The system will allow the user to make payment.
* The system will confirm the booking through sending notification.
* The system will allow the admin to define new parking slots, modify the data of existing parking slots, viewing the data of all existing parking slots and also delete the parking slots.
* The system will scan the car number plate and verify it through database and then lift up the barrier.
* The system will turn off the status of the booked parking slots.

# Non-Functional Requirements

* The system should be able to handle the concurrent requests from different users.
* The system should provide confidentiality for user data.
* The system should be stable and reliable enough to handle the exceptions.
* The system should be available for 24/7 of the time to handle the concurrent request of the users.
* The system should permit only authorize users to ensure its security.
* The system should be efficient enough to handle the concurrent request to the user.
* Interface and the system itself should be user friendly so that the customer will feel it easy to use.
* The system will authenticate the user by verifying the credentials to database.

# Use Case Model

In the Unified Modeling Language (UML), a use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system. Following are the use cases of the EzParking.

**3.3.1 Use Case Diagram:**



**Fig 3.1: Use Case diagram**

* + 1. **Actors Description:**

**We have four types of actors.**

**Driver:**

Driver is the primary actor which directly interacts with the app. He can make booking, payment and can cancel the booking. He has to register and authenticate himself**.**

**Admin:**

Admin registers and authenticates himself. He manages the database which contains information regarding parking lots.

**System:**

System performs all the action that are being automated in our project. It displays available time slots, verifies number plate and allows entry off car according to that.

**Database:**

Database stores all the information and data fetched through it as well.

* + 1. **Use Case Description:**

|  |  |
| --- | --- |
| **Use Case ID:** | ID-01 |
| **Use Case Name:** | Authentication |
| **Actors:** | User, System |
| **Description:** | User will provide its credentials and authenticated by the system through database. |
| **Trigger:** | When user click on login app |
| **Preconditions:** | username and password must be provided by the users. |
| **Postconditions:** | User will login successfully. |
| **Normal Flow:** | Credentials will be entered by users  Credentials will be authenticated from database of the system  User will be successfully login |
| **Alternative Flows:** | If user is not already registered, then user will first sign up  User will then provide credentials  User will again be authenticated through database |
| **Exceptions:** | If user will be not authenticated, error message will be appeared. |
| **Special Requirements:** | None |
| **Assumptions:** | None |
| **Notes and Issues:** | User will have only 3 attempts for login. After 3 attempts user will have to wait for 30 seconds to retry. |

**Table 3.1: Authentication**

|  |  |
| --- | --- |
| **Use Case ID:** | ID-02 |
| **Use Case Name:** | Booking |
| **Actors:** | User, System |
| **Description:** | System will display time slots and user will book time slot. System will show payment option and after payment system will send confirmation message to the user. |
| **Trigger:** | When user want to place a booking. |
| **Preconditions:** | User should be authenticating.  User should have Wi-Fi connection. |
| **Postconditions:** | The system will book the time slot. |
| **Normal Flow:** | User will first login then request for booking System will display time slots and user will book time slot. System will show payment option and after payment system will send confirmation message to the user. |
| **Alternative Flows:** | None |
| **Exceptions:** | if user cancel the booking then payment will not be refundable |
| **Special Requirements:** | None |
| **Assumptions:** | Android app will be connected to the system. |
| **Notes and Issues:** | if user want to book the slots then user must be send request for booking. |

**Table 3.2: Booking**

|  |  |
| --- | --- |
| **Use Case ID:** | ID-03 |
| **Use Case Name:** | Lift barrier |
| **Actors:** | System |
| **Description:** | The car driver reaches the parking lot. The camera on the lot detects the number plate. System scans the number plate, verifies it through database and lift barrier. |
| **Trigger:** | When car reaches the parking lot and camera detects the number plate. |
| **Preconditions:** | Booking must be done and payment have been made. |
| **Postconditions:** | System lifts the barrier and allow the car entry. |
| **Normal Flow:** | The car driver reaches the parking lot.  The camera on the lot detects the number plate.  System scans the number plate, verifies it through database and lift barrier. |
| **Alternative Flows:** | Booking is not made.  Payment is not made.  Number plate does not match. |
| **Exceptions:** | If number plate does not match, barrier would not lift. |
| **Special Requirements:** | None |
| **Assumptions:** | None |

**Table 3.3: Lift Barrier**

# Chapter 4: Design and Architecture

In this chapter we will discuss the design and architecture of our system.

**4.1. System Architecture**

As system design varies from system to system, therefore user need to have the architecture view of the whole system.

Presentation tier

Interface

Business logic tier

Application server

**Data management tier**

Database server

**Fig 4.1: System Architecture**

**4.2. System Design**

Systems design is the process of defining elements of a system like components, modules, architecture and their interfaces and data for a system based on the specified requirements. The purpose of the System Design process is to provide sufficient detailed data and information about the system. Following is the system design of the EzParking.

