Share Parking Space

Sana Ishfaq

Muqaddas Hameed

DEPARTMENT OF COMPUTER SCIENCES
COMSATS UNIVERSITY ISLAMABAD,
ATTOCK CAMPUS – PAKISTAN

SESSION 2017-2021

Share Parking Space

Undertaken By:

SANA ISHFAQ
CIIT/FA17-BCS-077/ATK

MUQADDAS HAMEED
CIIT/FA17-BCS-053/ATK

Supervised By:

MUHAMMAD JAMAL AHMED

A DISSERTATION SUBMITTED AS A PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF BACHELOR OF SCIENCE IN COMPUTER SCIENCE

DEPARTMENT OF COMPUTER SCIENCES
COMSATS UNIVERSITY ISLAMABAD,
ATTOCK CAMPUS – PAKISTAN

SESSION 2017-2021

UNDERTAKEN

We certify that this is our own work. The work has not, in whole or in part, been presented elsewhere for assessment. Where material has been used from other sources it has been properly acknowledged. If this statement is untrue, we acknowledge that we will have committed an assessment offence and shall be liable to punishable action under the plagiarism rules of HEC.

____
SANA ISHFAQ
FA17-BCS-077

____
MUQADDAS HAMEED
FA17-BCS-053

Dated: _________

Dated: _________

FINAL APPROVAL

Certified that we have read this project report submitted by Sana Ishfaq and Muqaddas Hameed and it is, in our judgment, of sufficient standard to warrant its acceptance by Department of Computer Science, Comsats University Islamabad, Attock Campus, for the BS in Computer Science.

Committee:

1. External Examiner ______________________________

(Examiner Name)
Designation
University Name

2. Supervisor ______________________________

(Muhammad Jamal Ahmed)

3. Chairperson ______________________________

(Chairperson Name)

4. Dean/Director ______________________________

(Dean/Director Name)

DEDICATION

To my Loving Parents and Teachers

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 MAM TAHIRA SADAF. Without her personal supervision, advice and valuable guidance, completion of this project would have been doubtful. We are deeply indebted to them for her encouragement and continual help during this work.

We are grateful and thankful to our parents, who're always being supportive to us, helped and kept us motivated throughout the journey from then to now with this project. Not only that but they also gave us moral support and financial help. In true meanings, we would have not made till now without them. We are truly thankful to our whole family for being supportive and helping us in every aspect to be mentioned.

SANA ISHFAQ MUQADDAS HAMEED

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

PROJECT BRIEF

PROJECT NAME	SHARE PARKING SPACE
ORGANIZATION NAME	COMSATS ATTOCK
OBJECTIVE	TO LEARN ANDRIOD STUDIO
UNDERTAKEN BY	SANA ISHFAQ (FA17-BCS-077) MUQADDAS HAMEED (FA17-BCS-053)
SUPERVISED BY	SIR M. JAMAL AHMED LECTURER COMPUTER SCIENCE COMSATS ATTOCK
STARTED ON	07 NOVEMBER 2020
COMPLETED ON	09 JULY 2021
COMPUTER USED	CPU:CORE(TM) m3-7Y30

SOURCE LANGUAGE	JAVA
OPERATING SYSTEM	ANDROID
TOOLS USED	ANDRIOD STUDIO & SDK | JUPYTER | FIREBASE | Star UML

ABSTRACT

Nowadays, especially in big cities, one of the biggest problems is finding a parking space. The parking lot parked during the busy time is completely crowded and crowded which makes it very confusing for the car owner to look for his car in the parking lot. To overcome this difficulty, it is important to develop and improve the current parking system.

Very few people rely on public transport these days and as the population grows, the demand for vehicles also increases. Traffic collisions, parking, billing and most importantly the safety of a person's car are some of the most stressful things that passengers face every day. Due to the large number of vehicles and restricted areas, which often have unplanned parking spaces, people are forced to look for safer parking spaces. This leads to depression, anxiety and a lot of wasted time. Parking is currently hand-operated with several security and operational issues. The system depends on the quality of the staff which is often less skilled and has a high level of inefficiency.

So we came up with a solution. Everyone uses smart phones. Our idea is to install another component with Smart phones. This android application focuses on an effective parking system that will help you to find nearest parking spot and also view whether space is available or not. If a reservation is available, then he can book it for a period of time. Smart parking will allow car parking, monitoring and management to be done more efficiently in real time which can also help to reduce environmental pollution. This android application manages parking system in non-commercial(homes) areas. The user will have a nice and easy interface for parking reservation.

In addition, the app will have various AI-based features such as the Recommendation system, live tracking location etc.

TABLE OF CONTENTS

List of Figures

List of Tables

Chapter 1

Introduction

1.1 INTRODUCTION

Share Parking Space is an android application. Its aim is to utilize vacant places for parking hence to increase parking spaces. It helps user to book the nearest parking spaces of their area. When user search for the parking space, application recommend the nearest parking spaces to the user on his/her map of specified city with different prices and rating so that user can book parking space of his/her own choice. Another feature is the rental system. If someone parks his/her vehicle for a certain time, then another person can use their vehicle for that time and the buyer will pay the amount according to that certain time. So, both sides will get benefit from this application. Recommendation system is used. If user search for the parking place, this application recommends the nearest parking places and recently visited places by him/her.

1.1.1 Brief

Our main aim is to design the application through which parking places increases and people book parking places online. The methodology we follow in our project is iterative. As in Iterative method Software is built in Iterations and every module is examined in repeated cycles. In every cycle we can easily add, remove and update features for our project. It will also help us in future work to add more feature in our Project.

Main tool used in our project is Android studio. It is IDE used for computer programming languages. The purpose for Android is to speed up your progress and help you build the highest quality apps for every Android device. Secondly, we used Star UML for diagrams. It is software modeling tool used to draw different diagrams. It is used to illustrate different approaches and framework used in project. Diagrams of our project like use case, activity diagram, sequence diagram are created with help of this tool.

Python is also used for recommendation system. For implementation Jupyter is used. The architecture of Jupyter is language independent. The decoupling between the client and kernel makes it possible to write kernels in any language. Jupyter brings a lightweight interface for kernel languages that can be wrapped in Python.

1.2 RELEVANCE TO COURSE MODULES

Course Name	Course Code	Details
Mobile Application Development(MAD)	CSC445	In this subject we studied IDE Android Studio for development of static and dynamic application that will helps us during design of our Application.
Object Oriented Programming (OOP)	CSC241	In this subject we have studied the concepts of java that we using in the implementation of design in Android Studio.
Database Systems (DBS)	CSC271	In this subject we have studied how to maintain record in a database that helps us during maintaining database if our application.

Table 1.1 Course Relevance.

1.3 PROJECT BACKGROUND

Nowadays, especially in big cities, one of the biggest problems is finding a place to park. Timed parking busy hours are completely filled and congested causing great confusion for the car owner to demand his car in parking lot. To overcome this difficulty, it is important to develop and improve the current parking system. Traffic congestion is one of the most common problems in developing lands. As a result of the road people lose their value

time in their busy schedule. One of the main reasons for traffic congestion is parking on the side of the road we need to upgrade to improve the parking system to reduce traffic congestion. In our application we try to resolve all these problems. Through the application parking places increases and people could easily find nearest parking places to their area.

1.2 LITERATURE REVIEW

In this section we are going to review and study Knowledge of already developed system that are related to

Smart Parking:

The Smart Parking app helps to find parking space using live information from the Smart Parking sensor system installed in the parking areas around the city. The Smart Parking app reads your GPS-based location and directs you to your chosen location.

