LOVELY PROFESSIONAL UNIVERSITY

Software Requirements Specification (SRS)

Vehicle Parking Assistance.

Submit To

Aditi Sharma

"In partial fulfilment of the requirements for the Master of Computer Application".

PREPARED BY		
STUDENT'S NAME	**ROLL NUMBER**	**REGISTRATION NO.**
EKHLAKH AHMAD	RD2215B67	12209166
SUKHDEEP KAUR	RD2215B65	12209123

Lovely Professional University,
Punjab(India)

Contents

Introduction

By helping both the user/customer park their car in the parking area efficiently without requiring human assistance (such as with parking fines and payments, locating a parking level, etc.), this Vehicle Parking Assistant makes managing a parking area easier.

The manager is responsible for overseeing and managing specific parking spaces, including managing payments and keeping records, etc. Guards contribute to distinguishing the locations and keeping track of the automobiles in the parking spaces when the cars come in and go out.

1.1 Purpose

A system called vehicle assistance is used to manage vehicles in various parking locations, such as those at hospitals, schools, colleges, universities, businesses, and shopping centers. As people rush in and out at various times throughout the day, many locations feature congested parking. It's challenging for parking management to keep track of the many cars that scurry in and out of the parking. Additionally, people have trouble parking their cars, much like in our customary systems People must stand in line to obtain the parking permit, which wastes their time in locating free parking for their vehicles.

1.2 Scope

The "Vehicle Parking Assistant" under evaluation will carry out the following duties:

1. Give the manager(s), guard(s), and users the login ID and password.
2. Continue to keep the data that the manager(s), guards, and users have provided.
3. Assists in managing and upkeep of the database's automotive information for the manager(s).
4. Assists in checking the vehicle before parking it.
5. Aids guards and users in identifying available parking floors.
6. Aids in online payment.
7. Assists the security personnel in managing parking spaces with available space.

1.3 Objectives/Aims

This software would produce the following anticipated outcomes:

1. Parking management will be simpler for customers because everything will be automated, including issuing tickets, locating parking spaces, managing accounting, improving safety, etc. It will oversee parking according to the kind of vehicle and allocate parking levels considering how long the car has been parked, which would save time.
2. As this program would recognize the vehicles' license plates, it would aid in recovering Unregistered or stolen vehicles It might also pick up on a bogus license plate.
3. The entire design of this software would almost eliminate human error.
4. The ultimate objective is to simplify parking management for clients by design that is user-friendly, simple enough for non-technical people to utilize, and minimizes human Correcting problems will greatly improve the experience for regular users.

1.4 Challenges

We shall face the following difficulties:

1. The entrance has cameras installed.
2. Hardware and computer connections.
3. Acquiring New Skills.
4. Take a picture of the license plate.
5. Verify bicycles and autos and keep dedicated lanes for them.

1.5 Learning Outcomes

The following difficulties shall we face:

1. The entrance was outfitted with cameras.
2. The computers' hardware connections.
3. Acquiring fresh tools.
4. Photograph the license plate.
5. Verify bicycles and automobiles and keep distinct lanes for them.
6. At any time, we shall be aware of the status of the vehicles parked in our lot. Additionally, we'll have details regarding their owners. We will be aware of their admission and exit times as well as financial information.

1.6 Nature of Final Product

Parking Assistants would manage the use of human beings in helping and maintaining the parking of automobiles in a parking area.

1.7 Business Value

This system will target businesses that want to improve the parking facilities either in their offices, malls, hospitals, schools, and universities. This will help them to improve their security and will help them to utilize their parking space more efficiently. Moreover, paid parking owners and customers can have details of their payments. The parking owners will also know at any time how many cars are there in their parking area. This system will ensure that all the cars in the parking are verified. Owners can maintain their earning history. Our earnings will be 1% of the total revenue generated by this system.

2. General Description

2.1 Product Specifications

2.1.1 Join and log in:

The manager is required to sign up for the web-based application. Users and guards must both register for the mobile application. Users can access the web application and mobile application after authentication.

2.1.2 Management Assistant Module:

It will help the manager keep track of the cars and users' records while forcing the manager to handle the database. Additionally, it will help the management to keep the payment records up to date and grant the permission to do so upon request.

2.1.3 The User Assistance Module:

This will allow users to examine their payment history and make online payments. Additionally, it will force people to look up available parking floors before parking their automobiles.

2.1.4 The Guard Assistant:

It will help the guards determine whether the car has been verified or not. Additionally, it will assist the security guards in determining the free parking floors that are accessible, allowing them to find the floors. Guards will be able to gather information about visitors to the parking lot thanks to this as well. They were first. Guards will also be able to produce slips as a result.

2.1.5 Payment:

Users can pay online with its help.

2.1.6 Search Parking Floor:

Users using the appropriate application can look up the open parking floors.

2.1.7 Notifications:

Users who have or have not installed an application on their cell phones will be notified through email or text messages, about their contact information.

2.2 Classes and Features of Users

Managers, Guards, and guests are the users of this system. Each user must initially sign up to use the application. They will need to log in to both the web-based (for managers) and mobile applications (for guests/guards). They can access the information after logging in. The entrance to the Information is ensured based on the user's access privileges.

- **Manager**: The manager has access to and control over the database's records (upon request).
- **Guards**: Guards can determine whether or not an automobile has been confirmed. From the parking floors, they may alter the number of cars (depending upon their status in the parking area). Guards can print slips and will obtain the visitor's information (who is visiting for the first time).
- **Users**: Users who have the application may check the parking levels that are available and their payment history. They could pay online. Users who don't have an application will get their parking information by email or SMS.

2.3 Operating Environment

- **Operating System:** Android and Windows.
- **Hardware:** Mobile device with at least android 5.0, IOS 6 client/server system
- **Framework:** NodeJS
- **Database:** SQL

2.4 Constraints in Design and Implementation

- Only the management may use the website. The language of the website will be English.
- The application is for Guards and Guests. The application will be based on mobile.
- Users of web-based and mobile-based applications must be registered.

2.5 Hypotheses and Dependencies

1. All necessary hardware must be accessible.
2. The necessary budget should be considered and made accessible.
3. The code must be faultless.
4. The required specification document will be followed in the completion of all tasks.
5. The system will be easy to use.
6. The project should include all the platforms and software needed.
7. A database backup should be accessible.