**4.2.1 UML Structural Diagrams**

Following are the UML structural diagrams of our system:

**4.2.1.1 Component Diagram**



**Fig 4.2: Component Diagram**

**4.2.1.2. Package Diagram**



**Fig 4.3: Package Diagram**

* + - 1. **Deployment Diagram**

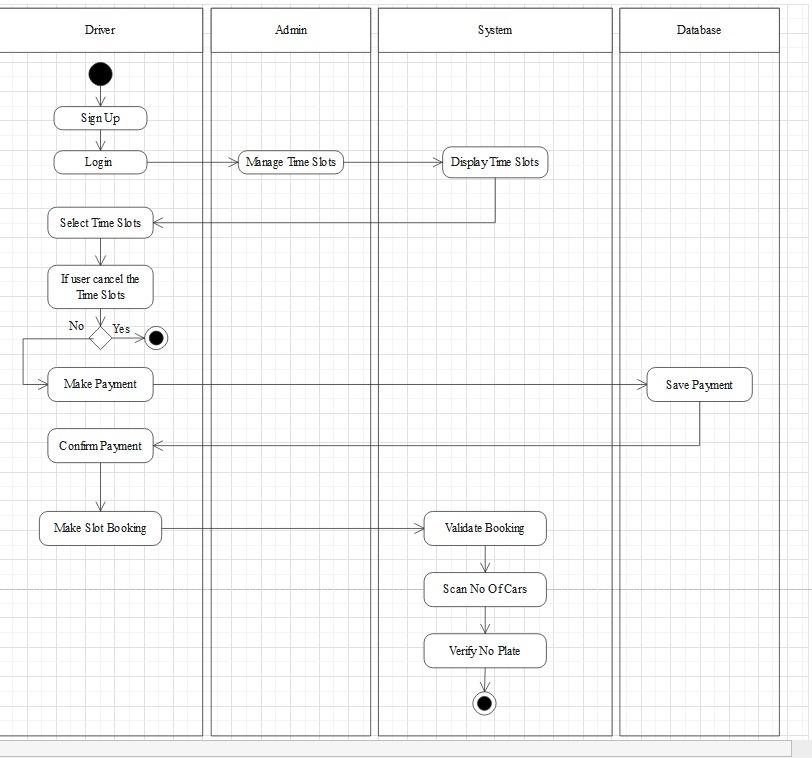


**Fig 4.4: Deployment Diagram**

**4.2.2 UML Behavioral Diagrams**

Following are the behavioral diagrams of our system:

**4.2.2.1 Activity Diagram**



**Fig 4.5: Activity diagram**

**4.2.2.2 State Machine Diagram**



**Fig 4.6: State Machine Diagram for booking**



**Fig 4.7: State Machine Diagram of manage parking lots**



**Fig 4.8: State Machine Diagram of barrier lifting**

**4.2.3 UML Interaction Diagrams**

Following are the UML interaction diagrams of our system:

**4.2.3.1 Sequence Diagrams**

A sequence diagram is a type of interaction diagram because it describes how and in what order a group of objects works together



**Fig 4.9: Sequence diagram for Signup**



**Fig 4.10: Sequence diagram for Login**



**Fig 4.11: Sequence diagram for Booking**



**Fig 4.12: Sequence diagram for Booking Cancellation**



**Fig 4.13: Sequence diagram for Barrier Opening**

**4.3 Node Structure**



**Fig 4.14: Node Structure**

**4.4 Class Diagram**

Class diagrams are the main building block in object-oriented modeling. They are used to show the different objects in a system, their attributes, their operations and the relationships among them.



