

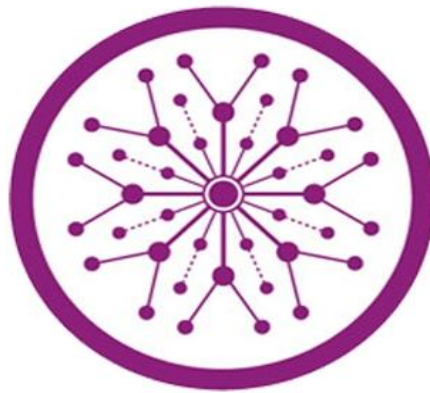
Smart Parking App

Final Year Project

Session 2018-2022

A project submitted in partial fulfillment of the degree of

BS in Computer Science



Department of Computer Science

Faculty of Computer Science & Information Technology

The Superior University Lahore

Fall 2021

Type (Nature of project)	[☒] Development [☐] Research [☐] R&D			
Area of specialization	Android Application			
FYP ID	FYP-BCSM-S21-015			
Project Group Members				
Sr.#	Reg. #	Student Name	Email ID	*Signature
(i)	BCSM-S18-005	Asadullah	Bcsm-s18-005@superior.edu.pk	
(ii)	BCSM-S18-028	Amarah Amjad	Bcsm-s18-028@superior.edu.pk	
(iii)	BCSM-S18-053	Uzma Nisar	Bcsm-s18-053@superior.edu.pk	

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

Plagiarism Free Certificate

This is to certify that, I **Asadullah** S/D of **Asmatullah Zahid**, group leader of FYP under registration no **BCSM-S18-005** at Computer Science Department, The Superior College, Lahore. I declare that my FYP report is checked by my supervisor.

Date: **2/21/2022** Name of Group Leader: **Asadullah**

Signature: _____

Name of Supervisor: Mr. Talha Amjad

Co-Supervisor: Ms Amna Zeeshan

Designation: Lecturer

Designation: Lecturer

Signature: _____

Signature: _____

HoD: Dr. Irfan-u-Din

Signature: _____

Smart Parking System

Change Record

Author(s)	Version	Date	Notes	Supervisor's Signature
	1.0		<Original Draft>	
			<Changes Based on Feedback from Supervisor>	
			<Changes Based on Feedback From Faculty>	
			<Added Project Plan>	
			<Changes Based on Feedback from Supervisor>	

APPROVAL

PROJECT SUPERVISOR

Comments: _____

Name: _____

Date: _____ Signature: _____

PROJECT MANAGER

Comments: _____

Date: _____ Signature: _____

HEAD OF THE DEPARTMENT

Comments: _____

Date: _____ Signature: _____

Dedication

We dedicate this project to all of our friends, family and teachers who have crossed this stage without any resources and faced these types of difficulties. We also dedicate this project to our supervisor **Mr. Talha Amjad** who supported and guide us throughout the project.

Acknowledgements

Above all, I owe much tribute to the Almighty God who gave us a life worth living and i thank Him for giving us the strength to accomplish this project. The success and accomplishment of this project stems from efforts and dedication offered by many individuals whose support was either direct or indirect. I thank all of them for their devotion.

Executive Summary

The Smart Parking Application aims at helping users to find the most suitable area for parking , make reservations and extend them , if required. It enables parking administrators to define and manage parking spaces as well as enables parking operators to authenticate users against their reservations when users enter the parking area. Users access location based information and request system services via mobile.

Table of Contents

Final Year Project	i
Plagiarism Free Certificate	ii
Smart Parking App.....	i
Final Year Project	i
Session 2018-2022	i
BS in Computer Science	i
Plagiarism Free Certificate	ii
Dedication	v
Acknowledgements	vi
Executive Summary	vii
Table of Contents	viii
List of Figures	xi
List of Tables	xii
Chapter 1	1
Introduction.....	1
1.1. Background	2
1.2. Motivations and Challenges	2
1.3. Goals and Objectives.....	2
1.4. Existing Solutions	2
1.5. Gap Analysis	3
1.6. Proposed Solution	3
1.7. Project Plan	4
Work Breakdown Structure	4
1.7.1. Roles & Responsibility Matrix	4
1.7.2. Gantt Chart	5
1.8. Report Outline	5
Chapter 2	6
Software Requirement Specifications	6
2.1. Introduction	7
2.1.1. Purpose	7
2.1.2. Document Conventions	7
2.1.3. Intended Audience and Reading Suggestions	8
2.1.4. Product Scope	8
2.1.5. References	8
2.2. Overall Description	8
2.2.1. Product Perspective	9
2.2.2. User Classes and Characteristics	9
2.2.3. Operating Environment	9
2.2.4. Design and Implementation Constraints.....	10
The customer should be verified with its email.	10
2.2.5. Assumptions and Dependencies	10
2.3. External Interface Requirements	11
2.3.1. User Interfaces	11
2.3.2. Hardware Interfaces.....	11

2.3.3. Software Interfaces	11
2.3.4. Communications Interfaces	11
2.4. System Features.....	11
Chapter 3.....	14
Use Case Analysis.....	14
3.1. Use Cases Description.....	17
Chapter 4.....	18
System Design	18
4.1. Architecture Diagram	19
4.2. Domain Model.....	20
4.3. Entity Relationship Diagram with data dictionary	21
4.4. Class Diagram	23
4.5. Sequence / Collaboration Diagram	24
4.5.1. Login Sequence / Collaboration Diagram.....	24
4.6. Operation contracts	27
4.7. Activity Diagram.....	27
4.8. State Transition Diagram	30
4.9. Component Diagram	33
4.10. Deployment Diagram	34
Chapter 5.....	35
Implementation	35
5.1. Important Flow Control/Pseudo codes	36
5.2. Components, Libraries, Web Services and stubs	37
5.3. Deployment Environment	37
5.1. Tools and Techniques.....	37
Development Tools	37
Software Designing Tool	38
Testing Tool	38
Documentation Tools.....	38
Techniques	38
Programming Techniques	38
Web Application Techniques.....	38
Testing Techniques	38
5.2. Best Practices / Coding Standards.....	38
Chapter 6.....	40
Testing and Evaluation	40
6.1. Use Case Testing.....	41
6.2. Equivalence partitioning	42
6.3. Boundary value analysis.....	43
6.4. Data flow testing	43
6.5. Unit testing	43
6.6. Integration testing.....	43
6.7. Performance testing.....	43
6.8. Stress Testing	43
Chapter 7.....	44
Summary, Conclusion and Future Enhancements	44

7.1. Project Summary	45
7.2. Achievements and Improvements	45
7.3. Critical Review	45
7.4. Lessons Learnt.....	46
7.5. Future Enhancements/Recommendations	46
Appendices.....	47
Appendix A: Information / Promotional Material	48
Reference and Bibliography	53

List of Figures

1.1	Caption of first figure of first chapter	4
1.2	Caption of second figure of first chapter	5
3.1	Caption of first figure of third chapter	17
4.1	Caption of first figure of fourth chapter	19
4.2	Caption of second figure of fourth chapter	20
4.3	Caption of third figure of fourth chapter	22
4.4	Caption of fourth figure of fourth chapter	23
4.5	Caption of fifth figure of fourth chapter	25
4.6	Caption of sixth figure of fourth chapter	26
4.7	Caption of first figure of fifth chapter	31
4.8	Caption of second figure of fifth chapter	32
4.9	Caption of first figure of fifth chapter	33
4.10	Caption of first figure of fifth chapter	34

List of Tables

1.1	label of first table of first chapter	4
2.1	label of first table of second chapter	9
4.1	label of first table of fourth chapter	27
6.1	label of first table of sixth chapter	41
6.2	label of second table of sixth chapter	42
6.3	label of third table of sixth chapter	43

Chapter 1

Introduction

Chapter 1: Introduction

Smart parking system enables users to reserve their parking places . Once the reservation is confirmed, the users will receive the one time password to enter the parking area and the lot number for parking. If they exceed the time to reach the parking area, the password will expire and the reservation will be cancelled. The system is subdivided into Reservation and Access system.

1.1. Background

Locating a parking spot during peak hours in most populated areas like shopping malls, universities, exhibitions or convention centers is difficult for the drivers. The difficulty rises from not knowing where the available spots may be at that required time. Smart parking is a solution to metropolitan cities to reduce congestion, cut vehicle emission totals and save persons' time by helping them in finding a spot to park.

1.2. Motivations and Challenges

Motivation to doing this is that, there is no such platform exists that is why we decided to take this project which will help people in future, and the challenge behind this project is as there is no related project exists so that we were not have a single idea how to start and where to get started. And which things or components should be added in it.

1.3. Goals and Objectives

Goals behind this application(Smart Parking System) involves the use of real-time data and applications that allow users to monitor available and unavailable parking spots. The goal is to automate and decrease time spent manually searching for the optimal parking floor, spot and even lot.

1.4. Existing Solutions

Existing parking systems are working manually in Pakistan, A person needs to present physically at certain place and ensure his parking place by his own. We are going to ensure the automated

parking system .A person can reserve his parking from anywhere at anytime through smart parking app. The payment and parking will be much easier as compared to the existing systems

1.5. Gap Analysis

If we talk about the gap analysis, I would like to say nothing is hundred percent perfect at start with the passage of time things are getting better, and our plan is to launch this app based application with such convenient way that everyone would like it and show some eager to use it.

1.6. Proposed Solution

Smart Parking is a parking system, usually a new one that is equipped with special structured devices (things) to detect the available parking slots at any parking area. This is an application based on Internet of Things (IoT) that in Real-Time environment have sensors and devices embedded into parking spaces, transmitting data on the occupancy status; and the vehicle drivers can search for parking availability using their mobile phones or any information system that is attached to the vehicle. Hence the driver would know where there is an available spot to park his vehicle in less time, reducing the energy consumption and air pollution.

Information gathering

Information gathering is done by interviewing the users. For the development of smart parking system a lot of research and important input from various websites and users was needed. Hence the following questionnaires were provided to them.

- What are the difficulties you are facing in the manually system?
- What all new things you want to be included in the proposed solution?
- In what way you are storing your information?

