

---

---

# EZPARK

---

---

An Online Parking Management System



CREATED BY

2ND YEAR GROUP CS44

*University Of Colombo School of Computing*

October 30, 2023

**Details of Project Supervisor, Co-supervisor, Advisors and Clients****Project Supervisor**

Name of the supervisor : Ms. Akna Delgodararchchi

**Project Co-Supervisor**

Name of the co-supervisor : Mr. Ashen Kavinda

**Attendance of the group members**

Student Name	Index No.	Signature
K.D.T.Srikantha	21001901	
P.S.Hansika	21000778	<u>Sohani</u>
A.J.P.G. Pehesara	21001316	<u>Gamindu Pehesara</u>
C.D. Sriwarnasinghe	21001911	

# Table of Contents

## 1 Project Introduction

1.1 Domain Description

1.2 Project Goal and Objectives

1.3 Project Constraints

1.4 Project Assumption

## 2 Feasibility Study

2.1 Technical Feasibility

2.2 Economic Feasibility

2.3 Legal and Ethical Feasibility

2.4 Operational Feasibility

2.5 Scheduling Feasibility

## 3 Requirements

3.1 Stakeholders

3.2 Functional Requirements

3.2.1 All Users

3.2.2 Admin

3.2.3 Driver

3.2.4 Parking owner

3.2.5 Merchandiser

3.2.6 Security

3.3 Quality Attributes

3.3.1 Performance

3.3.2 User Friendliness

3.3.3 Security

3.3.4 Availability

3.3.5 Modifiability

3.3.6 Accuracy and Reliability

3.4 UseCase Diagram

3.4.1 UseCase Narratives

3.4.1.1 All Users

3.4.1.2 Parking Owner and Merchandiser

3.4.1.3 Driver

3.4.1.4 Admin

3.4.1.5 Security

## 4 Proposed System's Architecture

4.1 Component Diagram

4.2 Component Description

## 5 System's Design

5.1 Class Diagrams

[5.2 ER Diagram](#)

[5.3 Sequence Diagrams](#)

[5.3.1 All users](#)

[5.3.2 Parking Owner and Merchandiser](#)

[5.3.3 Driver](#)

[5.3.4 Security](#)

[5.3.5 Admin](#)

[5.4 Activity Diagrams](#)

[5.4.1 All users](#)

[5.4.2 Parking owner and Merchandiser](#)

[5.4.3 Driver](#)

[5.4.4 Security](#)

[5.4.5 Admin](#)

[5.5 State transition diagrams](#)

[6 User Interface Flow Diagram using wireframes](#)

# 1 Project Introduction

## 1.1 Domain Description

As urbanization grows and more people move to cities, own vehicles, and the amount of free land decreases, the demand for parking spaces has increased over the past few years. So, people often face difficulties finding suitable parking spaces in densely populated areas such as Colombo District.

- Difficult to find parking spaces during rush hours.
- Retail businesses often face difficulties providing enough parking spaces, especially during seasonal times.
- Utilization of empty lands in urban areas.

To address this recurring problem, we came up with an idea of a parking management system, eZpark.



- This parking system is a project that seeks to efficiently manage and organize parking spaces in a designated location, such as residences, empty land, etc.
- Users in our system will be driver, parking space owner, merchant, admin, and security.
- In the system, a parking owner can be an owner of an already existing parking lot or a bare land owner, or an owner of a household who is willing to give their extra spaces in the house garden.
- Parking owners or a merchant can register their parking in the system.
- Drivers can search for a normal parking space by location as well as merchant's parking by its name. And the driver can see the availability of parking in real-time.

- Also, merchants can manage their own parking areas as well as merge nearby parking spaces as their own parking spaces and provide additional parking facilities for their customers through this system.
- The proprietors will also provide features to administer all their empty lands and provide parking facilities for people through the system.
- The basis of the parking system is a centralized software platform that incorporates all the components.
- It manages data, measures occupancy, generates reports, and allows administrators to monitor and control the system.

As the final product, this will be a functional, reliable web-based application where vehicles will be able to locate a parking location more straightforwardly and simply. Also, proprietors will be able to promote and manage their property with simplicity. And merchants will be able to administer their own parking places and will also be able to provide more parking facilities for their consumers.

## 1.2 Project Goal and Objectives

Ultimate goal of the system is to provide hassle-free parking while optimizing space utilization and ensuring safety.

- Provide real-time information on parking availability.
- Optimize space utilization in urban areas.
- Increase the customer base of merchants by providing convenience to consumers.

## 1.3 Project Constraints

- The system should be developed from scratch since no frameworks are allowed
- Geographical Constraints - Physical location of the user should be within Colombo District
- Security Constraints - The security of the vehicle depends on the security measures taken by the landowner
- Social (Language) Constraints - The system will be available only in English.
- Real time production workflow can not be viewed through the system as all processes are done manually.

## 1.4 Project Assumption

- All system users have basic digital literacy
- All system users have a digital device with a stable internet connection
- All system users have the software application installed on their device

## 2 Feasibility Study

### 2.1 Technical Feasibility

In Technical Feasibility, we concentrate on whether the project can be implemented in the given technical aspect (Is it possible to implement given software and hardware, and are technology skills available?). Project members have capabilities in these technologies and languages (technologies are described in the Technology section). And also, these technologies and platforms have enough resources and community support. The hardware of the initiative is our customers' personal devices (laptops, phones, tablets, etc.). We host the software in the cloud. Then anyone can use the software. Therefore, the project can be implemented using the available technologies and technical talents of the members.

### 2.2 Economic Feasibility

Economic feasibility refers to the analysis of the project's cost-effectiveness, which determines if it is practical to implement our system. Apart from the Google Maps API, and Hosting Expenses, all the above-mentioned software and tools that are used for development reasons are free and open-source (Some have free trial versions, and other hosting and APIs have lower costs). The development team will use their own devices and other hardware components; there is no additional hardware cost. The expense can be covered with the earnings we keep from the clients as well as the service providers throughout the year. Since there are no major expenses, the total cost of the system is tolerable. Hence, the system is economically feasible.

### 2.3 Legal and Ethical Feasibility

In this section, we consider the legal and ethical requirements of the project and confirm the legal and ethical requirements. Analyzing barriers to the legal and ethical implementation of projects, data protection acts, etc. are included in this section.

Users of the system are unable to access the personal information of any other users or service providers.

Under no circumstances will the customer-related service records that have been gathered and stored be released to a third party.

Payment records of customers and service providers can be accessed and administered only by the system administrator. The whole system will be developed using free and open-source resources, and any external code fragments will be acknowledged and given credit to their rightful proprietor. Although Sri Lanka does not have any consolidated and/or specific laws on data protection yet, in early 2021, the LDD released a final draft version of an Act to Provide for the Regulation of Processing of Personal Data (2021) ('the Draft Bill'). According to sections 9 and 10 of the draft bill, collecting and retaining data for specific purposes with prior knowledge is feasible.

Hence, according to the draft, there are no constraints on constructing the system in compliance with the rules and regulations relating to data protection. Since there are neither legal nor ethical issues, the initiative is legally and ethically feasible.

## 2.4 Operational Feasibility

This section refers to the analysis of how well the proposed solution solves the problems, and satisfies the requirements identified in the requirements gathering phase.

- Considering the issues identified, the proposed system addresses all the functionalities essential for the current system.
- Making sure that platform's user interface is interactive, user-friendly, and accessible in order to attract donors to readily participate and donate.
- only the basic technical knowledge is needed to interact with the system.

Our solution is a mobile responsive web application, which necessitates the following for usage and operation:

- Users are required to have an active internet connection.
- Access requires a mobile phone or computer with a recent web browser installed.
- Basic familiarity with IT and internet usage is enough.
- Making payments entails an intermediate understanding of online and card-based transactions. If there is no such knowledge the platform will provide the knowledge needed.

According to the above facts the system caters to all the requirements identified. Hence the system is operationally feasible.

## 2.5 Scheduling Feasibility

This feasibility looks at if the project's goals can realistically be fulfilled within the given timeframe, including tasks, resources, and probable delays.

- The development of the system should be completed during the coming 8 months.
- The timeline for the project is created efficiently.
- Requirement collecting phase is concluded and all the requirements of the system have been clearly identified.
- The tasks will be spread equitably among the members in order to ensure that the timeline can be met.
- According to that, the estimated man hours for the project completion might be indicated as follows,
  - Working hours per week per member= 10 hours
  - Weeks per 8 months = 32
  - Working hours per member for 8 months = 320 hours
  - No of members = 4
  - Minimum working hours for the project = 1280 hours
- We have decided to use the iterative waterfall methodology for the development process as the requirements are almost stable at the analysis step and there won't be any big changes to the requirements that can affect the timetable of the project.

As well as we can finish off a task one after another while we can adjust the previously done tasks by backtracking if needed.

Hence scheduling the project is feasible.

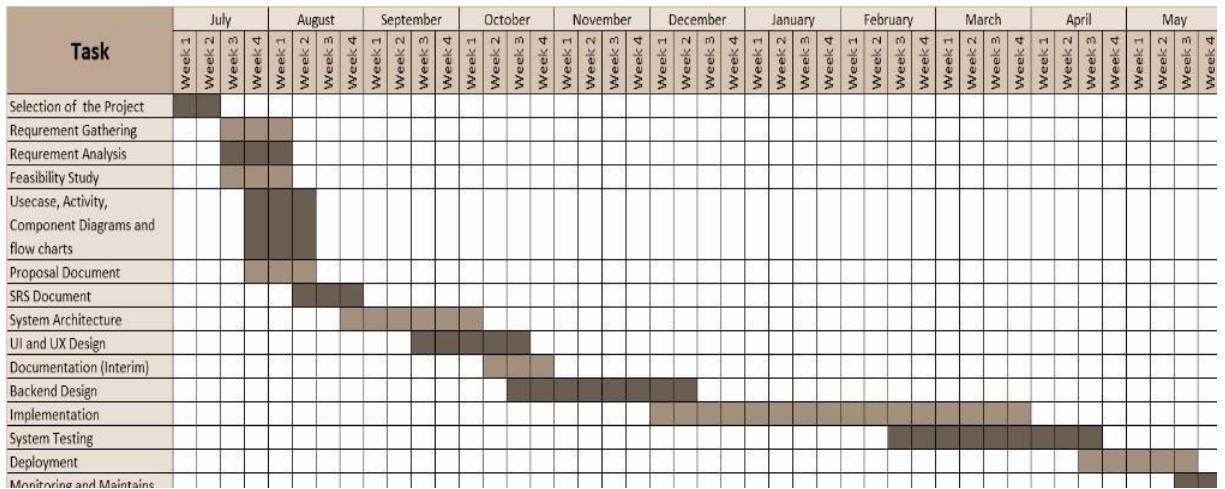


Figure 1: gantt chart

## 3 Requirements

### 3.1 Stakeholders

In requirement identification, the users of the system have been identified as follows.

- Driver
- Parking owner
- Merchandiser
- Admin
- Security

### 3.2 Functional Requirements

The following functionalities have been recognized regarding the operations that each actor performs using the system. Each functional requirement is stated under the identified actors of the system.

#### 3.2.1 All Users

- Should be able to register (except admin).
- Should be able to login.
- Should be able to logout.
- Will be able to view the profile (except admin).
- Will be able to update the profile (except admin).
- Will be able to delete the profile (except admin).

- Will be able to change the password (except admin).

### 3.2.2 Admin

- set a price limit
  - o admin can set a maximum price limit for parkings.
- do Access controlling
  - o admin can delete or suspend reported accounts and monitor the overall activity log of the system.
- verify parking
  - o verify details regarding to parkings.
- set an income rate
  - o the percentage of a parking payment goes to the system.
- view reports
  - o admin can view income reports .

### 3.2.3 Driver

- search parking slots (by location or by merchandiser)
  - o driver can search for an available parking slot by location or merchandiser's name.
- register a vehicle
  - o driver can register new vehicles to the system as well as delete or update vehicles.
- view available parking slots
  - o driver can see available parking slots in a parking space.
- complain about parking
  - o driver can review and provide feedback about the parking.
- subscribe to the package
  - o driver can subscribe to weekly, monthly packages according to their requirement.
- manage the package
  - o driver can unsubscribe or change the current subscribed package
- rate (security, parking, or both)
  - o driver can rate not only parking but also security.
- view parking history
  - o driver can view his/her parking history

- scan QR
  - When a driver enters a parking lot, driver has to scan a QR code to measure the time period he/she spends in the parking slot.
- make a payment
  - When a driver leaves the spot he/she has to scan the QR code to confirm the leaving and it calculates the total amount to pay and driver can pay through the QR or manual. After payment is successful parking slot will reavailable.