**Fig 4.15: Class Diagram**

**4.5 Communication Design Protocol**

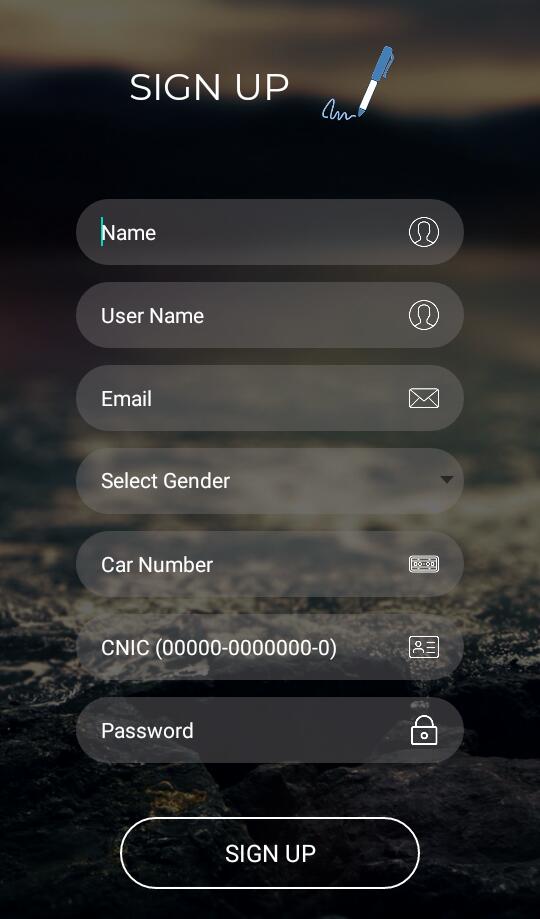


**4.6 Interfaces of EZParking App:**

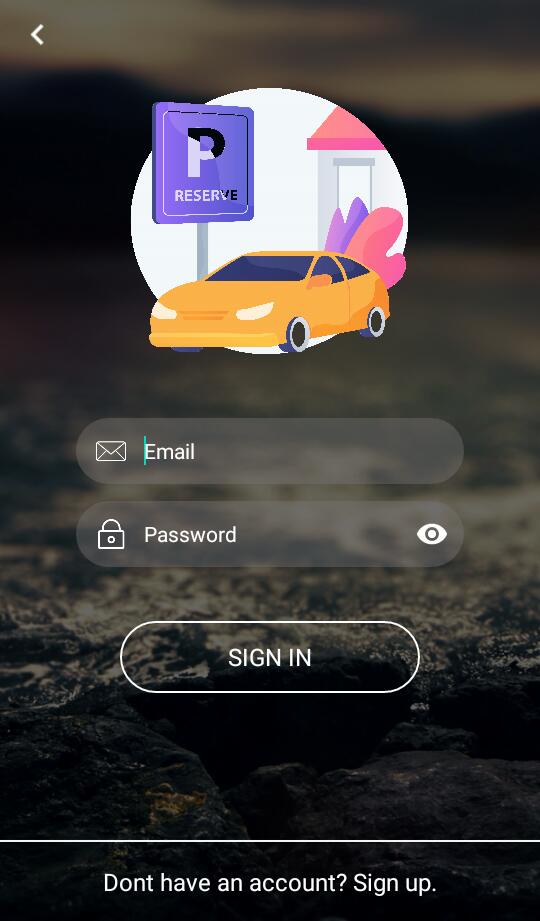
* **Splash Screen:**



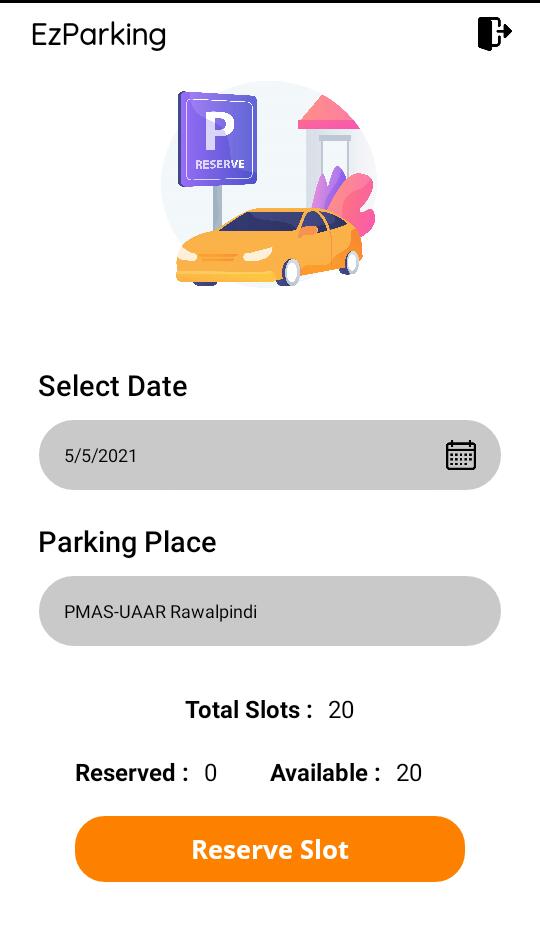
* **Signup:**



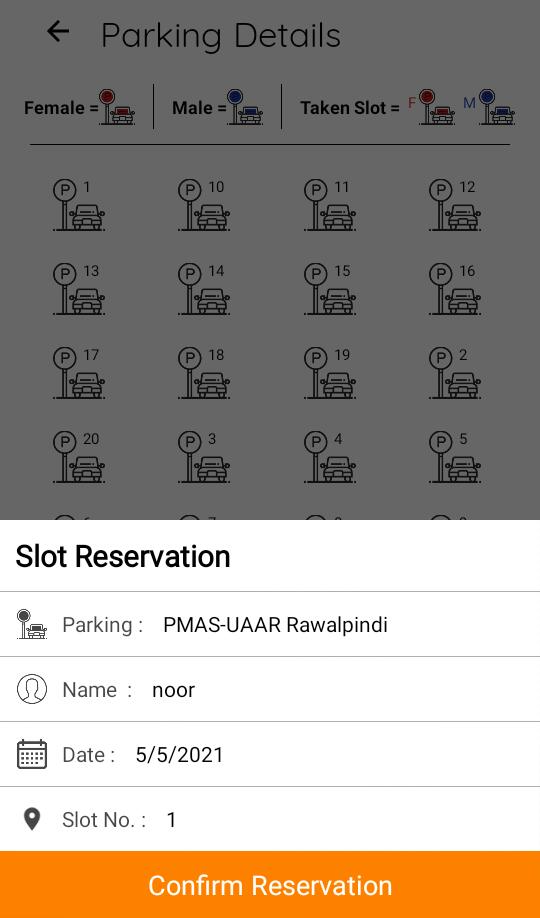
* **Login:**



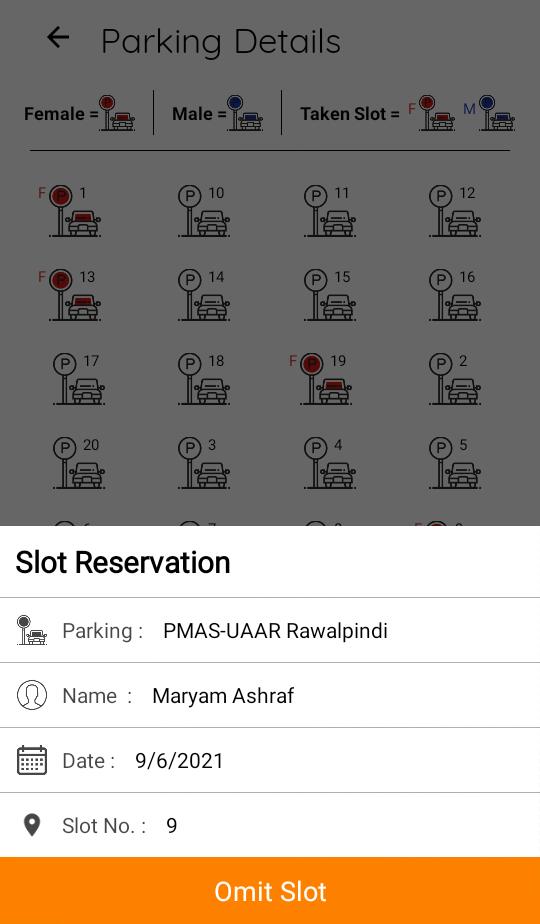
* **Reserve Seat:**



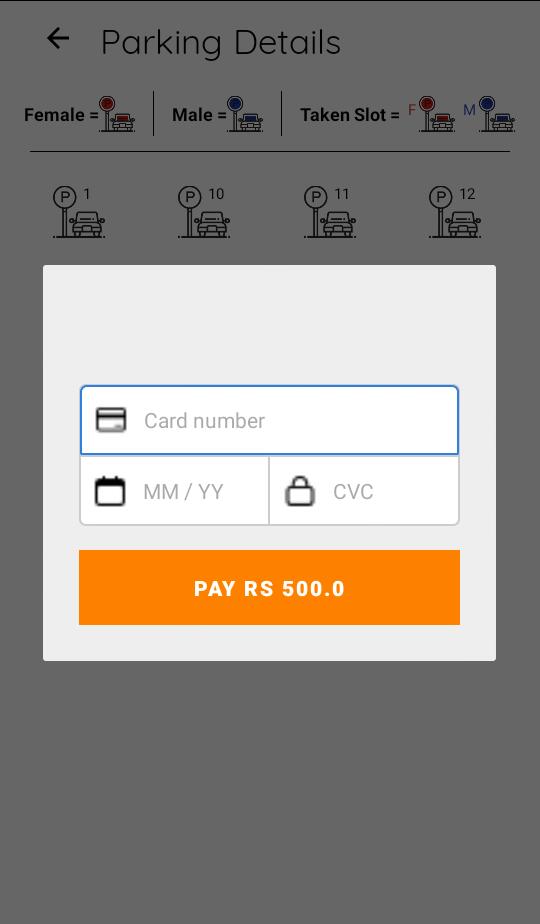
* **Confirm Reservation**



* **Cancel Slot:**



* **Payment:**



* **Hardware:**



# Chapter 5: Implementation

This chapter will discuss implementation details supported by UML diagrams (if applicable). You will not put your source code here. Any of the following sections may be included based on your project.

# Communication Protocol Implementation



**Explanation:**

This project EZ Parking System is concerned with three different technologies. We developed an android app with front end in xml and backend in java. the driver registers himself and can book any available parking lot. The driver have to make payment for booking confirmation and the driver can cancel the booking before making payment.

We also have an admin panel which is managed through our desktop application. Hardware is also involved in it. When a driver reaches booked parking lot. Its car number plate is captured by camera and the number plate is then matched with database and then allows the car to enter the parking lot by lifting the barrier.