1.7. Project Plan

The project plan of our project:

Dashboard design 30 days

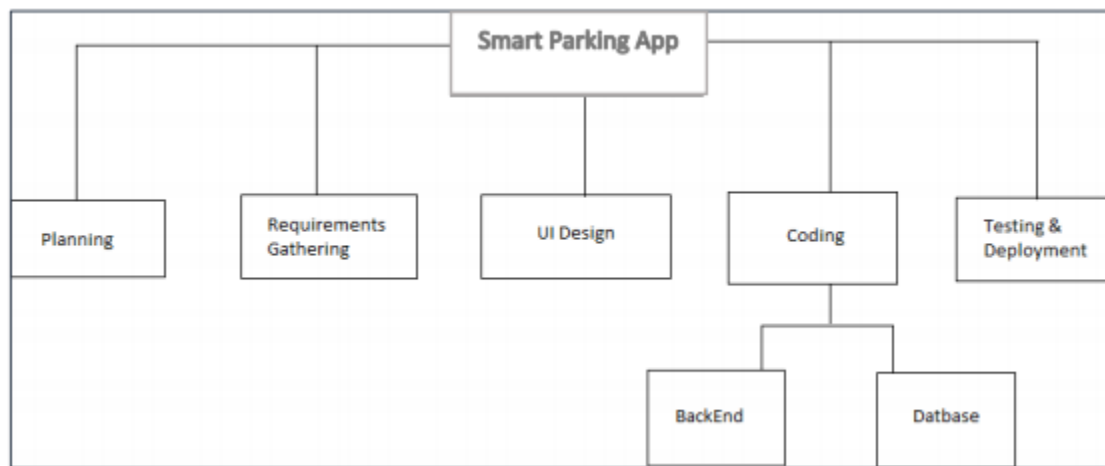
Requirement gathering 30 days

UI design 60 days

Coding 180 days

Testing and implementation 30 days

Work Breakdown Structure

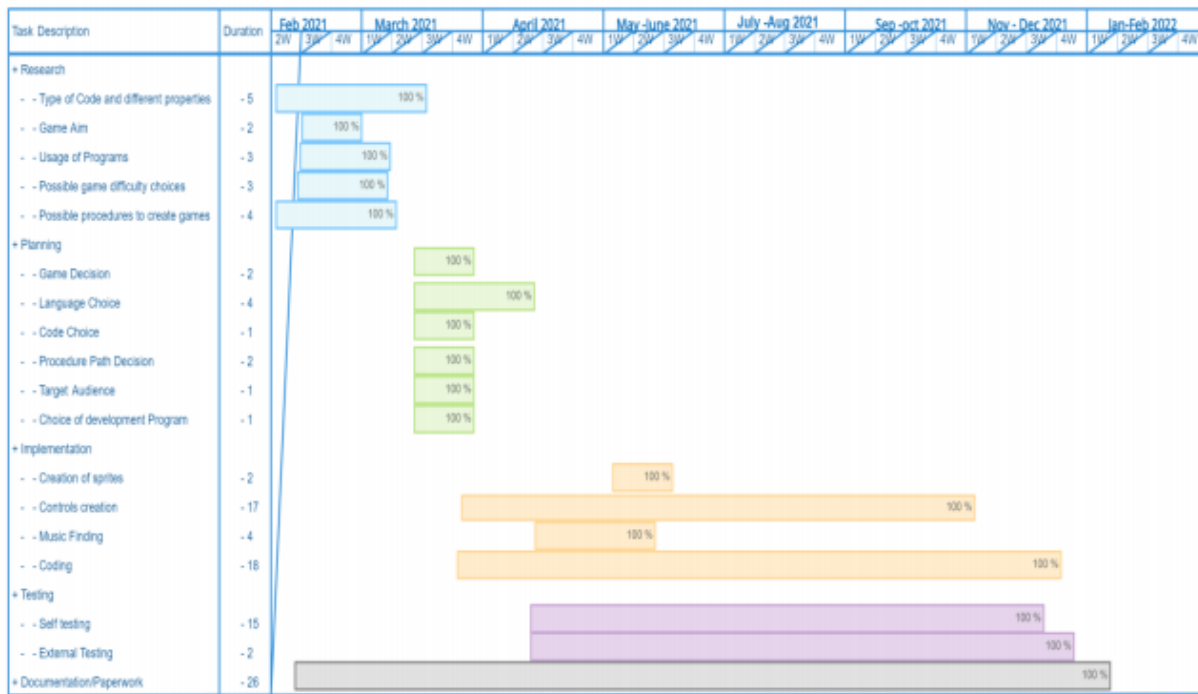


1.7.1. Roles & Responsibility Matrix

The purpose of roles & responsibility matrix is to identify who will do what.

WBS #	WBS Deliverable	Activity #	Activity to Complete the Deliverable	Duration (# of Days)	Responsible Team Member(s) & Role(s)
1	Planning	1		10 Days	All Members
2	Requirement Gathering	2		30 Days	All Members
3	UI Design	3	Android Studio	60 Days	All Members
4	Coding	4	Android Studio	180 Days	All Members
5	Testing and Implementation	5		30 Days	All Members

1.7.2. Gantt Chart



1.8. Report Outline

To make the document more effective and readable I used Calibri font style and font size of 12, and main headings are bold with the size of 16, and sub-headings are bold with the size of 14. Line spacing in paragraphs is 1.5.

Chapter 2

Software Requirement Specifications

Chapter 2: Software Requirement Specifications

2.1. Introduction

This section offers a summary of the smart parking system project, including goals and objectives, project scope, general system details, and some major constraints associated with the intended platform.

2.1.1. Purpose

The purpose of this document is to describe the external and internal behavior of the system. In this document, every kind of requirement will be finalized. It describes the functional and non-functional requirement, design requirements and also provides the other detail which will clear the whole idea about the project.

2.1.2. Document Conventions

Document contains headings and paragraphs.

First Heading: is bold, Font is Calibri and Size is 16.

Second Heading: is bold, Font is Calibri and Size is 14.

Paragraph: Font is Calibri, Size is 12 and line spacing is 1.5.

DB: Database

ERD: Entity Relationship Diagram

SRS: Software Requirement Specification

API's: Application programming interface

Ui: user interface

ERD: entity relationship diagram

OS: operating system

HI: Hardware interface

SI: Software interface

CI: Communication interface

2.1.3. Intended Audience and Reading Suggestions

In the first section purpose of the document is described where every reader can understand the purpose of this document. In the second section, we described the product scope where administrators can get the idea of the system. In the overall description operator and client can clear the whole idea of the systems. Developers can understand the requirements of the system. Designers can understand the design requirements in the user interface requirement sections. The tester can prepare test cases for testing the features of the system. Client can verify their functional requirements.

2.1.4. Product Scope

This project is very useful for the parking provide the services to their devices through the mobile application and in future insha ALLAH in web application. In this application all the operator will show their exact location.

2.1.5. References

- www.apsinvo.com
- <https://www.smartparking.com/smartpark-system/smart-app>
- https://www.temjournal.com/content/94/TEMJournalNovember2020_1357_1363.pdf
- https://www.researchgate.net/publication/308186280_Smart_Parking_Application

2.2. Overall Description

This section offers perspective of the smart parking system project as well as the user classes and characteristics, the operating environment of the product, the constraints in production and designs and the assumptions & dependencies of the project.

2.2.1. Product Perspective

This product belongs to smart parking system. There are many kinds of software products in this field like as best parking, parking panda etc. They are IOT based device linked with website and mobile application systems. Our system is a large system that includes parking.

2.2.2. User Classes and Characteristics

We defined different user classes in the below table. The company owner's not required high-level knowledge for interacting with the system. The company Manager or Admin should have technical knowledge for interacting with the system. The operator should know how to communicate on the internet.

Classes	Functionality Privilege	Technical/Non- Technical	Education Level
Company Owner	Reports Generation and Device parking	Technical or Non- Technical	Master
Company Manger	Record Management, Confirm shifting, Set Schedules and Devices parking system.	Technical	Master
Web operator	View parking slots info and online parking place.	Non-Technical or Technical	Bachelor's
Mob operator	Request booking slots,.	Technical or Non- Technical	Matriculation

2.2.3. Operating Environment

Our client is currently using CORE I5 System with 8GB RAM, 500 GB Hard Drive, and a 2.8 GHz Processor. Window 7 is installed in the System and Google Chrome Application is also installed

in the system where our software will run. On the other side, we will choose our specification where we will deploy our system on the cloud.

2.2.4. Design and Implementation Constraints

Manage the location data may be it will create memory storage limitation trouble for us.

We need to be careful when we call the APIs because we need efficient response time.

The user interface should be very simple which can be understandable for a full non-technical person.

We need HTTPS for secure communication.

We have to deal with Firebase Database.

The Naming Convention approach should be the same throughout the development.

MVC approach should be followed in programming which will help for efficient memory management.

Data should be encrypted when it comes from the user to the databases.

The customer should be verified with its email.

2.2.5. Assumptions and Dependencies

We depend on the online location API's where parking devices can be seen online on web application. In the future, if these APIs will not work then we will have to change this requirement or registration. If new technology or techniques arrive in the future and our users would demand the new technology then we will change the system requirements. In the future due to large data maybe we will have to need an advanced database system to manage our records.

2.3. External Interface Requirements

This section offers an overview of the UI , HI, SI and CI of the project smart parking system.

2.3.1. User Interfaces

The user interface (UI) for an Android app is built as a hierarchy of layouts and widgets. The layouts are View Group objects, containers that control how their child views are positioned on the screen. Widgets are View objects, UI components such as buttons and text boxes

2.3.2. Hardware Interfaces

There is no hardware interface yet.

2.3.3. Software Interfaces

Smart parking development implies an IoT-based system that sends data about free and occupied parking places via web/mobile application. ... The main idea was the creation of smart parking using the Internet of Things and ultrasonic sensors, where available parking places could be displayed in a web application

2.3.4. Communications Interfaces

The Smart Parking System would enable vehicle occupancy, monitoring and managing of available parking space in real-time that reducing human effort and cost. The communication interfaces between the system and the user. The user interfaces can be classified as android application and the system are easy to use.

2.4. System Features

User interface, admin interface, parking owner interface and other features are included. The first must-have feature is notifications to interact with users with the app. Inform customers about the available place, successful payment, completed reservation, etc. Push notifications help to interact with users better.

To build a GPS smart parking app is impossible without a GPS feature. Users will be able to select the available spots on the map.

Admin module Functional Requirements

FR1: Admin can define new parking areas; specify a range of parking slots.

FR2: Admin can update data of existing parking areas.

FR3: Admin can view the information of all registered parking areas.

FR4: Admin can manage the transaction details.

User Functional Requirements

Register: User must have to enter his/her full name, valid email and unique password along with at least one numerical numbers.

Login: A user should not be able to use this application without entering the valid email and password.

