PARKING RESERVATION SYSTEM

BY

Saifullah-Uz-Zaman

Supervised By

Dr Muddassar Azam Sindhu

Department of Computer Science,

Quaid-I-Azam University, Islamabad, Pakistan.

Session (2015-2020)

Table of Contents

List of Figures

List of Tables

Chapter 1 Software Project Management Plan

1.1 Introduction

This introduction provides background information for the rest of the document. It briefly describes the project, the client deliverables, the project milestones and expected document changes.

1.2 Project Overview

The Parking Reservation System is a web service as well as android based application that will help the Students, Faculty and visitors of Quaid-i-Azam University to reserve parking space before their arrival.

1.3 Project Deliverables

There are three deliverables for this project

- SPMP and SRS of the project.
- Complete documentation of the project.
- Implementation of the project with complete documentation by end of next semester (Spring 2019).

1.4 PROJECT ORGANIZATION

Project organization consists of software process model, roles and responsibilities and tools and techniques.

1.4.1 Software Process Model

I'll use Incremental Model as a software process model for this project due to following reasons:

- This model is flexible, less costly to change scope and requirements.
- It is easier to test and debug during a smaller iteration.
- Easier to manage risks.
- It is easy to check the progress after each iteration.
- Generate working software earlier during software lifecycle.

1.4.2 Roles and Responsibilities

I'll be doing everything related to this project as it is a single person's project.

1.4.3 Tools and Techniques

Following are the tools used in this project

- MS Word
- Project Libre
- ARGO UML
- Android Studio
- Sublime Text
- Google Map API
- QAU CMS API

1.5 PROJECT MANAGEMENT PLAN

		Name	Duration	Start	Finish	Predecessors	Resource Names
1		**Create Parking Reservation System**	113 days	**11/12/18 8:00 AM**	**4/17/19 5:00 PM**		
2		Problem understanding	1 day	11/12/18 8:00 AM	11/12/18 5:00 PM		
3		**Make Software Project Management Plan**	4 days	**11/13/18 8:00 AM**	**11/16/18 5:00 PM**	**2**	**Saif;PC;MS Word**
4		Write Introduction	1 day	11/13/18 8:00 AM	11/13/18 5:00 PM		
5		Define Project Organization	2 days	11/13/18 8:00 AM	11/14/18 5:00 PM		
6		Define Project Management Plan	2 days	11/15/18 8:00 AM	11/16/18 5:00 PM	5	Project Libre
7		**Make Requirements document**	58 days	**11/19/18 8:00 AM**	**2/6/19 5:00 PM**	**6**	**Saif;PC;MS Word**
8		**Make Software Requirement Specification Document**	25 days	**11/19/18 8:00 AM**	**12/21/18 5:00 PM**		
9		Give Introduction and Overview	1 day	11/19/18 8:00 AM	11/19/18 5:00 PM		
10		Define Scope	1 day	11/19/18 8:00 AM	11/19/18 5:00 PM		
11		Define Purpose and objective	1 day	11/19/18 8:00 AM	11/19/18 5:00 PM		
12		Review and refine scope and plan	2 days	11/20/18 8:00 AM	11/21/18 5:00 PM	11	Project Supervisor
13		Identify Specific Requirements	1 day	11/22/18 8:00 AM	11/22/18 5:00 PM	12	
14		Explain External Interfaces	1 day	11/23/18 8:00 AM	11/23/18 5:00 PM	13	
15		Identify Use Cases	2 days	11/26/18 8:00 AM	11/27/18 5:00 PM	14	
16		Make UseCase Diagram	1 day	11/28/18 8:00 AM	11/28/18 5:00 PM	15	ArgoUML
17		Review and Refine UC Diagram	2 days	11/29/18 8:00 AM	11/30/18 5:00 PM	16	Project Supervisor
18		Define UseCase descriptions	2 days	11/29/18 8:00 AM	11/30/18 5:00 PM	16	
19		Review and Refine UC Description	2 days	12/3/18 8:00 AM	12/4/18 5:00 PM	18	Project Supervisor
20		Define System Attributes	2 days	12/3/18 8:00 AM	12/4/18 5:00 PM	18	
21		Identify DataBase Requirements	1 day	12/5/18 8:00 AM	12/5/18 5:00 PM	20	
22		Identify Entities	2 days	12/6/18 8:00 AM	12/7/18 5:00 PM	21	Saif
23		Make Entity Relation Diagram	2 days	12/10/18 8:00 AM	12/11/18 5:00 PM	22	Saif;PC;ArgoUML
24		Review and Refine ERD	2 days	12/12/18 8:00 AM	12/13/18 5:00 PM	23	Project Supervisor
25		Make System Sequence Diagrams	2 days	12/14/18 8:00 AM	12/17/18 5:00 PM	24	ArgoUML
26		Review and refine SSD	2 days	12/18/18 8:00 AM	12/19/18 5:00 PM	25	Project Supervisor

Figure 1.1 Project Plan (a)

		Name	Duration	Start	Finish	Predecessors	Resource Names
27		Make Domain Model	2 days	12/18/18 8:00 AM	12/19/18 5:00 PM	25	ArgoUML
28		Review and Refine SRS	2 days	12/20/18 8:00 AM	12/21/18 5:00 PM	27	Project Supervisor
29		**Make Software Design Description Document**	**16 days**	**12/24/18 8:00 AM**	**1/14/19 5:00 PM**	**28**	**Saif;PC;MS Word**
30		Give Introduction and Overview	1 day	12/24/18 8:00 AM	12/24/18 5:00 PM		
31		Make Activity Diagrams	4 days	12/24/18 8:00 AM	12/27/18 5:00 PM		ArgoUML
32		Review and Refine Activity Diagram	2 days	12/28/18 8:00 AM	12/31/18 5:00 PM	31	Project Supervisor
33		Make System Architectural Design	2 days	12/24/18 8:00 AM	12/25/18 5:00 PM		ArgoUML
34		Review and Refine Architecture Diagram	2 days	12/26/18 8:00 AM	12/27/18 5:00 PM	33	Project Supervisor
35		Make Sequence Diagrams	2 days	12/28/18 8:00 AM	12/31/18 5:00 PM	33;34	ArgoUML
36		Review and Refine SD	2 days	1/1/19 8:00 AM	1/2/19 5:00 PM	35	Project Supervisor
37		Identify Classes	2 days	1/3/19 8:00 AM	1/4/19 5:00 PM	36	
38		Make Class Diagram	2 days	1/7/19 8:00 AM	1/8/19 5:00 PM	37	ArgoUML
39		Review and Refine Class Diagram	2 days	1/9/19 8:00 AM	1/10/19 5:00 PM	38	Project Supervisor
40		Review and Refine Software Design Description	2 days	1/11/19 8:00 AM	1/14/19 5:00 PM	39	Project Supervisor
41		**Make Software Test Document**	**6 days**	**1/15/19 8:00 AM**	**1/22/19 5:00 PM**	**40**	
42		Make Test Cases	4 days	1/15/19 8:00 AM	1/18/19 5:00 PM		MS Word
43		Review and Refine Test Document	2 days	1/21/19 8:00 AM	1/22/19 5:00 PM	42	Project Supervisor
44		Review Analysis and Design Document	10 days	1/23/19 8:00 AM	2/5/19 5:00 PM	43	Project Supervisor
45		Provide 1st Deliverable	1 day	2/6/19 8:00 AM	2/6/19 5:00 PM	44	
46		**Project Implementation**	**50 days**	**2/7/19 8:00 AM**	**4/17/19 5:00 PM**	**45**	**Android Studio;Saif;Proj...**
47		Create user Interfaces(Front End)	10 days	2/7/19 8:00 AM	2/20/19 5:00 PM		Saif;Android Studio
48		Create Database	40 days	2/7/19 8:00 AM	4/3/19 5:00 PM		Saif;Android Studio
49		Create Back End	30 days	2/21/19 8:00 AM	4/3/19 5:00 PM	47	Saif;Android Studio
50		Make Database Connectivity	10 days	4/4/19 8:00 AM	4/17/19 5:00 PM	48;49	Saif;Android Studio
51		Provide 2nd Deliverable	1 day	11/12/18 8:00 AM	11/12/18 5:00 PM		

Figure 1.2 Project Plan (b)