### 3.2.4 Parking owner

- register parking
  - land owners can register their parkings on the app.
- set parking slot capacity
  - land owners can set how many slots available in the parking according to the vehicle type.
- set prices
  - land owner can set a price below the maximum price limit for their parking depending on the vehicle type.
- set the availability of security officers
  - landowner can display the availability of the security officer of his /her parking space.
- set availability of the parking space
  - landowner can set a parking space is unavailable if it's occupied.
- update the availability of the parking slots
  - Whenever a vehicle leaves the premises and pays manually, the land owner can enable the availability of the empty slot.
- add a security officer
  - parking owner can check security profiles and request its service if needed.
- manage parking slots(update , add , delete)
  - can add more parking slots and also delete or update.
- create packages
  - can create various packages, weekly, monthly wise.
- view security profiles
  - land owner can search and view profiles of security officers.
- view merchants' requests

- o land owner can view merchants' requests and respond to them.
- view reports - parking owner can view automatically generated reports.

### 3.2.5 Merchandiser

- request nearby parking
  - o merchandisers can request nearby parking from parking owners.
- make payments
  - o merchandiser can pay through the app to the land owner for attached parking.
- register parking -
  - o merchandiser also can register their parking space on the app to optimize the utilization of the parking.
- set parking slot capacity
  - o can set how many slots available in the parking according to the vehicle type.
- set the availability of security officers
  - o can display the availability of the security officer of his/her parking space.
- set availability of the parking space
  - o can set a parking space is unavailable if it's occupied.
- update the availability of the parking slots
  - o whenever a vehicle leaves the premises and pays manually, the land owner can enable the availability of the empty slot.
- add a security officer
  - o parking owner can check security profiles and request its service if needed.
- view security profiles
  - o can search and view profiles of security officers.
- manage parking slots(update , add , delete)
  - o can attach more parking spaces and also delete or update.
- view reports
  - o Merchandiser can view reports, how many vehicles arrives to their attached parking.

### 3.2.6 Security

- request for a job
  - Security can register to the system and request for available jobs.
- view available jobs
  - Security can view posted security jobs.
- update availability of parking slots
  - Security can update the availability of a parking slot.
- view capacity
  - Security officer can view the capacity of a parking, how many vehicles can park according to their type.

## 3.3 Quality Attributes

- Performance
- User Friendliness
- Security
- Availability
- Modifiability
- Accuracy and Reliability

### 3.3.1 Performance

- System should respond to the events without any significant delay that may affect the overall usability of the system.
- Use optimized algorithms.
- Database shall be normalized to reduce data redundancy and by that it shall allow retrieval of data efficiently by minimizing the delay.
- Responsiveness: We recognize the importance of a system that swiftly reacts to user inputs. To achieve this, we will implement efficient coding practices and employ technologies that minimize processing delays. Rigorous testing will be conducted to

gauge response times under different loads, enabling us to identify bottlenecks and optimize system performance.

- **Concurrent User Capacity:** A crucial aspect of our system's utility is its ability to accommodate multiple users simultaneously. We will engineer the system architecture to handle concurrent user loads effectively. Scalability will be a guiding principle as we design and test the system. System will smoothly run for a large number of students.( roughly 5000)

### 3.3.2 User Friendliness

- **User-friendly interfaces:** In order to achieve a user-friendly interface, we will focus on simplifying navigation and minimizing complexity. By implementing consistent design elements, users will have a familiar experience across various interactions within the system. Also, there are more symbols. That will help to have a smooth process even for users who with less IT and English skills
- Confirmation and pop up messages shall provide a clear idea of the actions that the user performs.
- **Streamlining the Registration Process:** Our approach involves a review of the existing registration workflow to identify inefficiencies. By collaborating closely with stakeholders, we will work to reengineer the registration process, removing any unnecessary steps that hinder smooth user interactions. This approach serves a dual purpose: it minimizes the occurrence of unnecessary rejections and significantly reduces the administrative burden.

### 3.3.3 Security

In our security strategy, we have implemented the following measures to ensure the safety of our system and user data:

- **Password Protection:** We are implementing measures to ensure that passwords are never visible during any point in time. This ensures that passwords remain confidential and protected from unauthorized access.
- **Manual Verification:** Our security protocols include manual verification procedures to ensure that only legitimate users gain access to the system. This verification step adds an extra layer of protection against unauthorized access attempts.
- Every login is processed with authentication of credentials
- OTP being sent to customers via email when recovering/changing passwords.

### 3.3.4 Availability

- We can use reliable Hosting Providers to host our app.
- We use mirror servers to ensure the availability.
- The system will have a 100% uptime except in the case of maintenance or crashes.

### 3.3.5 Modifiability

- We use structured architectural patterns to implement, such as MVC architecture, and it helps to increase the scalability of the system.
- Version Control System: We utilize a version control system to keep track of changes made to the system's codebase over time. This not only enables efficient collaboration among our development team but also ensures that any modifications are traceable and reversible if necessary.
- Proper Documentation: We prioritize maintaining comprehensive documentation that covers system architecture, codebase structure, and functionality details. This documentation serves as a valuable resource for both current and future developers, facilitating a smoother understanding of the system and making future updates or enhancements more seamless.

### 3.3.6 Accuracy and Reliability

- Error handling .
- Conduct comprehensive testing at all levels to ensure the reliability and accuracy of our system.
- The users' information will be protected and not be shared with third parties.
- The information provided will be up-to-date and reliable.
- The server will be hosted on a cloud platform and will be accessible at any time.
- The system will be continuously checked for errors and bugs to improve reliability.

## 3.4 UseCase Diagram

The below shown figure shows the UseCase Diagram of the system

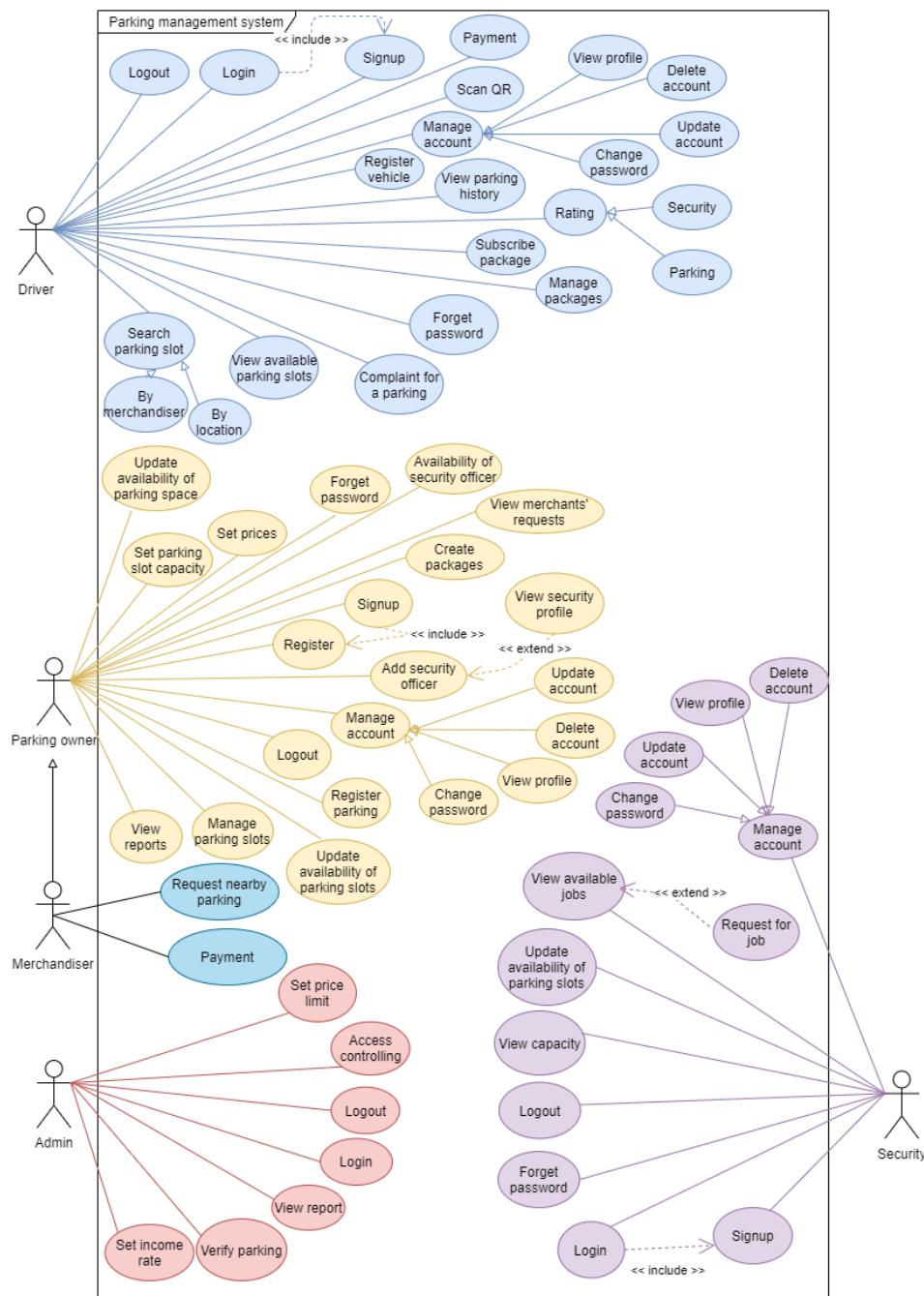


Figure 2: UseCase Diagram

UseCase Diagram link - [UseCase Diagram](#).

### 3.4.1 UseCase Narratives

The below shown tables shows the usecase narratives of the system

### 3.4.1.1 All Users

<b>Use Case</b>	SignUp
<b>Use Case ID</b>	01
<b>Actor</b>	Driver, Parking owner, Merchandiser, and security
<b>Description</b>	All users except admins can register with the system.
<b>Precondition</b>	The user has not been registered in this system before.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Select the "Register" option.</li> <li>2. Select the "Driver" or "Parking Owner" or "Merchandiser" or "Security" option.</li> <li>3. Enter registration details.</li> <li>4. Validate the information.</li> <li>5. Verify the contact number or email. The system updates itself and notifies</li> </ol>
<b>Post Condition</b>	Parking owner profile created
<b>Alternative Flows</b>	<p>Entered invalid data.</p> <ul style="list-style-type: none"> <li>• The system displays an error</li> <li>• The user can enter data again</li> </ul> <p>Incorrect contact number or email.</p> <ul style="list-style-type: none"> <li>• OTP is not received</li> <li>• The user can enter data again</li> </ul> <p>Incorrect OTP</p> <ul style="list-style-type: none"> <li>• OTP verification fails</li> <li>• User can resend OTP again</li> </ul>

Figure 3: Sign up

<b>Use Case</b>	Login
<b>Use Case ID</b>	02
<b>Actor</b>	Driver, Parking owner, Merchandiser, Security, and admin
<b>Description</b>	All users can login to the system.
<b>Precondition</b>	All users except admin should be registered with the system before logging in.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Select "Login" option</li> <li>2. Enter the user name and password.</li> <li>3. The system validates the user name and password.</li> <li>4. The system displays "Login Successful".</li> <li>5. The system provides the relevant interface.</li> </ol>
<b>Post Condition</b>	The user is logged into the system.
<b>Alternative Flows</b>	<p>If the provided user name or password is incorrect,</p> <ul style="list-style-type: none"> <li>• Error message displayed</li> <li>• The user can re-enter credentials or reset password</li> </ul>

Figure 4: Login

<b>Use Case</b>	Logout
<b>Use Case ID</b>	03
<b>Actor</b>	Driver, Parking owner, Merchandiser, Security, and admin
<b>Description</b>	All users can log out of the system.
<b>Precondition</b>	Logged in through their accounts.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Users can click the "Logout" option.</li> <li>2. Users log out of the system.</li> <li>3. Display "Logout Successfully".</li> </ol>
<b>Post Condition</b>	Users log out of the system.
<b>Alternative Flows</b>	None

Figure 5: Logout

<b>Use Case</b>	View Profile
<b>Use Case ID</b>	04
<b>Actor</b>	Driver, Parking owner, Merchandiser, and security
<b>Description</b>	Already-registered users can view their profiles.
<b>Precondition</b>	Logged in through their accounts.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Select the "Manage account" option</li> <li>2. Click the "View profile" option.</li> <li>3. Systems display their profile details.</li> </ol>
<b>Post Condition</b>	None
<b>Alternative Flows</b>	None

Figure 6: View Profile

<b>Use Case</b>	Update account
<b>Use Case ID</b>	05
<b>Actor</b>	Driver, Parking owner, Merchandiser, and security
<b>Description</b>	Already registered users can update profile details that were entered in the initial registration process.
<b>Precondition</b>	Already registered as a driver, a parking owner, a merchandiser, or a security in the system. I logged in through their account.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Select the "Manage account" option.</li> <li>2. Click the "Update account" option</li> <li>3. Make the necessary changes.</li> <li>4. The system validates the changes.</li> <li>5. If the contact number or email is changed, validate the information.</li> </ol> <p>The system updates the database and notifies</p>
<b>Post Condition</b>	Profile data has been updated.
<b>Alternative Flows</b>	<p>The data entered is invalid</p> <ul style="list-style-type: none"> <li>• An error message is displayed.</li> <li>• Users can re-enter or cancel.</li> </ul>