Select/Reserve Parking: If user wants to select/reserve his/her parking. She/he can't do it without payment.

Payment method: A user can pay its parking charges through easy paisa easily.

Mapping: If user wants to see map regarding his/her selected parking. She/he can see it easily.

Non Functional Requirements:

- **Performance Requirements**

System must be fast and reliable. It should not cause any errors within the system.

- **Reliability Requirements**

The reliability of smart parking app is portability and compatibility and maintainability.

- **Usability Requirements**

Smart parking app has appropriate user interface and adequate information to

Guide the user in order to use the app.

- **Security Requirements**

This application provides user authentication and authorization so that only the legitimate users are allowed to use the website.

- **Safety Requirements**

Data and records must be kept securely. Unwanted access would not be allowed.

- **scalability Requirements**

Smart parking system can further modify in future.

Chapter 3

Use Case Analysis

Chapter 3:

Use Case Analysis

Functional requirements of the system are described using use case view. The Following use case diagrams summarize the functional requirements of the system and explain it in more convenient way.

Use Case Model:

The use case system is a graphic description of the interactions among the elements of smart parking system. It represents the methodology used in system analysis to identify, clarify, and organize system requirement of smart parking system. The main actors of smart parking app in the use case diagram are:

- Admin
- User
- Easy paisa

Who perform the different type of use cases such as:

Admin

- Login
 - User authentication
- Check user
- Delete user
- View history
- Delete history
- Manage payment

User

- Login
 - User authentication
- Check slots
 - Positioning
- Confirm slot

- Payment

Easy paisa

- Pay dues through easy paisa
- Check credit balance



3.1. Use Cases Description

Use Case Name	login /signup
Use case Description	A user login to the system or web application to access the functionality of the system.
Actors	User , Admin,bank
Pre-Condition	System must be connected to the network.
Post-Condition	After got login, user get notification of successful login. User can select, confirm parking slots ,and pay dues

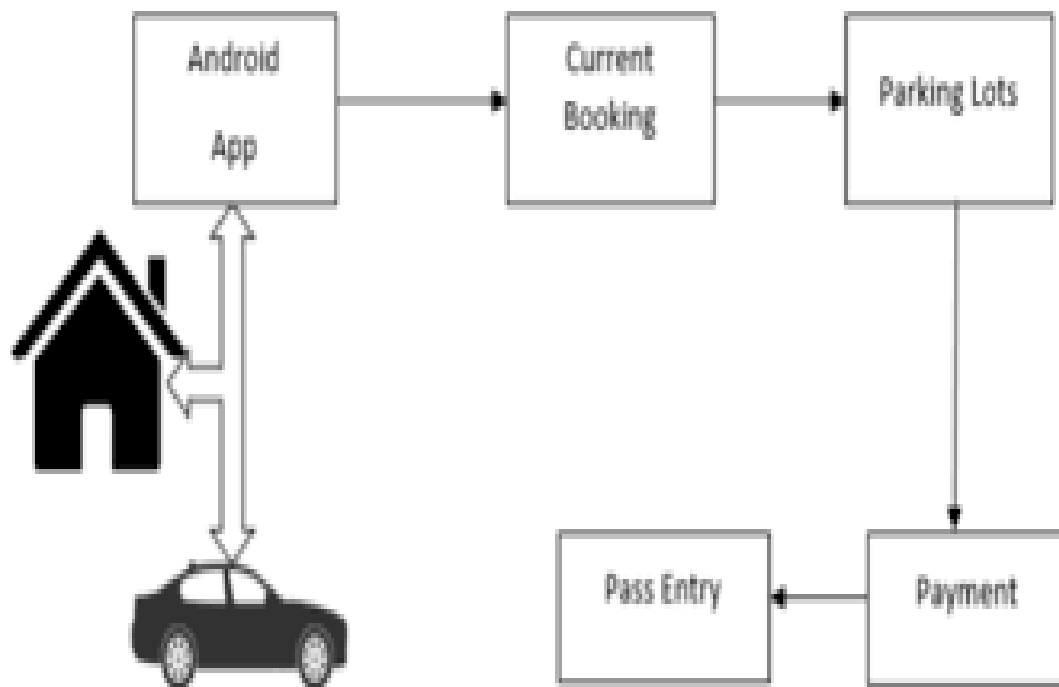
Chapter 4

System Design

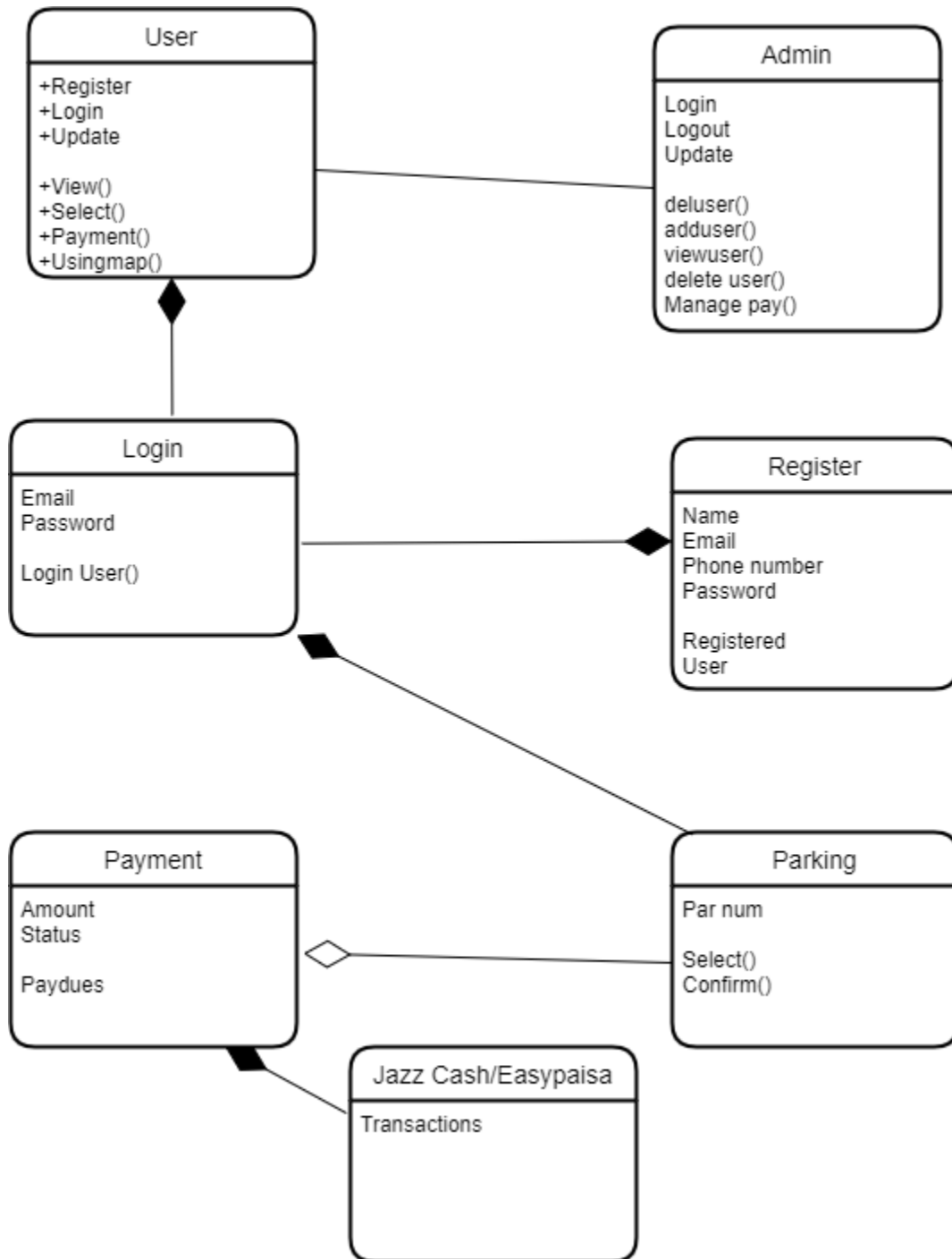
Chapter 4: System Design

System design is the process of designing the elements of a system such as the architecture, modules and components, the different interfaces of those components and the data that goes through that system.

4.1. Architecture Diagram



4.2. Domain Model



4.3. Entity Relationship Diagram with data dictionary

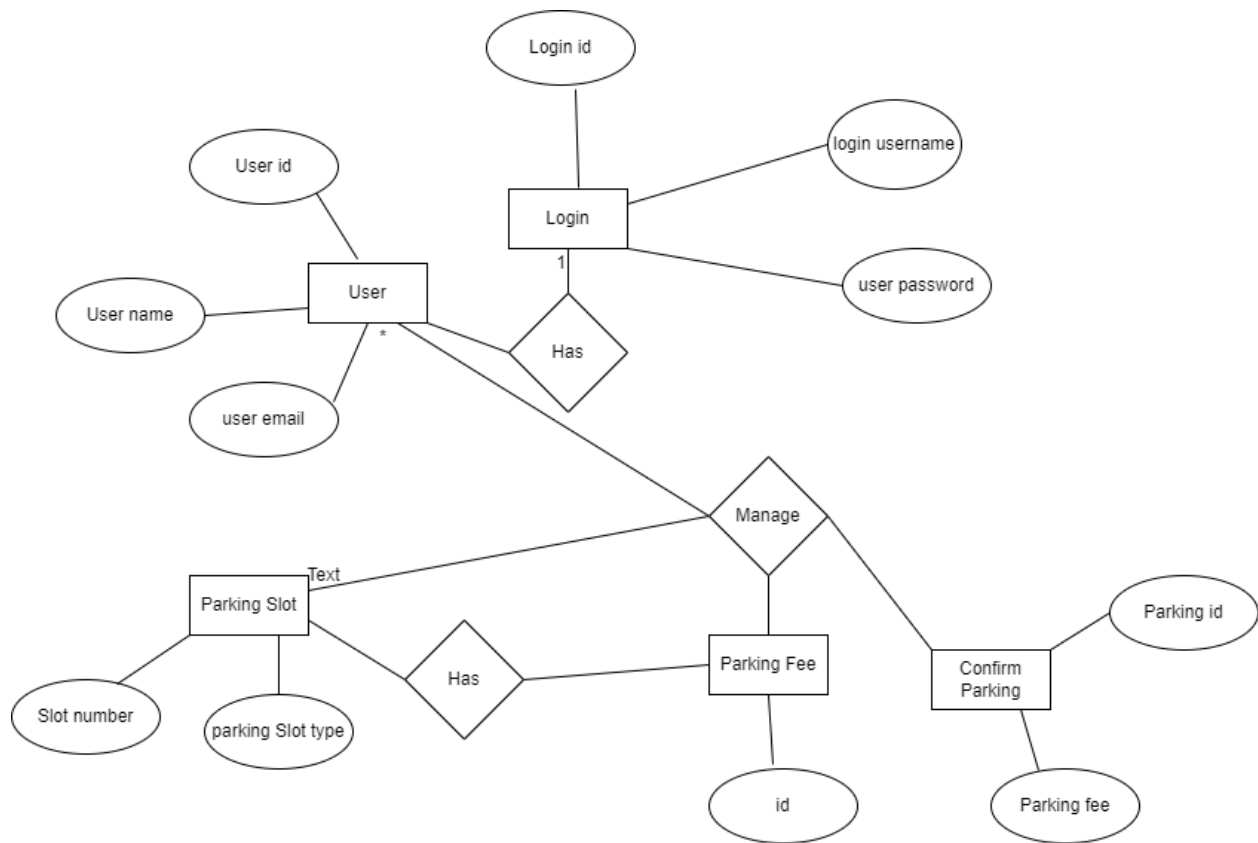
The ER diagram represents the model of smart parking system entity. The ER diagram of smart parking app shows all the visual instruments of database table and the relations between parking, parking slots etc. It used structure data and to define the relationships between structured data group of smart parking system functionalities.