3. Functional Requirements

3.1 Manager Login and Manage User Information:

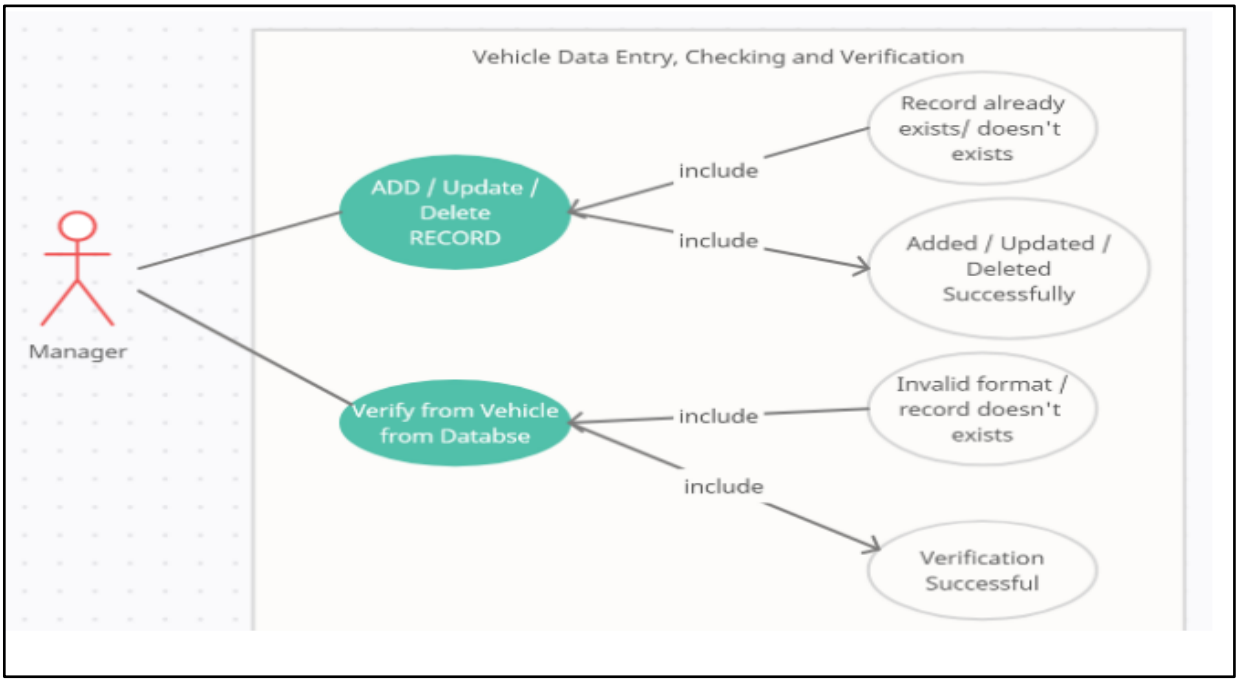
Identifier	**Manager**
Purpose	To prevent security breaches, only pre-registered users can sign in. This also ADDS a user to the database.
Priority	High
Pre-conditions	There is no manager registration, thus the database has the manager's credentials saved.
Post-conditions	The manager may add or change user information.

Typical Procedural Steps

S.no.	Actor Action	System Response
1	ADD User	New user added
2	Update User	User information updated
3	Delete User	User record removed
4	Bulk User Addition(upload file)	**amount**users added from the uploaded file

Alternate mode of Action

S.no.	Actor Action	System response
1	ADD User	• User already exists • Please fill in the credentials • User verification failed
2	Update User	The user does not exist
3	Delete User	The user does not available
4	Bulk User Addition(Upload file)	• File not found • File Format not supported • File is empty



3.2 Registration

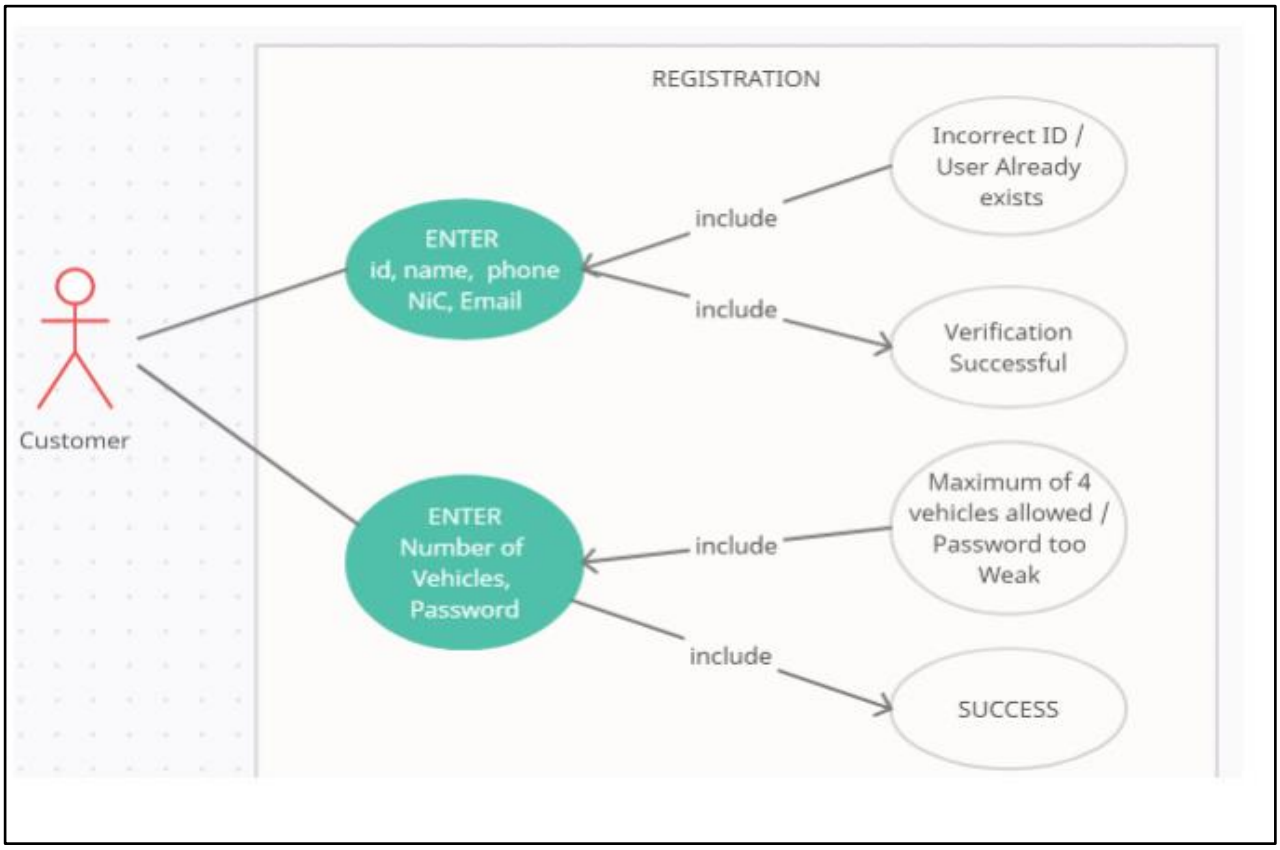
Identifier	**Customer/Manager**
Purpose	Users can register through the app and use it
Priority	Low
Pre-conditions	The user should have a unique ID provided by the manager for the registration
Post-conditions	The homepage should provide all the relevant information and a greeting for users.