Figure 7: Update Account

<b>Use Case</b>	Delete account
<b>Use Case ID</b>	06
<b>Actor</b>	Driver, Parking owner, Merchandiser, and security
<b>Description</b>	Already-registered users can delete their profiles.
<b>Precondition</b>	Users are logged in to the system.
<b>Main Flow</b>	<p>1. Select the "Manage account" option.      2. Click the "Delete account" option.      2. The system gets a confirmation from the user.      3. The system checks whether there is ongoing activity for the user.      4. If not, the user's password is verified.      The system updates the database and notifies</p>
<b>Post Condition</b>	The user's profile was removed.
<b>Alternative Flows</b>	<p>The user has ongoing activity          • An error message is displayed</p> <p>Entered invalid password          • An error message is displayed          • The user can re-enter or cancel</p>

Figure 8: Delete Account

<b>Use Case</b>	Change password
<b>Use Case ID</b>	07
<b>Actor</b>	Driver, Parking owner, Merchandiser, and security
<b>Description</b>	Users can change their current password.
<b>Precondition</b>	Logged in through their accounts
<b>Main Flow</b>	<p>1. Select the "Manage account" option.      2. Click the "Change Password" option.      2. The system asks for the current password.      3. The user enters the current password.      4. The system validates the current password.      5. The system asks for the new password two times.      6. The system validates and updates the new password.      7. The system displays "Password Changed Successfully".</p>
<b>Post Condition</b>	Next, login with the new password.
<b>Alternative Flows</b>	<p>Enter an existing password as new password          • The system displays, "This password is already in use."          • The user can re-enter or cancel.</p> <p>Enter two different passwords as a new password two times.          • An error message is displayed.          • The user can re-enter or cancel</p>

Figure 9: Change Password

<b>the Case</b>	Forgot password
<b>Use Case ID</b>	08
<b>Actor</b>	Driver, Parking owner, Merchandiser, and security
<b>Description</b>	Users can reset their current password.
<b>Precondition</b>	Enter the user name.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Select the "Forgot password" option.</li> <li>2. The user selects a contact number or email for the recovery pin.</li> <li>3. The system sends an OTP.</li> <li>4. The user resets the password.</li> <li>5. The system validates and updates the new password.</li> <li>6. The system displays "Password is reset Successfully".</li> </ol>
<b>Post Condition</b>	Login again to the system using a new password.
<b>Alternative Flows</b>	<p>Invalid OTP.</p> <ul style="list-style-type: none"> <li>• An error message is displayed.</li> <li>• The user can re-enter.</li> </ul> <p>Invalid password.</p> <ul style="list-style-type: none"> <li>• An error message is displayed.</li> <li>• The user can re-enter.</li> </ul>

Figure 10: Forget Password

### 3.4.1.2 Parking Owner and Merchandiser

<b>Use Case</b>	Register parking
<b>Use Case ID</b>	09
<b>Actor</b>	Parking owner, Merchandiser
<b>Description</b>	Register their property.
<b>Precondition</b>	Registered as a parking owner or merchandiser.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Select the "Register Parking" option.</li> <li>2. Enter the registration details.</li> <li>3. Verify the property.</li> <li>Update the system and notify.</li> </ol>
<b>Post Condition</b>	Property shows as a parking lot.
<b>Alternative Flows</b>	<p>Invalid details</p> <ul style="list-style-type: none"> <li>• An error message is displayed.</li> <li>• The user can re-enter the details.</li> </ul>

Figure 11: Register Parking

<b>Use Case</b>	Set prices
<b>Use Case ID</b>	10
<b>Actor</b>	Parking owner
<b>Description</b>	Set the prices for each vehicle type for a specific time limit. Set the prices for land that is rented to a merchant.
<b>Precondition</b>	Registered as a parking owner and also has a registered property.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Select the "Set prices" option.</li> <li>2. Set the prices for each vehicle type for one hour.</li> <li>3. If a parking space is rented to a merchant, set the rent. The system verifies each price and updates the database.</li> </ol>
<b>Post Condition</b>	Show the prices in the system.
<b>Alternative Flows</b>	<p>Set prices exceed the price limit</p> <ul style="list-style-type: none"> <li>• An error message is displayed.</li> <li>• The user can reset the prices.</li> </ul>

Figure 12: Set Prices

<b>Use Case</b>	Set parking slot capacity
<b>Use Case ID</b>	11
<b>Actor</b>	Parking owner, Merchandiser
<b>Description</b>	Set the number of vehicles of each type that can park on the property.
<b>Precondition</b>	Registered as a parking owner or a merchandiser and also has a registered property.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Select the "Set parking slot capacity" option.</li> <li>2. Enter the vehicle types and the capacity of each type.</li> <li>3. Verify and update the database.</li> </ol>
<b>Post Condition</b>	Show the capacity of the system.
<b>Alternative Flows</b>	None

Figure 13: Set Parking Slot Capacity

<b>Use Case</b>	Create package
<b>Use Case ID</b>	12
<b>Actor</b>	Parking owner
<b>Description</b>	Create a package for drivers.
<b>Precondition</b>	Registered as a parking owner and also has a registered property.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Select the "Create package" option.</li> <li>2. Set package details such as time period, prices, etc.</li> <li>3. The system verifies the package.</li> <li>4. Update the database.</li> </ol>
<b>Post Condition</b>	Show packages in each parking space through the system.
<b>Alternative Flows</b>	<p>Set prices exceed the price limit</p> <ul style="list-style-type: none"> <li>• An error message is displayed.</li> <li>• The user can reset the prices.</li> </ul>

Figure 14: Create Package

<b>Use Case</b>	Availability of security officer
<b>Use Case ID</b>	13
<b>Actor</b>	Parking owner, Merchandiser
<b>Description</b>	Parking owners or merchandisers can update the system to indicate whether the security officer is available or unavailable.
<b>Precondition</b>	Registered as a parking owner or a merchandiser and also has a registered property.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Select the "Availability of a security officer" option.</li> <li>2. Set the availability.</li> <li>3. Update and notify the system.</li> </ol>
<b>Post Condition</b>	Show whether the security officer is available or not.
<b>Alternative Flows</b>	None

Figure 15: Availability of Security Officer

<b>Use Case</b>	View security profile
<b>Use Case ID</b>	14
<b>Actor</b>	Parking owner, Merchandiser
<b>Description</b>	Parking owners or merchandisers can view the security officer's profile
<b>Precondition</b>	Logged as a parking owner or a merchandiser
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Select the security officer</li> <li>2. Select the "View the security officer's profile" option.</li> <li>3. View the profile</li> </ol>
<b>Post Condition</b>	None
<b>Alternative Flows</b>	None

Figure 16: View Security Profile

<b>Use Case</b>	Add security officer
<b>Use Case ID</b>	15
<b>Actor</b>	Parking owner, Merchandiser
<b>Description</b>	If there is no security officer, the parking owner or merchandiser can see the requested or available profiles of security officers and select an officer for the parking.
<b>Precondition</b>	Registered as a parking owner and also has a registered property.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Click on the "Add security officer" option.</li> <li>2. The system shows the requested security officers' profiles as well as other available profiles of security officers.</li> <li>3. Check the profiles and identify conforming officers among them.</li> <li>4. The system sends a compliance notification to the security officer.</li> <li>5. Update and notify the system.</li> </ol> <p style="text-align: center;">•</p>
<b>Post Condition</b>	A uniformed officer is assigned to that parking space.
<b>Alternative Flows</b>	<p>The selected officer is already assigned to a parking space.</p> <ul style="list-style-type: none"> <li>• The system displays, "That requested officer is not available. Select another officer.".</li> <li>• The user can select another one.</li> </ul>

Figure 17: Add Security Officer

<b>Use Case</b>	Manage parking slots
<b>Use Case ID</b>	16
<b>Actor</b>	Parking owner, Merchandiser
<b>Description</b>	Parking owners or merchandisers can add, remove, or change the quantity of vehicles in each vehicle type that they initially assigned under the "set parking slot capacity" option.
<b>Precondition</b>	Under the "set parking slot capacity" option, the vehicle quantity for each type has been set initially.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Select the "Manage parking slots" option.</li> <li>2. Make the necessary changes.</li> <li>3. Update and notify the system.</li> </ol>
<b>Post Condition</b>	Update the capacity of the parking lot.
<b>Alternative Flows</b>	None.

Figure 18: Manage Parking Slots

<b>Use Case</b>	Availability of parking spaces
<b>Use Case ID</b>	17
<b>Actor</b>	Parking owner, Merchandiser
<b>Description</b>	Parking owners or merchandisers can update the system to indicate whether the parking space is open or closed.
<b>Precondition</b>	Registered as a parking owner or a merchandiser and also has a registered property.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Select the "Availability of parking spaces" option.</li> <li>2. Set the availability.</li> <li>3. Update and notify the system.</li> </ol>
<b>Post Condition</b>	Show whether the parking space is open or closed.
<b>Alternative Flows</b>	None

Figure 19: Availability of Parking Spaces

<b>Use Case</b>	View report
<b>Use Case ID</b>	18
<b>Actor</b>	Parking owner, Merchandiser, Admin
<b>Description</b>	Parking owners or merchandisers can view the system's generated reports.
<b>Precondition</b>	Registered as a parking owner and also has a registered property.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Select the "View Report" option</li> <li>2. The system generates the report.</li> <li>3. displays the report.</li> </ol>
<b>Post Condition</b>	Report is displayed.
<b>Alternative Flows</b>	None

Figure 20: View Report

<b>Use Case</b>	Request nearby parking
<b>Use Case ID</b>	19
<b>Actor</b>	Merchandiser
<b>Description</b>	Merchandisers can request nearby packs and merge them into their own parking.
<b>Precondition</b>	Registered as a merchandiser.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Select the "Request nearby parking" option.</li> <li>2. View nearby parking spaces.</li> <li>3. Request for packing.</li> <li>4. Update and notify the system.</li> </ol>
<b>Post Condition</b>	Merge nearby parking with the merchant's parking.
<b>Alternative Flows</b>	None

Figure 21: Request Nearby Parking

<b>Use Case</b>	View merchants' requests
<b>Use Case ID</b>	20
<b>Actor</b>	Parking owner
<b>Description</b>	Parking owners can view and accept the merchandiser's requests.
<b>Precondition</b>	Registered as a parking owner or a merchandiser and also has a registered property.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Select the "View merchants' requests" option.</li> <li>2. The system displays the requests of merchants.</li> <li>3. Select the merchandiser and accept the request.</li> <li>4. Update and notify the system.</li> </ol>
<b>Post Condition</b>	Merge the parking space with the merchandiser.
<b>Alternative Flows</b>	None

Figure 22: View Merchants Request

<b>Use Case</b>	Payment
<b>Use Case ID</b>	21
<b>Actor</b>	Merchandiser
<b>Description</b>	Merchandisers can make payments for rented packing through the system.
<b>Precondition</b>	Registered as a merchandiser and requested nearby parking.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Select the "payment" option.</li> <li>2. Make payments.</li> <li>3. Update and notify the system.</li> </ol>
<b>Post Condition</b>	Merge nearby parking with the merchant's parking.
<b>Alternative Flows</b>	None

Figure 23: Payment

<b>Use Case</b>	Update availability of parking slots
<b>Use Case ID</b>	22
<b>Actor</b>	parking owner,Merchandiser,Security
<b>Description</b>	Parking owners, merchandisers, or security can update the availability of the parking slots when drivers make a cash payment.
<b>Precondition</b>	Login as a parking owner, security guard, <del>security</del> or merchandiser.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Select the “Update availability of parking slots” option.</li> <li>2. Update the availability of parking slots.</li> </ol>
<b>Post Condition</b>	Changed the availability of parking slots
<b>Alternative Flows</b>	none

Figure 24: Update Availability of Parking Slots

<b>Use Case</b>	Search Parking
<b>Use Case ID</b>	23
<b>Actor</b>	Driver, Merchandiser
<b>Description</b>	Search parking slots using location and merchandiser name
<b>Precondition</b>	Users are logged in to the system.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. The user can type the location or merchandiser name and click the “Search” button</li> <li>2. Select the parking lot and navigate to it</li> </ol>
<b>Alternate Flow</b>	The data entered is invalid <ul style="list-style-type: none"> <li>• Error message displayed</li> </ul>
<b>Post Condition</b>	The system gives the data to specified location

Figure 25: Search Parking

### 3.4.1.3 Driver

<b>Use Case</b>	Rating
<b>Use Case ID</b>	24
<b>Actor</b>	Driver
<b>Description</b>	Drivers can rate to the security officers and parkings
<b>Precondition</b>	Drivers are logged into the system.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Drivers can select actor (Security officer or Parking)</li> <li>2. Drivers can select the rating level and submit</li> </ol>
<b>Alternate Flow</b>	None
<b>Post Condition</b>	The system updates the rating level of each actor

Figure 26: Rating

<b>Use Case</b>	Subscribe package
<b>Use Case ID</b>	25
<b>Actor</b>	Driver
<b>Description</b>	Drivers can search and subscribe packages given by the parking owner
<b>Precondition</b>	Users are logged in to the system.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. The user can search for packages parking owner provides</li> <li>2. Select the packages</li> </ol>
<b>Alternate Flow</b>	None
<b>Post Condition</b>	Update the database

Figure 27: Subscribe Package

<b>Use Case</b>	Complaint for a parking
<b>Use Case ID</b>	26
<b>Actor</b>	Driver
<b>Description</b>	Drivers can complain about the parkings
<b>Precondition</b>	Drivers are logged into the system.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. The user can click “Complaint” option</li> <li>2. Give a complaint and submit</li> </ol>
<b>Alternate Flow</b>	None
<b>Post Condition</b>	Update the database.