Smart parking system entities and their attributes

- **Login entity:** Attributes of login are login_id, login_user_name, login_password.
- **User entity:** Attributes of user are user_id, login_user_name, user_email.
- **Parking slot entity:** Attributes of parking are p_id, ps_type.
- **Confirm Parking entity:** Attributes of confirm parking are p_id and p_fee.
- **Parking fee entity:** Attribute of parking fee is pf_id.

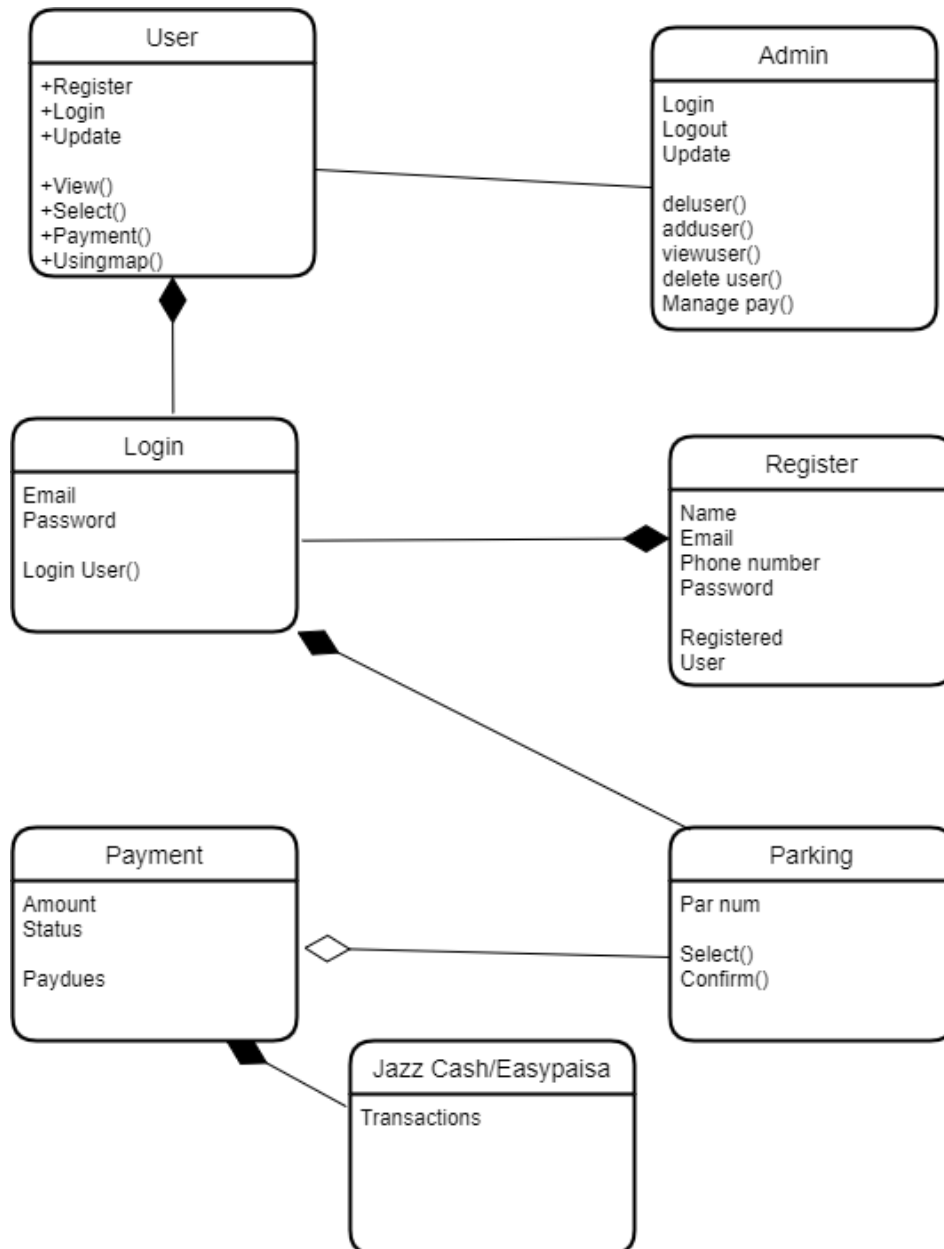
Description of Smart parking system database

- Each entity contains primary key and unique key
- There are one-to-one and one-to-many relationships available between parking fee and parking slot.
- All the entities smart parking app are normalized and reduced duplicity of records.
- We have implemented indexing on each tables of smart parking system tables for fast query execution.



4.4. Class Diagram

A class diagram is a picture for describing generic descriptions of possible systems. Class diagrams and collaboration diagrams are alternate representations of object models. This contain icons representing classes, interfaces, and their relationships. We can also create one or more class diagrams to depict classes contained by each package in our model; such class diagrams are themselves contained by the package enclosing the classes they depict; the icons representing logical packages and classes in class diagrams.



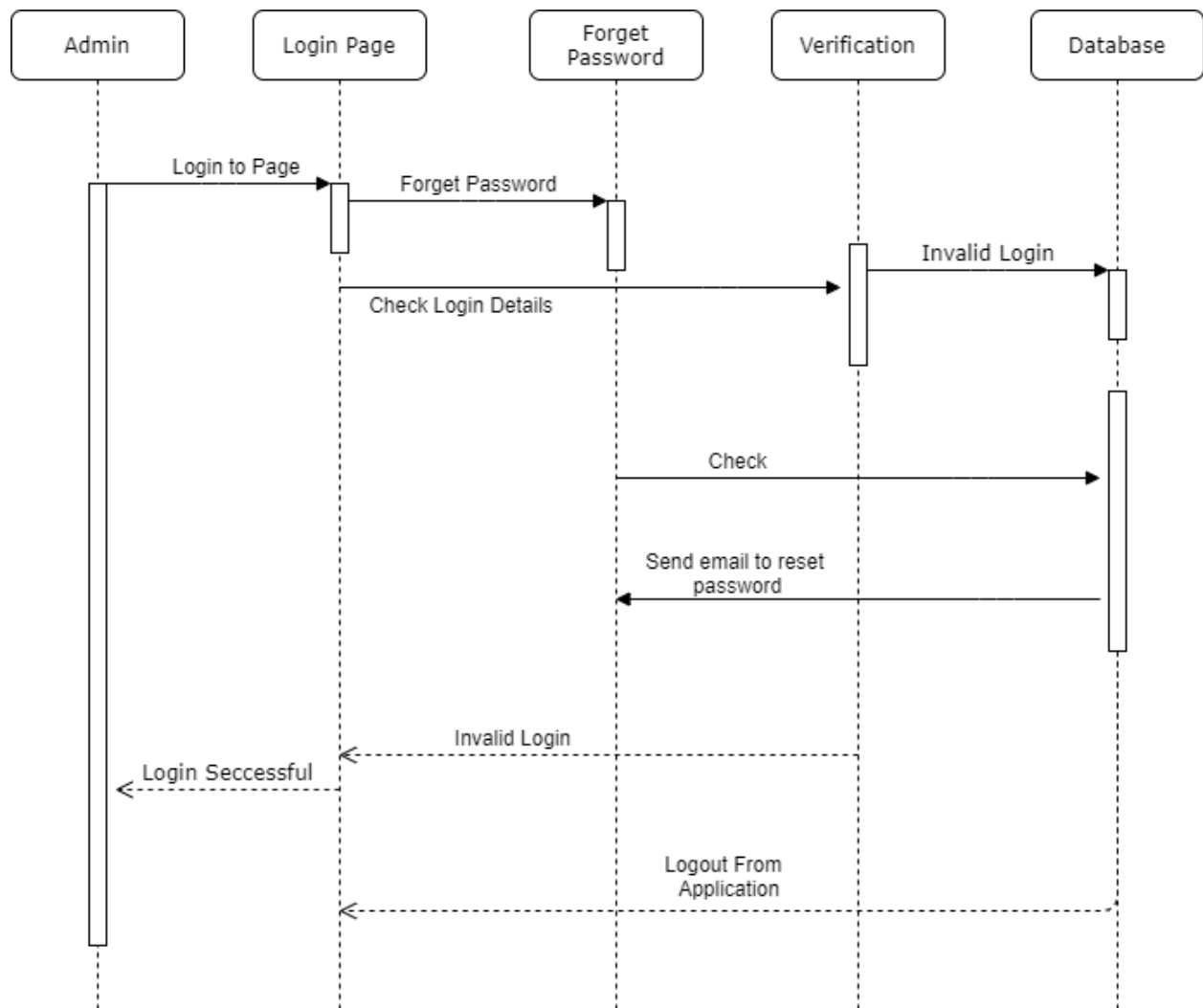
4.5. Sequence / Collaboration Diagram

This is sequence diagram of smart parking app which shows the interaction between the objects. The instance of class objects involved in this sequence diagram of smart parking app are as follow.