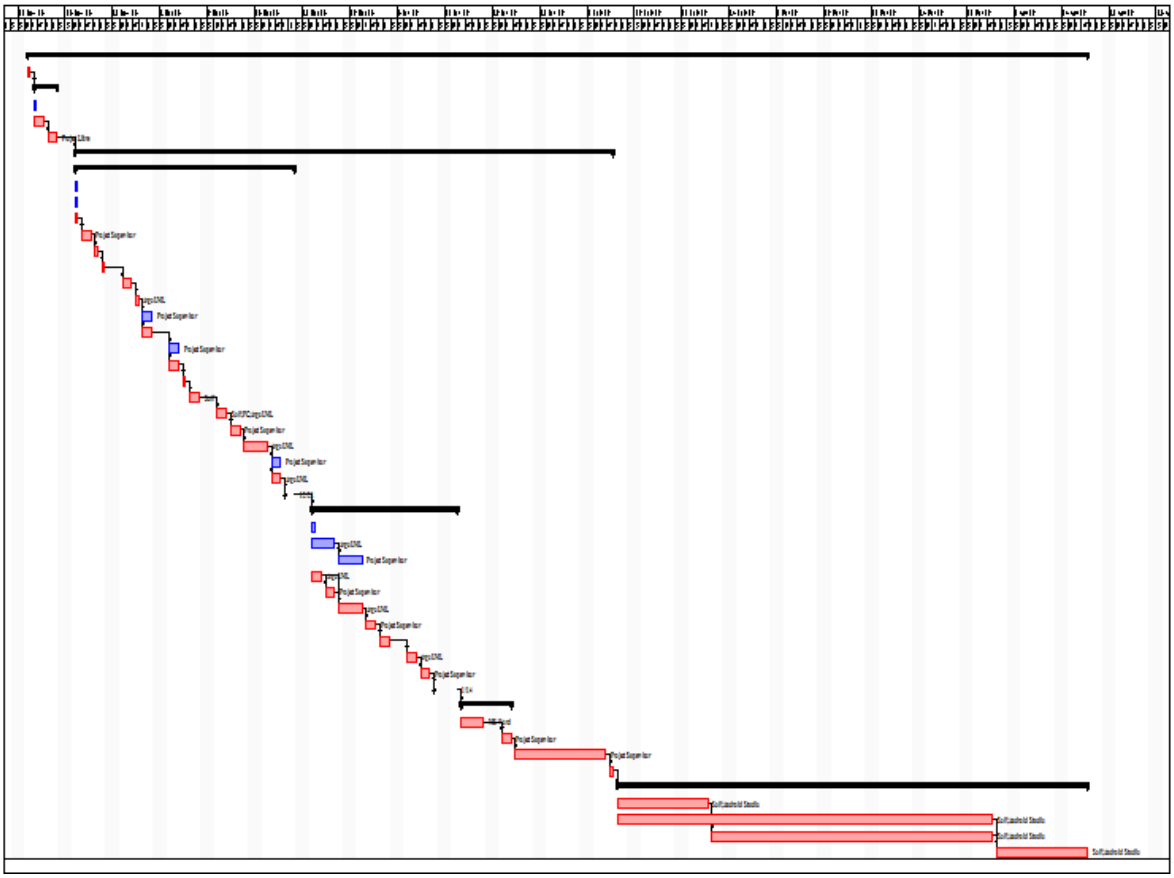

Figure 1.3 Gantt Chart

Chapter 2 Software Requirements Specification

2.1 Introduction

This chapter covers the software requirement specification of Android based Application. World has seen a tremendous increase in Android users. To provide leisure to the Android users there are almost 2.2 million apps available on Google Play Store [2]. Day-by-day, Android Application world is expanding as more and more people are opting for Android platform and Using numerous applications in their devices.

2.2 Project Overview

Quaid-e-Azam University had faced some challenges regarding the parking of vehicles in the university restricted area and physical struggle while parking by Teachers, Students and visitors, wastage of time, congestion and collision. The Parking Reservation System (PRS) is a system that enables Teachers, Students and visitors to reserve a parking space. It also allows the Teachers, Students and visitors to view the parking status at Quaid-e-Azam University. It was developed because the congestion and collision of the vehicle. The parking is booked for a specific period of time. And if the person did not reach the particular parking place in time than it got reserved by someone else. Specific locations are reserved for faculty members. Therefore, the project aimed at solving such problems by designing a web based and Android system that will enable the Teachers, Students and visitors to make a reservation of available parking space at Quaid-e-Azam University.

2.3 Purpose

In the modern society, there is increasing number of vehicles. This create problems and increase difficulty to find open spaces in QAU, as well as the increasing need to devote larger areas of land for additional parking spaces.

The three main problems that the increasing number of vehicles and the decreasing efficiency of QAU parking lots are

- Valuable time wasted from inconvenient and inefficient parking lots.
- More fuel consumed while idling or driving around parking lots, leading to more CO_2 emissions being produced.
- Potential accidents caused by abundance of moving vehicles in disorganized parking lots.

2.4 Scope

This system majorly solved the congestion, collision and save time during parking activities. The system will use data of faculty and Students to provide the facility. There are the following purpose and objectives to make online parking system app.

- Providing a simple web service and android application for parking vehicles.
- Booking for parking slot at home.
- Can search nearby places using the given locations in the app.
- Make easy to automate parking owner and customer.
- To save user time in search of parking space available in such a long parking area.
- To provide a graphical view of the parking spaces.
- Students and Faculty member login through their CMS, while visitors give some personal information (like name, phone no, car no, car name etc) during Signup.
- While booking a parking place the user must mention the time span of parking.
- The user get notification fifteen minutes before the parking time ends.
- The user can also exceed the time limit of booked parking.
- User can also cancel the booked parking.

2.5 Specific Requirements

2.5.1 External Interface Requirement:

2.5.1.1 User Interfaces:

Initially the user is presented with login page. If the user is already logged in, they'll be taken to the homepage of the system where they will be able to perform actions available to them.

2.5.1.2 Hardware Interfaces:

Computer/Laptop/Mobile.

2.5.1.3 Software Interfaces:

Web Browser, Mobile Application.

2.5.1.4 Communication Protocols

As the application is standalone so no communication protocol is required. But for downloading from Android Play Store, HTTPS protocol is required. Other than downloading from Google play store no other communication protocol is required.

2.6 Software Product Features

The product is a web application. It'll have login facility only for students and teachers. Users of the system will be Faculty, students and visitors. Multiple users will be able to use the system at the same time. The system will be backed by a database which will be used to store relevant data.

2.6.1 Product Functions

- Manage parking slot requests.
- Manage maintenance complaints and feedback.
- Manage Applicants list.
- Manage vehicles data.

2.7 Use Case Diagram

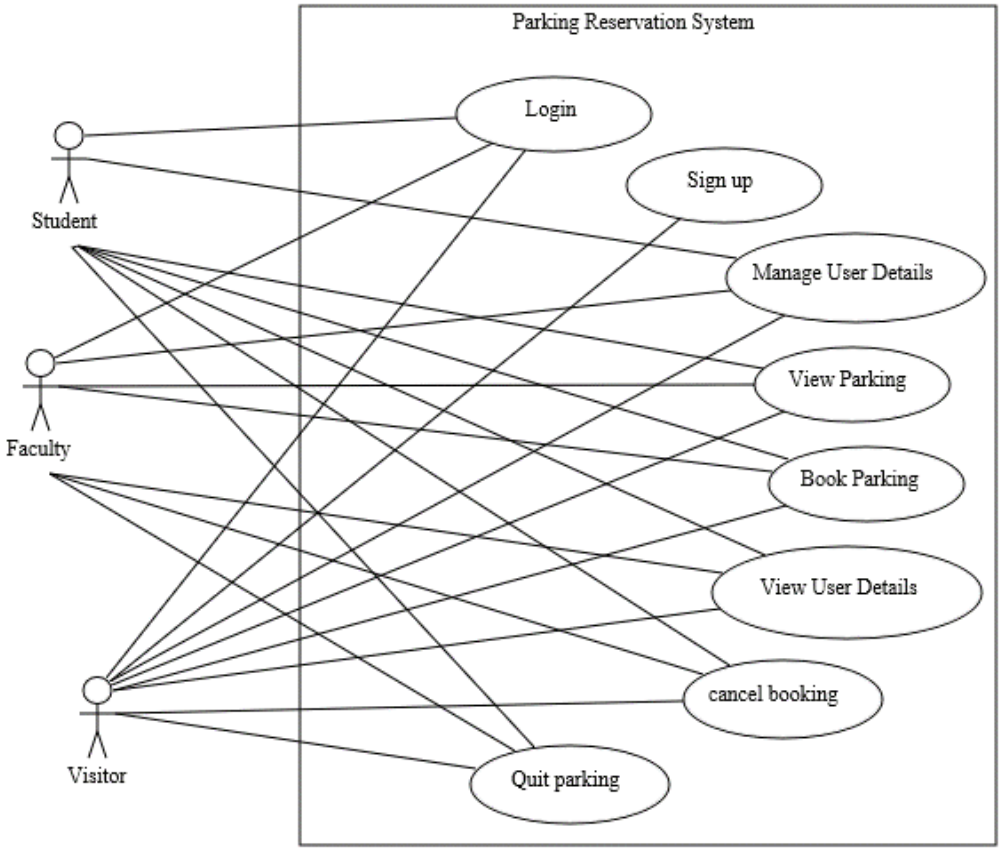

Figure 2.1 Use Case Diagram

2.8 Use Cases Description

UC-1: Login

Table 2.1 Login

ID	UC1
Name	Login
Primary Actor	Student, Faculty
Pre-Condition	User executed the application.
Post-Condition	User Login Successfully.
Main Success Scenario	1. User open the PRS App from the mobile. 2. User Selects the option Student or Faculty. 3. User fill up login details (username, password). 4. User clicks on login button.
Alternative flows or Extensions	1a. if a user enters invalid user name. 1. System asks user to enter user name again. 2. If user enters the invalid password. 2a. System signals invalid password and ask to re-enter password.
Frequency	Could be nearly continuous.

UC-2: View Parking

Table 2.2 View Parking

ID	UC2
Name	View Parking
Primary Actor	Student, Faculty, Visitor
Pre-Condition	User login Successfully.
Post-Condition	User see all the parking slots in the QAU.
Main Success Scenario	1. User see the whole parking areas in the QAU. 2. User selects the parking area in which he/she park his vehicle.
Alternative flows or Extensions	1a. User does not see the parking slots. 1b. Google API down.
Frequency	Could be nearly continuous.