Figure 28: Complaint for a Parking

<b>Use Case</b>	View available parking slots
<b>Use Case ID</b>	27
<b>Actor</b>	Driver
<b>Description</b>	Drivers can view availability of the parking
<b>Precondition</b>	Drivers are logged into the system.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. The user can click on the parking spot he wants.</li> <li>2. View</li> </ol>
<b>Alternate Flow</b>	None
<b>Post Condition</b>	Update the database.

Figure 29: View Available Parking Slots

<b>Use Case</b>	Manage packages
<b>Use Case ID</b>	28
<b>Actor</b>	Driver
<b>Description</b>	Drivers can unsubscribe or change the package
<b>Precondition</b>	Drivers are logged into the system.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. The user can click "Manage Packages" option</li> <li>2. Select the option you want</li> </ol>
<b>Alternate Flow</b>	None
<b>Post Condition</b>	Update the database

Figure 30: Manage Packages

<b>Use Case</b>	payment
<b>Use Case ID</b>	
<b>Actor</b>	Driver
<b>Description</b>	Drivers can payment
<b>Precondition</b>	Drivers are logged into the system and scanned QR.
<b>Main Flow</b>	1.The user can click "pay" option 2.system generate the cost 3.select payment method. 4.Do payment 5.update the system and release the slot
<b>Alternate Flow</b>	None
<b>Post Condition</b>	If do a cash payment then security or owner need to be release the slot

Figure 31: Payment

<b>Use Case</b>	Register vehicle
<b>Use Case ID</b>	28
<b>Actor</b>	Driver
<b>Description</b>	Drivers can unsubscribe or change the package
<b>Precondition</b>	Drivers are logged into the system.
<b>Main Flow</b>	1.The user can click "Register vehicle" option 2.Enter the details. 3. Verify the vehicle. Update the system and notify.
<b>Alternate Flow</b>	Invalid details <ul style="list-style-type: none"> <li>• An error message is displayed.</li> <li>• The user can re-enter the details.</li> </ul>
<b>Post Condition</b>	Update the database.

Figure 32: Register Vehicle

<b>Use Case</b>	Scan QR
<b>Use Case ID</b>	28
<b>Actor</b>	Driver
<b>Description</b>	Drivers can scan QR when enter a parking for calculate the cost
<b>Precondition</b>	Drivers are logged into the system.
<b>Main Flow</b>	1.The user can click "Scan QR" option 2.Scan the QR
<b>Alternate Flow</b>	None
<b>Post Condition</b>	Unavailable the parking slot

Figure 33: Scan QR

#### 3.4.1.4 Admin

<b>Use Case</b>	Set the price limit
<b>Use Case ID</b>	29
<b>Actor</b>	Admin
<b>Description</b>	Admin controls maximum price limit for a parking slot
<b>Precondition</b>	Logged as admin
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Go to the Dashboard.</li> <li>2. Go to the set price limit option.</li> <li>3. Set the maximum price.</li> <li>4. Submit data.</li> </ol>
<b>Post Condition</b>	Updated the database
<b>Alternative Flows</b>	none

Figure 34: Set the Price Limit

<b>Use Case</b>	Set profit rate
<b>Use Case ID</b>	30
<b>Actor</b>	Admin
<b>Description</b>	Admin can set the profit rate. How much percentage to keep for providing the service.
<b>Precondition</b>	Logged as admin
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Go to the Dashboard.</li> <li>2. Select the profit rate</li> <li>2. Set percentage.</li> <li>3. Submit.</li> </ol>
<b>Post Condition</b>	Data has been updated.
<b>Alternative Flows</b>	none

Figure 35: Set Profit Rate

<b>Use Case</b>	Access controlling
<b>Use Case ID</b>	31
<b>Actor</b>	Admin
<b>Description</b>	Admin can control the accounts of users
<b>Precondition</b>	Logged as admin
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Admin can suspend users.</li> <li>2. Admin can delete user accounts.</li> <li>3. View account.</li> </ol>
<b>Post Condition</b>	None
<b>Alternative Flows</b>	None

Figure 36: Access Controlling

<b>Use Case</b>	Verify parking space
<b>Use Case ID</b>	32
<b>Actor</b>	Admin
<b>Description</b>	Check details to validate the parking space.
<b>Precondition</b>	Logged as admin
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Check registered details.</li> <li>2. Validate details.</li> <li>3. Accept or decline the request.</li> <li>4. Register to the system.</li> </ol>
<b>Post Condition</b>	Added to the database successfully.
<b>Alternative Flows</b>	None

Figure 37: Verify Parking Space

### 3.4.1.5 Security

<b>Use Case</b>	Request for job
<b>Use Case ID</b>	33
<b>Actor</b>	Security
<b>Description</b>	Security officers can request for security jobs
<b>Precondition</b>	Logged as security
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Select the “View available jobs” option.</li> <li>2. Select a suitable job and click “Request”.</li> </ol>
<b>Post Condition</b>	The system sends a notification to the parking owner
<b>Alternative Flows</b>	none

Figure 38: Request for Job

<b>Use Case</b>	View available jobs
<b>Use Case ID</b>	34
<b>Actor</b>	Security
<b>Description</b>	Security officers can view available jobs
<b>Precondition</b>	Logged as security
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Select the “View available jobs” option</li> <li>2. View jobs</li> </ol>
<b>Post Condition</b>	The system sends a notification to the relevant parking owner
<b>Alternative Flows</b>	none

Figure 39: View Available Jobs

<b>Use Case</b>	View capacity
<b>Use Case ID</b>	35
<b>Actor</b>	Security
<b>Description</b>	Security officers can view capacity of the parking space
<b>Precondition</b>	Logged as security
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Select the “View capacity” option.</li> <li>2. View the capacity of the relevant parking space.</li> </ol>
<b>Post Condition</b>	The system shows available parking spaces.
<b>Alternative Flows</b>	none

Figure 40: View Capacity

## 4 Proposed System's Architecture

### 4.1 Component Diagram

The below shown figure shows the Component Diagram of the system

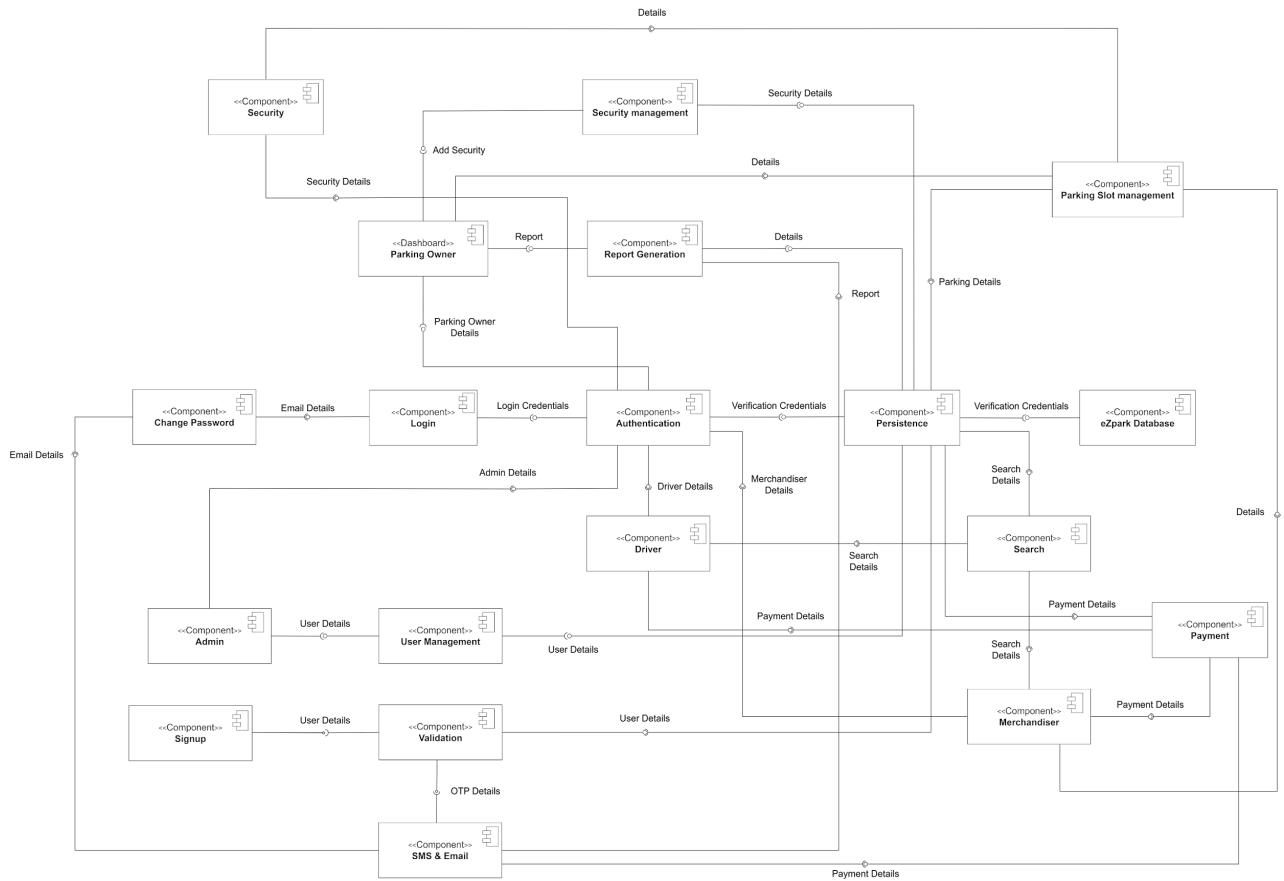


Figure 41: Component Diagram

Component Diagram link - [Component Diagram](#).

### 4.2 Component Description

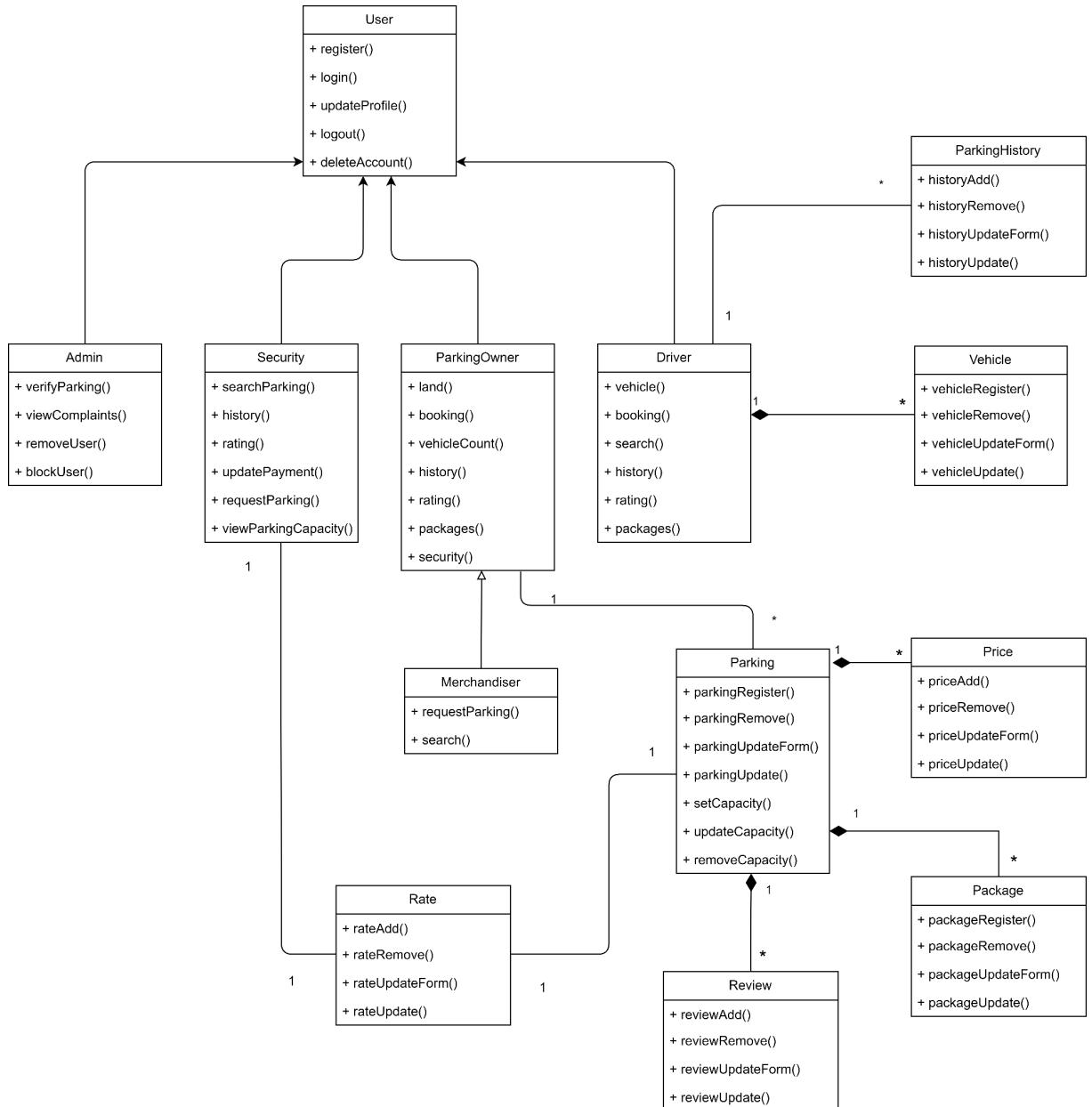
Component	Description
Login	login UI
Authentication	validate user login details
Change password	allow to change the password
SMS and E-mail	to send an OTP to verify registration and incase of forgotten password and the case of payment and to send the

	generated report to the user
Admin	admin dashboard and UIs
User management	admin can manage users according to their behaviour
Signup	user provide informations to register to the system
Validation	validate inserted details of user and response and trigger the SMS and Email component to notify the user
Driver	driver functionalities and UI
eZpark database	store all the data of the system
Persistence	create middle state between database and app as a best practice
Search	driver and merchandiser search for a parking and get the results according to their requirements from the middle state
Payment	handle payment
Parking slot management	parking owner and merchandiser manage parkings through this component
Report generation	this component generate revenue reports for land owner and merchandiser
Parking owner	parking owner functionalities and UI
Merchandiser	merchandiser functionalities and UI, also can request parking spaces from land owners to attach to their parking
Security	security functionalities and UI
Security management	this component manage the manual payments and mange details of existing parking spaces