# Component Diagram



**Explanation:**

In component diagram it has been shown that the driver registers himself and book parking lot and confirm it through payment module. The admin register himself and manage the dashboard. The camera captures the number plate, matches it through template matching and then motor lifts the barrier.

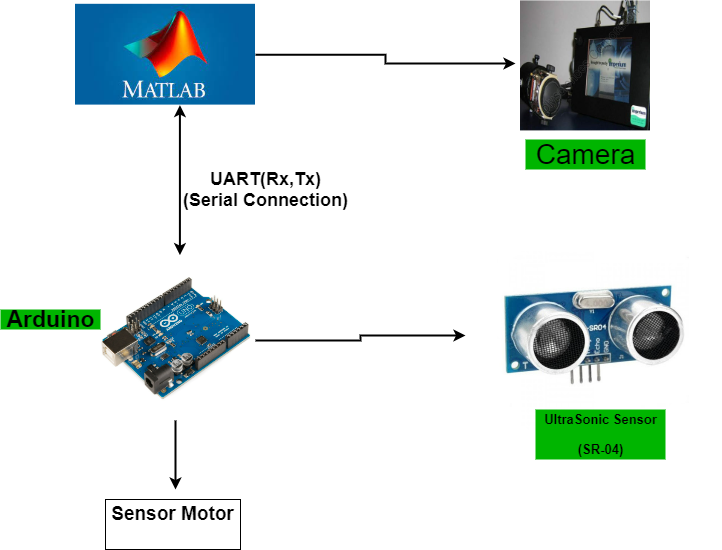
**System Component Diagram:**



**Explanation:**

We have android app for which we use firebase database. For desktop app we have used SQL Management Studio database. The car number plate is captured by camera and Arduino pass the image to MATLAB and here the image is matched from database through template matching techniques and if the image matches, the motor will lift the barrier.

**Hardware Component Diagram:**



**Explanation:**

In our hardware system we used A4tech camera and sensor SR-04. MATLAB and Arduino have a UART serial connection. The car number plate is captured by camera and Arduino pass the image to MATLAB and here the image is matched from database through template matching techniques and if the image matches, the motor will lift the barrier.