Typical Procedural Steps

S.no.	**Actor Action**	**System Response**
1	Enter Unique ID	ID verified
2	Enter Full Name	Verified
3	Enter Phone No	Verified
4	Enter NiC	4 digits code verified
5	Enter Email	Verified
6	Enter the number of vehicles you owned	Saved
7	Enter Password	Saved
8	Press Register	Register, Pease Login now

Alternate Mode of Action

S#	**Actor Action**	**System Response**
1	Enter Unique ID	• Incorrect ID • ID/User already exists
2	Enter Full Name	• Please enter valid alphabets only • Numeric digits not allowed
3	Enter Phone NO.	• Please enter the correct/full digits • Phone No failed to verify • Wrong verification code
4	Enter NIC	• Please enter the right format • Please enter the complete digits
5	Enter E-mail	• Verification failed • Invalid format
6	Enter Number of vehicles you own.	• Maximum of 4 vehicles are allowed • Only Numerical digits allowed, no decimals allowed
7	Enter Password	• Password is too weak
8	Press Register	• User is already registered • Connection problem please try again
		•



3.3 Vehicle Data Entry, Checking and Verification:

Identifier	**Manager**
Purpose	• To ADD a new Vehicle record • To verity the vehicles from the local database and check it on Traffic police database
Priority	HIGH
Pre-conditions	• User is logged in • Vehicle should have a valid number plate with correct format • User has Access to local database
Post-conditions	Vehicle data should be added successfully and displayed in tabular form to the user with a verification successful signal.

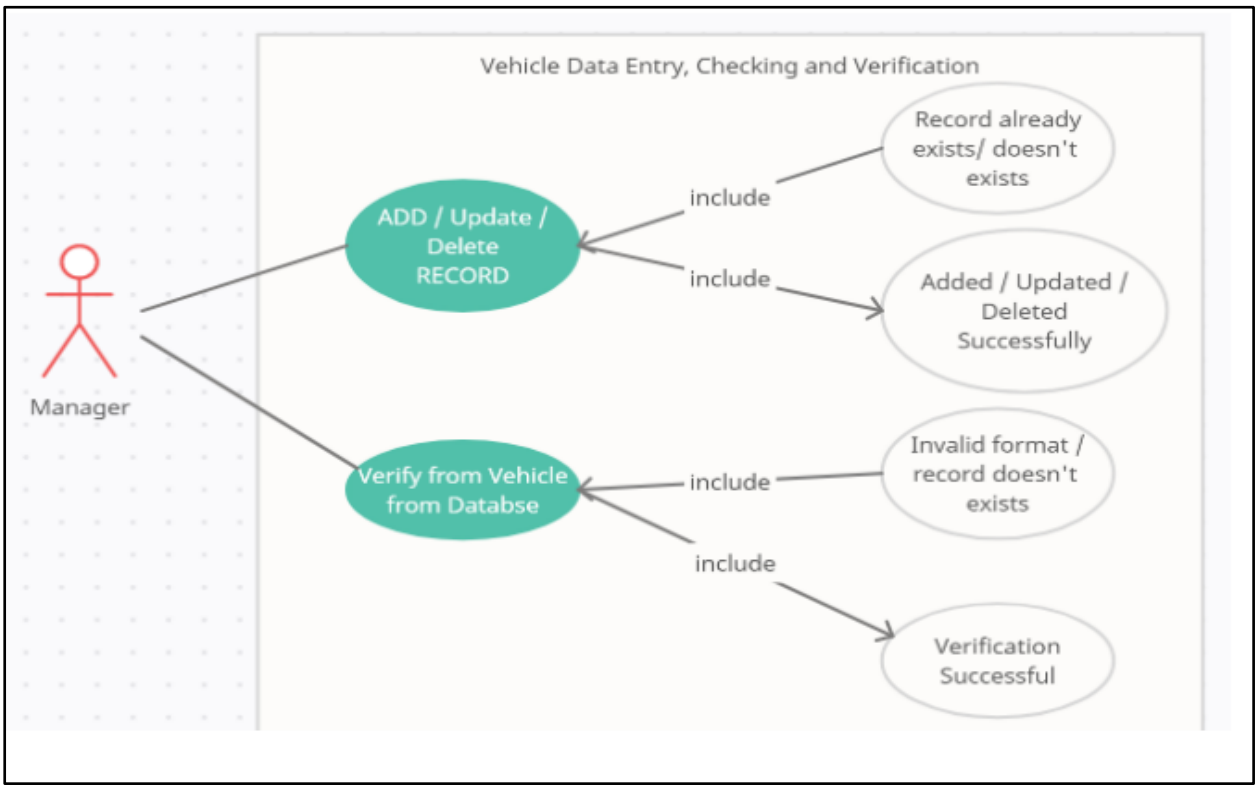
Typical Course of Action

#S	**Actor Action**	**System Response**
1	ADD Record	Record Added
2	Update Record	Record Update
3	Delete Record	Record Deleted
4	Verify from Local database	Record exists in the database

5	Check on Traffic Police database	Record Matched
6	ADD Records in bulk(upload file)	**amount** Records added from the file uploaded

Alternate Course of Action

#S	Actor Action	• System Response
1	ADD Record	• Record already exists • Invalid Number Plate format
2	Update Record	• Updation failed • Invalid format • Record doesn't exist
3	Delete Record	• Record doesn't exist
4	Verify from Local Database	• Verification failed, Record doesn't exist.
5	Check on Traffic Police Database	• Record doesn't exist but it does in local databases • Record doesn't exist elsewhere. • Server ERROR, Please try again shortly
6	ADD Records in bulk (upload file)	• File not found • File format not supported • File is empty