# 5 System's Design

## 5.1 Class Diagrams

Class diagram link :[class diagram](#)



## 5.2 ER Diagram

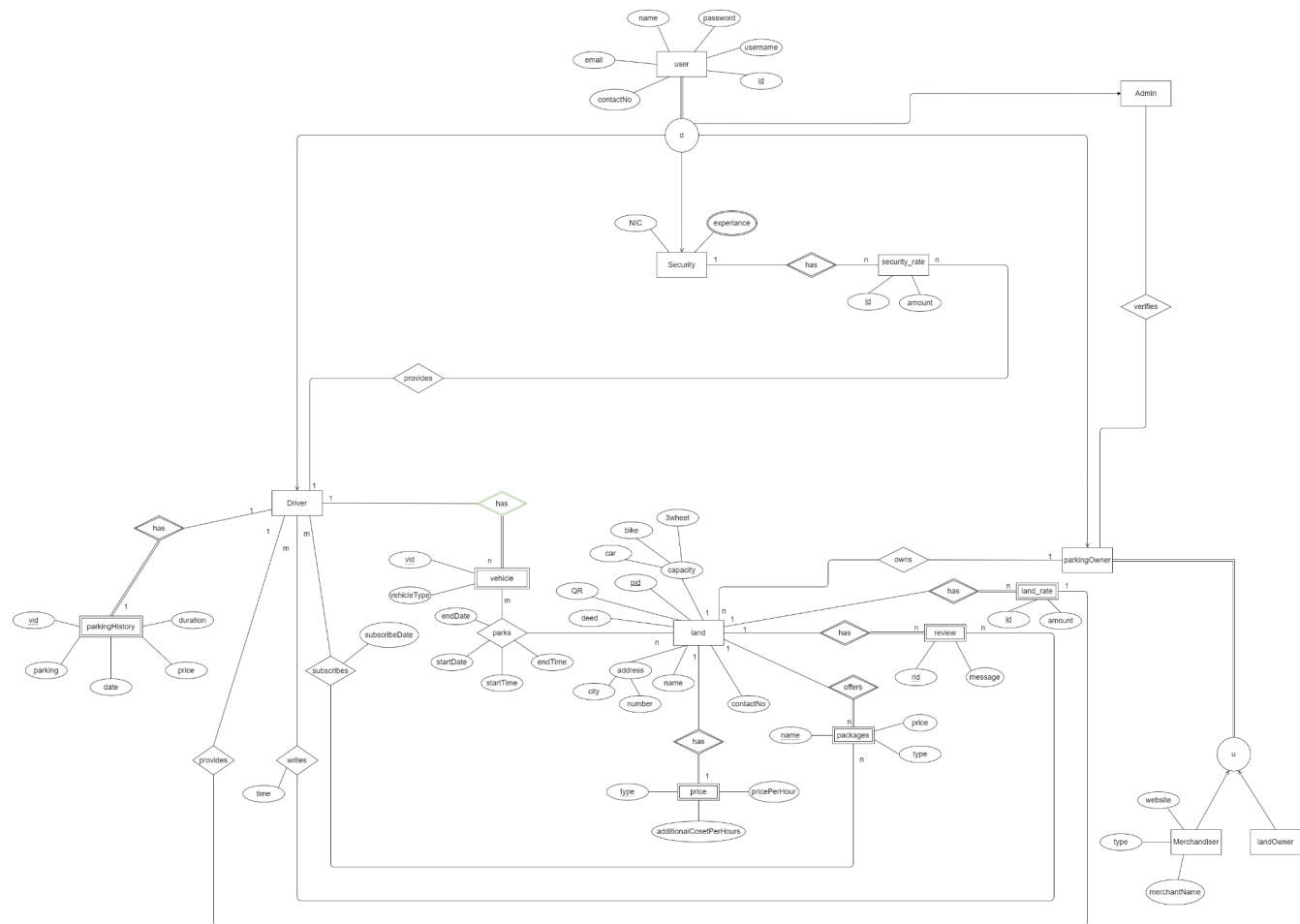


Figure 42: ER diagram

ER Diagram link :[ER diagram](#)