Parkopedia Parking:

The name Parkopedia is a combination of parking words and encyclopaedia. Following are the features of the app are

Find a parking space using your location or by entering an address

* Get directions directly to space or entry

* See real-time parking available where available (Premium upgrade required)

* Find opening hours, current prices, payment methods and more

* Quickly reduce your choice of parking using filters such as street parking only, free of charge, accepted credit cards, covers, etc.

SMS Parking:

SMS Parking is an application that allows you to pay for public parking easily with features such as locating and sending the appropriate short text message (SMS) to the corresponding service number.

1.5 ANALYSIS FROM LITERATURE REVIEW

Application Name	Features	Yes/No	My Application
	Online Booking	Yes	Yes
	Live Vehicle Tracking by user	No	Yes

Smart Parking	Recommendation system	No	Yes
	Utilization of vacant places for Parking	No	Yes
	Vehicle Rental system	No	Yes
	Cash on Delivery/Online Payment	Yes	Yes
Parkopedia Parking	Online Booking	Yes	Yes
	Live Vehicle Tracking by user	Yes	Yes
	Recommendation system	No	Yes
	Utilization of vacant places for Parking	No	Yes
	Vehicle Rental system	No	Yes
	Cash on Delivery/Online Payment	Yes	Yes
SMS Parking:	Online Booking	Yes	Yes
	Live Vehicle Tracking by user	No	Yes
	Utilization of vacant places for Parking	No	Yes
	Vehicle Rental system	No	Yes
	Cash on Delivery/Online Payment	Yes	Yes

Table 1.2 Iterative Methodology.

1.6 METHODOLOGY & SOFTWARE LIFECYCLE

We are using Iterative Methodology in our Application (Share Parking Space). As in Iterative Method Software is built in Iterations and every module is examined in repeated cycles. In every cycle we can easily add, remove and update features for our project. It will also help us in future work to add more feature in our Project.

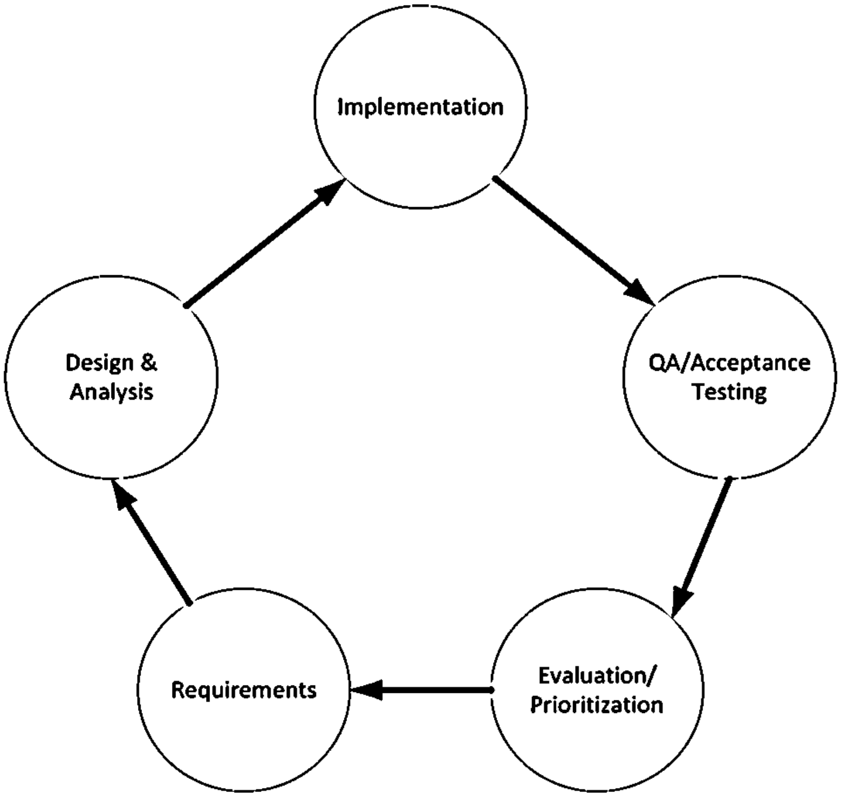


Figure 1.1 Iterative Methodology

1.7 RATIONALE BEHIND SELECTED METHODOLOGY

We have selected Iterative Approach for our project. As in iterative approach every module is explained in a detailed way and in repeated cycles. If requirements change during development, then we can easily handle these changes in next iteration. It will also help us in future work to add more features in a project.

Chapter 2

PROBLEM DEFINITION

2.1 PROBLEM STATEMENT

As the population increases, the demand for the vehicles also increases. Now a days very few people rely on public transport. Due to increase in vehicles traffic jam is one of the common problem in the developing countries. One of the prime reason of traffic jam is parking vehicles on the road side so we need to evolve to develop a parking system. If anyone wants to park his/her vehicle, most of the time wasted in search of parking spaces. Especially, reserved parking area in commercial areas are full in busy hours. This leads to the stress, anxiety and a wastage of time

Q1: How to know about different parking places of your area?

Q2: How to help Users to Book parking places online and at reasonable prices?

2.2 DELIVERABLE AND DEVELOPMENT REQUIREMENTS

Main Deliverables of our project are:

Sr. No	Deliverable Names	Description	Owner	Status
1-	Plan Project Plan	Identify the milestones and timeline of project.	CIIT/ATK	Complete
2-	Recommendation system	Recommend nearest and recently visited parking places.	CIIT/ATK	In Progress
3-	User Panel	Online Booking, Online payment, put car on rent.	CIIT/ATK	In Progress
4-	Service Provider Panel	Provide vacant places for parking.	CIIT/ATK	In Progress
5-	Customer	Wants vehicle on rent.	CIIT/ATK	In Progress

Table 2.1 Deliverable Status.

Development requirements include tools and technologies that we are using in our system.

- Java Programming Language
- Android Studio & SDK
- Firebase
- Star UML
- Google Maps
- Flask Framework
- Python
- Unsupervised Machine Learning Algorithm.

Chapter 3

REQUIREMENT ANALYSIS

3 REQUIREMENT ANALYSIS

The System Requirement Specification is the official statement for any system to get Functional, Non-Functional and Operational Requirements. During the process of development of any System, it serves as a contract between the developer and the customer. The developer would be capable to build the system based on specified requirements as the enlists and necessary requirements specification for the system development is enough. SRS dependence based on functional and Non-functional requirements and inclusion of set of use cases that describe interactions between user that software must provide. Software requirements specification establishes the relation between contractors or suppliers and customers on what the software product supposed to do as well as not expected to do.

3.1 USE CASE DIAGRAM(S)

The system's use case shows the user a full view of the scheme and how actors would interact with the system. It is a graphical depiction of the interactions among the elements of the system.

User Use case:

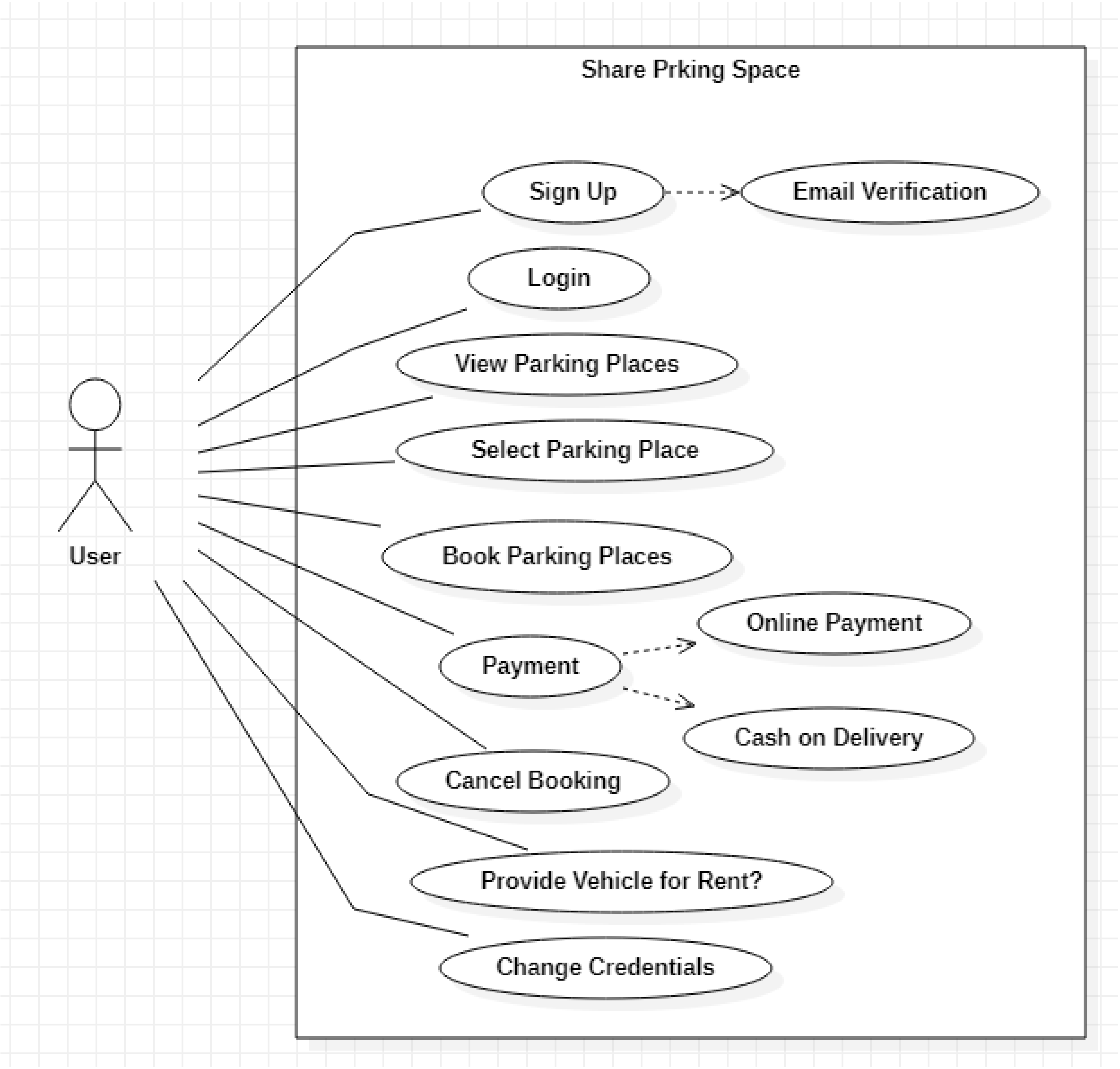


Figure 3.1 User Use case Diagram

Detailed Use Case:

3.1.1 Sign_Up

Use Case Name	Sign_Up
Actor	User
Description	To make his/her account in the application user have to registered by putting required information.

Pre-condition	User have to install the Application.
Post-condition	User have account in the application and he/she is capable to search parking places and book parking lots.

Table 3.1.1 Sign Up

3.1.2 Email Verification

Use Case Name	Email Verification
Actor	User
Description	User should verify account through his/her email.
Pre-condition	User wants to register in the application
Post-condition	User have verified account in the application.

Table 3.1.2 Email Verification

3.1.3 Login

Use Case Name	Login
Actor	User
Description	User should login with the email and password to interact with the application.
Pre-condition	User have a registered account in the application.
Post-condition	User can interact with the application and perform actions like search parking places, make bookings etc.

Table 3.1.3 Login

3.1.4 View Parking Places

Use Case Name	View Parking Places
Actor	User
Description	User can view the parking places.
Pre-condition	User have account in the application.
Post-condition	User can request for the booking/reservation.

Table 3.1.4 View Parking Places

3.1.5 Select Parking Place

Use Case Name	Select Parking Place
Actor	User
Description	User can select the parking place nearest to its destination.
Pre-condition	User have a list of parking lots at his/her near places with specialty and price.
Post-condition	User can view the service provider profile to know about his/her rating.

Table 3.1.5 Select Parking Place

3.1.6 Book Parking Place

Use Case Name	Book Parking Place
Actor	User
Description	User can view available parking places and then book parking place.
Pre-condition	User should book the parking place.
Post-condition	User should wait for the approval of request for his reservation/booking.

Table 3.1.6 Book Parking Place

3.1.7 Online Payment

Use Case Name	Online Payment
Actor	User
Description	User can select online payment option to pay online after reserving parking lot/place.
Pre-condition	User wants to get online payment service.
Post-condition	User pay for booking online.

Table 3.1.7 Online Payment

3.1.8 Cash on Delivery

Use Case Name	Cash on Delivery
Actor	User
Description	User select cash on delivery option to get cash on delivery service through application.
Pre-condition	User wants to get cash on delivery service.
Post-condition	User pay at the time of booking.

Table 3.1.8 Cash on Delivery

3.1.9 Cancel Booking

.

Use Case Name	Cancel Booking
Actor	User
Description	User may select the cancel boking option for the cancellation.
Pre-condition	User wants to cancel the booking from the admin.
Post-condition	User booking should be cancelled.

Table 3.1.9 Cancel Booking

3.1.10 Parked Vehicle for Rent?

Use Case Name	Cancel Booking
Actor	User
Description	User may select the cancel boking option for the cancellation.
Pre-condition	User wants to cancel the booking from the admin.
Post-condition	User booking should be cancelled.

Table 3.1.10 Parked Vehicle for Rent?

3.1.11 Change Credentials

Use Case Name	Change Credentials
Actor	User
Description	User can make any change in his/her profile like email and password or a phone number.
Pre-condition	User should select the edit option to update anything in his/her profile.
Post-condition	User should login with updated details.

Table 3.1.11 Change Credentials

Service Provider Use Case:

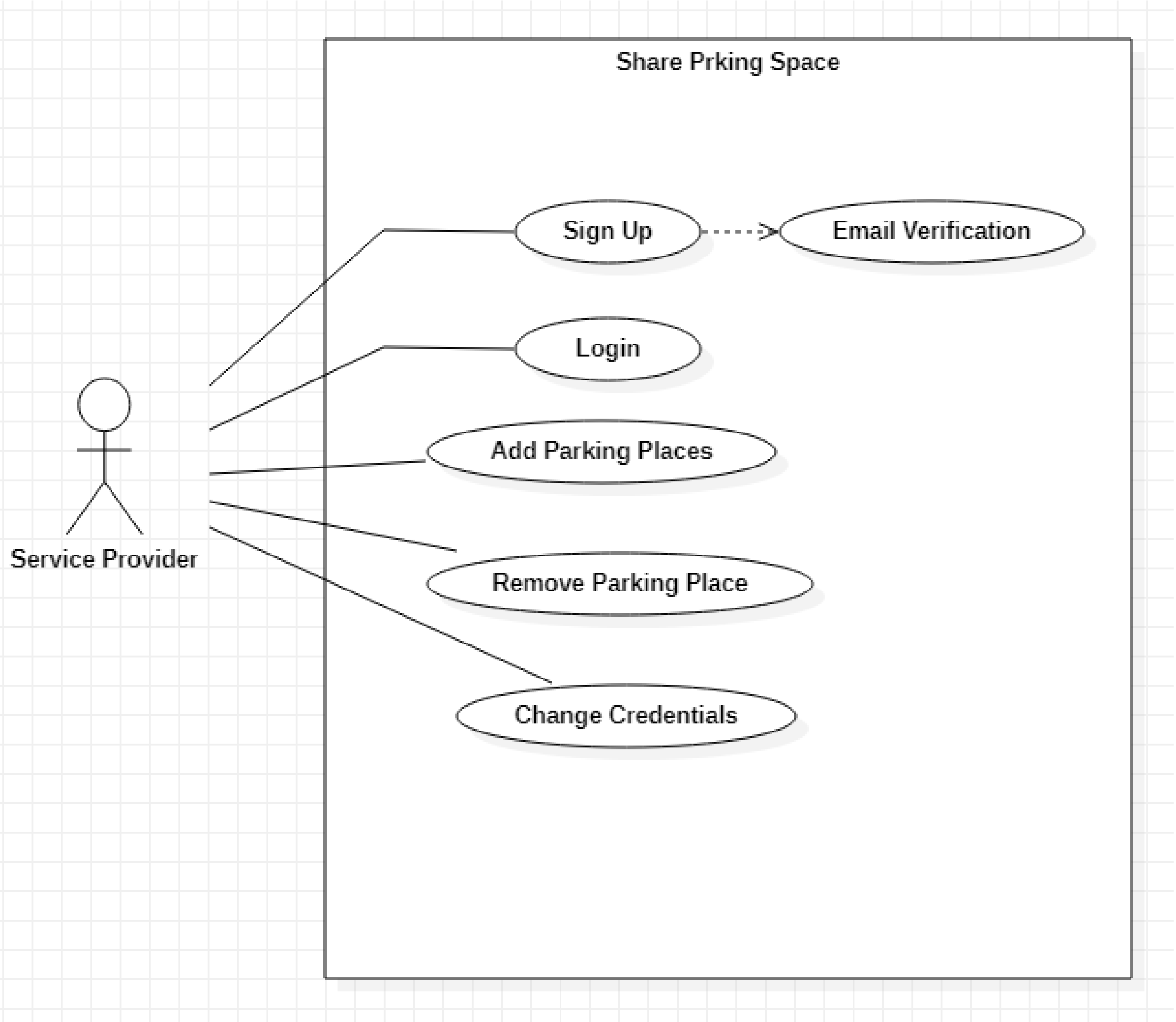


Figure 3.2 Service Provider Use case Diagram

Detailed Use Case:

3.2.1 Sign_Up

Use Case Name	Sign_Up
Actor	Service Provider
Description	To make his/her account in the application user have to registered by putting required information.
Pre-condition	Service Provider have to install the Application.

Post-condition	Service Provider have account in the application and he/she is capable to add and remove parking places.

Table 3.2.1 Sign-up

3.2.2 Email Verification

Use Case Name	Email Verification
Actor	Service Provider
Description	Service Provider should verify account through his/her email.
Pre-condition	Service Provider wants to register in the application
Post-condition	Service Provider have verified account in the application.

Table 3.2.2 Email Verification

3.2.3 Login

Use Case Name	Login
Actor	Service Provider
Description	Service Provider should login with the email and password to interact with the application.
Pre-condition	Service Provider have a registered account in the application.
Post-condition	Service Provider can interact with the application and perform actions like add or remove parking places etc.

Table 3.2.3 Login

3.2.4 Add Parking Places

Use Case Name	Add Parking Place
Actor	Service Provider
Description	Service Provider can add parking places in the application.
Pre-condition	Service Provider must have parking places.
Post-condition	Users can get information whether a parking place available in particular area.

Table 3.2.4 Add Parking Place

3.2.5 Remove Parking places

Use Case Name	Remove Parking Place
Actor	Service Provider
Description	Service Provider can remove parking places in the application.
Pre-condition	Service Provider have not that parking places in their area.
Post-condition	Service Provider should book these parking places.

***Tab*le 3.2.5 Remove Parking Place**

3.2.6 Change Credentials

Use Case Name	Change Credentials
Actor	Service Provider
Description	Service Provider can make any change in their profile like email and password.
Pre-condition	Service Provider should select the edit option to update anything in the profile.
Post-condition	Service Provider should login with updated details.

Table 3.2.6 Change Credentials

Parked Vehicles for Rent:

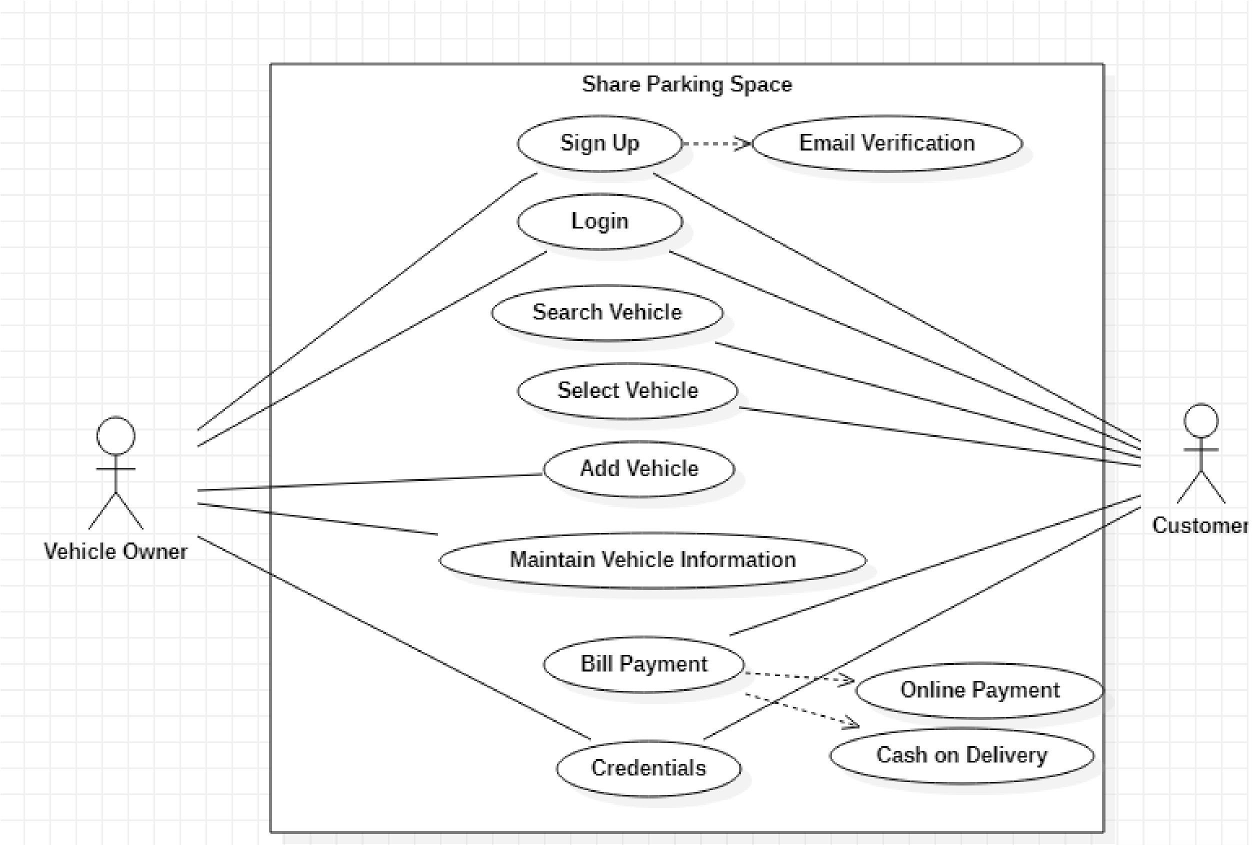


Figure 3.3 Parked vehicles for Rent Use case Diagram

Detailed Use Case:

3.3.1 Sign_Up

Use Case Name	Sign_Up
Actors	Vehicle Owner , Customer
Description	To make his/her account in the application user have to registered by putting required information.
Pre-condition	Vehicle Owner , Customer have to install the Application.
Post-condition	Vehicle Owner, Customer have account in the application and he/she is capable to add vehicles and maintain vehicles information.