- login
- Parking slots
- Confirm parking
- Parking fees

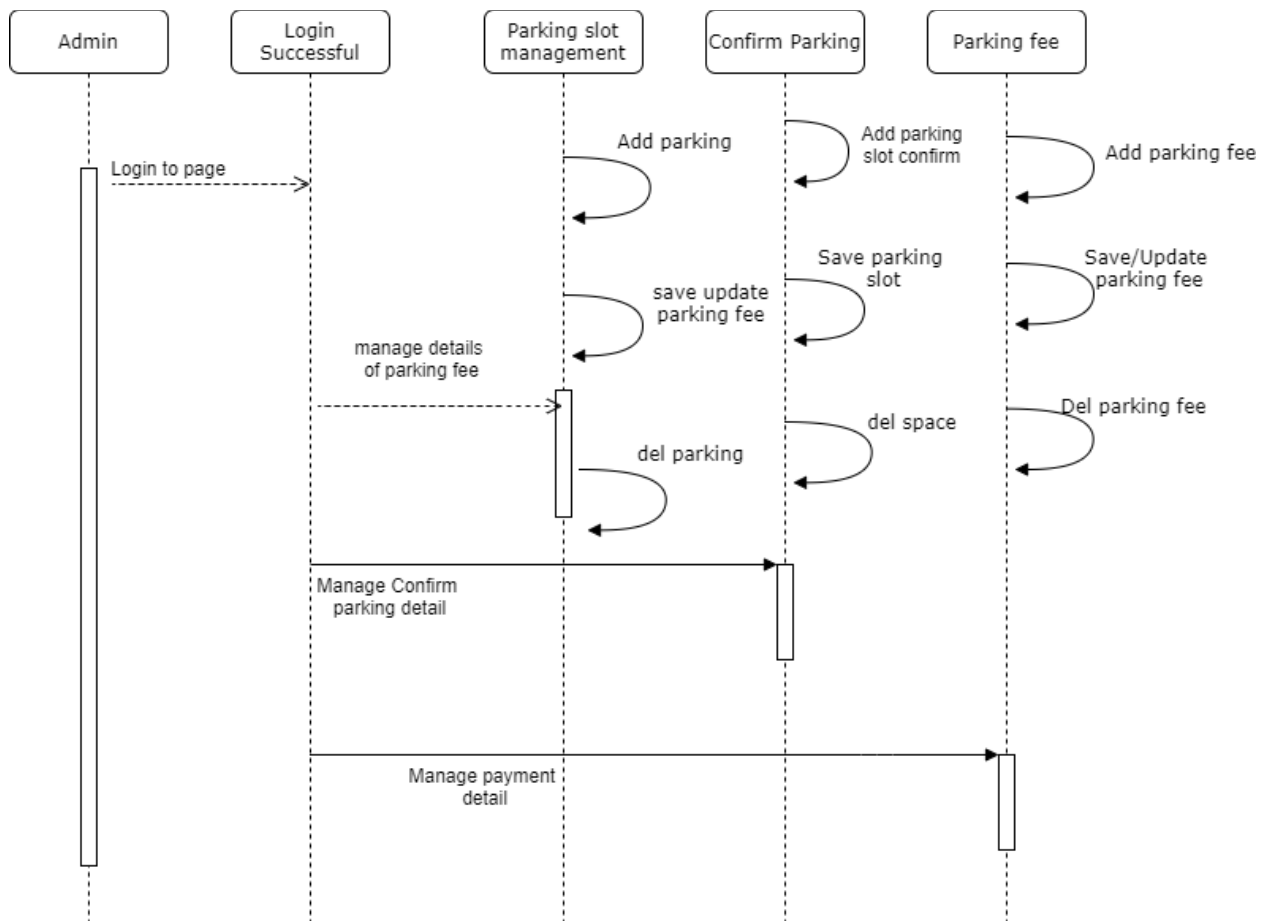
4.5.1. Login Sequence / Collaboration Diagram

This is the login diagram of smart parking app, where admin will be able to login in their account using their credentials. After login user can manage all the operations. This diagram demonstrate hoe the login page works in a smart parking app.



This is sequence diagram of smart parking app which shows the interaction between the objects. The instance of class objects involved in this sequence diagram of smart parking app are as follow.

- login
- Parking slots
- Confirm parking
- Parking fees



4.6. Operation contracts

Operation	login /signup
Actors	User , Admin and bank
Pre-Condition	System must be connected to the network.
Post-Condition	After got login, user get notification of successful login. User can select, confirm parking slots ,and pay dues

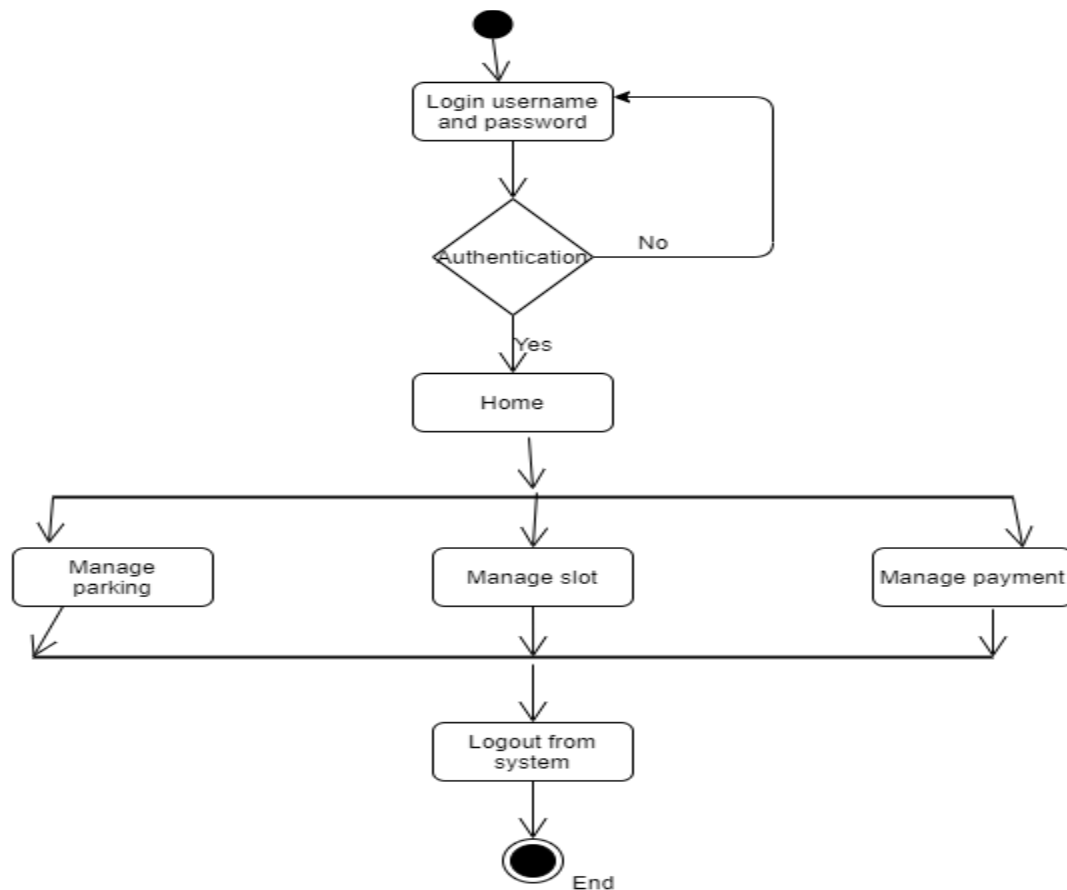
4.7. Activity Diagram

This is the activity diagram of smart parking app which shows the flows between the activity of parking slots, payment login and logout. The main activities involved in activity diagram of smart parking app are as follow:

- Parking activity
- Manage slot activity
- Manage payment activity

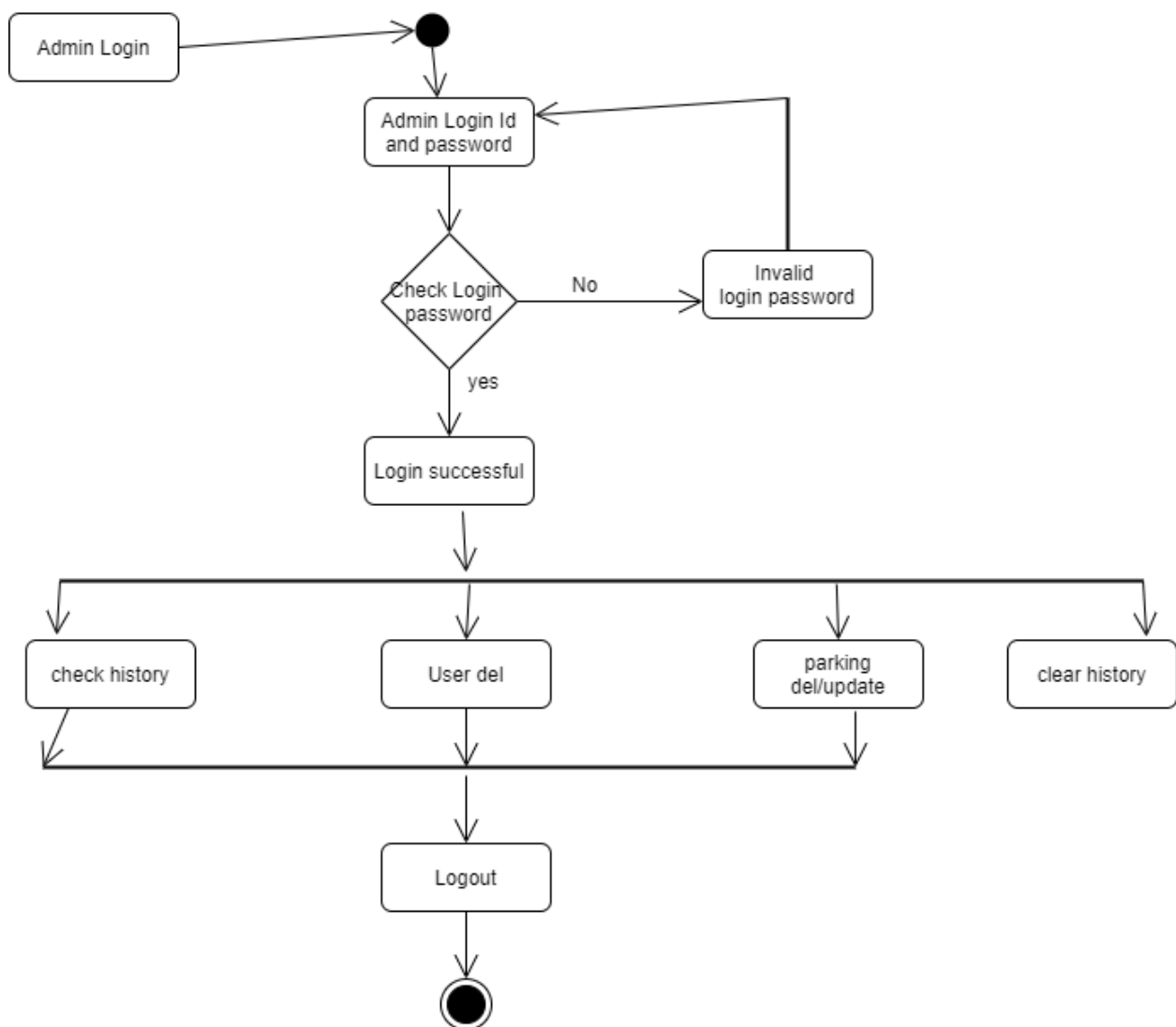
4.7.1. Features of activity diagram

- User will be able to search and generate parking slots and parking fees.
- It shows the full description and flow of parking slots, confirm parking slots and manage payment dues.