## 5.3 Sequence Diagrams

### 5.3.1 All users

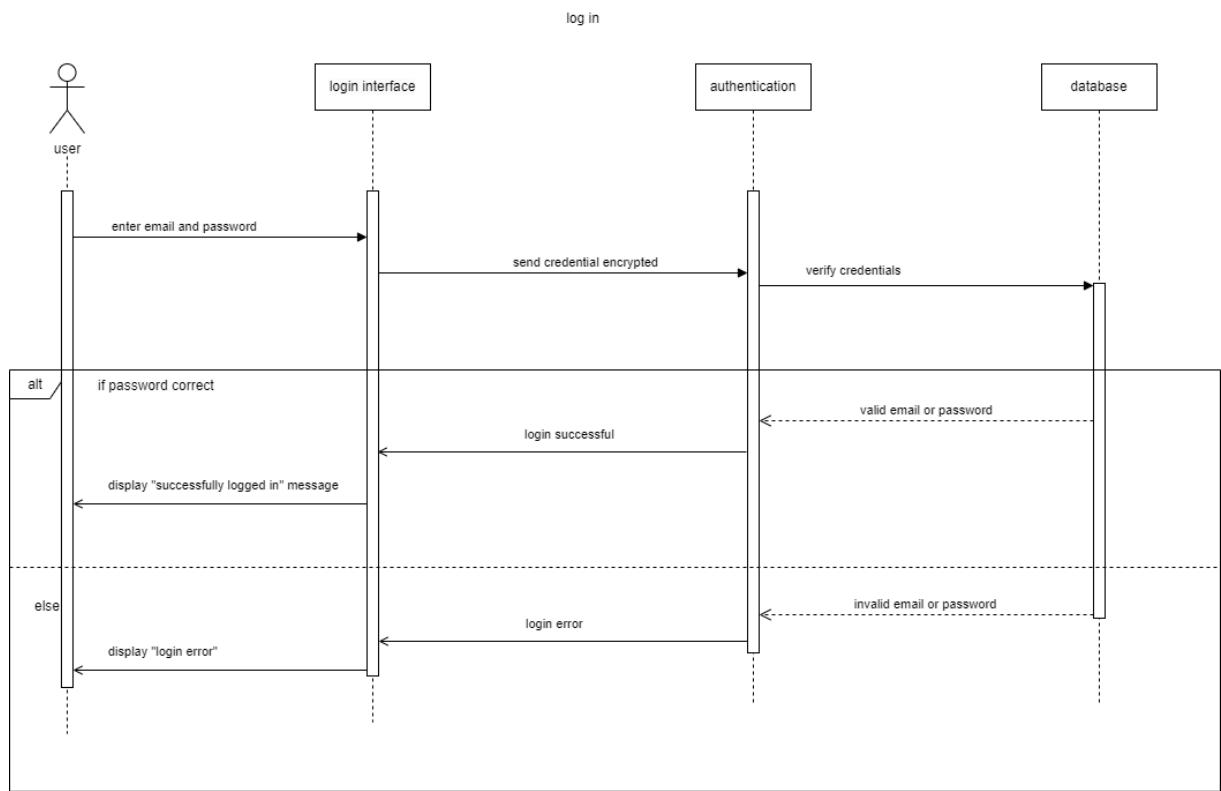


Figure 43:Log In

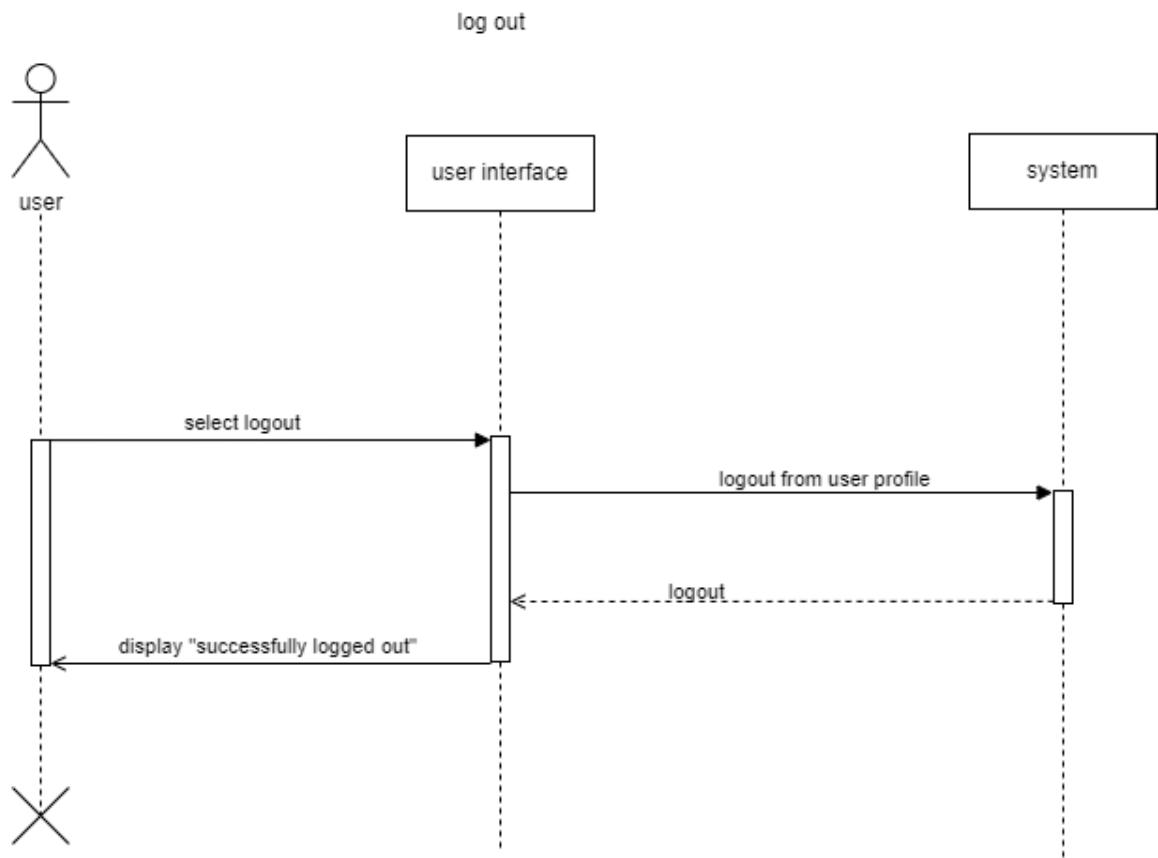


Figure 44:Log Out

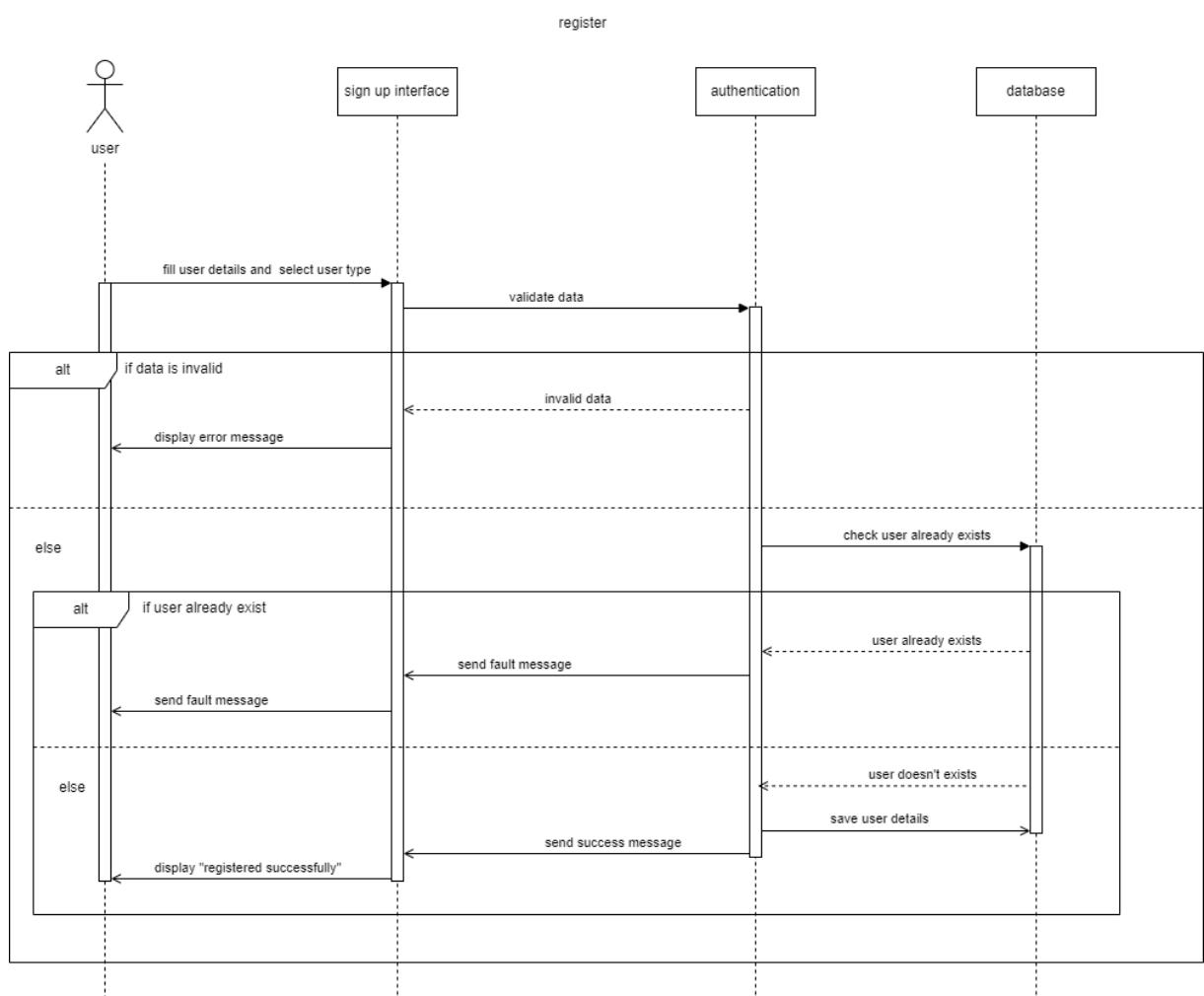


Figure 45: Register

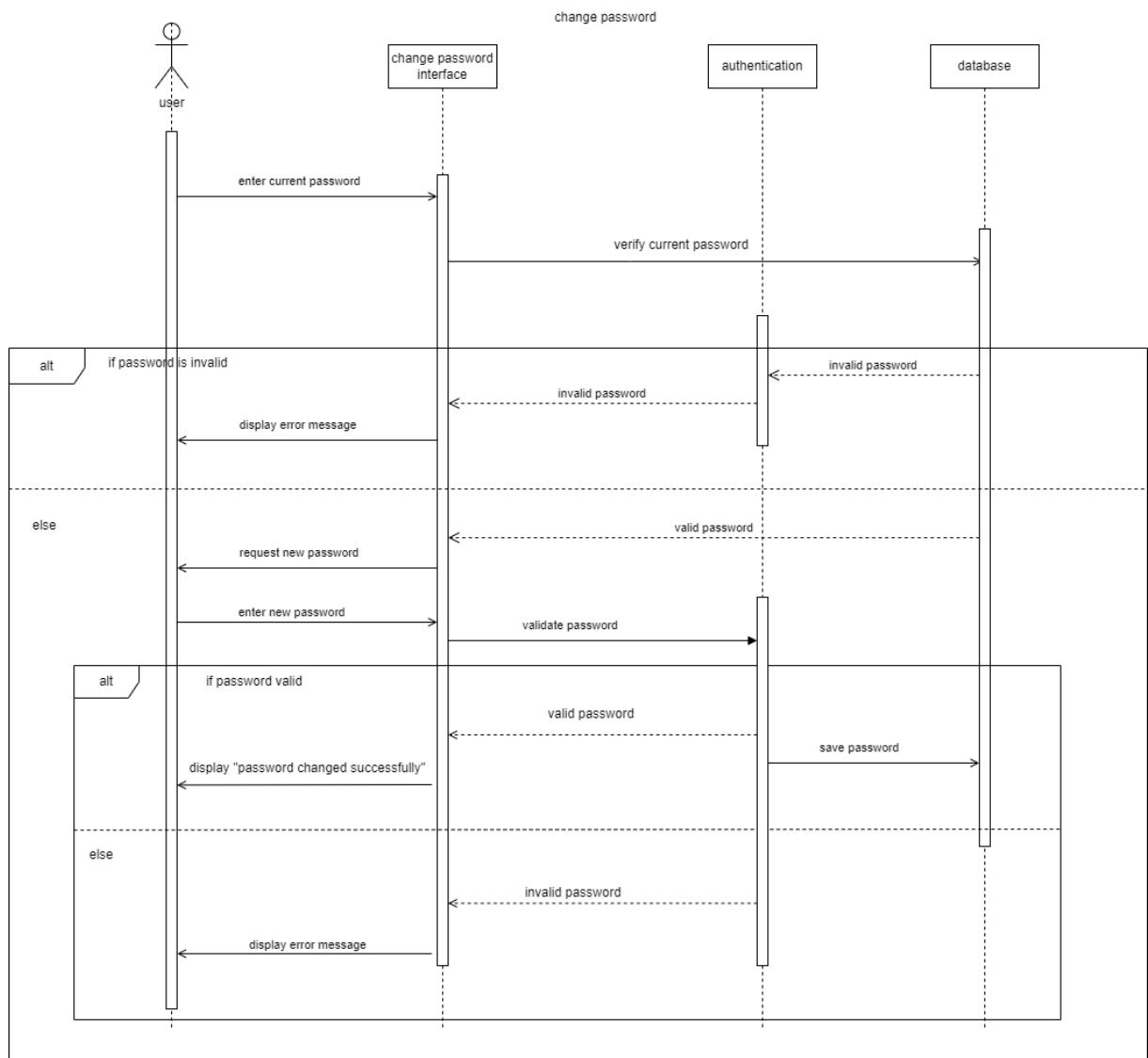


Figure 46:Change Password

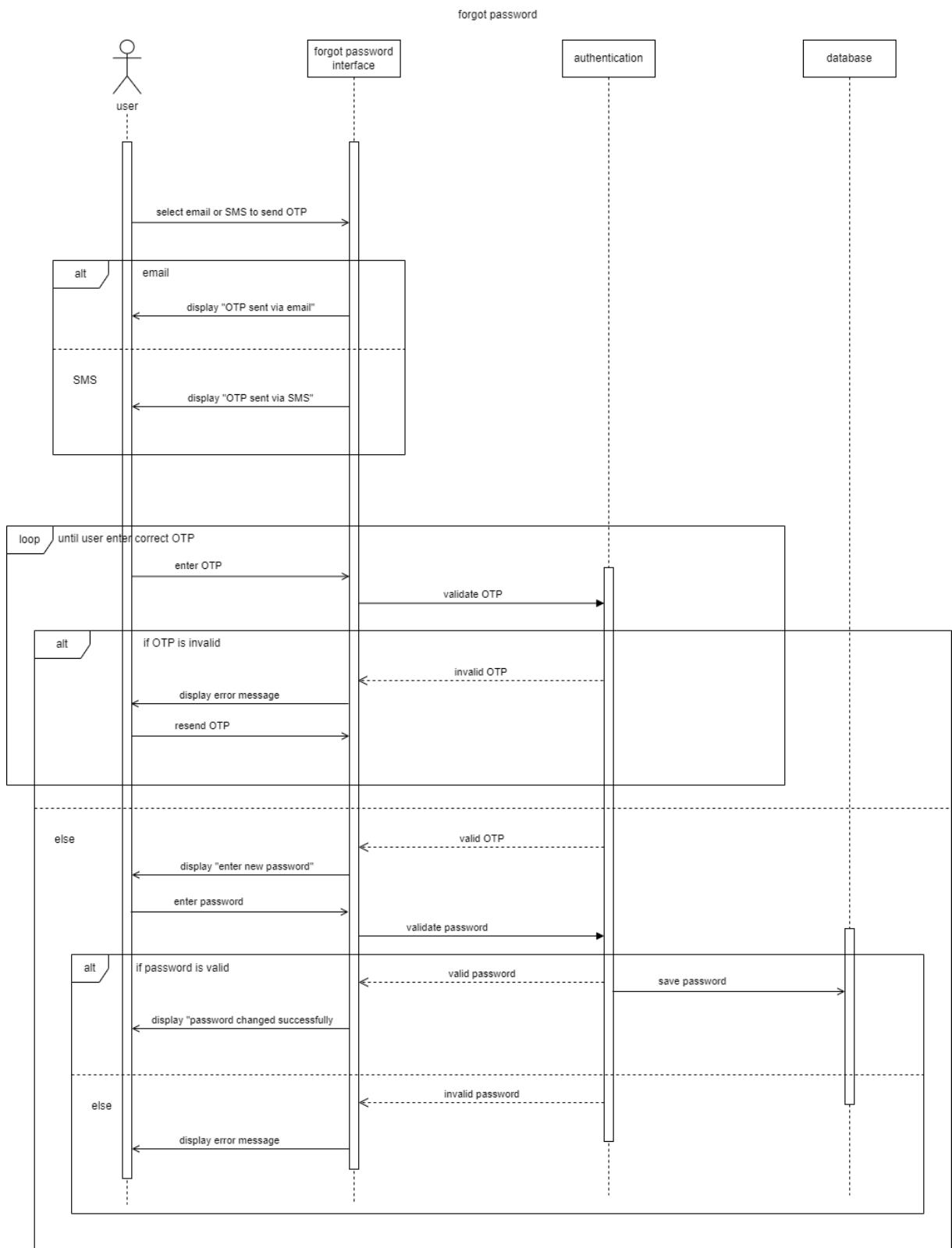