Table 3.3.1 Sign-Up

3.3.2 Email Verification

Use Case Name	Email Verification
Actors	Vehicle Owner **,** Customer
Description	Vehicle Owner**,** Customer should verify account through his/her email.
Pre-condition	Vehicle Owner**,** Customer wants to register in the application
Post-condition	Vehicle Owner**,** Customer have verified account in the application.

Table 3.3.2 Email Verification

3.3.3 Login

Use Case Name	Login
Actors	Vehicle Owner**,** Customer
Description	Vehicle Owner**,** Customer should login with the email and password to interact with the application.
Pre-condition	Vehicle Owner**,** Customer have a registered account in the application.
Post-condition	Vehicle Owner**,** Customer can interact with the application and perform actions like adding vehicles, maintain vehicles information etc.

Table 3.3.3 Login

3.3.4 Search Vehicle

Use Case Name	Search vehicle
Actors	Vehicle Owner**,** Customer
Description	Customer should search for the vehicle that he wants for rent.
Pre-condition	User have a list of vehicles at his/her near places.
Post-condition	User can view the details of the vehicle before selection.

Table 3.3.4 Search Vehicle

3.3.5 Select Vehicle

Use Case Name	Select vehicle

Actor	User
Description	User can select the vehicle.
Pre-condition	User can select the vehicle.

Table 3.3.5 Select Vehicle

3.3.6 Add vehicle

Use Case Name	Add Vehicle
Actor	User
Description	User can add his/her vehicles for rent.
Pre-condition	User provide vehicles for rent.
Post-condition	Other users can get information that required vehicle is available or not.

Table 3.3.6 Add Vehicle

3.3.7 Online Payment

Use Case Name	Online Payment
Actor	Customer
Description	Customer can select online payment option to pay online after getting a vehicle on rent.
Pre-condition	Customer wants to get online payment service.
Post-condition	Customer pay for booking online.

Table 3.3.7 Online Payment

3.3.8 Cash on Delivery

Use Case Name	Cash on Delivery
Actor	Customer
Description	Customer select cash on delivery option to get cash on delivery service through application.
Pre-condition	Customer wants to get cash on delivery service.

Post-condition	Customer pay at the time of booking.

Table 3.3.8 Cash on Delivery

3.3.9 Credentials

Use Case Name	Change Credentials
Actors	Vehicle Owner**,** Customer
Description	Vehicle Owner**,** Customer can make any change in their profile like email and password.
Pre-condition	Vehicle Owner**,** Customer should select the edit option to update anything in the profile.
Post-condition	Vehicle Owner**,** Customer should login with updated details.

Table 3.3.9 Credentials

3.4 Functional Requirements:

These are constraining that must be followed during the coding and without satisfying these constrains solution is not valid. Functional requirements describe the expected behavior of the system. These are the core system requirement and without implementing any of these requirements the system is supposed to be incomplete.

3.4.1 Sign Up

Identifier	FR-1
Title	Sign Up
Requirement	User, service provider, customer shall be able to register their accounts in the application.
Source	The Requirement originates from user.
Rationale	Through this requirement user have a registered account in the application and can interact with the application.

Business Rule (if Required)	Any restriction, policy, rule that the particular requirement must be fulfilled through its functional behavior.
Dependencies	FR-2
Priority	High

Table 3.4.1 Sign Up

3.4.2 Login

Identifier	FR-2
Title	Login
Requirement	User, service provider, customer shall be able to interact with the application.
Source	The Requirement originates from user.
Rationale	Through this requirement user can perform his/her required actions.
Business Rule (if Required)	Any restriction, policy, rule that the particular requirement must be fulfilled through its functional behavior.
Dependencies	FR-3, FR-4, FR-5
Priority	High

Table 34.2 Login

3.5 Non-Functional Requirements:

A non-functional requirement shows and specify how the system performs a certain functionality within scope of its requirements specified. In other words, behavior and limit of system based on its functionality describe that how a system will perform.

Non-functional requirements generally specify the system quality attributes or characteristics. Following are some non-functional requirements to ensure our system efficiency.

3.5.1. Security

Name	NF-1: Security
Summary	User, service provider, customer can be logged into the system.

Table 3.5.1 Security

3.5.2. Performance

Name	NF-2: Performance
Summary	Retrieving all the data must be fast. Retrieving and showing records in visualized forms shall not take more than 5 seconds.

Table 3.5.2 Performance

3.5.3. Reliability

Name	NF-3: Reliability
Summary	System should be available for its user 24/7. Fault tolerance is enough to tackle uncertainty. System is design for exception handling to avoid failure.

Table 3.5.3 Reliability

3.5.4. Usability

Name	NF-4: Usability
Summary	Usability is crucial in the Web App as well as in Android Apps. Make sure that the system is user-friendly that everyone can use it. The system shall have an international language with concise statements.

Table 3.5.4 Usability

3.5.5. Scalability

Name	NF-5: Scalability
Summary	Application would be enough capable to perform and handle high users traffic while performing the tasks.

Table 3.5.5 Scalability

3.5.6. Portability

Name	NF-6: Portability
Summary	Website is compatible will all popular browsers.

Table 3.5.6 Portability

Chapter 4

Design and Architecture

4 Design and Architecture:

In this chapter we describe our design that is used in system implementation. Design of system is developed according to the requirements collected in previous section. Result of this activity ends in systems architecture showing components and internal structure of system.

4.1 System Architecture

This phase describes the components of system and connection between them. It is conceptual model of system.

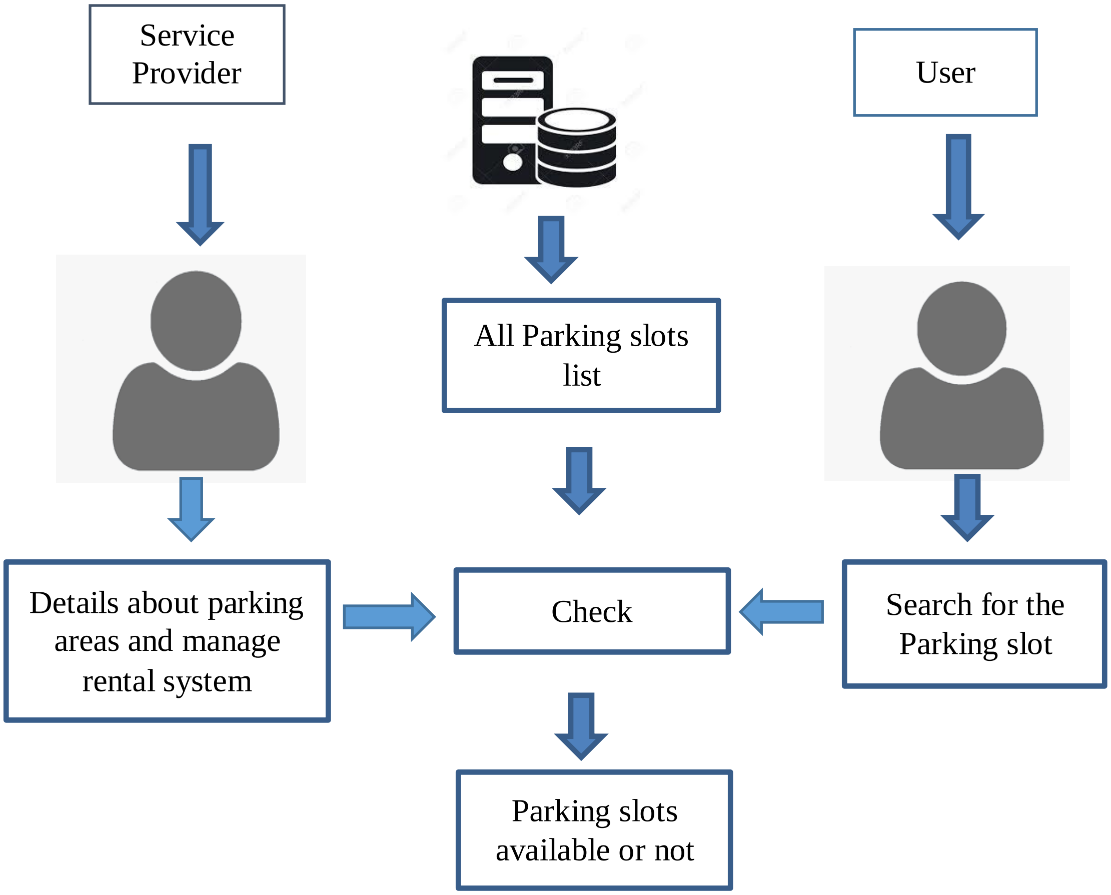

Service Provider
User
All Parking slots list
Details about parking areas and manage rental system
Check
Search for the Parking slot
Parking slots available or not