3.4 Manage Records:

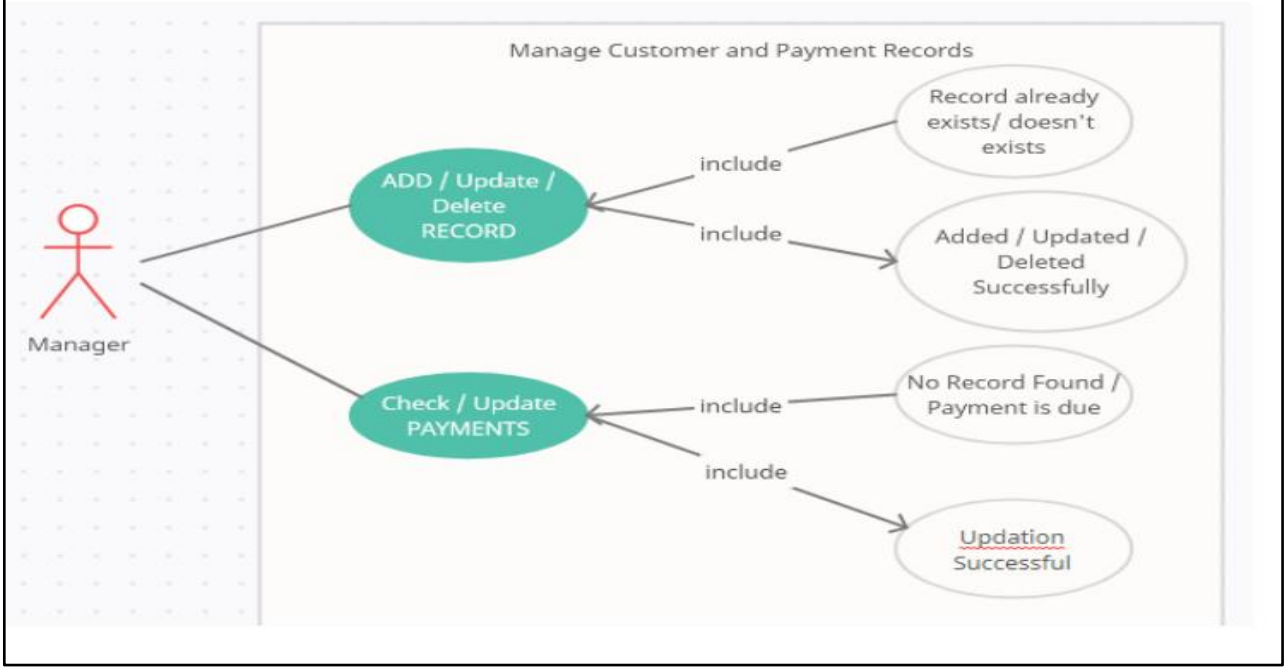
Identifier	**Manager**
Purpose	Manage Payments and Customer records
Priority	Moderate
Pre-conditions	Only an manager/Manager can access and must be logged-in
Post-conditions	Records added/updated/deleted successfully

Typical Course of Action:

S#	**Actor Customer**	**System Response**
1	ADD Customer Payments	Record added successful
2	Updated Customer Record	Record updated successful
3	Delete Customer Record	Record deleted successful
4	Check Customer Payments	Record displayed**
5	Update Customer Payments	Updation successful

Alternate Course Record:

S#	**Actor Action**	**System Response**
1	ADD Customer	User already exists
2	Updated Customer Record	User doesn't exist
3	Delete Customer Record	User doesn't exist
4	Check Customer Payments	• No record found • Payment Due
5	Update Customer Payments	No record found



3.5 Manage Vehicle traffic in the Parking:

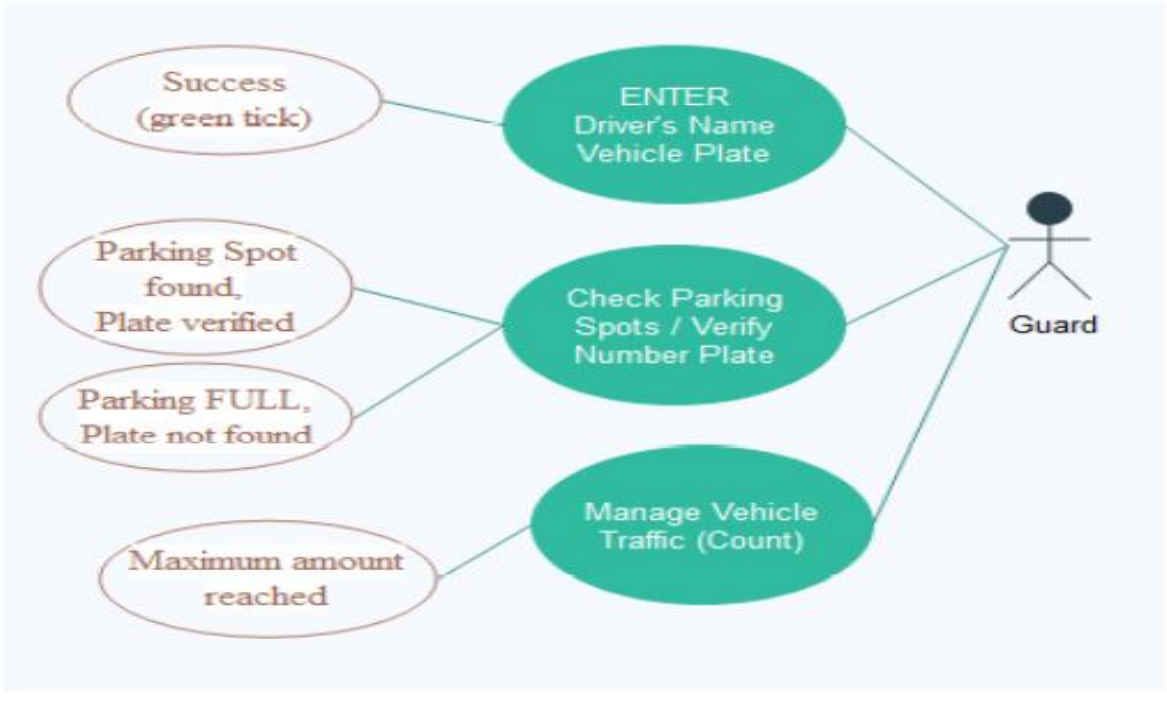
Identifier	Guard/Manager
Purpose	Issue a token on entrance for customer/guest who doesn't have a mobile phone or the APP installed
Priority	Moderate
Pre-conditions	Only Guards/Managers can access and must be logged-in
Post-conditions	Token issued and its record saved in Database