4.7.2. Login activity diagram

This is the login activity diagram of smart parking app which shows the flows of login activity where admin will be able to login using their user name and password. After login admin can manage all the operation on check user history, delete user, add user, parking history delete or update and manage the payment history. All the pages are secure and user can access these pages after login. The below diagram helps demonstrate how the login page working in smart parking app.



4.8. State Transition Diagram

State Transition diagram, shows the order of states underwent by an object within the system. It captures the software system's behavior. It models the behavior of a class, a subsystem, a package, and a complete system.

It tends out to be an efficient way of modeling the interactions and collaborations in the external entities and the system. It models event-based systems to handle the state of an object. It also defines several distinct states of a component within the system. Each object/component has a specific state.

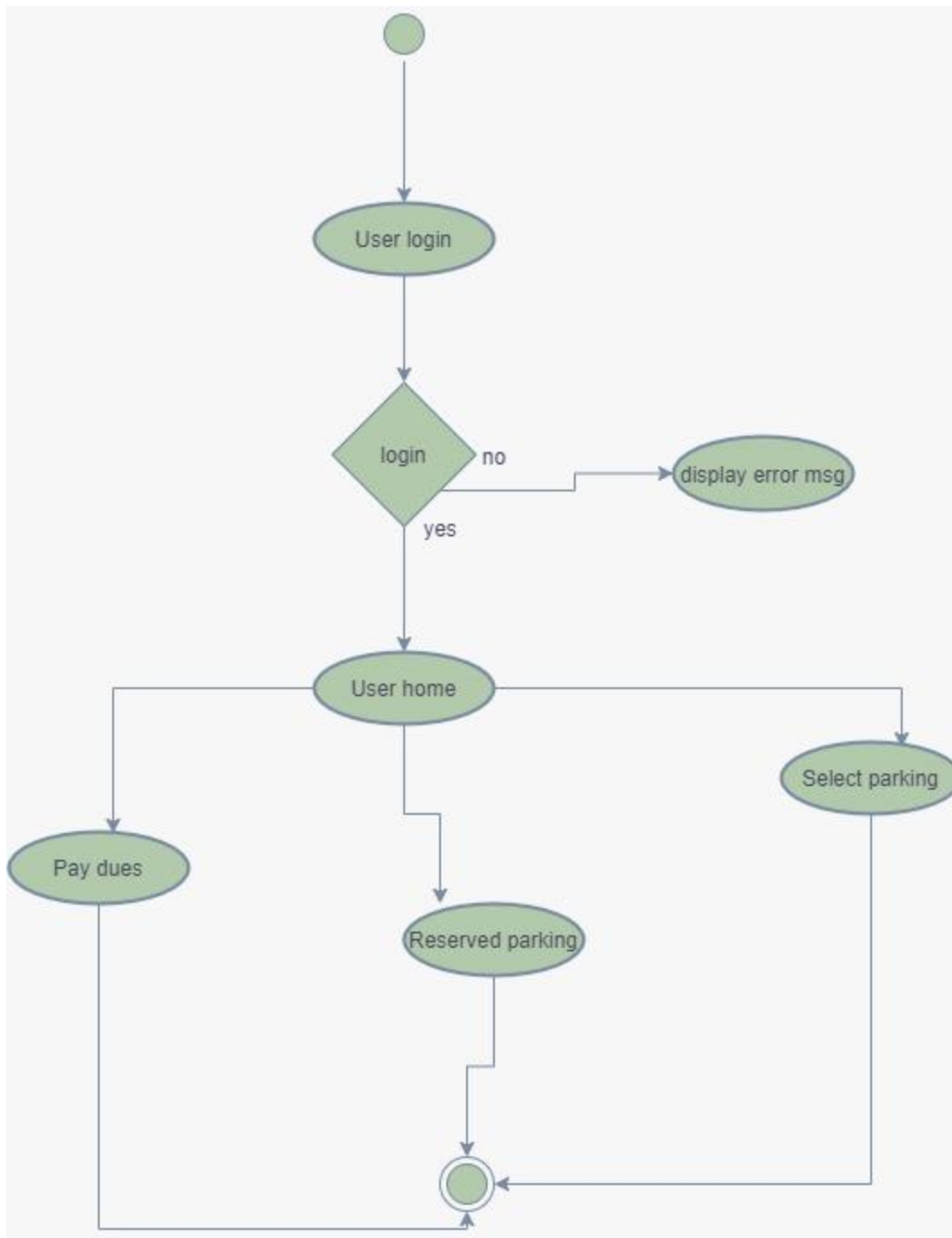
Smart parking app state transition diagram:

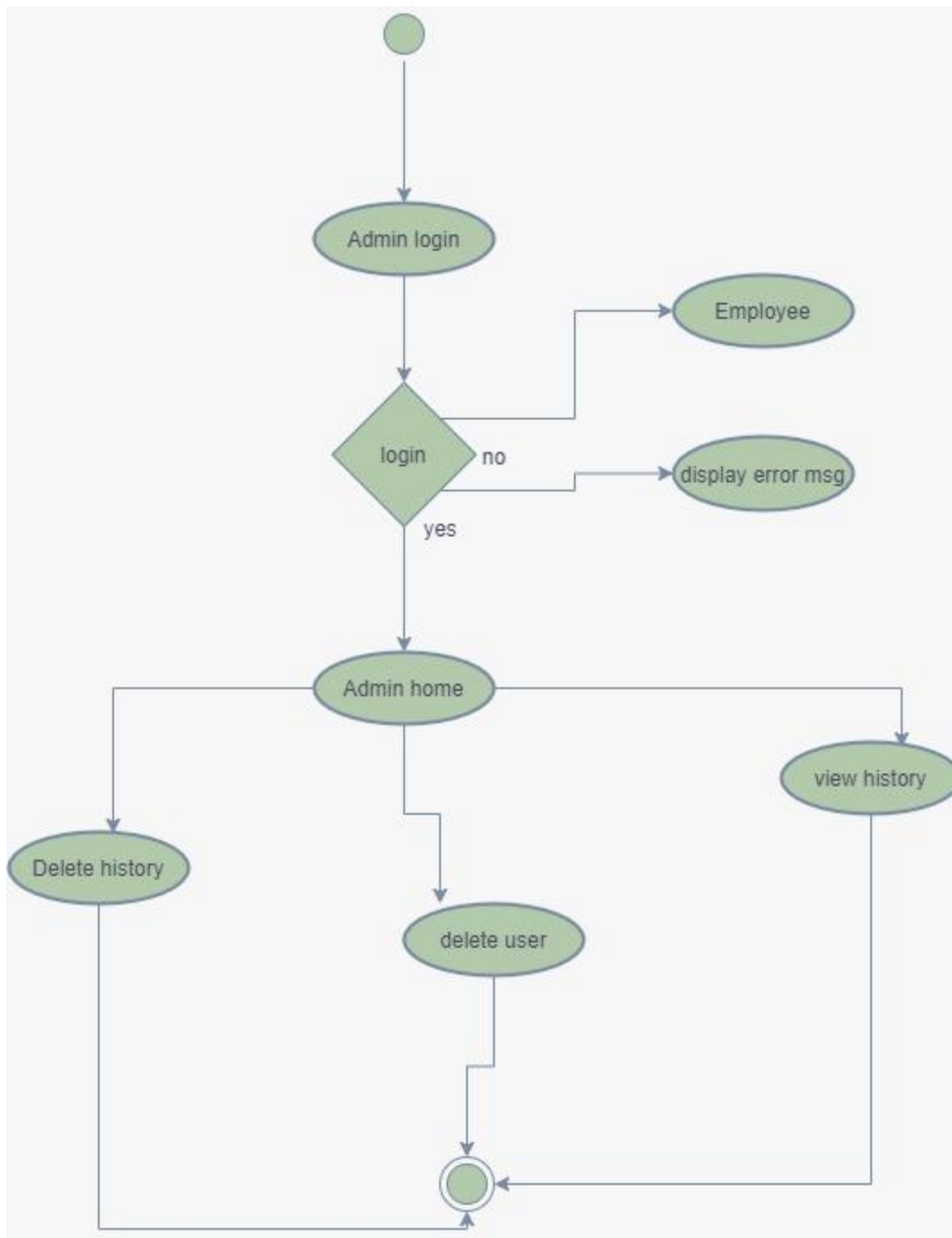
User

- User login
- User home
- Search parking
- Reserved parking
- Pay dues

Admin

- login
- Admin home
- View history
- Delete history
- Delete user/update
- Manage payment