Figure 4.1 System Architecture

4.2 Data Representation [Diagram + Description]:

Sequence Diagram:

Service Provider:

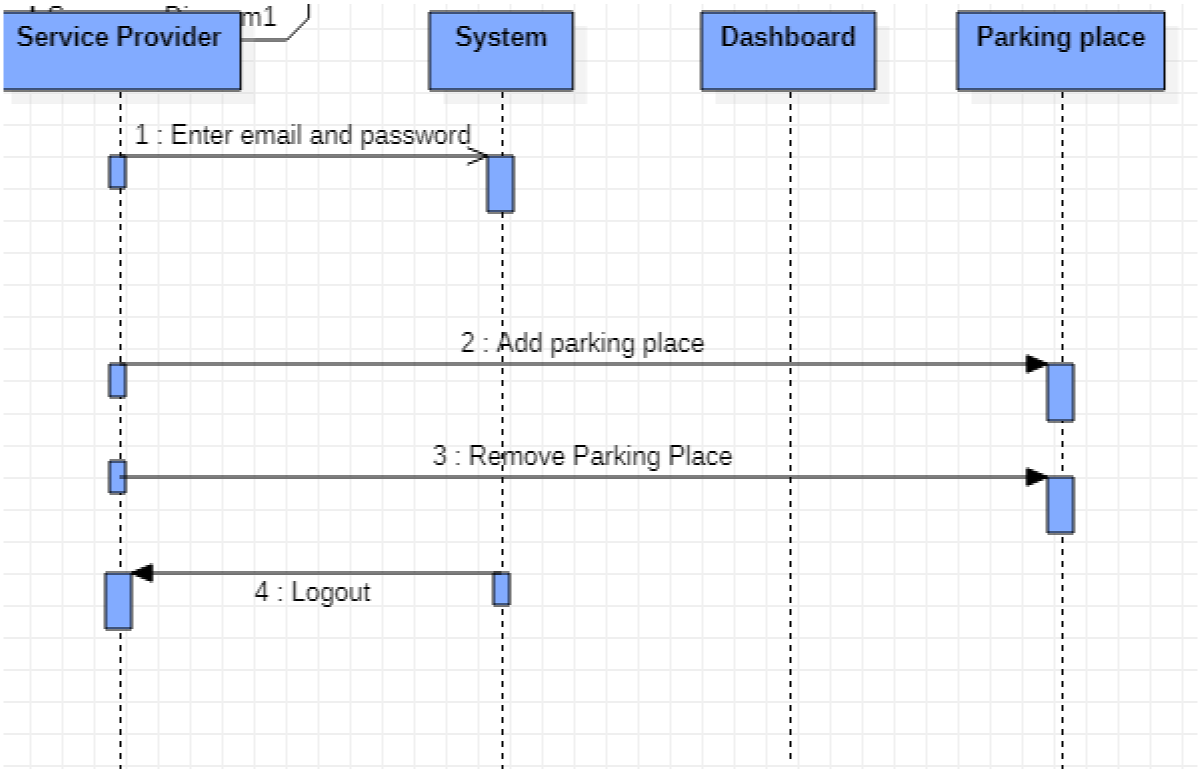


Figure 4.2.1 Service Provider

This sequence diagram shows that service provider login into the system through email and password. Service provide add or remove the parking places.

User:

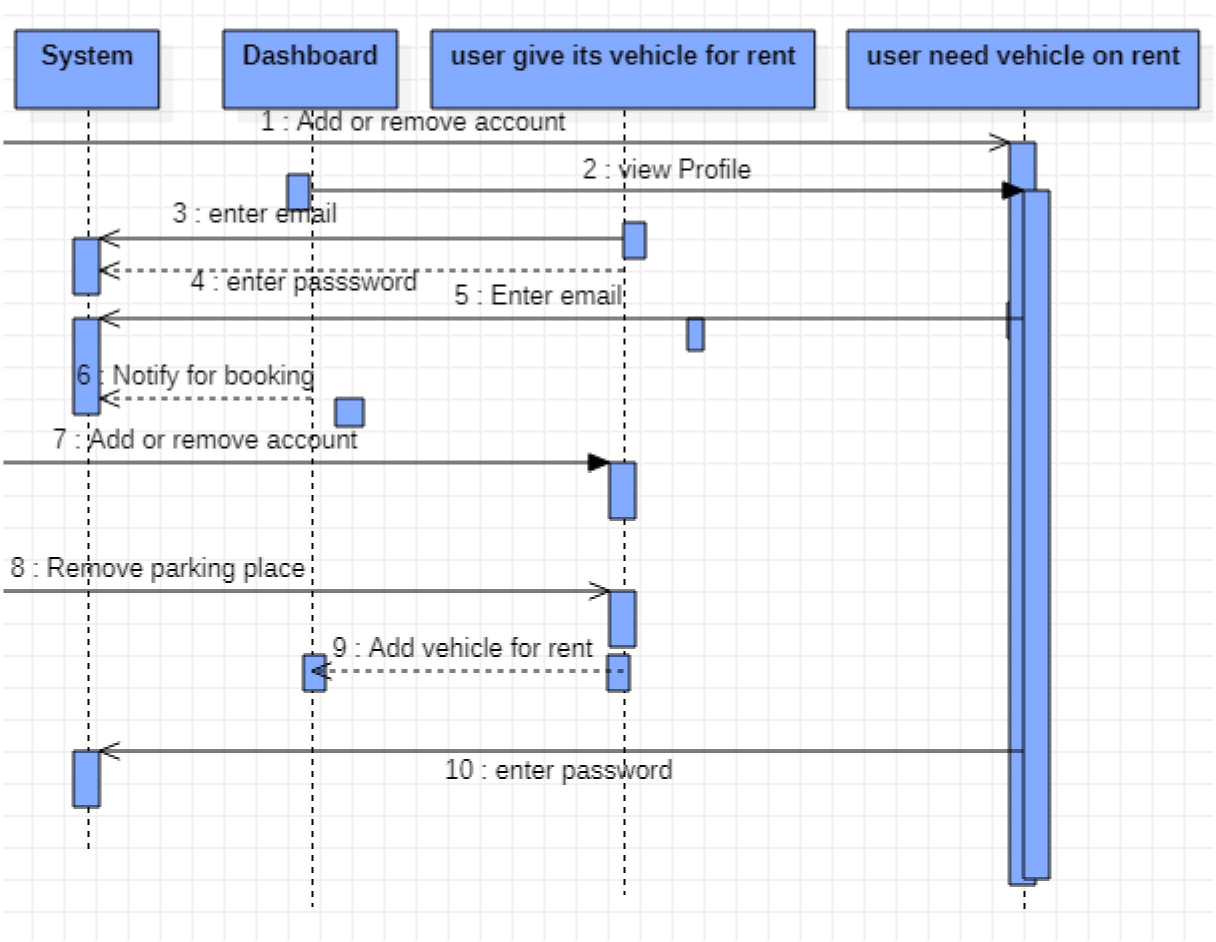


Figure 4.2.2 User

Description:

This sequence diagram shows that Users will login into the system through email and password. Users will get the list of available parking places and vehicles available for rent. User can select the parking place and then book the parking place. Customer search for the vehicle that he need , if available then he/she request for that vehicle on rent .