Typical Course of Action:

S#	Actor Action	System Response
1	Enter Name of the Driver	Green Tick
2	Enter Vehicle Number Plate	Green Tick
3	Check Number Plate	Verified
4	Manage Vehicle count	Increment/decrement
5	View Free parking floors	Pop-up shown

Alternate Course of Action:

S#	Actor Action	System Response
1	Enter Name of the Driver	• No numbers allowed • Can't leave this field empty
2	Enter Vehicle Number Palate	Invalid format
3	Check number Plate	Plate not found in the database
4	Manage Vehicle count	Limit exceeded
5	View Free parking floors	Parking FULL



3.6 Parking Reservation:

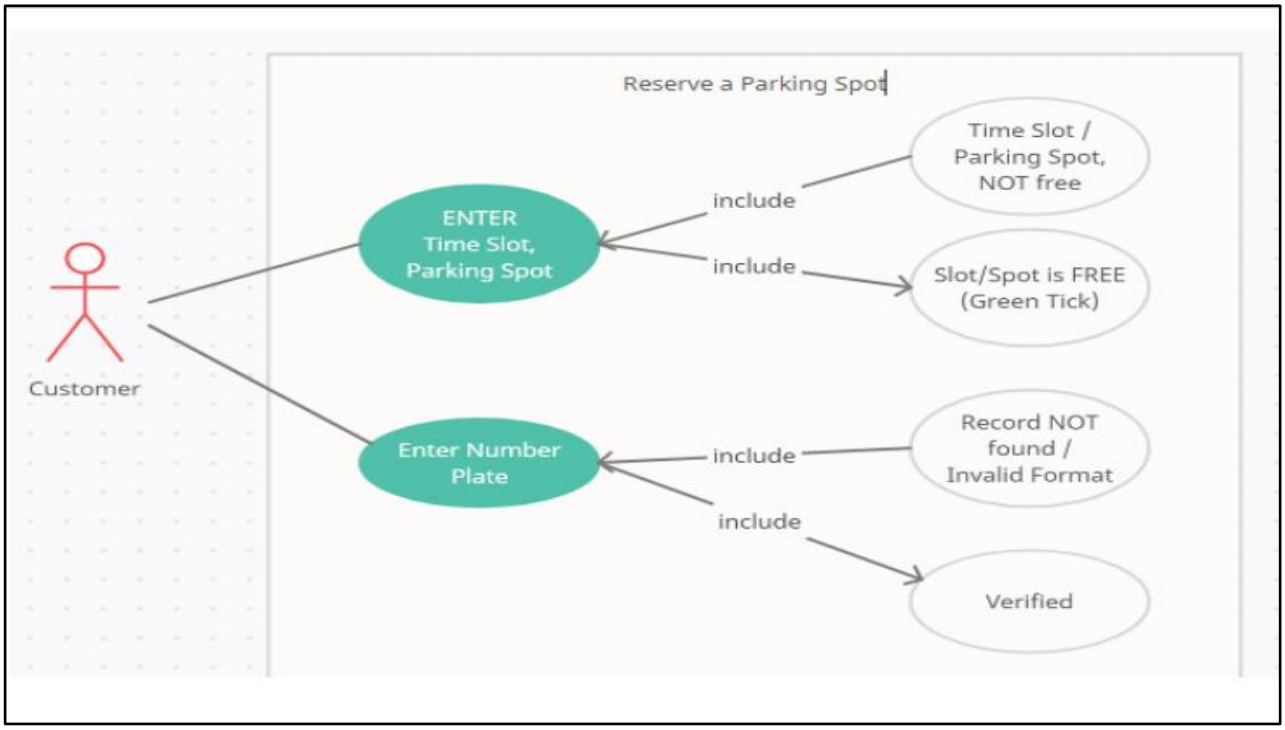
Identifier	**Customer**
Purpose	Reserve a parking floor
Priority	Moderate
Pre-conditions	User must be registered and must be a VIP
Post-conditions	Parking reserved and a timeslot is issued

Typical Course of Action:

S#	**Actor Action**	**System Response**
1	Enter the timeslot you want to reserve.	Slot is free (green tick)
2	Enter your desired parking floor	Spot is free (green tick)
3	Enter the plate of one of your registered vehicles you will be parking.	Verified.

Alternate Course of Action:

S#	**Actor Action**	**System Response**
1	Enter the timeslot you want to reserve	Timeslot isn't free
2	Enter your desired parking floor	Parking floor Taken
3	Enter the plate of one of your registered vehicles you will be parking.	• Invalid format • Vehicle not found



3.7 Payment methods:

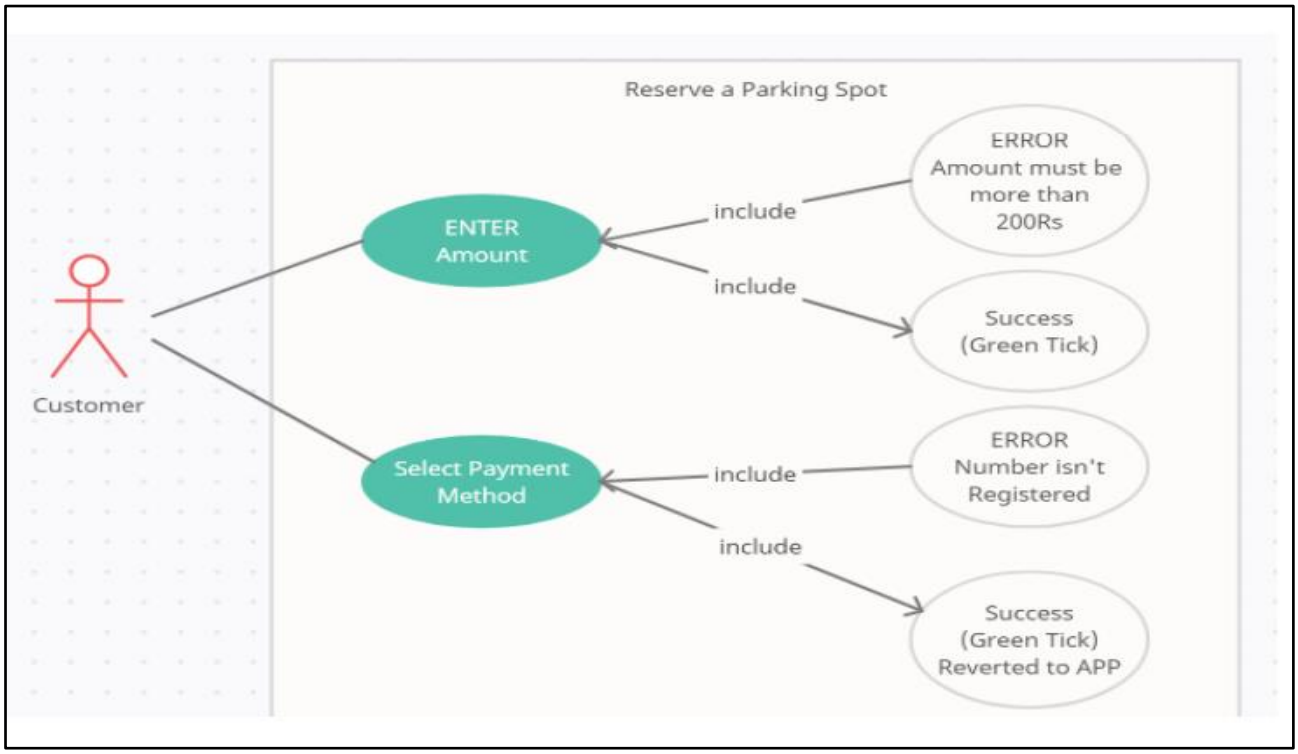
Identifier	Customer
Purpose	Pay for the parking through the APP
Priority	High
Pre-conditions	User must be registered and have a account on easy paisa/jazzcash
Post-conditions	Notification/SMS received and funds added