UC-3: Book Parking

Table 2.3 Book Parking

ID	UC3
Name	Book Parking
Primary Actor	Student, Faculty, Visitor
Pre-Condition	User login Successfully.
Post-Condition	User booked the parking slot in the parking area.
Main Success Scenario	1. User sees the parking slots in the parking area. 2. User selects the parking slot available. 3. User enter time span for parking.
Alternative flows or Extensions	1a. User selects the incorrect location.
Frequency	Could be nearly continuous.

UC-4: View User Details

Table 2.4 View User Details

ID	UC4
Name	View User Details
Primary Actor	Student, Faculty, Visitor
Pre-Condition	User login Successfully.
Post-Condition	User Modified the booking details.
Main Success Scenario	1. User selects the View User Detail button. 2. User Details Viewed on screen.
Alternative flows or Extensions	1a. User selects incorrect button.
Frequency	Could be nearly continuous. (depend on the user)

UC-5: SIGN-UP

Table 2.5 Sign-up

ID	UC5
Name	Sign-up
Primary Actor	Visitor
Pre-Condition	User executed the application.
Post-Condition	User Login Successfully.
Main Success Scenario	1. User open the PRS App from the mobile. 2. User Selects the option signup option. 3. User fill up signup details. 4. User clicks on signup option.
Alternative flows or Extensions	1a. User doesn't install the PRS App. 2a. User enters invalid signup details.
Frequency	Could be nearly continuous.

UC-6: Cancel Booking

Table 2.6 Cancel Booking

ID	UC6
Name	Cancel Booking
Primary Actor	Student, Faculty, Visitor
Pre-Condition	Booking created.
Post-Condition	User successfully cancel the booking.
Main Success Scenario	1. User open the PRS App from the mobile. 2. User selects the Cancel booking option.
Alternative flows or Extensions	1a. User selects incorrect button.
Frequency	Could be nearly continuous. (depend on the user)

UC-7: Cancel Parking

Table 2.7 Cancel Parking

ID	UC7
Name	Cancel Parking
Primary Actor	Student, Faculty, Visitor
Pre-Condition	User parked car Successfully.
Post-Condition	User successfully cancel the parking.
Main Success Scenario	1. User open the PRS App from the mobile. 2. User selects the Cancel Parking option.
Alternative flows or Extensions	1a. User selects incorrect button.
Frequency	Could be nearly continuous. (depend on the user)

UC-8: MANAGE USER DETAILS

Table 2.8 Manage User Detail

ID	UC8
Name	Manage User Details
Primary Actor	Student, Faculty, Visitor
Pre-Condition	User login Successfully.
Post-Condition	User Modified the User details.
Main Success Scenario	1. User selects the manage user detail option. 2. User updated the required information (name, phone no, car name, car no). 3. User selects the submit option.
Alternative flows or Extensions	1a. User selects incorrect button. 2a. User enters invalid details.
Frequency	Could be nearly continuous. (depend on the user)

2.9 **Domain Model**

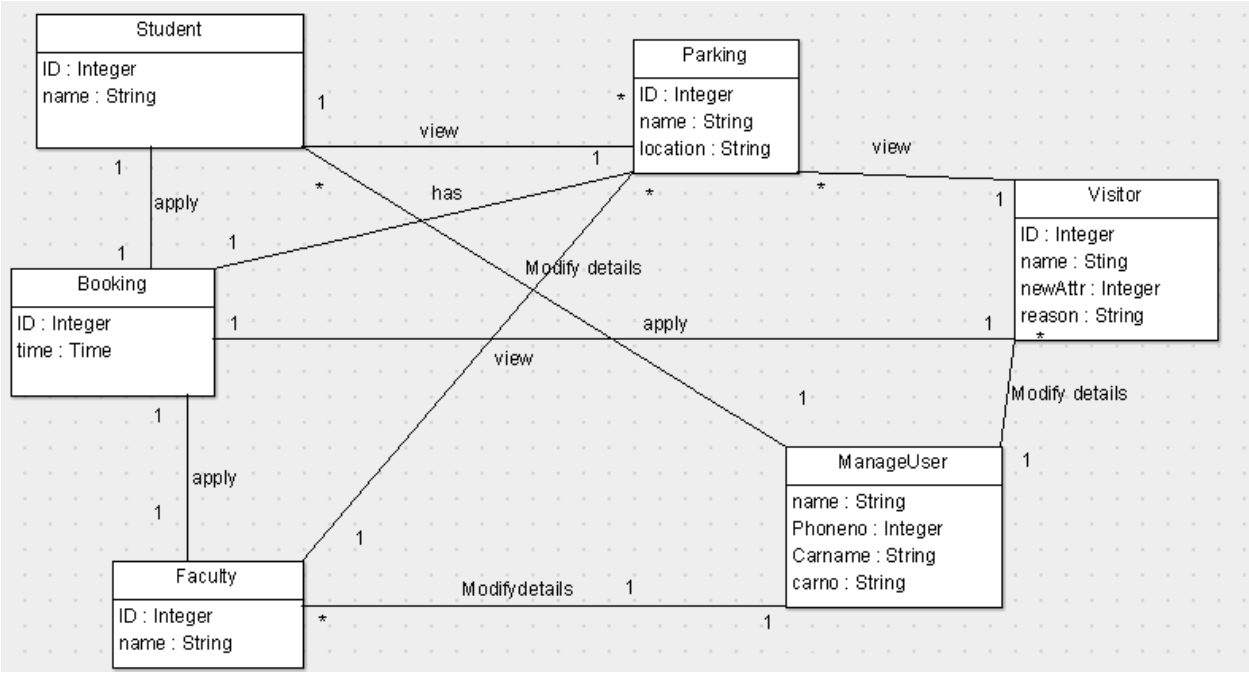

Figure 2.2 Domain model

2.10 **Software System Attributes**

Software system attributes define overall factors that affect run-time behavior, and user experience. To develop high quality application, software system attributes are the benchmarks that describe system's intended behavior within the environment for which it was built. Here is detail of some software system attributes.

2.10.1 **Reliability**

The system should never crash or hang, other than as the result of an operating system error. On the scale of software reliability, Parking Reservation System is 95% reliable.

2.10.2 **Availability**

Once the application is downloaded and installed, application will not take more than 10 seconds to reach the main menu upon launching.

2.10.3 **Security**

The system will have single interfaces for separate user access levels. They will once login to the system where they will have the authority to reserve the parking slot for their vehicle. The system

is dependent on data provided by the University. System will utilize data such as staff data and student data (id, rank, etc).

2.10.4 Maintainability

All program files shall include comments concerning authorship and date of last change. The code should be modular, to permit future modifications. Anticipated updates include changes to the sets of objects and their descriptions used during the application. These should be stored in a separate data file, rather than embedded in the program code.

2.10.5 Portability

The system should be portable to any other android system. No other specific portability requirements have been identified.

2.10.6 Fault Tolerance

The system would be able to respond adequately in case of a fault. A database backup mechanism will be in place to ensure that most of data remains intact in case of a database failure. The system should make sure no more than 2% of data is lost in case of database failure. The system will display error prompts on occurrence of any error.

2.10.7 Performance

The system should have a quick response time. A page load should not take more than 15 seconds. The system should also be able to handle multiple concurrent users. The maximum number of users at a single time could be teachers, students or the visitors and the system should be capable of handling the load.

2.11 Database Requirements

Firebase is a mobile and web app development platform that provides developers the facility to create real time database.

The firebase Real time database is a cloud-hosted NoSQL database that lets you store and sync between your users in real time.

WHY No SQL?

NoSQL is particularly useful for storing unstructured data, which is growing far more rapidly than structured data and does not fit the relational schemas of RDBMS.

There are four categories of NoSQL database:

- Key-value data stores
- Document stores
- Wide-column stores
- Graph stores

Firebase Realtime Database is of Document datatype and is really just one big JSON object that the developers can manage in real time.

2.12 **Additional Requirements**

There are additional requirements needed.

Chapter 3 Software Design Description

3.1 Introduction

The Software Design Document provides documentation which will be used to aid in software development by providing the details for how the software should be built. The SDD document contains the interface design, architecture diagram, sequence diagrams and class diagram. Within the Software Design Document narrative and graphical documentation of the software design for the project include use case models, sequence diagrams, collaboration models, object behavior models, and other supporting requirement information. It includes the description of how the software will meet the requirements.

3.1.1 Purpose

The purpose of the Software Design Document is to provide a description of the design of Parking Reservation System to allow for software development to proceed with an understanding of what is to be built and how it is expected to build. The Software Design Document provides information necessary to provide description of the details for the software and system to be built.

3.1.2 Design Overview

The software design document provides design details of the Parking Reservation System. The document contains a complete low-level description of the system, providing insight of the structure and design of each component. A system architecture is a conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system.

3.1.3 Requirements Traceability Matrix

Table 3.1 traceability matrix

Requirement ID	Requirement Name	Sequence Diagram	Interface
UC1	Login	Yes	Yes
UC2	Manage user details	yes	yes
UC3	View Parking	Yes	Yes
UC4	Book Parking	Yes	Yes
UC5	View User Details	Yes	No
UC6	Cancel Booking	Yes	Yes
UC7	Quit Parking	Yes	Yes

3.2 SYSTEM ARCHITECTURAL DESIGN

System Architecture Diagram is used to represent the components of system and interaction between them. Architectural design is the resolution of the requirements in the design of the software, the hardware and networking, operations, policies, and so forth.