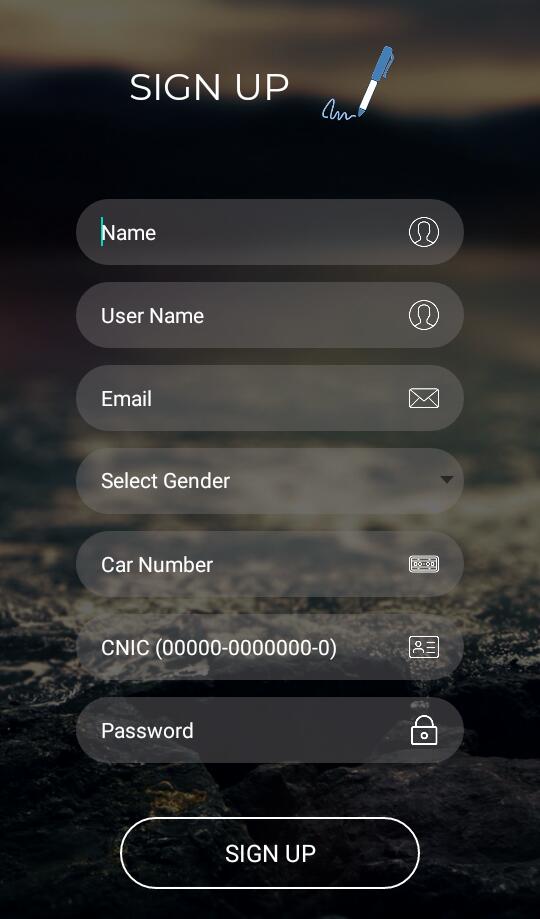
# User Interface

**Splash Screen**



This is a splash screen and it will be shown at the start of the application for 3 seconds.

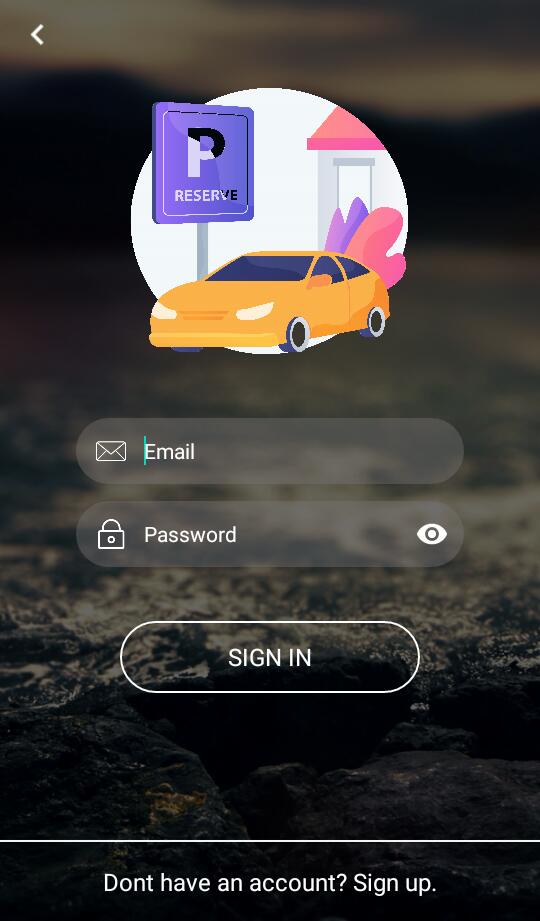
**Signup**



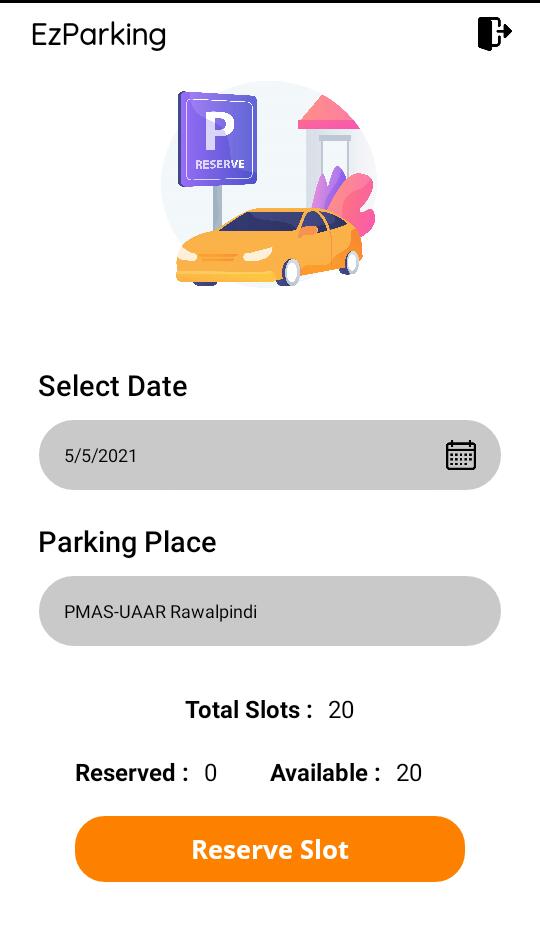
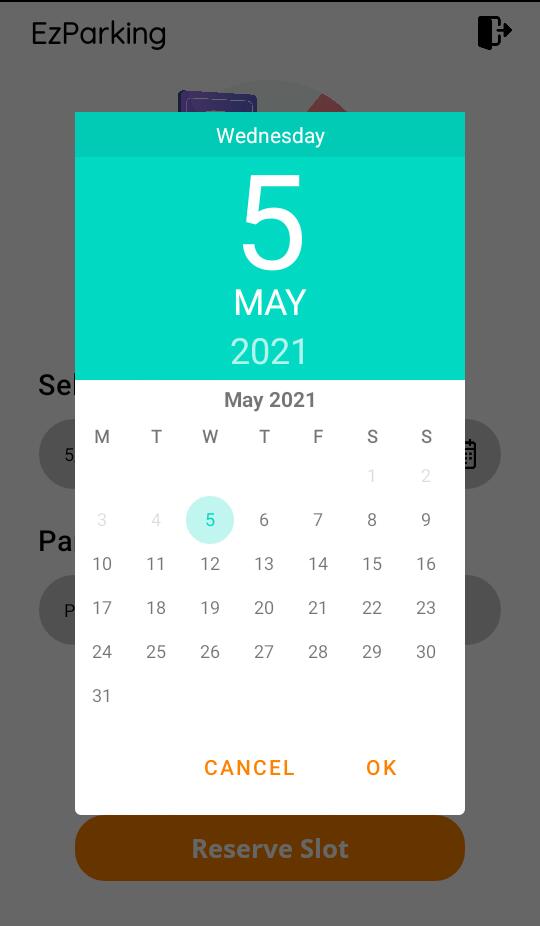
The user will enter the data for account creation. The validations applied on Signup page are given below:

1. Email should be valid.
2. Username should contain number and alphabets.
3. Name should only contain alphabets.
4. CNIC should be valid.
5. Password must be more than 5 digits.