Typical Course of Action:

S#	**Actor Action**	**System Response**
1	Enter Amount (minimum 200Rs)	Green Tick
2	Select payment method, easypaisa or jazzcash	Gree Tick
3	Pay now	Opening easypaisa/jazzcash app for final payment

Alternate Course of Action:

S#	**Actor Action**	**System Response**
1	Enter amount(minimum 200Rs)	• Error! Amount must be more than 200Rs • Invalid format, only integer value allowed
2	Select payment, easypaisa or jazzcash	Error! Your number isn't registered to the desired method
3	Pay now!	Error! App not installed



3.8 Send/Receive SMS and E-mail:

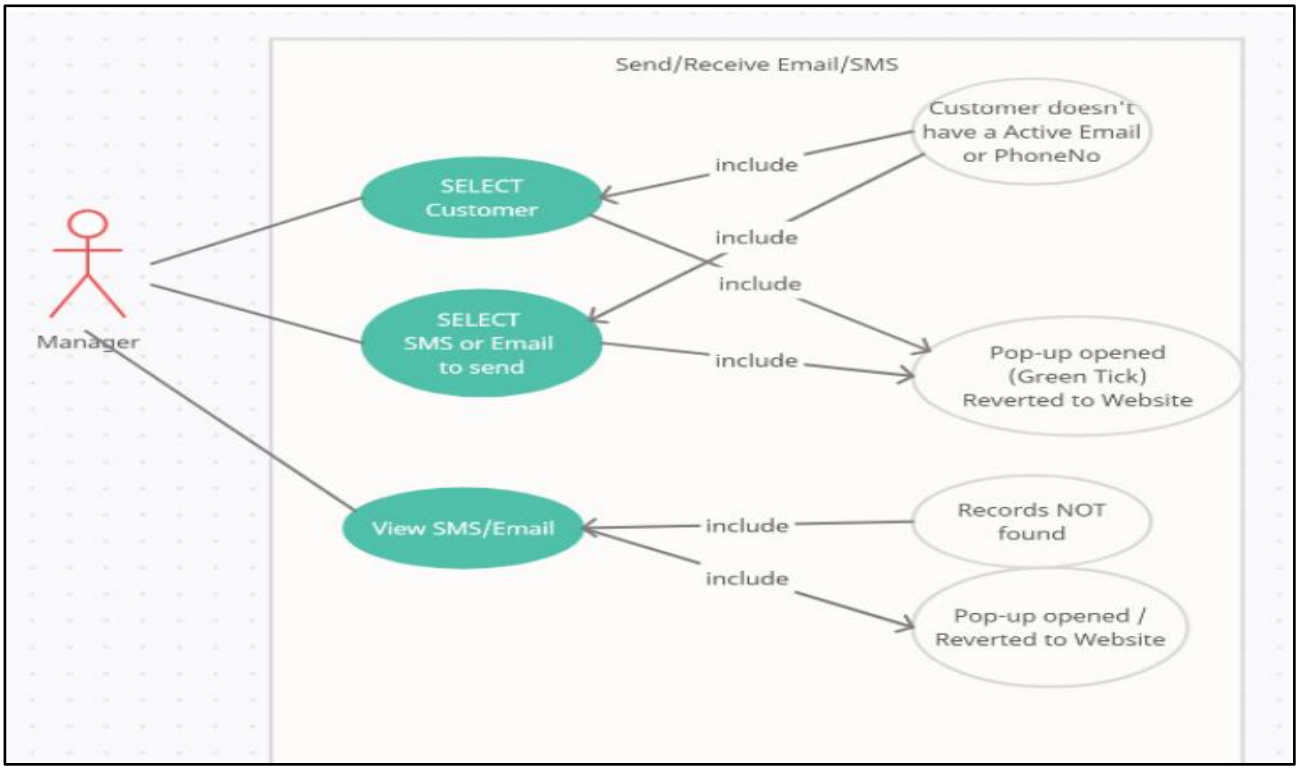
Identifier	Manager
Purpose	Send SMS/E-mail to customers about Queries
Priority	High
Pre-conditions	Only accessible by manager and must be logged-in and have a Number Verified
Post-conditions	SMS and E-mail sent and Record of is shown

Typical Course of Action:

S#	Actor Action	System Response
1	Select a customer	Greer Tick
2	Select E-mail or SMS	Email/SMS sender pop-up opened
3	SEND!	Green Tick
4	View E-mail/SMS	SMS record opened; E-mail website opened