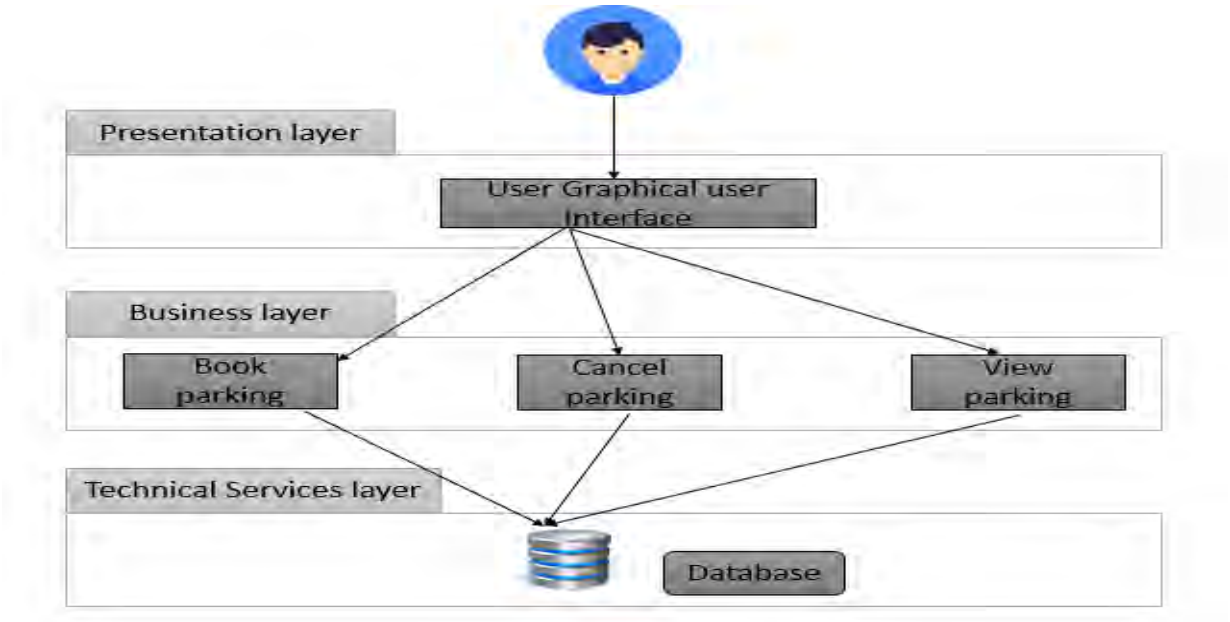

Figure 3.1 Architecture diagram

3.2.1 Chosen System Architecture

In this section, we describe chosen architecture of the system. The basic architecture of this system is 3-tier model. A 3-tier model uses the client/server computing model. It has three layers.

- Presentation Layer
- Application Layer
- Data Layer

3.2.2 Discussion of Alternative Designs

We are going to use the 3-Tier architecture instead of 2-Tier architecture which isolates the Application logic from the Data access tier. But for our application the security is more important than time. Therefore, in our scenario 3-tier architecture will be better.

3.2.3 System Interface Description

System interface describes the flow of resources. It is the logical characteristics of each interface between the software product and the hardware components of the system.

3.3 USER INTERFACE DESIGN

User interface is the logical characteristics of each interface between the software product and its users. In this section user interface of Parking Reservation System is discussed.

3.3.1 Description of the User Interface

In Parking Reservation System user can interact with the system by using touch screen interface of Android device. When user clicks on the system icon a splash screen will appear for 5 seconds and then main menu will appear. Main menu has two buttons Login or Signup. When user click on the Login or signup button, all the login requirements appear, and the user gets entered into the system. Then the user is asked to enter its car information completely. After that according to the user requirement he/she reserves the parking slot. When a parking slot got reserved the button appears green and when the user occupies its parking space than the button becomes red. In order to cancel any reservation, the user just cancel it by one click and after that the button appears white which shows the slot is available. If the user in any case want to update his/her information than they can easily come to their profile and update their information.

3.3.2 Screen Images

Following are few screen images of app.