Figure 47:Forgot Password

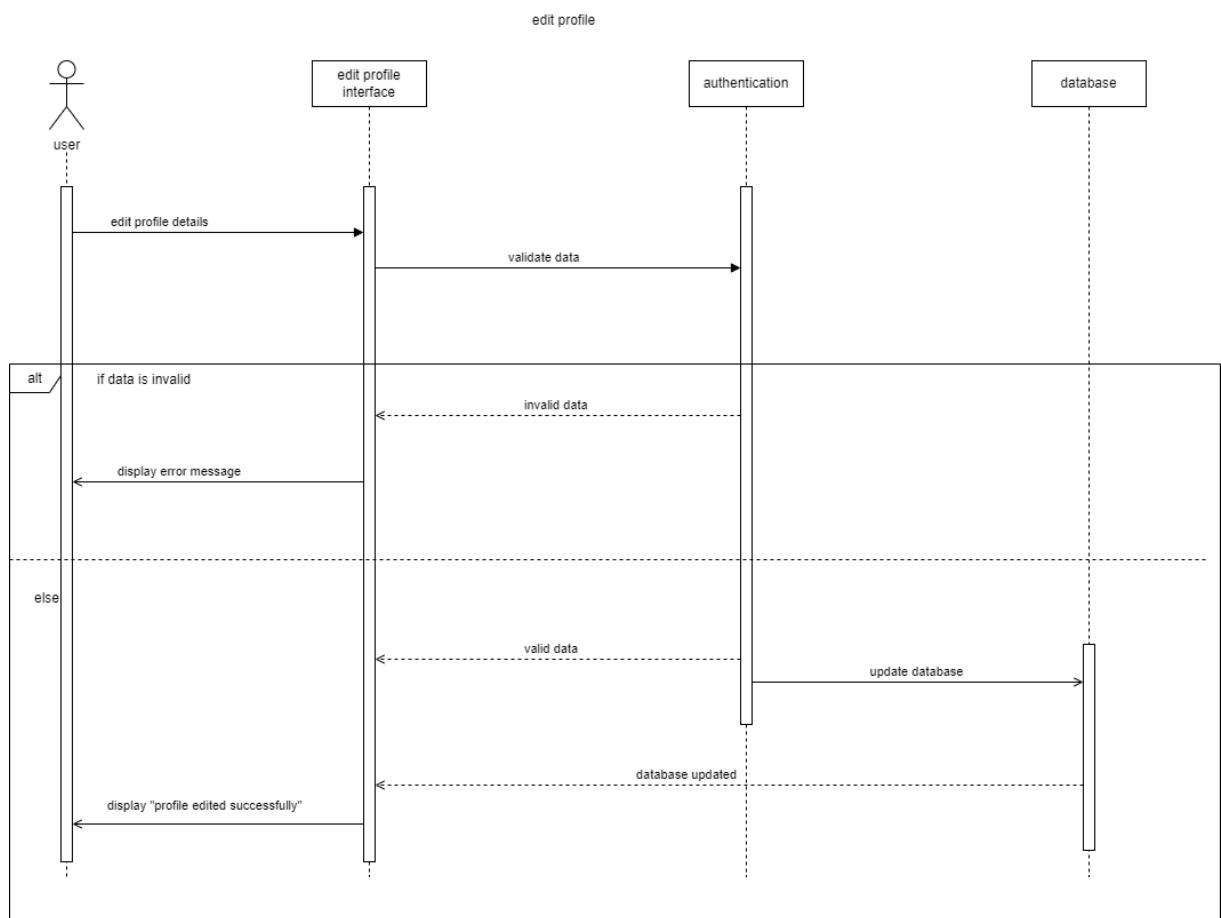


Figure 48: Edit Profile

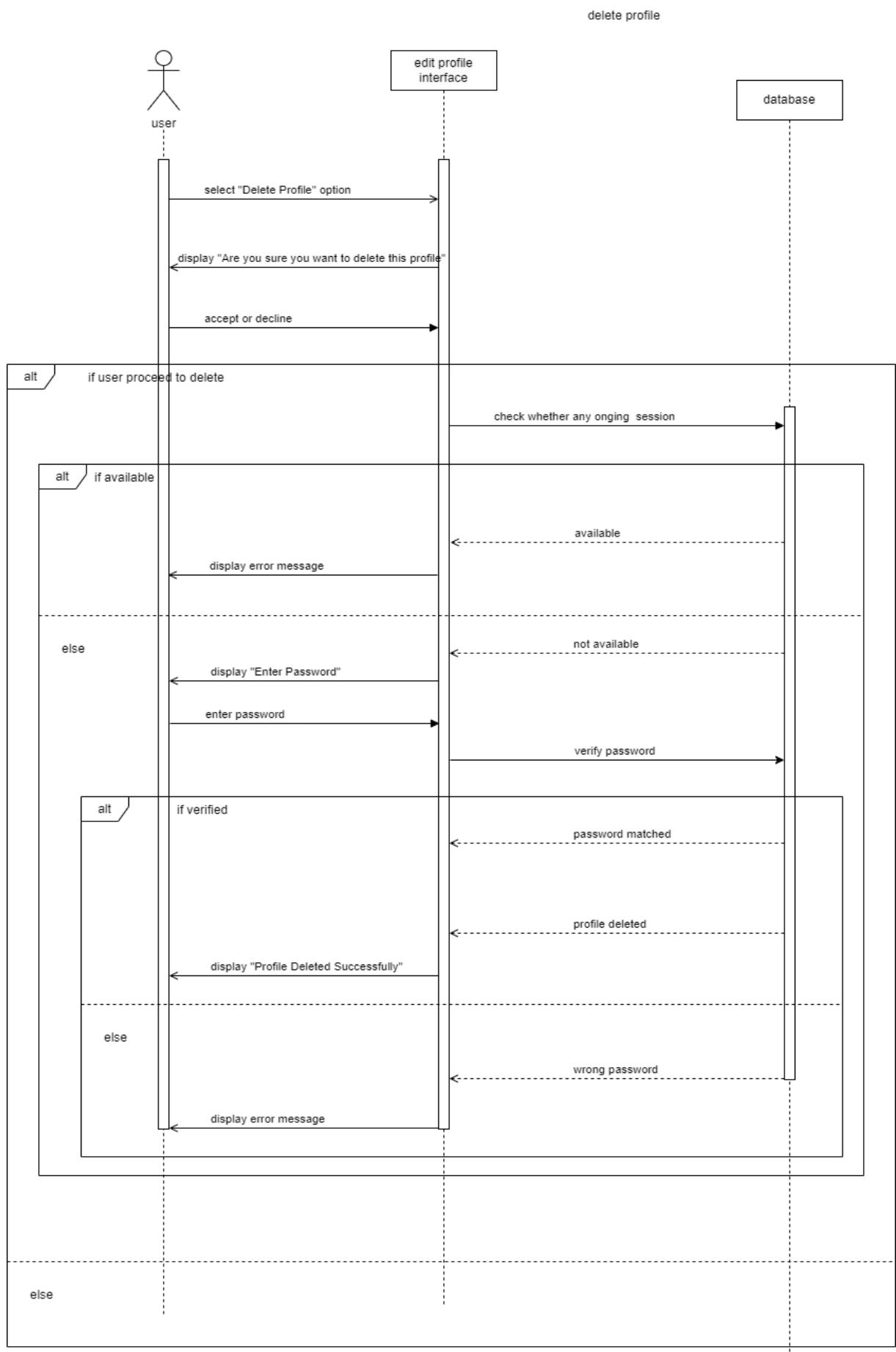


Figure 49:Delete Profile

### 5.3.2 Parking Owner and Merchandiser

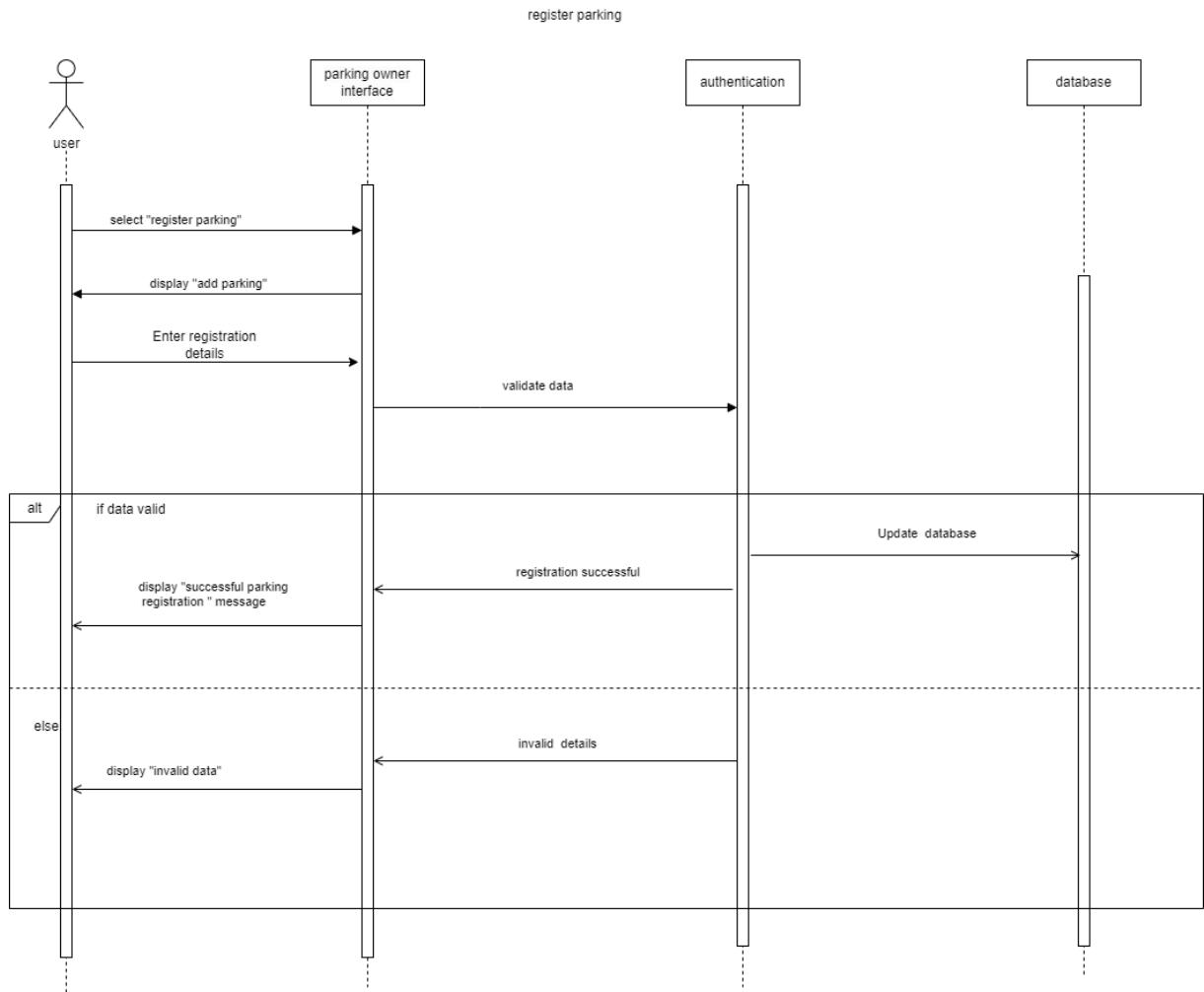


Figure 50: Register a parking space

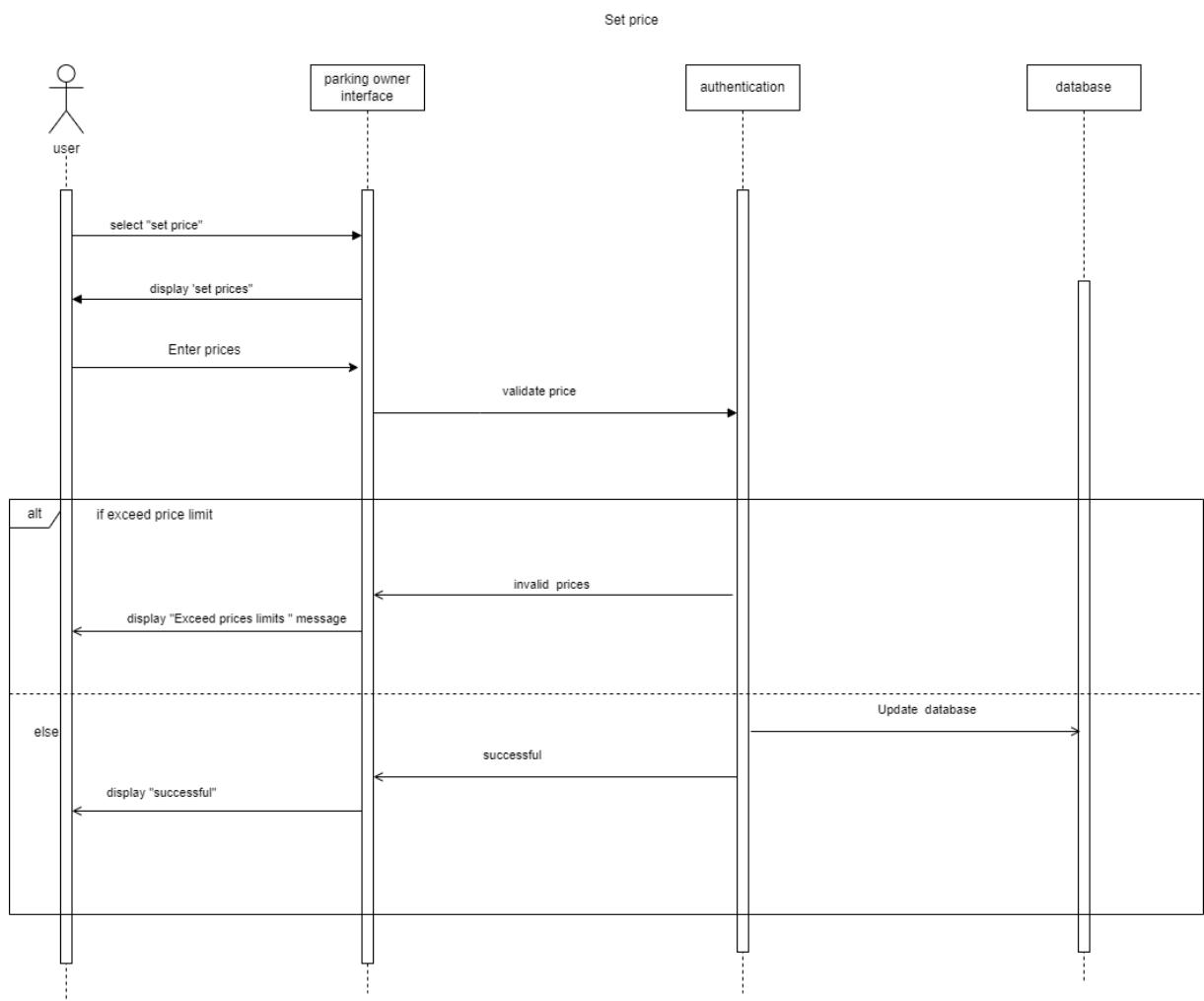