4.3 Process flow / representation

Activity diagram is used for representing the process flow. In this diagram, we check the graphical representation of process flow. Activity diagram is an important diagram in UML which is used to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity.

Starting symbol	●
Action state	▬
Decision	◆
Data flow	→
Final node	◉

Table 4.3.1 Activity diagram elements

Service Provider:

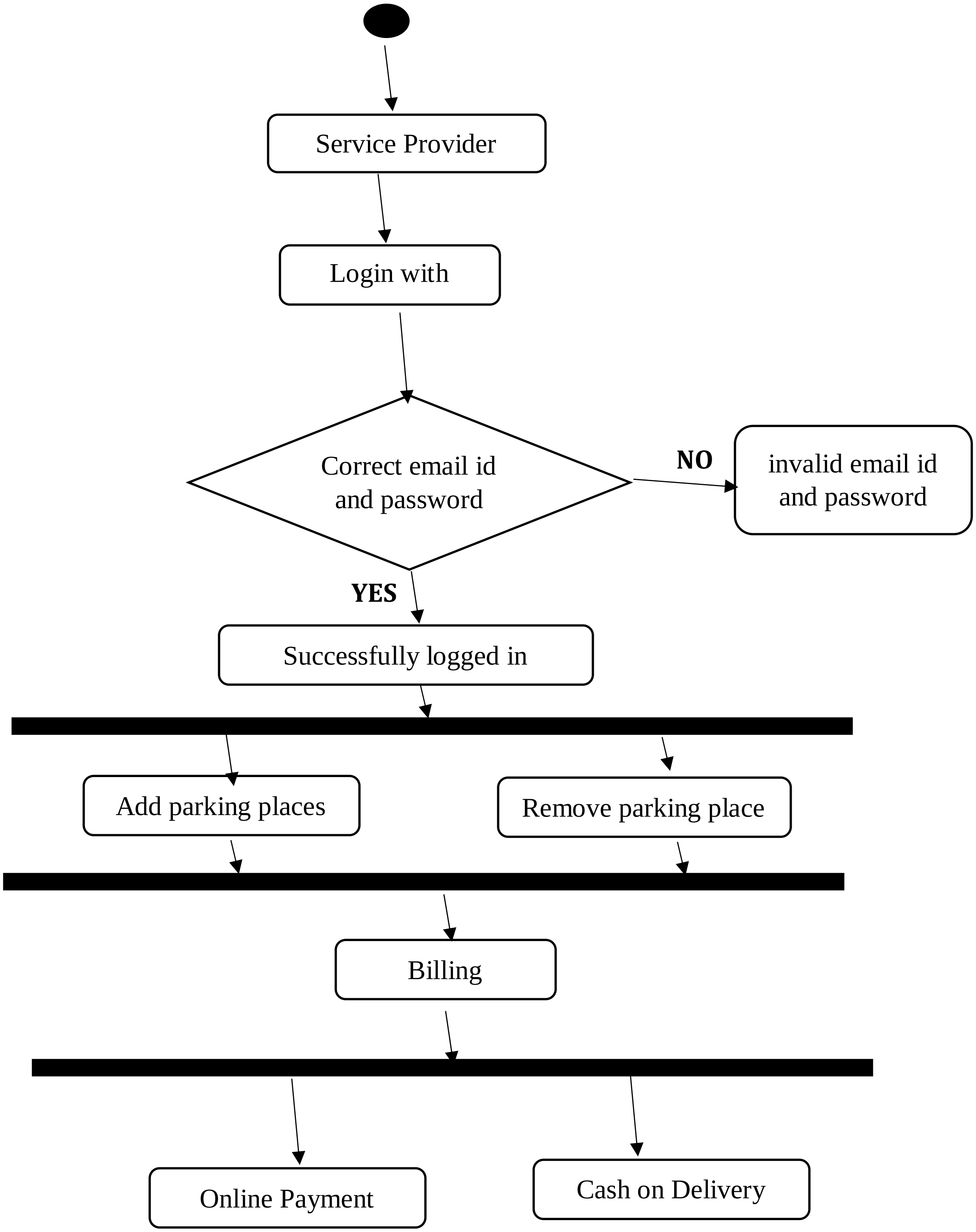
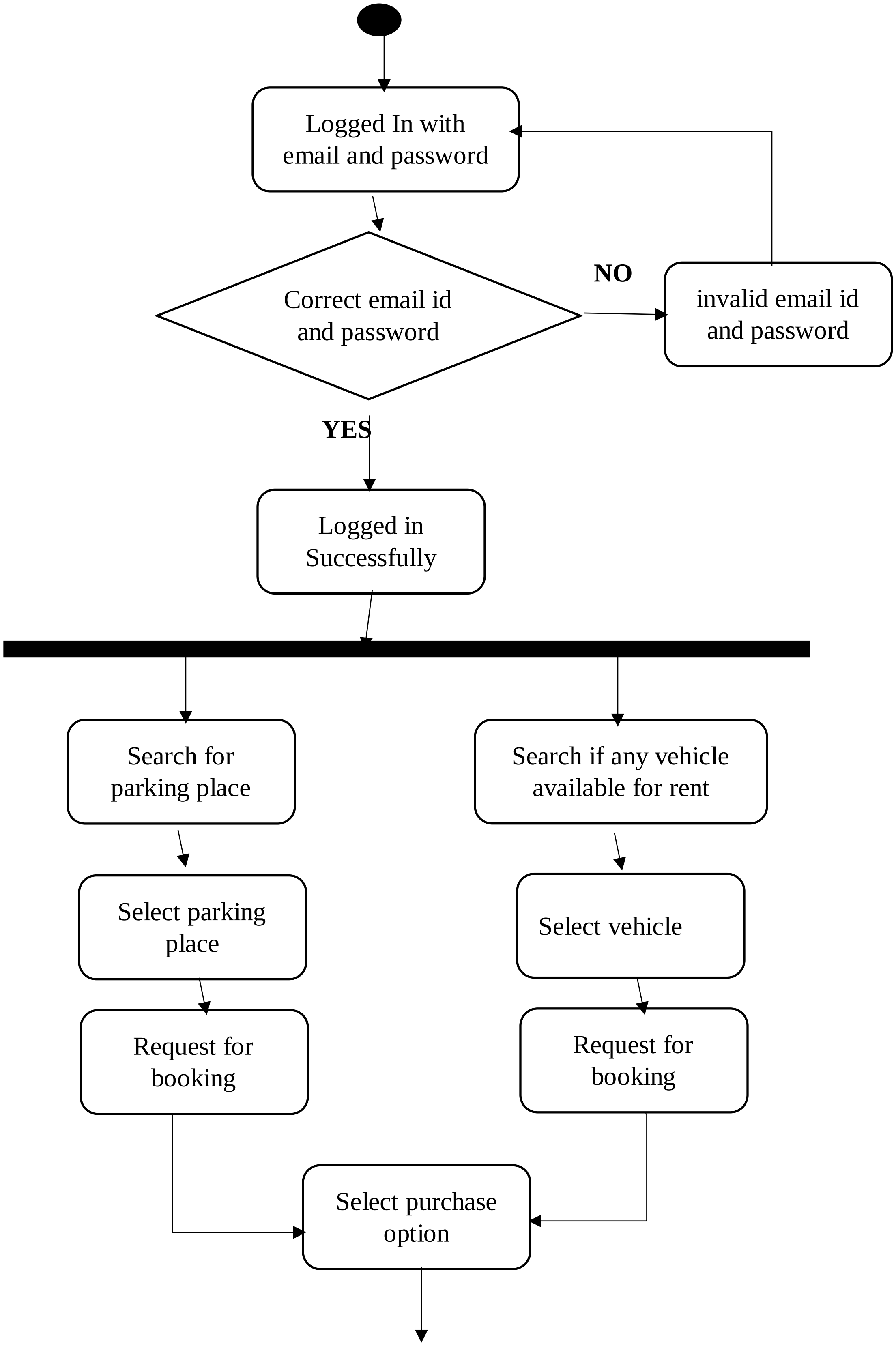