3.3.2.1 Home Screen

Figure3.2 Home Screen

3.3.2.2 **Sign-up**

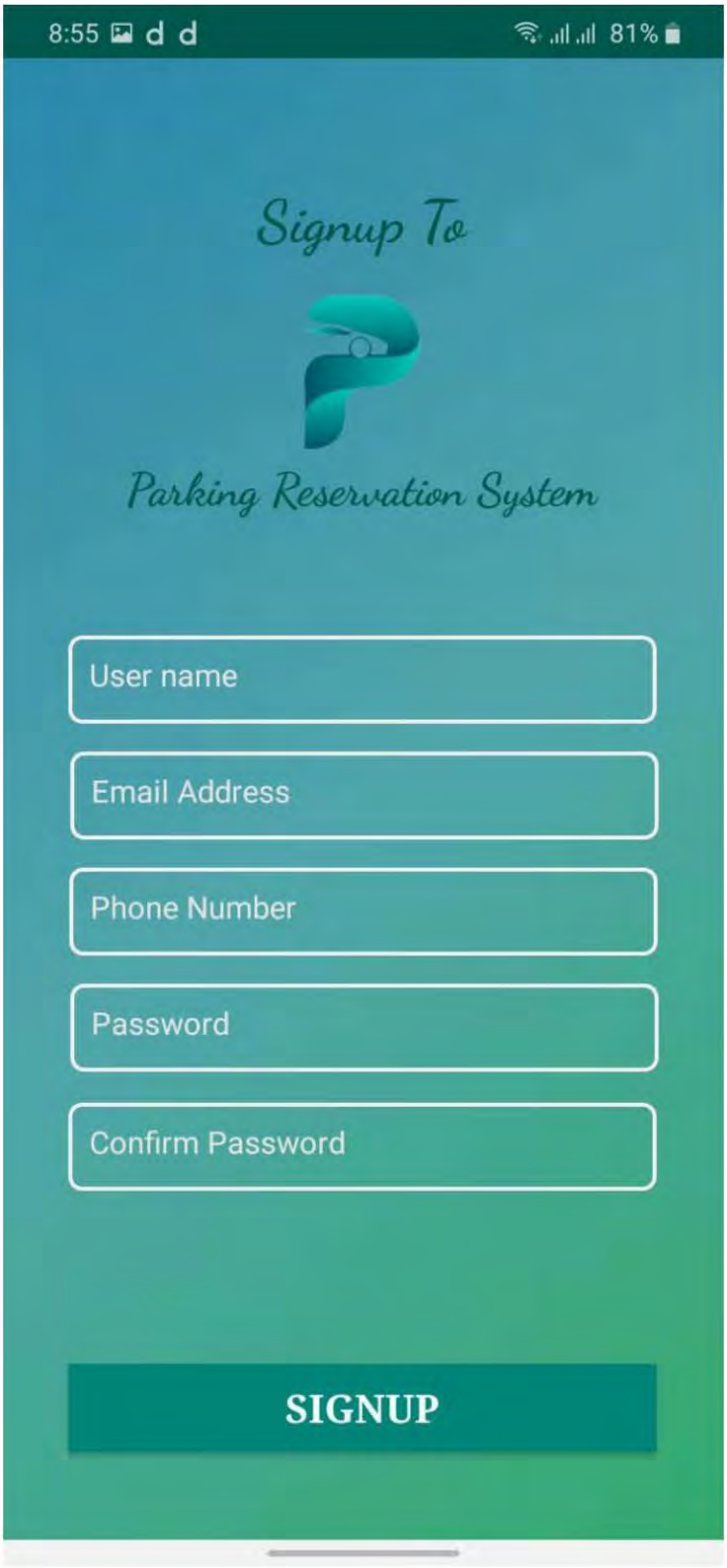

Figure 3.3 Sign-up

3.3.2.3 **Login**

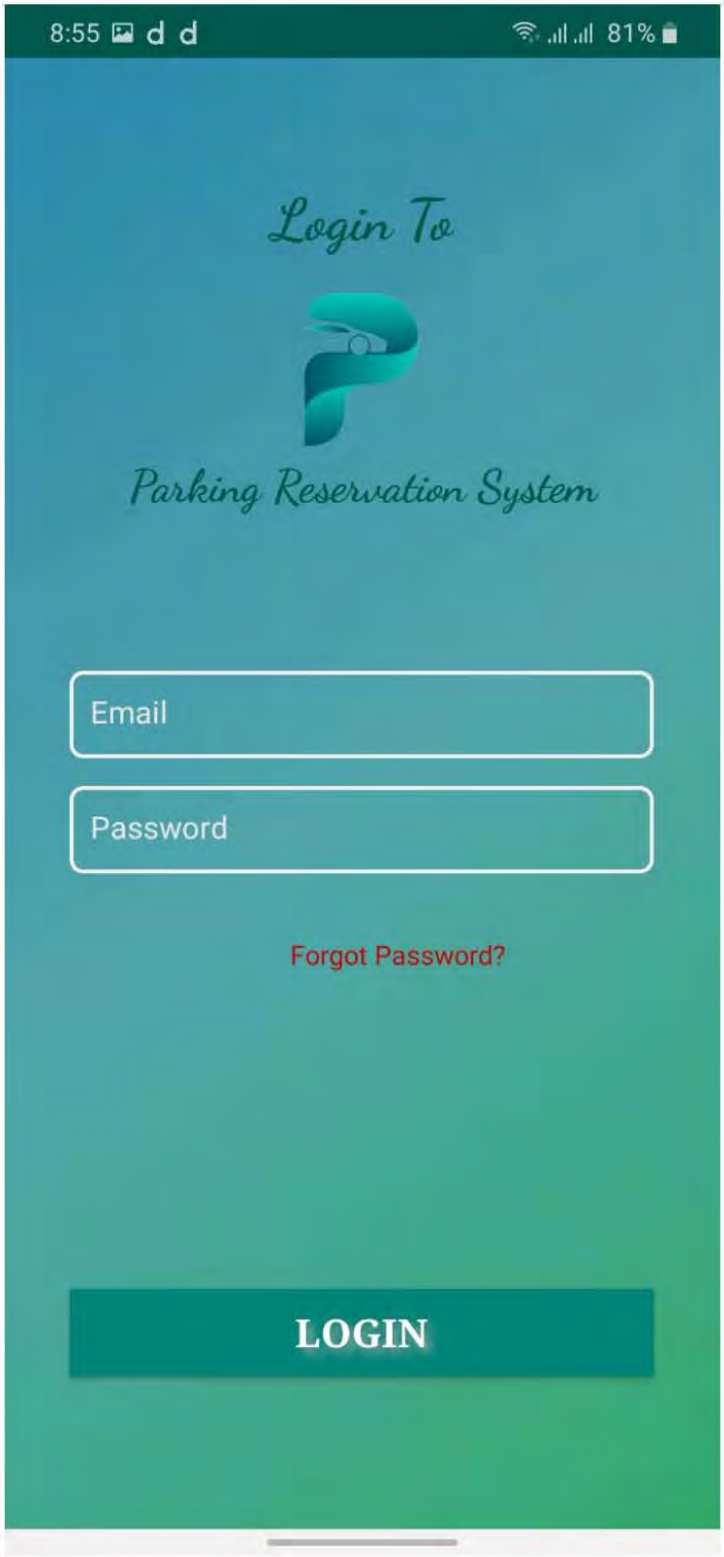

Figure 3.4 Log-in

3.3.2.4 **View Parking**

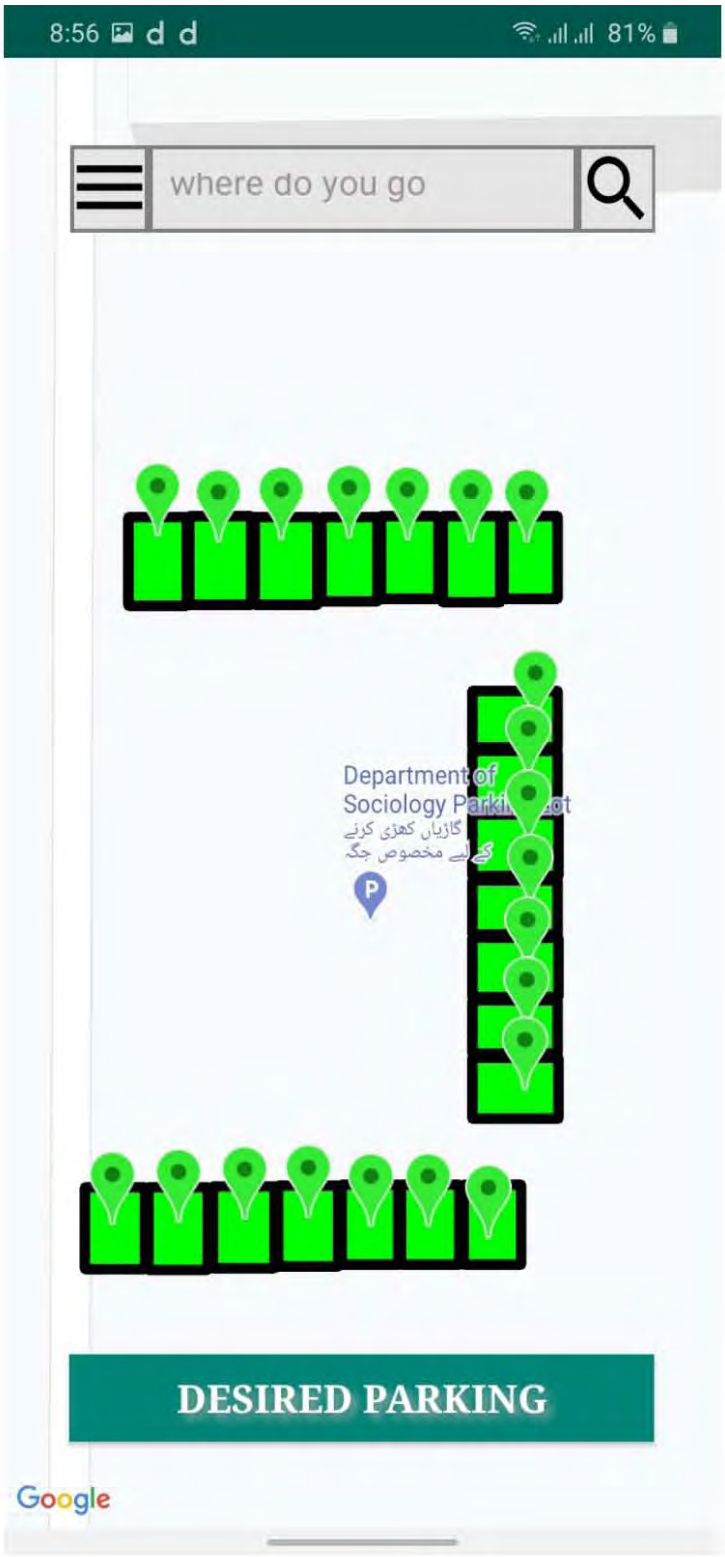

Figure 3.5 View Parking

3.3.2.5 **Book Parking**

Figure 3.6 Book Parking

3.3.2.6 **Desired Parking**

Figure 3.7 Desired Parking

3.4 Sequence Diagram

A sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It is a construct of a message sequence chart. A sequence diagram shows object interactions arranged in time sequence.