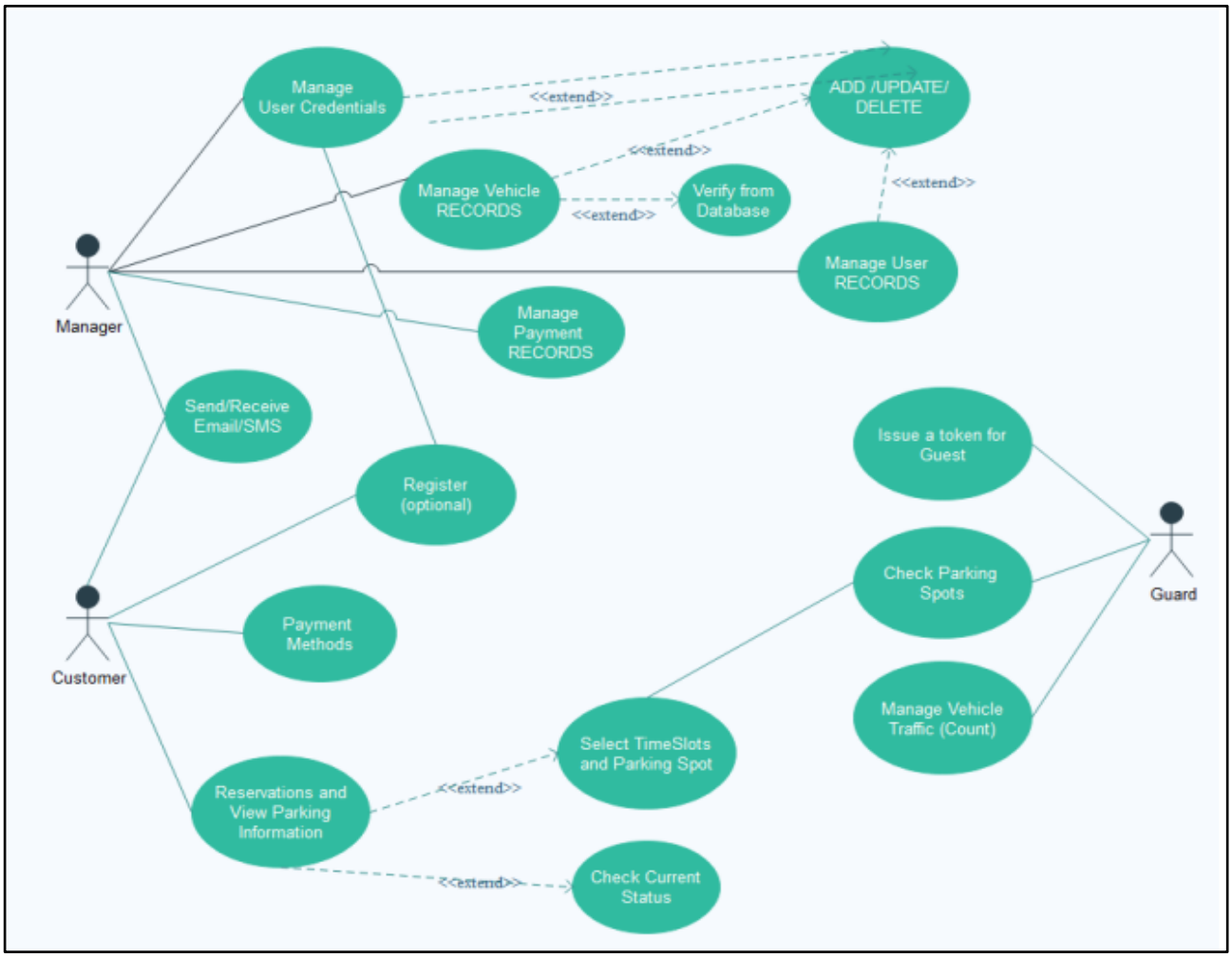
Alternate Course of Action:

S#	**Actor Action**	**System Response**
1	Select a Customer	• Customer doesn't have Active Phone No or E-mail
2	Select E-mail or SMS	-
3	SEND!	-
4	View E-mail/SMS	No records found