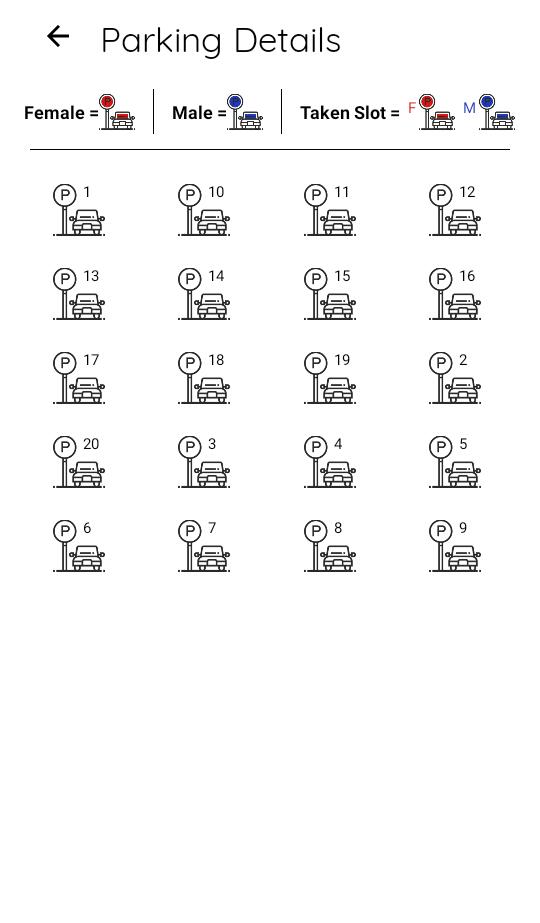
After checking the given validations and when user press “Signup” button, the user account will be created successfully.



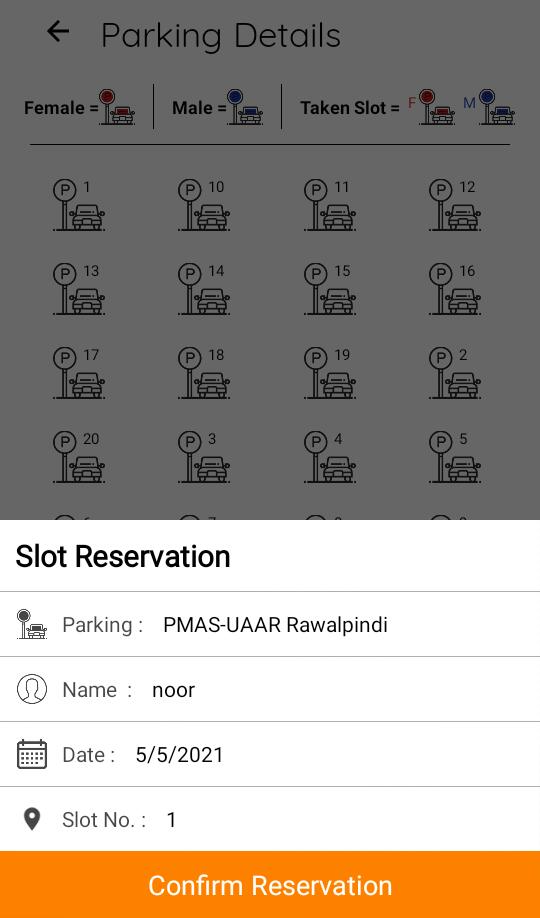
At login page the user will enter his valid email and password. If the email or password is invalid/empty the user will not be able to Log In to his/her account, and if the user does not have any account, then user should first create his/her account on clicking “Sign up” button.

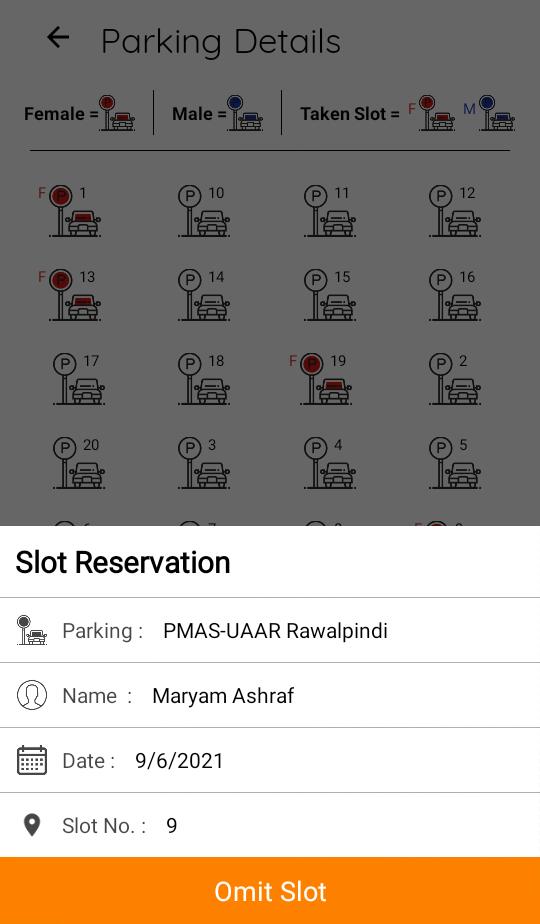
At this page the user will select the date according to his/her need. The records related to slot are given at this interface which helps the user to get know how many slots are available and reserved. The user will move to next interface by clicking the “reserve slot” button.