3.4.1 Login

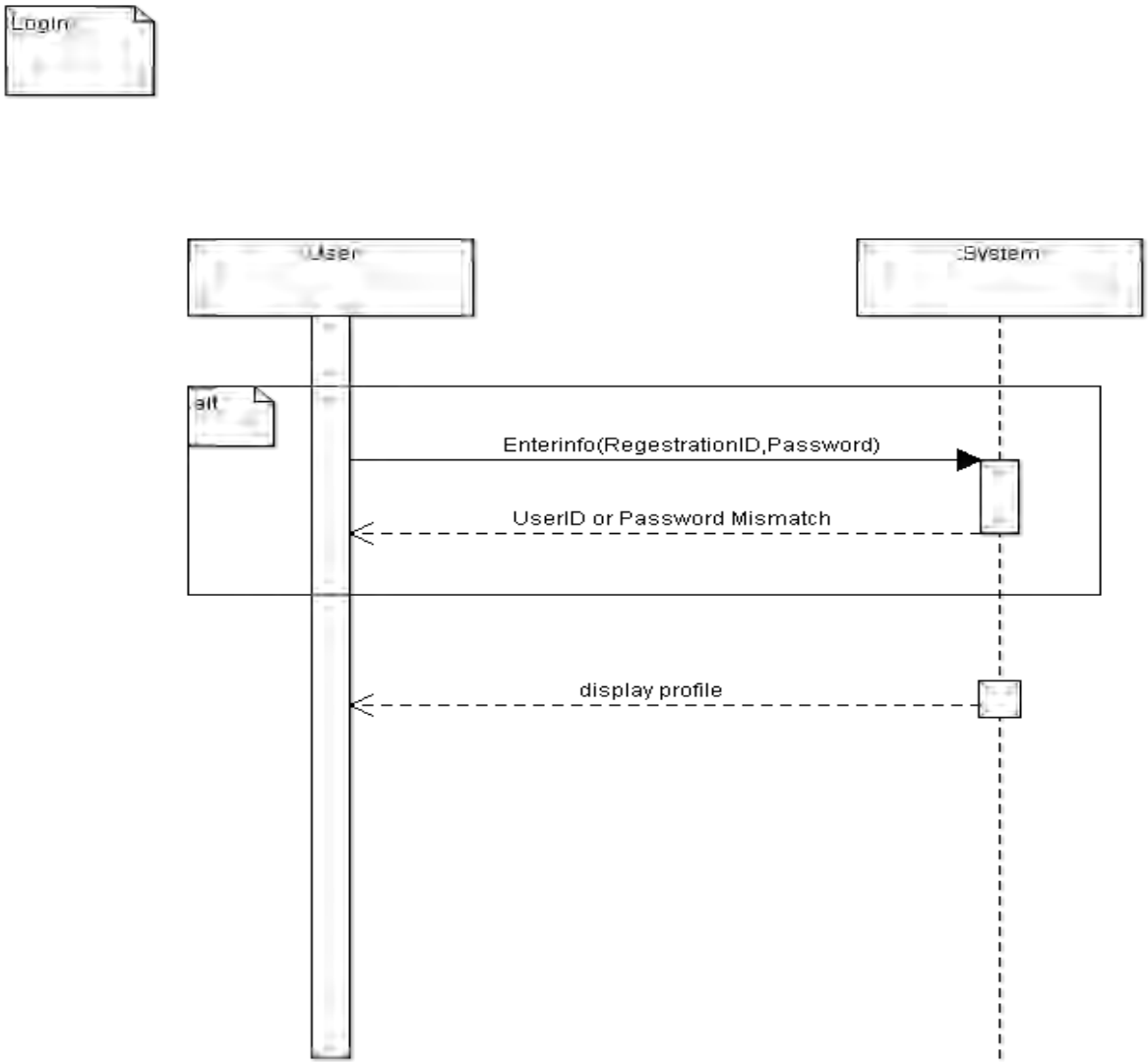

Figure 3.8 Login

3.4.2 **View Parking**

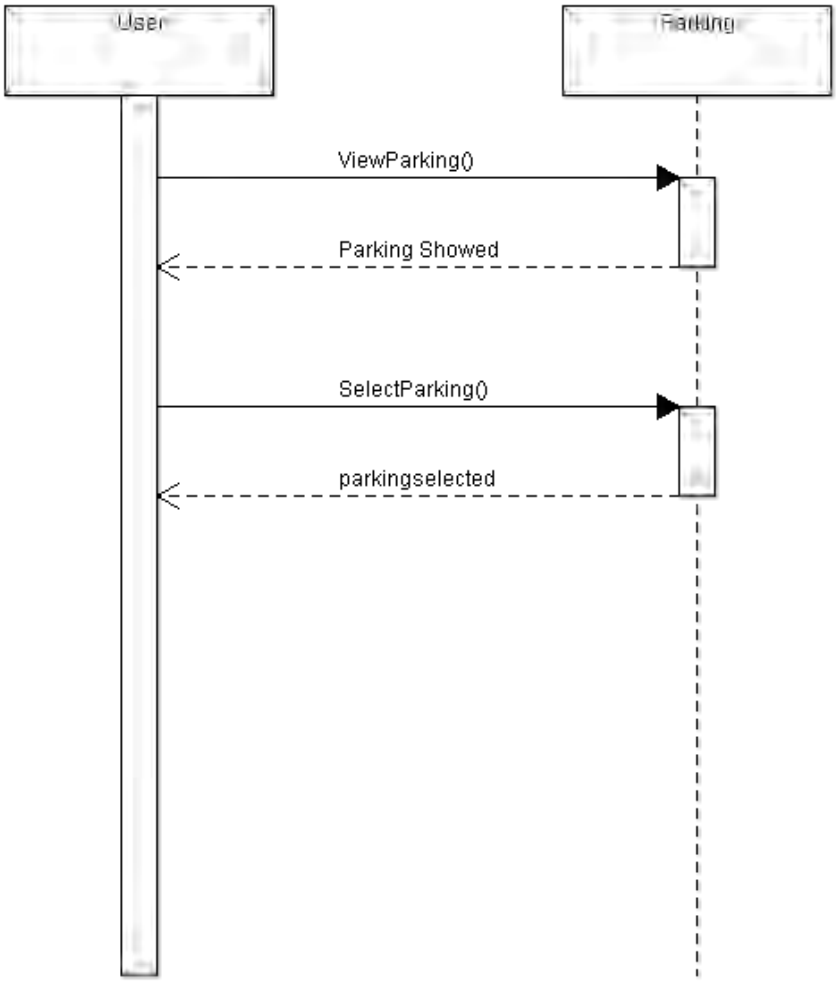

Figure 3.9 View Parking

3.4.3 **Book Parking**

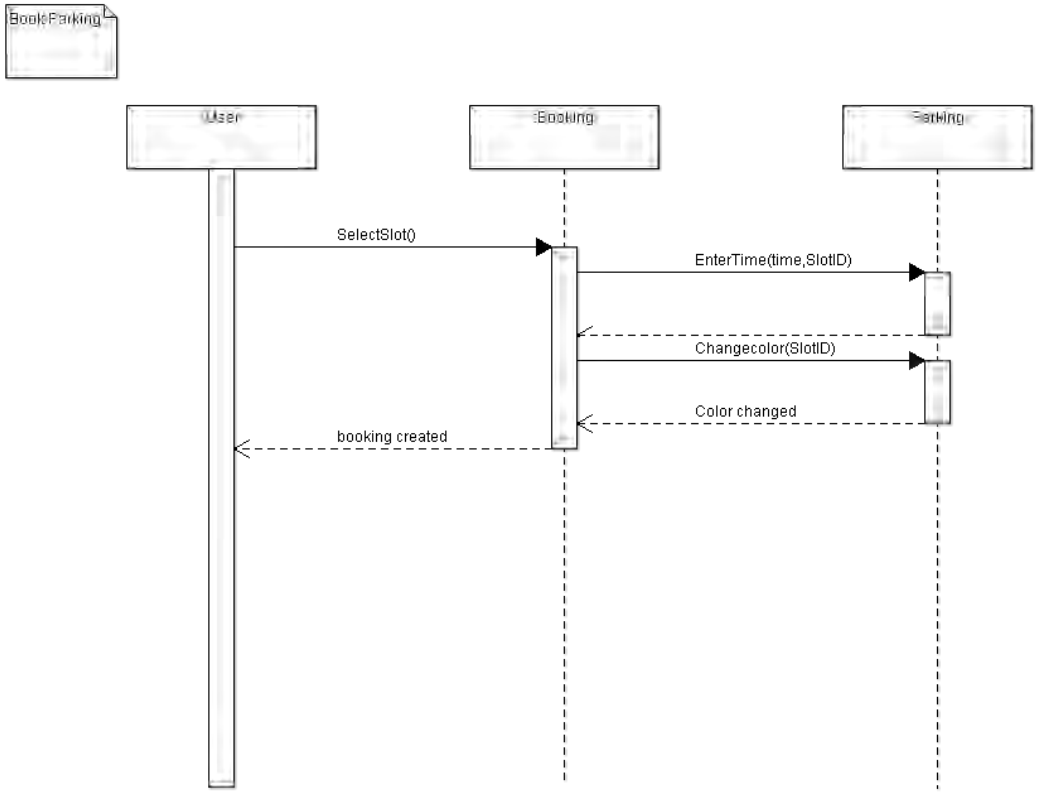

Figure 3.10 Book Parking

3.4.4 **Cancel Booking**

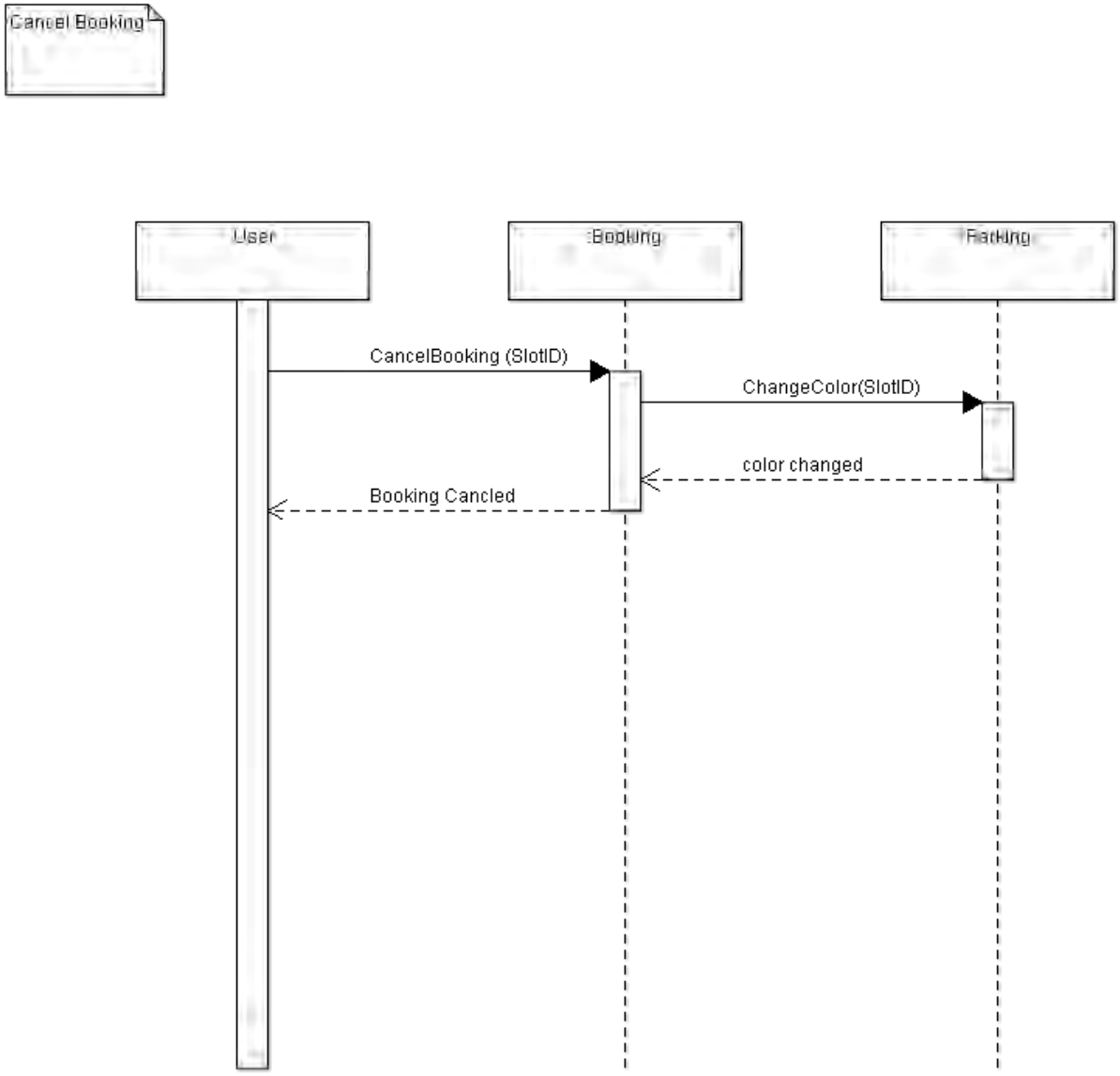

Figure 3.11 Cancel Booking

3.4.5 **Cancel Parking**