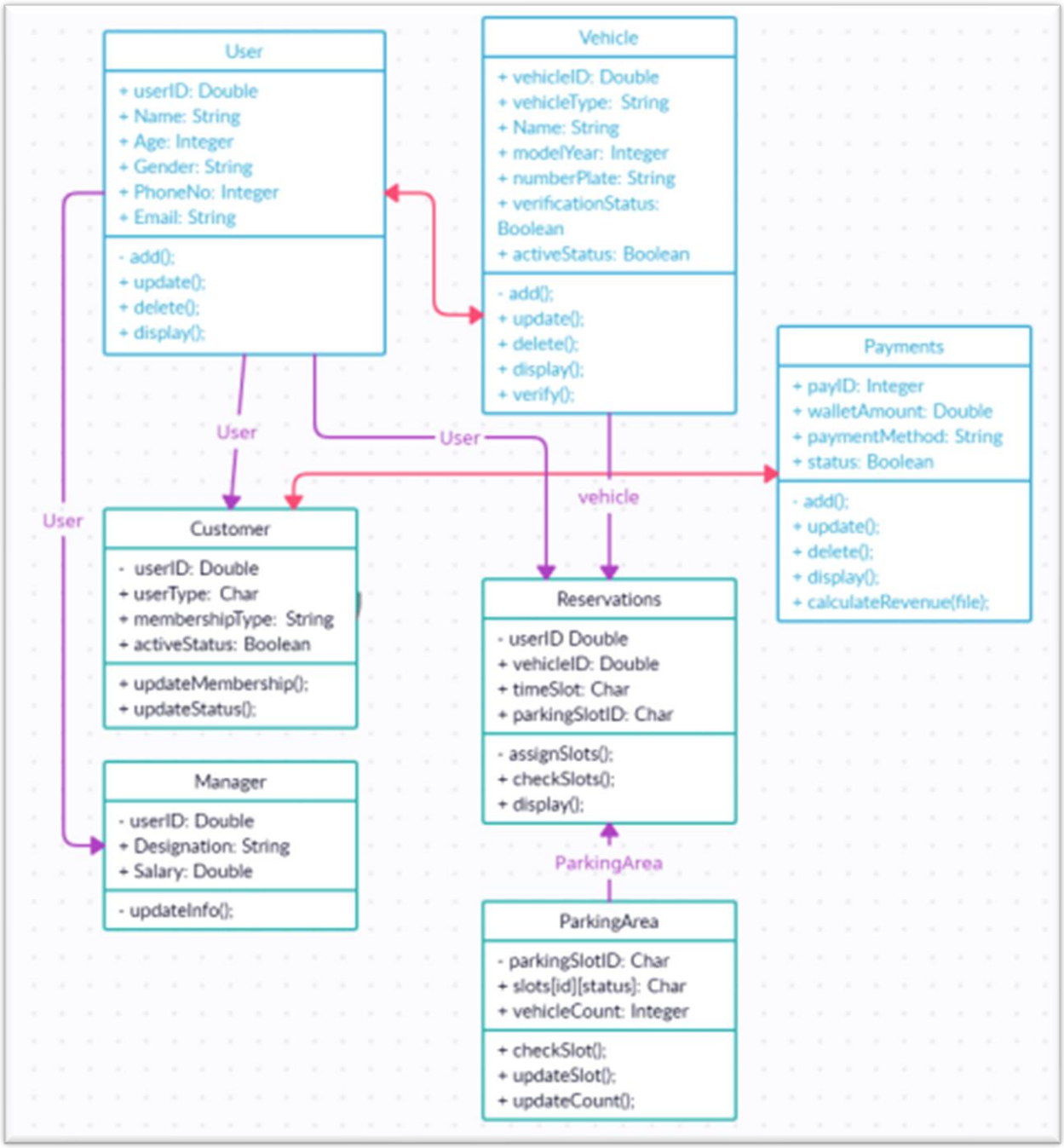


3.9 Requirements Analysis and Modelling:

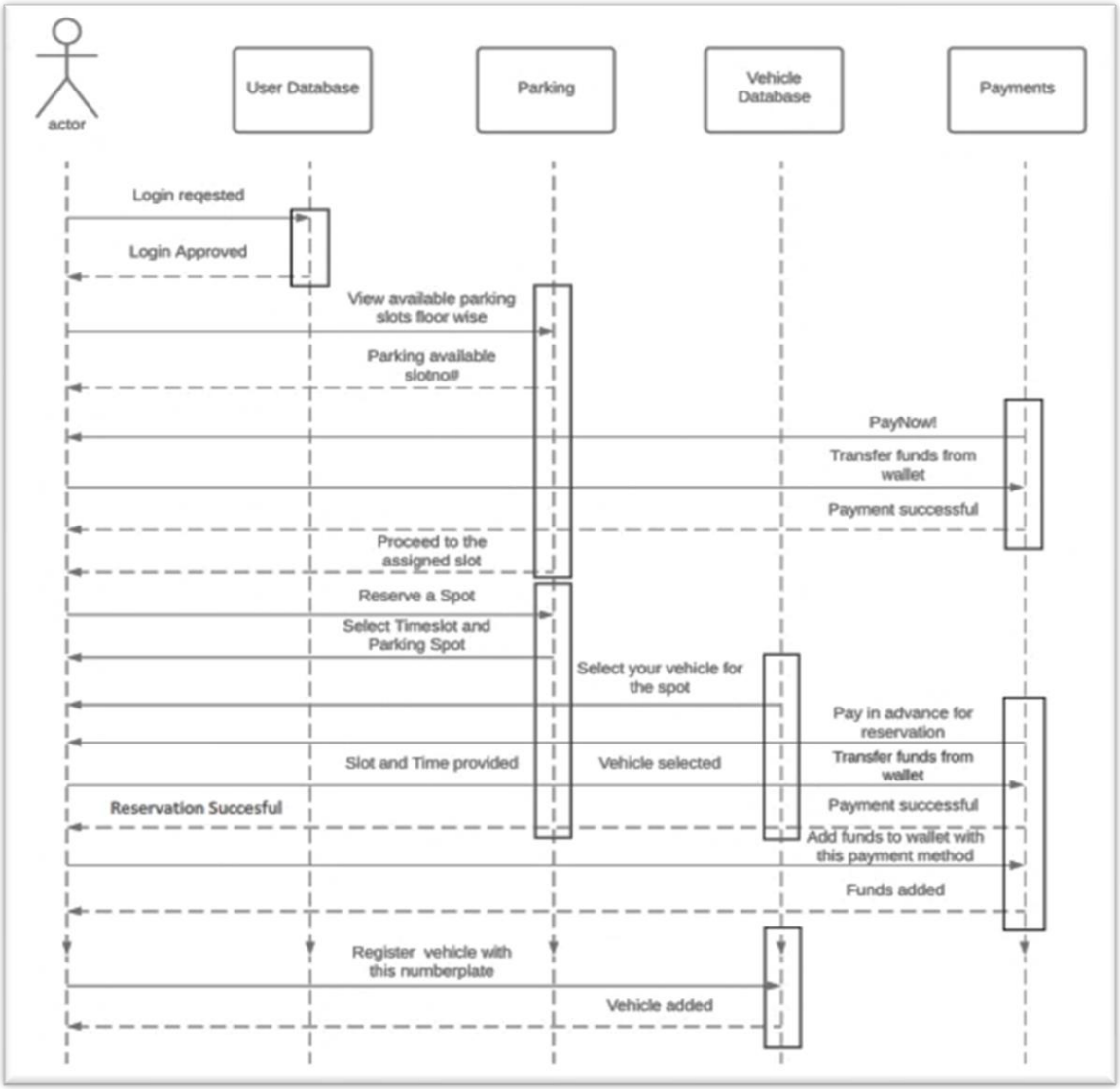
3.9.1 Use Case-



3.9.2 Class Diagram-



3.9.3 Sequence Diagram-



4. Non-functional Requirements

4.1 Performance requirements:

- Should support low storage; memory-efficient applications.
- For a service to be dependable, data servers must be accessible.
- There should be as little lag in communications as feasible.

4.2 Safety requirements:

- The database has to be backed up.
- Data must be accurate.
- Data safety and security.

4.3 Security requirements:

4.3.1 Data Privacy:

- Databases should be secured with encryption and kept secret.
- Authorized managers must be given access rights to the database.
- Only registered users will be granted access to the system.
- Users should log in before accessing the data.
- User authentication needs to come first.
- Email and a phone number, if a user is required.

4.3.2 User identification and authentication:

- Only registered users will be granted access to the system.
- Users should log in before accessing the data.
- User authentication needs to come first.
- Email and a phone number, if a user is required.

4.3.3 Security-related policies and guidelines:

- Every Process would follow the guidelines established by the executive branch.
- The regulations established by the government should be reflected in the policies.

4.3.4 Security for vehicles:

- There shouldn't be any unverified vehicles in our parking lot.
- At any time, the number of cars in the parking lot will be counted.

4.4 Additional Software Quality Attributes:

- **Usability:** The user interface of the application should satisfy the application's users. Additionally, the application ought to be faultless.
- **Order or Product Delivery:** The order must arrive at its destination within the estimated time frame.
- **Learnability:** The user of this system is able to comprehend the features and functions of the application with ease.
- **Availability:** Due to the high volume of simultaneous orders from customers, the server's and system and system's availability should be crucial for the application.
- **Correctness:** Since this application system allows three different user privileges, the login and registration processes should run smoothly.