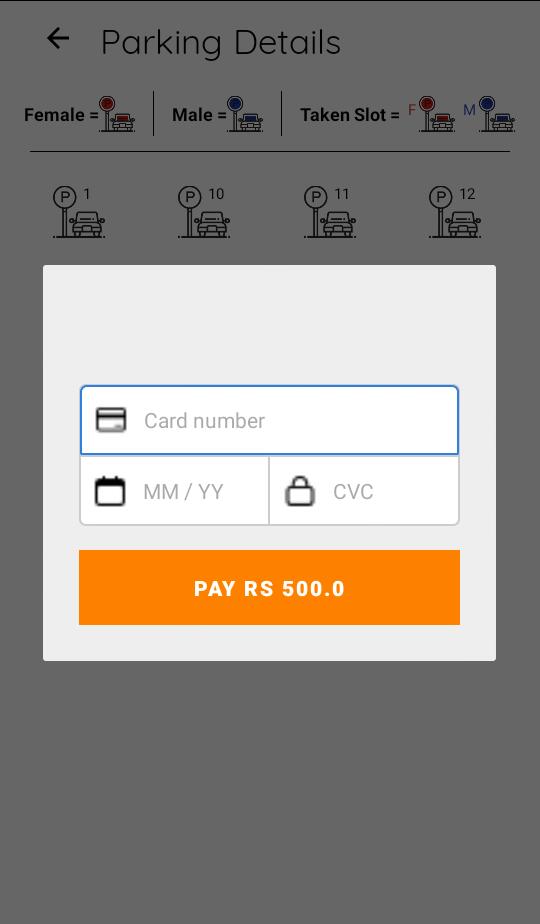
Parking details are given at this interface which helps the user to get know which slots are booked and which are not. White slot represents that slot is not booked yet. Red slot shows that the slot is reserved by female on the other hand blue slot represents that slot is booked by male. On clicking the white slot (un booked slot) a pop up appears.



This interface will display to the user at that time when user select the slot, this interface will show details of user like Name, Slot No and the date of his/her reservation. On Clicking the confirm reservation button the user will move to payment’s interface.



At this interface the user will be able to omit slot. By clicking the red/blue slot a pop up will appear from where the user will cancel the slot by clicking “omit slot” button.



At this interface the user will enter the record for payment and click the “pay” button, and the record will be saved in database.

# Chapter 6: Testing and Evaluation

**6.1. Verification:**

Verification is the process of checking that a software achieves its goal without any bugs. It is the process to ensure whether the product that is developed is right or not. It verifies whether the developed product fulfills the requirements that we have.

* + 1. **Functional Testing:**

Functional Testing is a type of black box testing whereby each part of the system is tested against functional specification/requirements.

* After giving right credentials we are able to login in our system.
* Our system gives error message when user enter incorrect card number.
* Our system successfully shows the record in database of registered users.

1. **Sign up**

|  |  |
| --- | --- |
| Tested By | Maryam Ashraf |
| Test type | Unit testing |
| Test case number | 01 |
| Test case name | Sign up |
| Test case description | This test case will check sign up form details |
| **Procedural Steps** | **Procedural Steps** |
| 1 | User will fill sign up form |
| 2 | All information must be entered correct |
| 3 | User account will be created |

**Sign up form**

1. **Login**

|  |  |
| --- | --- |
| Tested By | Maryam Ashraf |
| Test type | Unit testing |
| Test case number | 02 |
| Test case name | Login |
| Test case description | This test case will check login info |
| **Procedural Steps** | **Procedural Steps** |
| 1 | User will enter email and password |
| 2 | On clicking login button user will be logged in |
| 3 | If not able to login, then first have to create account |

**Login page**

1. **Forgot password**

|  |  |
| --- | --- |
| Tested By | Bisma Arshad |
| Test type | Unit testing |
| Test case number | 03 |
| Test case name | Forgot password |
| Test case description | This test case will check forgot password details |
| **Procedural Steps** | **Procedural Steps** |
| 1 | User will enter email and press send button |
| 2 | A dialog box appears confirming that the user wants to reset password or not |
| 3 | On selecting yes password will be reset |

**Forgot Password**

1. **Booking Status**

|  |  |
| --- | --- |
| Tested By | Bisma Arshad |
| Test type | Unit testing |
| Test case number | 04 |
| Test case name | Booking Status |
| Test case description | This test case will check Booking status details |
| **Procedural Steps** | **Procedural Steps** |
| 1 | User after login will move towards Booking Status page where he can see all the details. |
| 2 | On clicking reserve seat button app will ask user to confirm booking. |
| 3 | If user click confirm booking then popup message displayed. |

**Booking Status**

1. **Payment status**

|  |  |
| --- | --- |
| Tested By | Rimsha Masood |
| Test type | Unit testing |
| Test case number | 05 |
| Test case name | Payment status |
| Test case description | This test case will check Payment status details |
| **Procedural Steps** | **Procedural Steps** |
| 1 | After confirm booking, system ask user to make payment. |
| 2 | If user wants to cancel booking then before making payment user will cancel his/her booking. |
| 3 | If user cancel booking then system will go to page reserve seat. |
| 4 | If user make payment amount then system confirm his/her booking and display popup message. |

**Payment status**

1. **Admin**

|  |  |
| --- | --- |
| Tested By | Rimsha Masood |
| Test type | Unit testing |
| Test case number | 05 |
| Test case name | Admin |
| Test case description | This test case will check admin details |
| **Procedural Steps** | **Procedural Steps** |
| 1 | Admin can check and search user details |
| 2 | Details can be searched by name and email |
|  |  |

**Admin page**

* + 1. **Static testing:**

Static test techniques provide a great way to improve the quality and productivity of software development. It includes the reviews and provides the overview of how they are conducted.