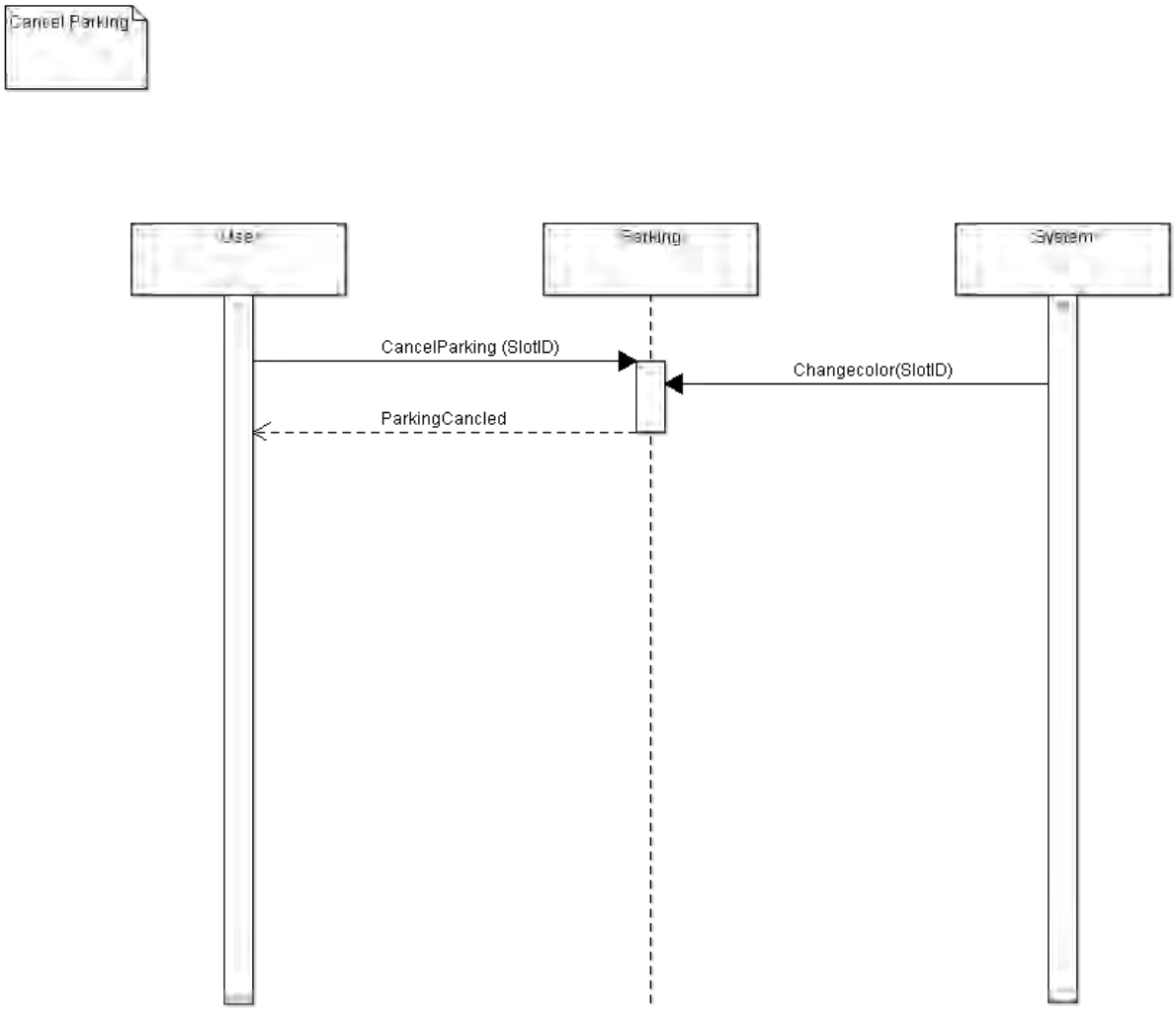

Figure 3.12 Cancel Parking

3.4.6 **Manage User Details**

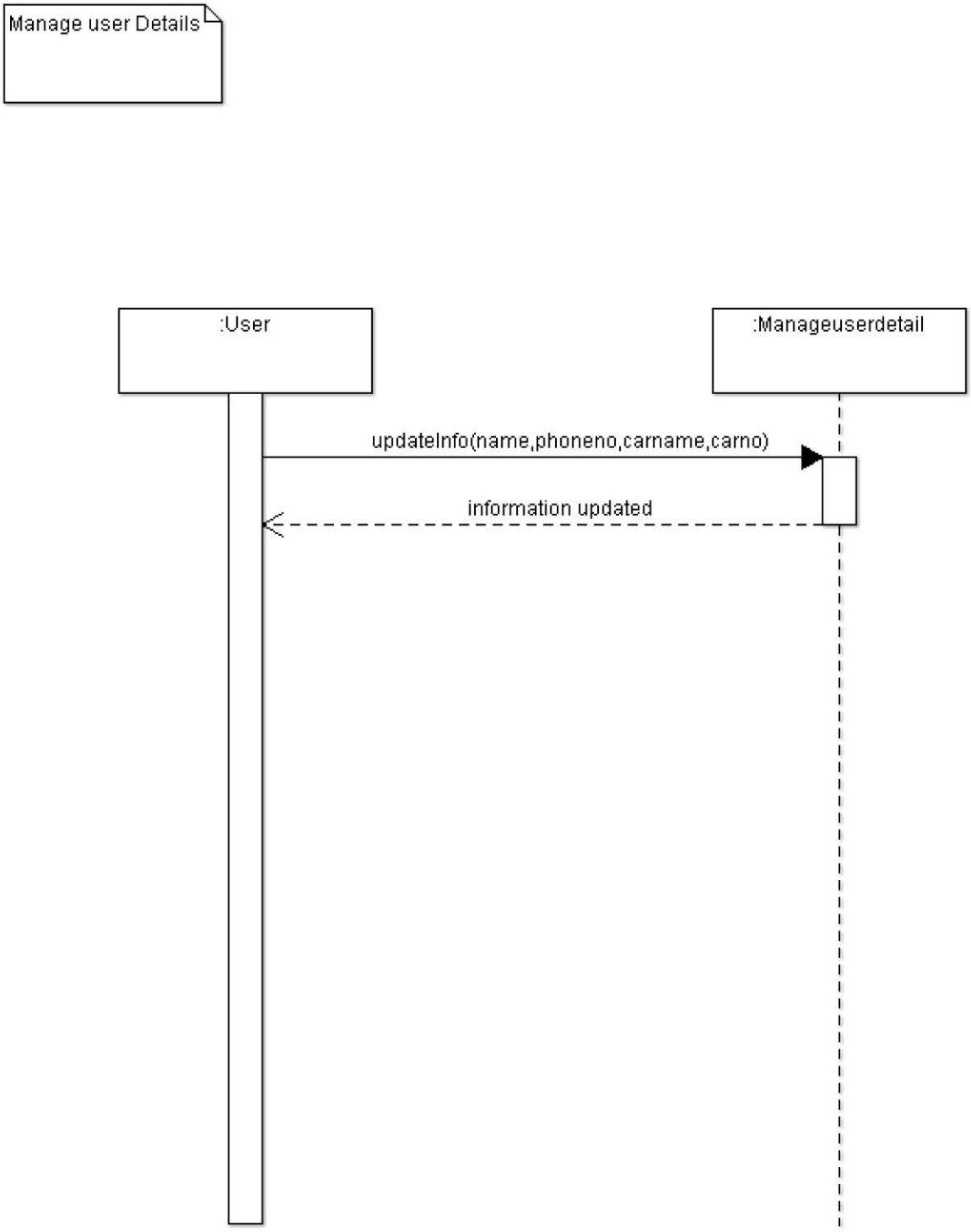

Figure 3.13 Manage User Details

3.5 **Class Diagram**

Class diagram shows the interrelationships including inheritance, association and aggregation, operations and attributes of the classes. Here is class diagram of the parking reservation system.