Figure 4.3.1 Service Provider

Users:


Logged In with email and password
Correct email id and password
NO
invalid email id and password
YES
Logged in Successfully
Search for parking place
Search if any vehicle available for rent
Select parking place
Select vehicle
Request for booking
Request for booking
Select purchase option

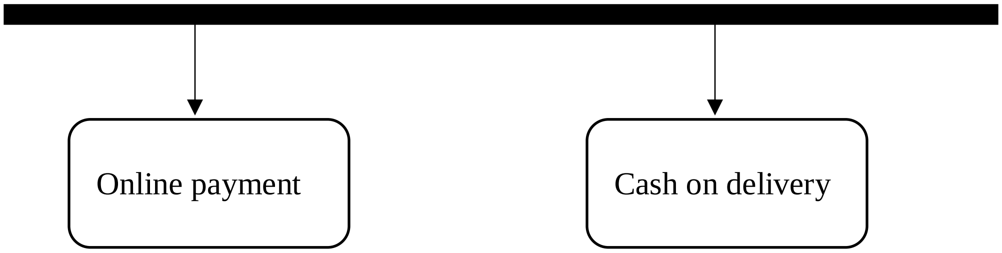


Figure 4.3.2 Users

4.4 Class Diagram

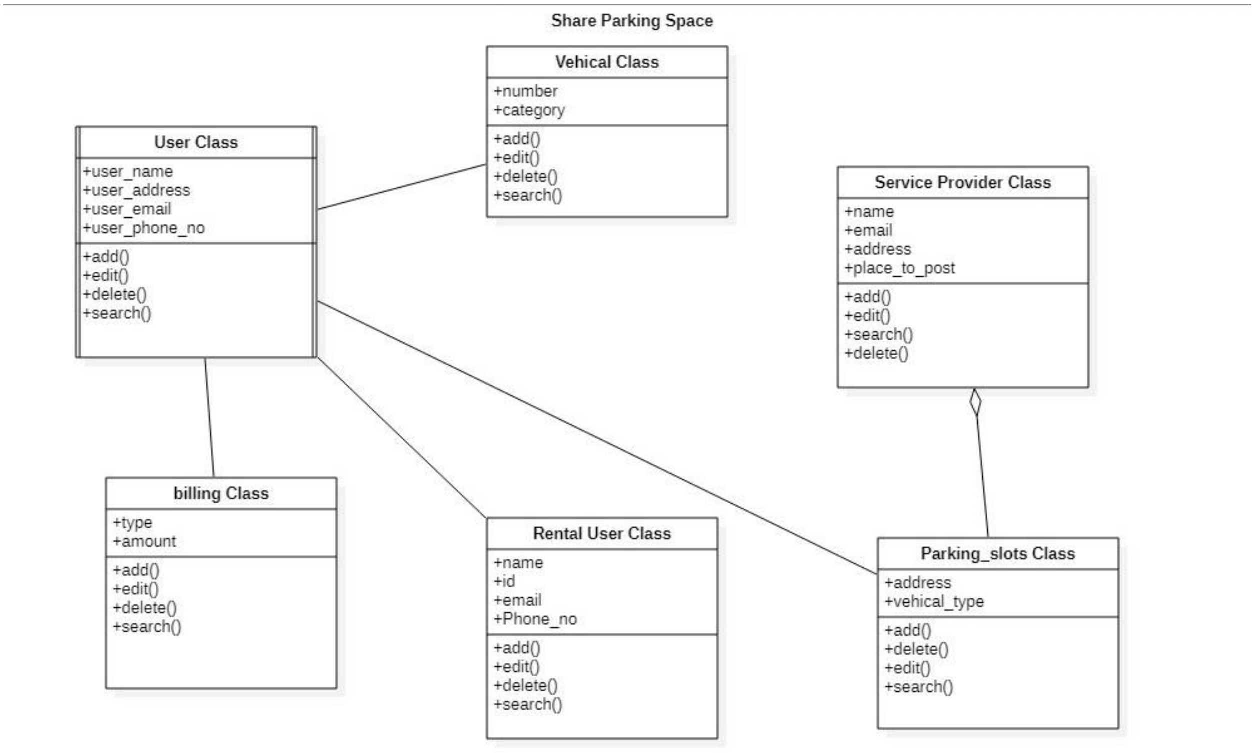


Figure 4.4 Class Diagram

4.5 ERD

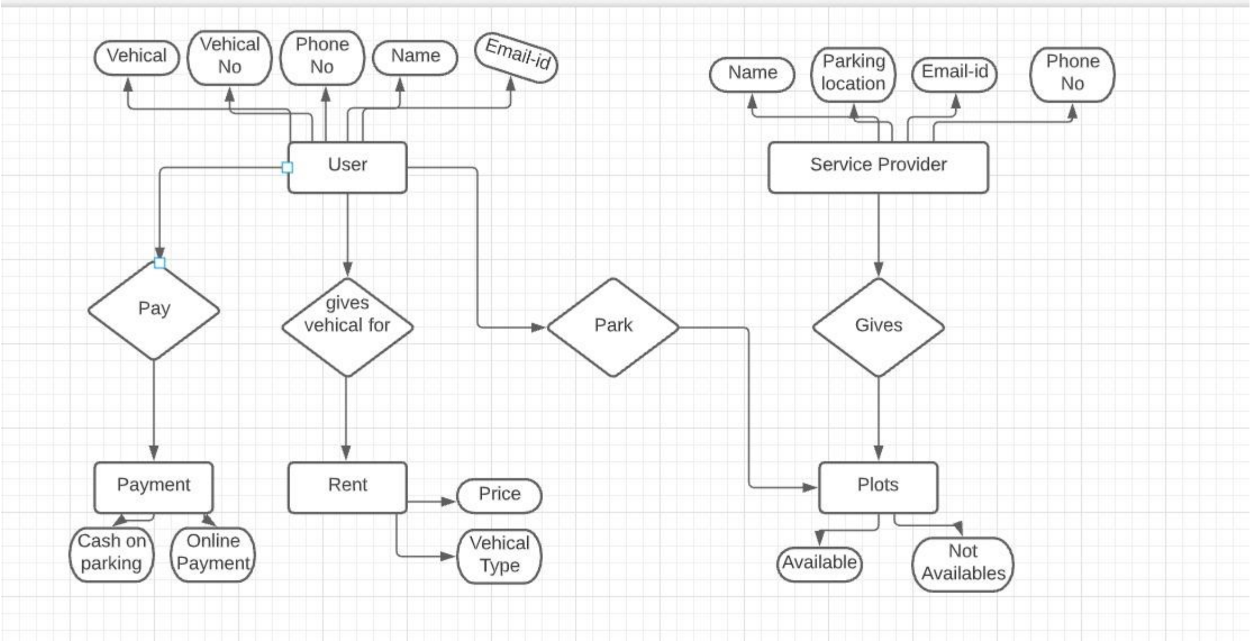


Figure :5.5 ERD