**6.2. Validation:**

Validation is the process of checking whether the software product is up to the mark or in other words product has high level requirements. It is the process of checking the validation of product i.e., it checks what we are developing is the right product. it is validation of actual and expected product. Validation involves white box testing is software testing technique in which internal structure, design and coding of software are tested to verify flow of input-output and to improve design, usability and security. E.g.; Payment and Booking module.

**6.3. Usability Testing:**

**Usability Testing** also known as User Experience (UX) Testing, is a testing method for measuring how easy and user-friendly a software application is. A small set of target end-users, use software application to expose usability defects. Usability testing mainly focuses on user's ease of using application, flexibility of application to handle controls and ability of application to meet its objectives. In our project some users test our system and they all are successfully registered. One user finds the defect in our payment module that you should add some other ways to enter amount.

* 1. **Module/Unit Testing:**

Unittesting is a confusing part of the software development process. Unit testinginvolves individually testing unit of code separately to make sure that it works on its own, independent of the other units.  Unit testing is essentially a set of paths, test performed to examine the several different paths through the modules. Unit testing is remarkably done by programmers with the help of Unit framework. Unittesting is usually an automated process and performed within the programmers IDE. Unittesting is an action used to validate that separate units of source code remains working properly. In our project we test the object detection, payment and booking modules individually for better results.

* 1. **Integration Testing:**

**Integration Testing** is defined as a type of testing where software modules are integrated logically and tested as a group. A typical software project consists of multiple software modules, coded by different programmers. The purpose of this level of testing is to expose defects in the interaction between these software modules when they are integrated. In our project, we integrate Payment and Booking modules and they show results successfully.

* 1. **System Testing:**

SystemTesting is carried out on the whole system in the context of either system requirement specifications or functional requirement specifications or in the context of both. System testing tests the design and behavior of the system and also the expectations of the customer. It is performed to test the system beyond the bounds mentioned in the [software requirements specification (SRS)](https://www.geeksforgeeks.org/software-engineering-quality-characteristics-of-a-good-srs/). In our project we give different inputs as user required from our system initially and get ninety percent accuracy in it.

* 1. **Acceptance testing:**

Acceptance Testing is a method of software testing where a system is tested for acceptability. The major aim of this test is to evaluate the compliance of the system with the business requirements and assess whether it is acceptable for delivery or not. Acceptance Testing is the last phase of software testing performed after System Testing and before making the system available for actual use. In our system user check all details, initially face some defects but after negotiating and resolving some defects they accept our system.

* 1. **Stress Testing:**

**Stress Testing** is a type of software testing that verifies stability & reliability of software application. The goal of Stress testing is measuring software on its robustness and error handling capabilities under extremely heavy load conditions and ensuring that software doesn't crash under crunch situations. It even tests beyond normal operating points and evaluates how software works under extreme conditions.

* 1. **Hardware Configuration Testing:**

**Configuration Testing** is a testing technique in which the software application is tested with multiple combinations of hardware in order to evaluate the functional requirements and find out optimal configurations under which the software application works without any defects or flaws. We connect Camera with our system and run it successfully.

* 1. **Evaluation:**

Evaluation was done as we completed our project by correctly evaluated things and completed them on time.

* 1. **Deployment:**

Our project deploys in real time environment and it works accurate. There is proper exchange of information between system and application, hence deployment was also done.

* 1. **Maintenance:**

Maintenance was done as we fulfilled all the objectives of the project and maintained all the detection and correction of errors in the system.

# Chapter 7: Conclusion and Future Work

This chapter concludes the project and highlights future work.

# Conclusion

This project EZ Parking system is an automated parking system which has two main users. Driver and Admin.

Android application has registration, booking and payment module.

Desktop application has registration and dashboard management module.

In our hardware system we used A4tech camera, sensor SR-04, MATLAB and Arduino software for number plate recognition and template matching and motor for barrier lifting.

# Future Work

* We are planning to do following advancements in our project in future.
* Addition of parking time extension module
* Facilitating the user by providing booking cancellation option after payment as well and refunding of payment.
* Extension of the dedicated parking system to the parking’s of all the cities of all over the country.
* Improvements in the designing and performance of hardware.

# References

1. <http://ijcsit.com/docs/Volume%205/vol5issue03/ijcsit2014050370.pdf>
2. <https://uxdesign.cc/ui-ux-case-study-parkingapp-using-a-app-to-find-parking-and-pay-cd7b576b9bd6?gi=feea63ec0e09>
3. <https://res.mdpi.com/d_attachment/sustainability/sustainability-13-06031/article_deploy/sustainability-13-06031.pdf>
4. <https://ieeexplore.ieee.org/document/8743906>
5. <http://www.cse.msu.edu/~cse870/IEEEXplore-SRS-template.pdf>
6. <https://www.scaledagileframework.com/nonfunctional-requirements/>