4.9. Component Diagram

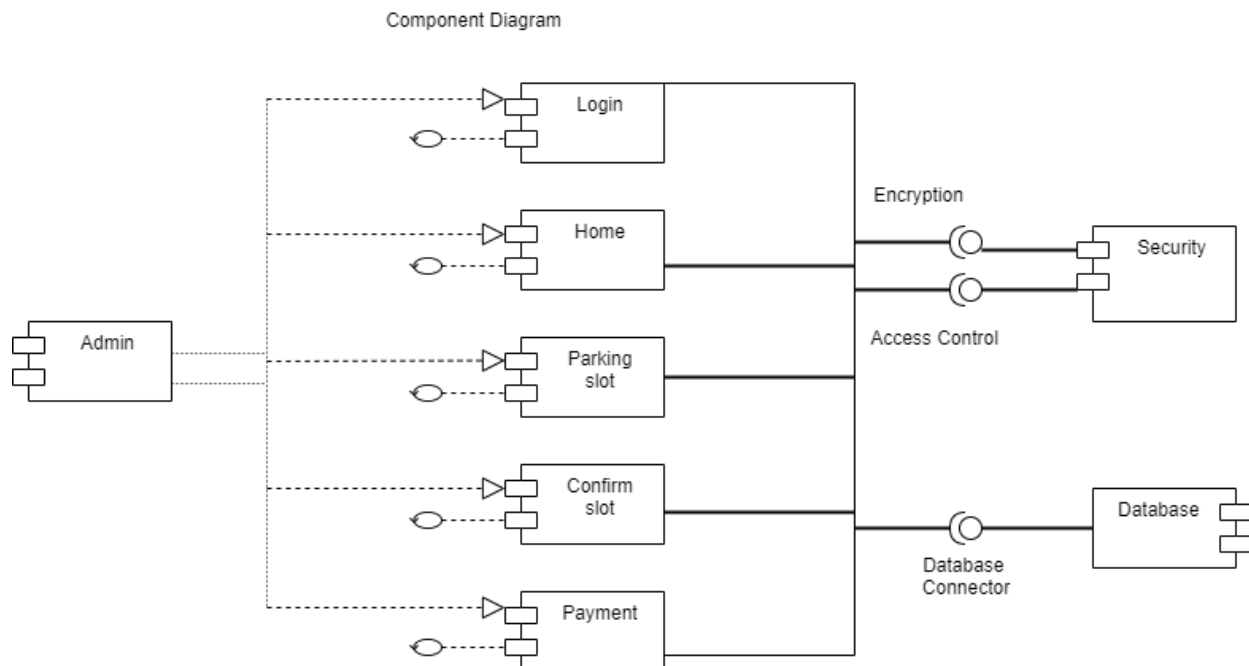
This is the component diagram of smart parking system which shows the components, provided and required interfaces, ports, and relationship between the parking slots, confirm slots and payment. This type of diagrams is used in component based development to describe system. Smart parking app component diagram, describes the physical components in a system.

Components of component diagram

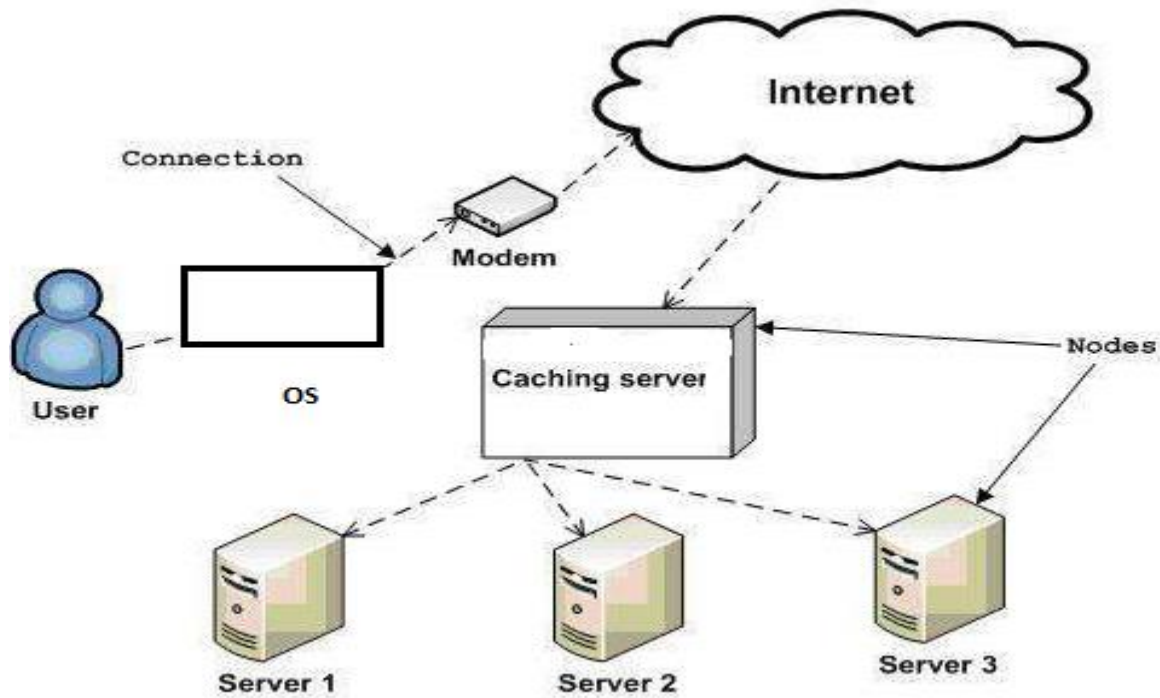
- Login component
- Home component
- Parking slot component
- Confirm slot component
- Payment component

Features of component diagram

- You can show the models the components of smart parking app.
- Model the database schema for smart parking app.
- Model the security of smart parking app.



4.10. Deployment Diagram



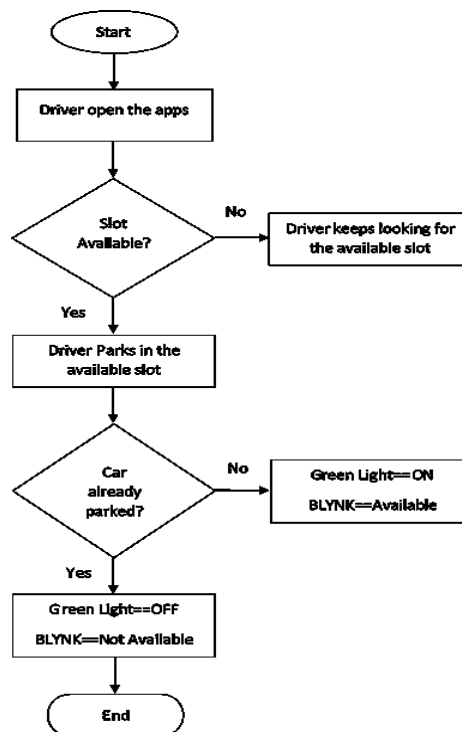
Chapter 5

Implementation

Chapter 5: Implementation

After defining the whole System design we are going to see how we will implement it, which types of tools & techniques that we are using to make it, which APP servers are required for our APPLICATION & which type of standards we should follow while coding.

5.1. Important Flow Control/Pseudo codes



5.2. Components, Libraries, Web Services and stubs

Components:

Domain name: Smart parking App

Stubs:

A method stub or simply stub in software development is a piece of code used to stand in for some other programming functionality. A stub may simulate the behavior of existing code (such as a procedure on a remote machine, such methods are often called mocks) or be a temporary substitute for yet to be developed code.

Libraries:

- RETROFIR(DEFINE OF INTERFACE)
- CHUCK
- TIMBLE
- RX JAVA

Web Services

- .NET
- JAVA

5.3. Deployment Environment

4. IDE: Google chrome, Mozilla Firefox, internet explorer, Microsoft edge
5. Operating system: Window 8, Windows 8.1, Window 10
6. Database: Firebase.

5.1. Tools and Techniques

Development Tools

Web Application

Visual Studio 2013, and Firebase.

Mobile Application

Android studio and firebase.

Software Designing Tool

Microsoft Visio

Testing Tool

Nibbler, Webpage Test, Load Test and Rubidium (Application)..

Fire

Documentation Tools

Microsoft Office.

Techniques

Programming Techniques

Web Application Techniques

JavaScript, JQuery, Firebase

Mobile Application Techniques

Java, Firebase.

Testing Techniques

We will use V model which will cover all these testing technique.

Unit Testing, Integration Testing, System Testing, and Alpha & Beta

Acceptance Testing.

5.2. Best Practices / Coding Standards

Coding standards should be best code should be simple and divided into different chunks that will make ease to modify code. Factorization of code is good habit it make your Application & website light and easily access able.

- Consistent Naming Scheme.
- Dry Principle
- Avoid Deep Nesting
- Limit Line length
- Files and Folder Organizations

- Capitalize SQL Special Words.
- Object Oriented Approach.
- Code Refactoring

Chapter 6

Testing and Evaluation

Chapter 6: Testing and Evaluation

Test & Evaluation (T&E) is the process by which a system or components are compared against requirements and specifications through testing.

6.1. Use Case Testing

User (Login)

Test Case ID	6,1	Test Case Description	Test the login button functionality		
Created By	Asadullah	Reviewed By	Ammarah Amjad	Test Module Name	User (Login)

Tester's Name	Uzma Nisar	Date Tested	23-dec-2021	Test Case (Pass/Fail/Not Executed)	Pass
----------------------	------------	--------------------	-------------	---	------

S #	Prerequisites:
1	Access to User Profile
2	
3	
4	

S #	Test Data
1	User id = asadullahdpr1@gmail.com
2	Password = Asad12345
3	
4	

Test Scenario Verify, the user can login the system or not

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to login page	Login page should open	As Expected	Pass
2	Enter User id & Password	Credential can be entered	As Expected	Pass
3	Click Submit	Customer is logged in	As Expected	Pass

User (Signup)

Test Case ID	6,2	Test Case Description	Test the signup button is working properly or not		
Created By	Asadullah	Reviewed By	Uzma Nisar	Test Module Name	User (Sign up)

Tester's Name	Ammarah Amjad	Date Tested	11-nov-2021	Test Case (Pass/Fail/Not Executed)	Pass
----------------------	---------------	--------------------	-------------	---	------

S #	Prerequisites:
1	User must enter required data
2	
3	
4	

S #	Test Data
1	User id
2	Password
3	
4	

Test Scenario

Verify, the user can signup the system or not

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to signup page	Signup page should open	As Expected	Pass
2	Enter User id & Password	Credential can be entered	As Expected	Pass
3	Click Submit	Customer is registered	As Expected	Pass

6.2. Equivalence partitioning

Username is alphabetic	valid
Username is not alphabetic	invalid
Email without '@' and '.com'	invalid
Email with '@' and '.com'	valid

6.3. Boundary value analysis

		Partition 1	Partition 2	Partition 3
1.	Password	Less than 8 character	1 – 7 character	< 12
2.	Email	Without @ and .com	With @ and .com	Start with numeric

6.4. Data flow testing

There is no conflict against any initialized variable in code. In data flow testing we tested the flow of data in program. Moreover, we tested all the relationships between different entities and variables.

6.5. Unit testing

We have checked our application on multiple systems and the result is very good . Application is very responsive on every computer.

6.6. Integration testing

We have tested each and every module we integrated in the system like easy paisa, jazz cash and map rooting as well. Easy paisa and jazz cash are used for payments on each reservation and map rooting is used to allocate targeted parking place. Every module is working properly and functional as well under all situations.