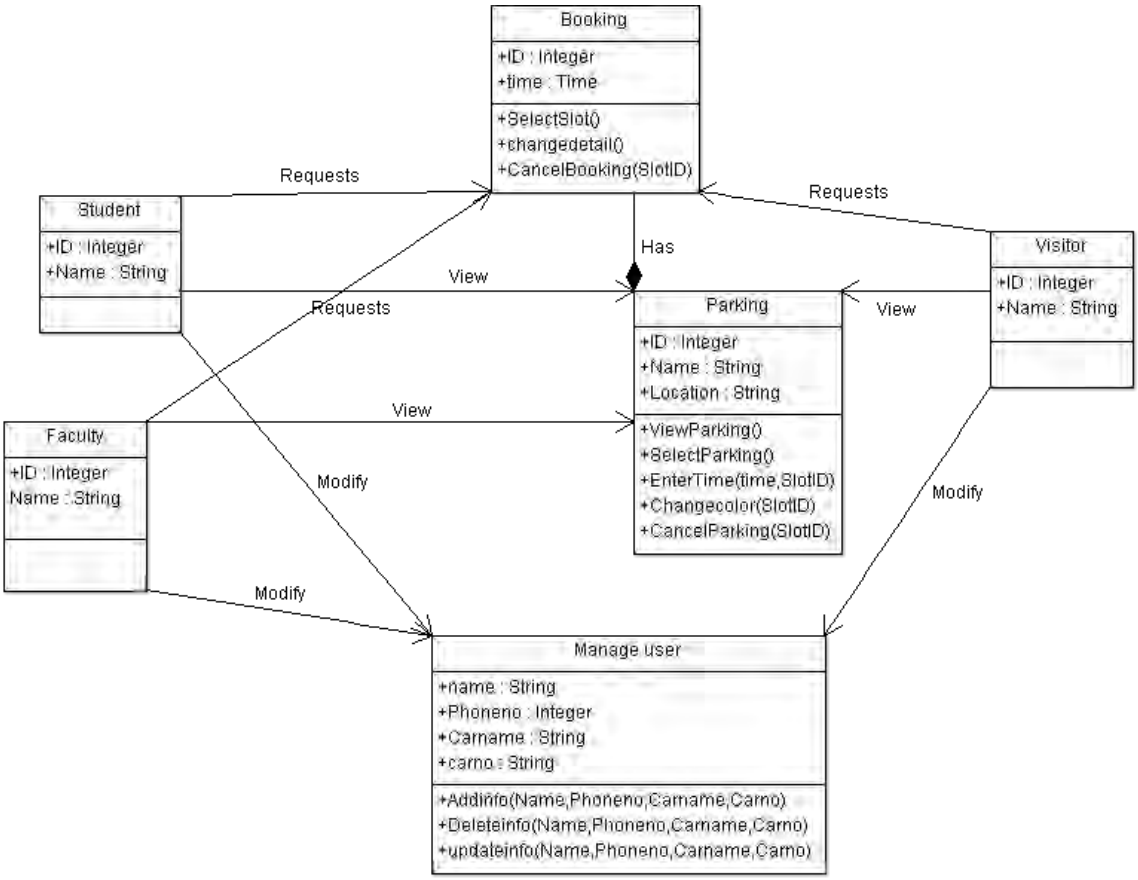

Figure 3.14 Class Diagram

3.6 **Activity Diagram**

Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another.

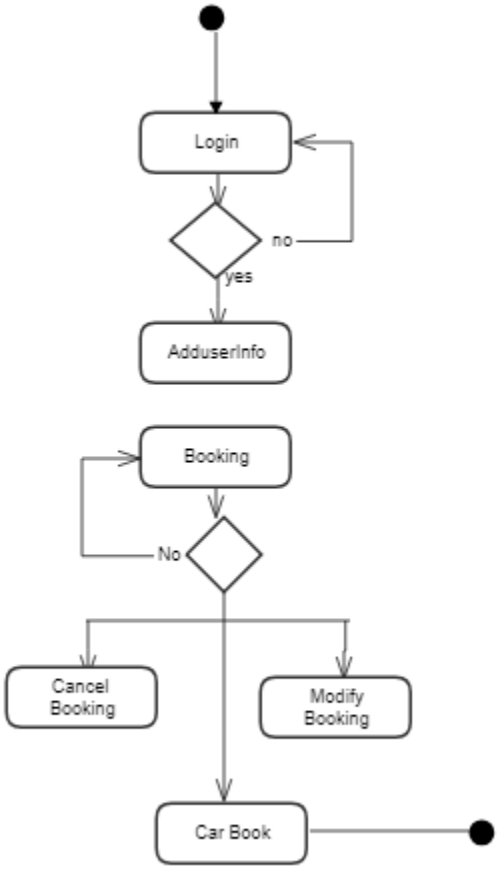

Figure 3.15 Activity Diagram

Chapter 4 Software Test Documentation

4.1 Introduction

Testing is the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not. In simple words, testing is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements. Software testing is a very important process that should be done during the development process because it is very useful to assess the quality of the product.

4.1.1 System Overview

It is a parking reservation system. In it the user can easily reserve and book a parking slot. The user first login/signup than enters the car details and further proceed in the system. For booking a slot color gets changed. Like when the slot is booked it becomes green and when occupied it becomes red. The user can cancel the booking by just one click and the slot color turns to white. The user can update his/her personal information like name, email, password and car information.

4.1.2 Test Approach

User acceptance testing (UAT) also called beta testing or end user testing, consist of a process of verifying that a solution works for the user [6]. It is not system testing (ensuring software does no crash and meets documented requirements), but rather is there to ensure that the solution will work for the user i.e. test the user acceptance the solution. In software development, UAT as one of the final stages of a project often occurs before a client or customer accepts the new system. Users of the system perform test in line with what occur in real life scenarios.

4.2 Test Plan

4.2.1 Features to be tested

Features to be tested are all according to user (student, visitor and faculty) prospective. For example,

- Start system
- Login/signup to the system
- Entering the car information
- Booking the parking slot
- Cancel the parking

- Cancel Booking

4.2.2 Features not to be tested

Features not to be tested are from the developer's point of view. For example

- How much power is consumed by the processor?
- How much memory is consumed by the game?
- Software risk factor
- Maintainability of the system

4.2.3 Testing tools and environment

As this is beta testing (testing by the user) so no specific tools and environment is required. All a user need is an Android device with application installed.

4.3 Test Cases

Following are the test cases of Parking Reservation System.

Test Case 1

Tester	User.
Purpose	Login/signup to the system.
Setup	User has started the application/tool.
Inputs	User enters registration id and password successfully.
	User enters wrong registration id or password.
Steps	1. User selects the Login/Signup option. 2. System display the registration id and password fields. 3. User enters the registration id and password. 4. User selects the login/Signup option. 5. System checks the id and password.
Expected Result	Login Successfully.
	System prompts “Incorrect id/password”.
Actual Result	As expected.
	As expected.
Pass/Fail	Pass.
	Fail.

Table 4.1 Login/Signup

Test Case 2

Tester	User.
Purpose	Entering the car information
Setup	User has started the application/tool and has logged in.
Inputs	User enters Name, Phone no, Car name, Car no successfully.
	User enters wrong Name, Phone no, Car name, Car no.
Steps	1. User login successfully. 2. User enter the Name, Phone no, Car name, Car no. 3. User selects the done option.
Expected Result	Information added Successfully.
	System prompts “Incorrect information”.
Actual Result	As expected.
	As expected.
Pass/Fail	Pass.
	Pass.

Table 4.2 Car Information

Test Case 3

Tester	User.
Purpose	Booking the parking slot
Setup	User has started the application/tool and has logged in.
Inputs	User selects the slot and enters time span for parking successfully.
	User selects the slot and enters wrong time span for parking.
Steps	1. User login successfully. 2. User selects the view parking option. 3. User selects the parking area. 4. User selects the parking slot. 5. User enter time span for parking. 6. User selects the book parking option.
Expected Result	Parking booked.
	System prompts “invalid slot or time ”.
Actual Result	As expected.
	As expected.
Pass/Fail	Pass.
	Fail.

Table 4.3 Book Parking

Test Case 4

Tester	User.
Purpose	Cancel the parking.
Setup	User has started the application/tool and has logged in.
Inputs	User selects the cancel parking option successfully.
	User doesn't select the cancel parking option.
Steps	1. User login successfully. 2. User selects the cancel parking option.
Expected Result	Parking cancelled.
	Nothing happens.
Actual Result	As expected.
	As expected.
Pass/Fail	Pass.
	Fail.

Table 4.4 Cancel Parking

Test Case 5

Tester	User.
Purpose	Cancel Booking.
Setup	User has started the application/tool and has logged in.
Inputs	User selects the cancel booking option successfully.
	User doesn't select the cancel booking option.
Steps	1. User login successfully. 2. User selects the cancel booking option.
Expected Result	Booking cancelled.
	Nothing happens.
Actual Result	As expected.
	As expected.
Pass/Fail	Pass.
	Fail.

Table 4.5 Cancel Booking

Chapter 5 Conclusion and Future Enhancements

5.1 Conclusion

The Parking Reservation System is a web service as well as android based application that will help the Students, Faculty and visitors of Quaid-i-Azam University to reserve parking space before their arrival. This application will save time of the users as they don't have to search for parking by roaming around. As for the teachers/faculty/visitors, they all get to park in their desired parking location allocated to them.

5.2 Future Enhancements

- Payment Transaction functionality should be added in future.
- Can be extended dynamically (for whole city parked places).