6.7. Performance testing

If we talk about the performance of our application it is good and very responsive as well . Response rate per second is less than 2 seconds on each and there is no code error in whole source code.

6.8. Stress Testing

Let's say if 100 at the same time along with the same food level which increases the stress on the website at a maximum load how will the website will perform under these circumstances.

Chapter 7

Summary, Conclusion and Future Enhancements

Chapter 7:

Summary, Conclusion & Future Enhancements

7.1. Project Summary

Smart parking project is completed so that I can explain the purpose of the app it is basically a mobile app. The main purpose of this app is providing a convenient parking to user and they can easily search, select and pay dues through smart parking app. Our software would generate the interface and its associated functionalities according to the requirement of organization. This system will provide many features in a single platform. That said, there is always room for improvement.

7.2. Achievements and Improvements

The biggest achievements here are that we were able to enhance our skills to the professional extend that we learned in four years of studying and apply it to this project. We learned software architecture design techniques, UML modeling, project management, testing and much more, and were able to apply it all in this project.

The next big achievement is the things we learnt during this project. New languages, frameworks, libraries, different software's for diagrams Database. All that will be useful for us in our futures. We learn Creative thinking

7.3. Critical Review

The critical element of our system is: Is our system is fast moving, give user friendly interface, not very costly, fulfill the specific requirements of the users at a runtime, has all the features which any organizations can require. To manage all these aspects it took lot of time and hard work.

7.4. Lessons Learnt

We absorb very much from this project. This project sharpens our skills in xml and android studio many other tools and many management concepts as well as how to deal with a problem and how to stick for finding the solution of any problem until you find. As well as technical skills this project also enhances our personal development skills such as team working, dedication.

7.5. Future Enhancements/Recommendations

As it has been already said, there is always room for further improvement. And since we plan to launch this website as our own startup, enhancements will keep coming.

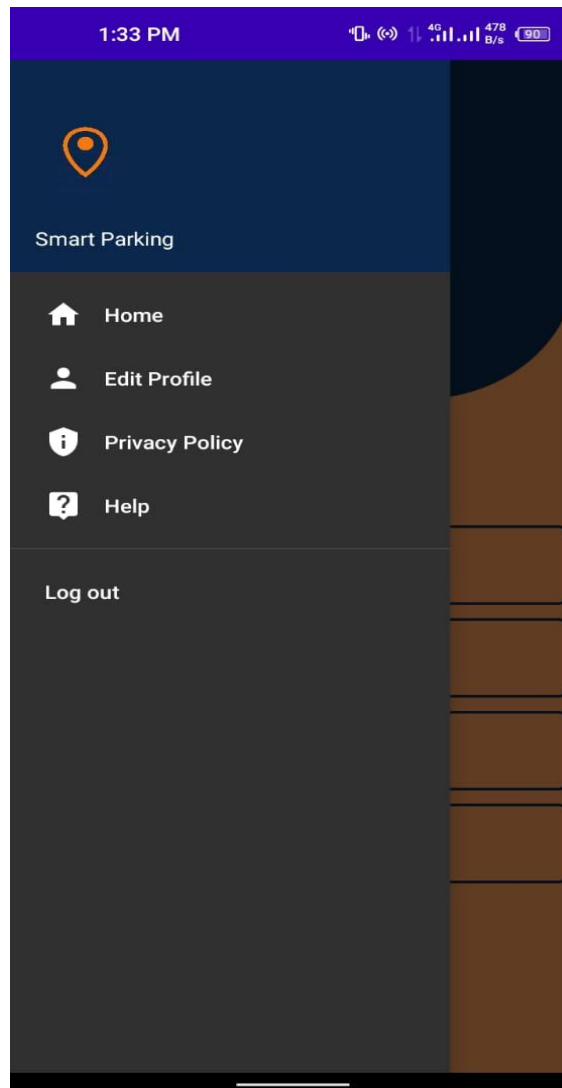
The system has been built on react native, but at some point we work on its efficiency, accessibility, flexibility, we are definitely going to scale it up which we believe ours is eventually going to become. We make it more user friendly interface.

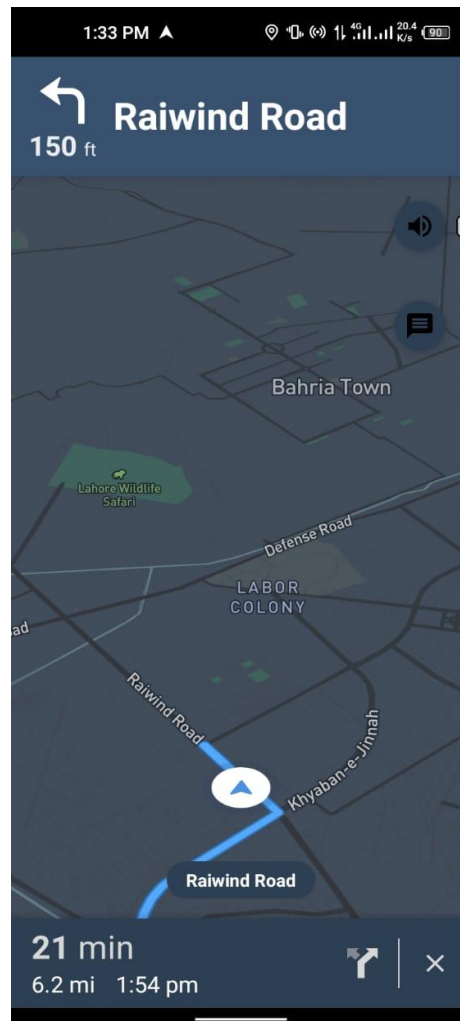
Appendices

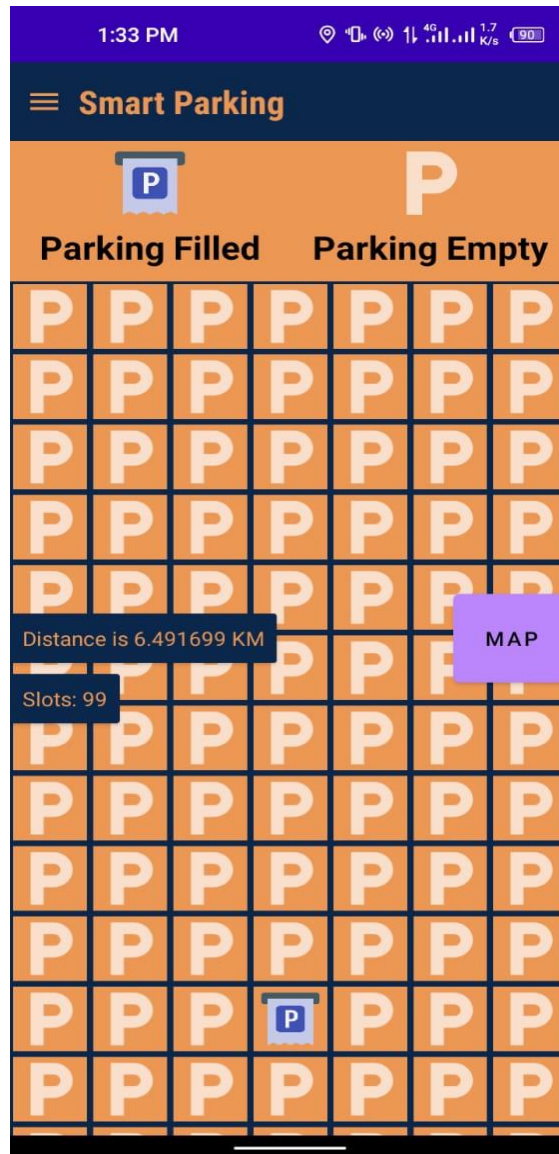
Appendix A: Information / Promotional Material

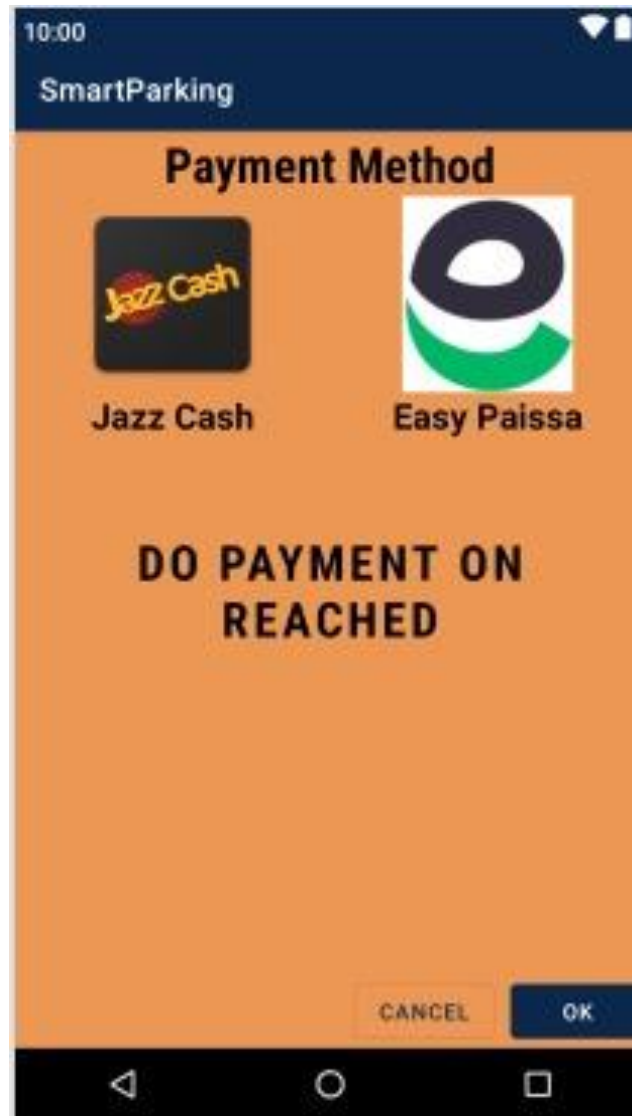
The appendix will describe the user related functionalities and user interface as well.

A.1.Apps ScreenShots









A.3.Standee



Reference and Bibliography

Reference and Bibliography

- [1] www.apsinvo.com
- [2] <https://www.smartparking.com/smartpark-system/smart-app>
- [3] https://www.temjournal.com/content/94/TEMJournalNovember2020_1357_1363.pdf
- [4] https://www.researchgate.net/publication/308186280_Smart_Parking_Application