Figure 51: Set Price

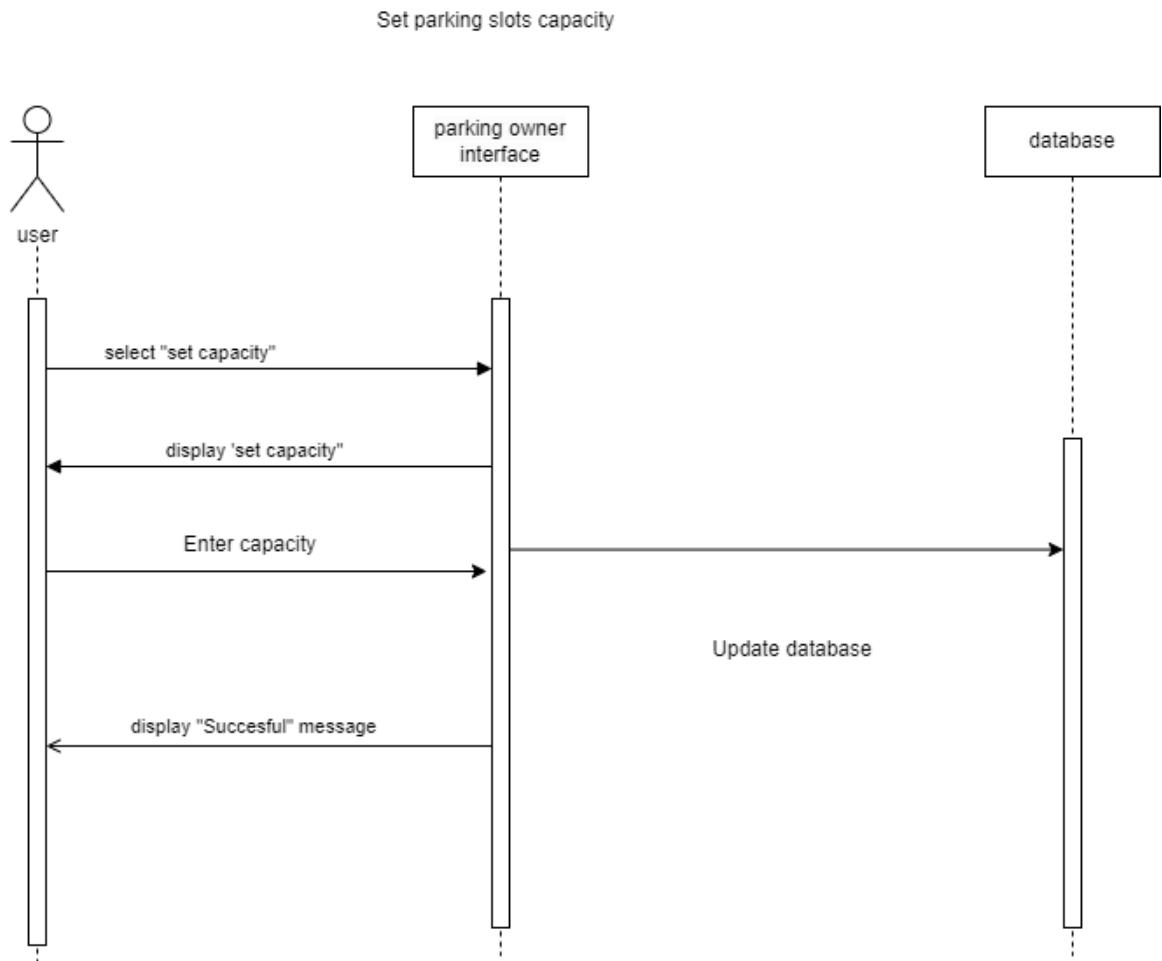


Figure 52: Set parking slot capacity

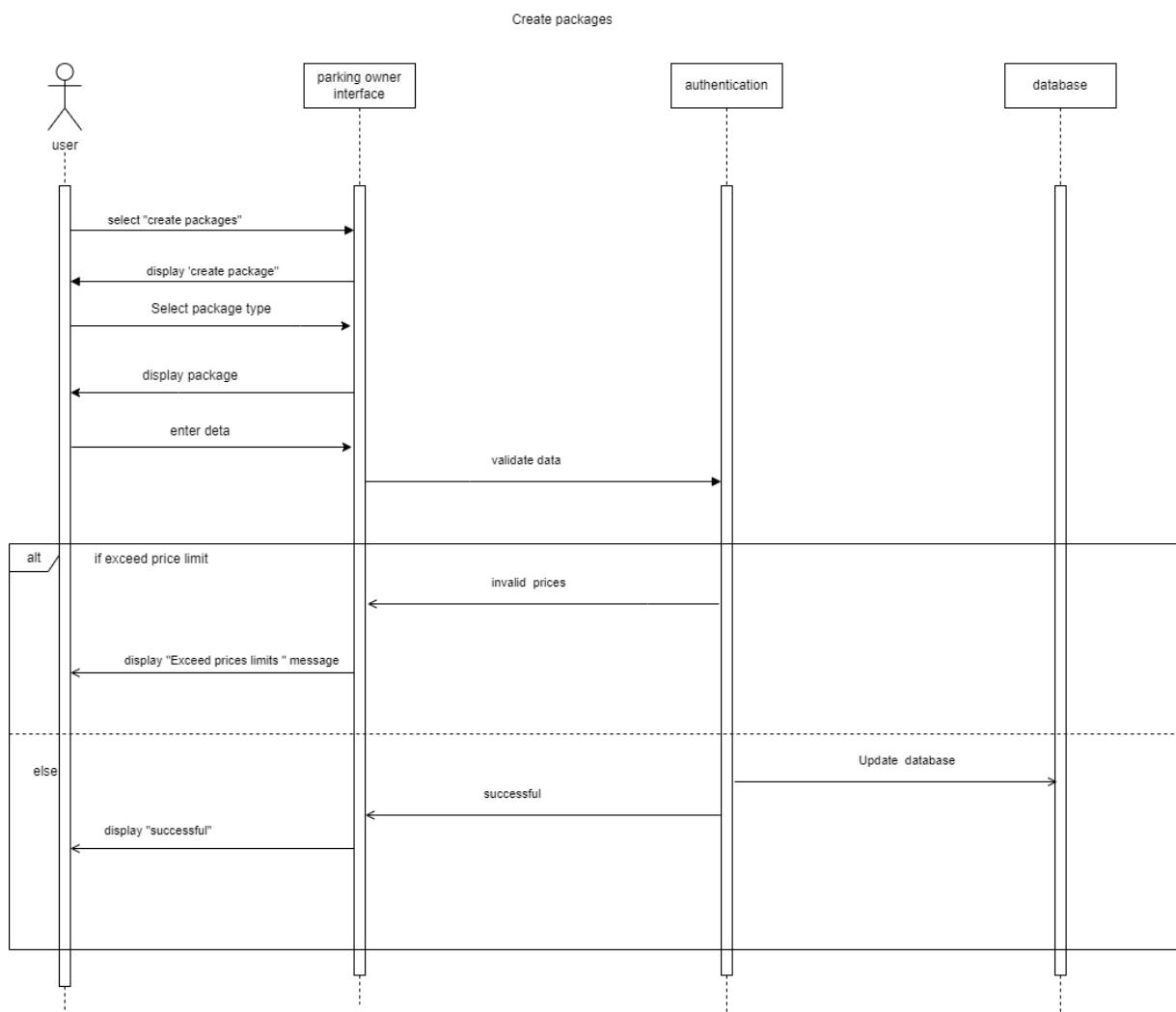


Figure 53:Create packages

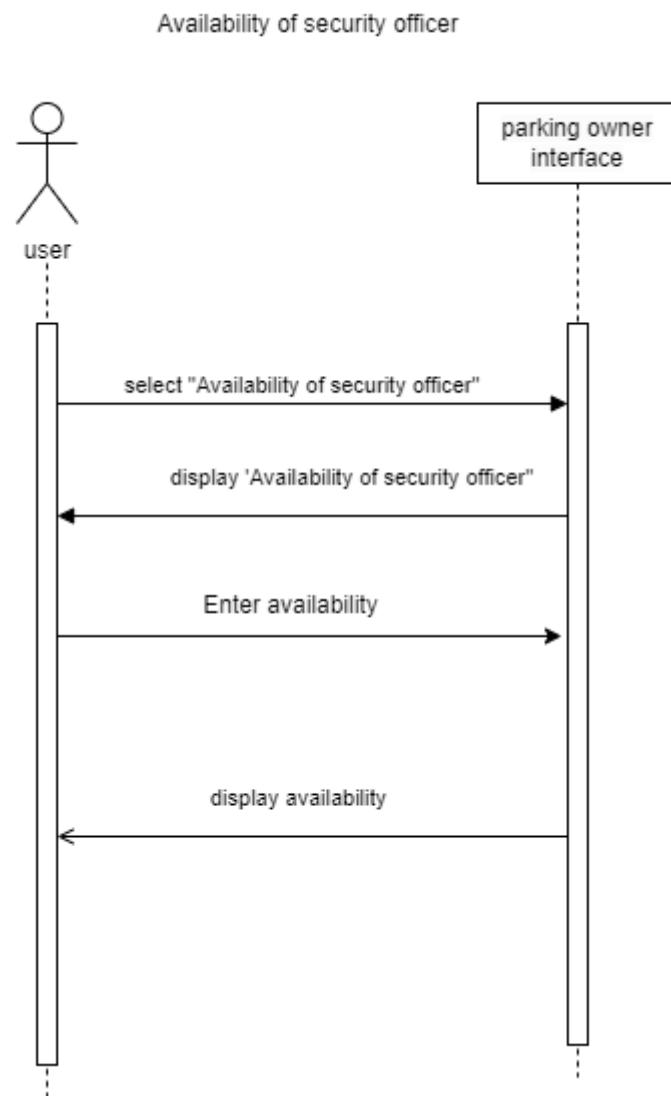


Figure 54: Availability of security Officer

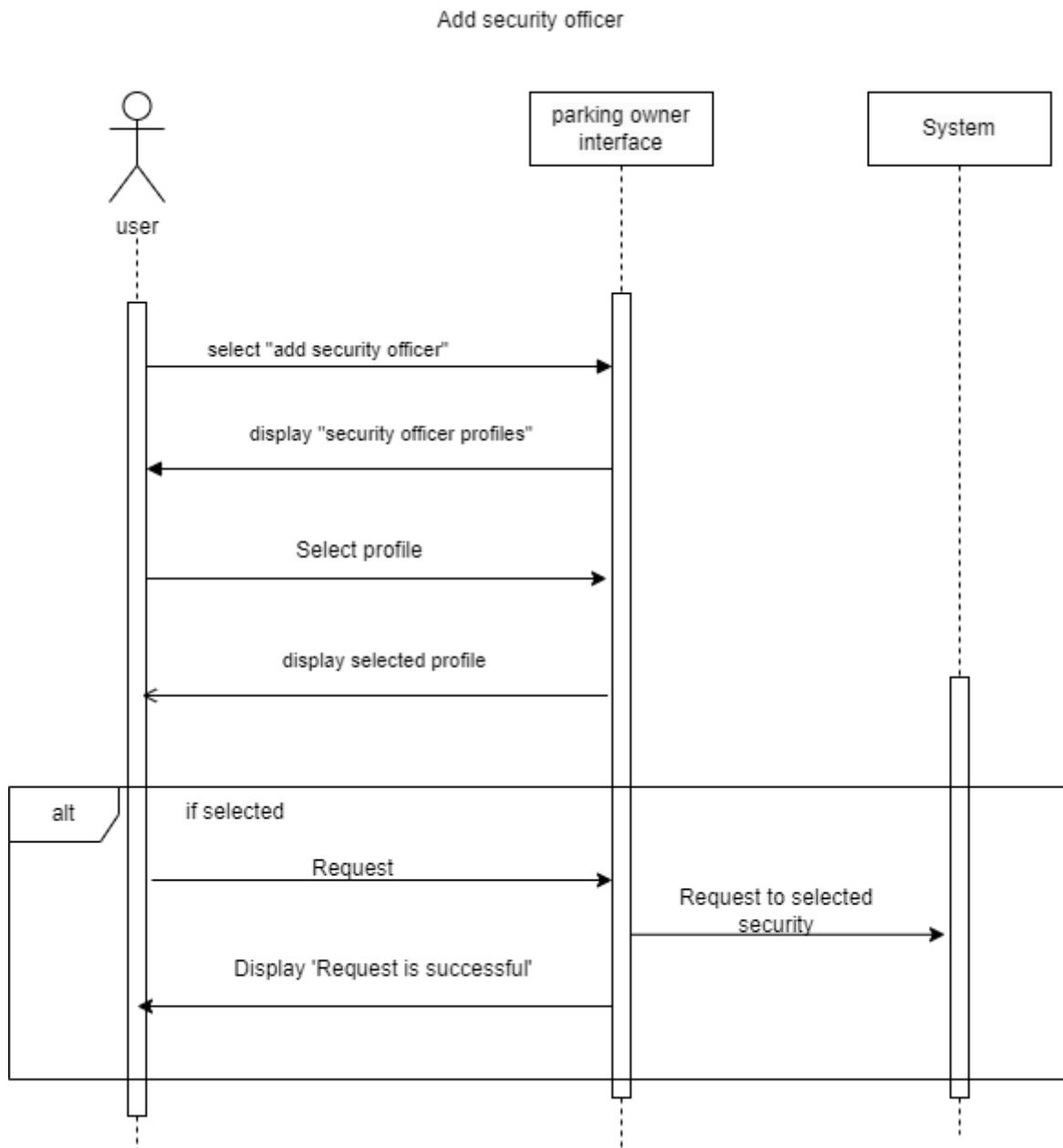


Figure 55: Add security officer

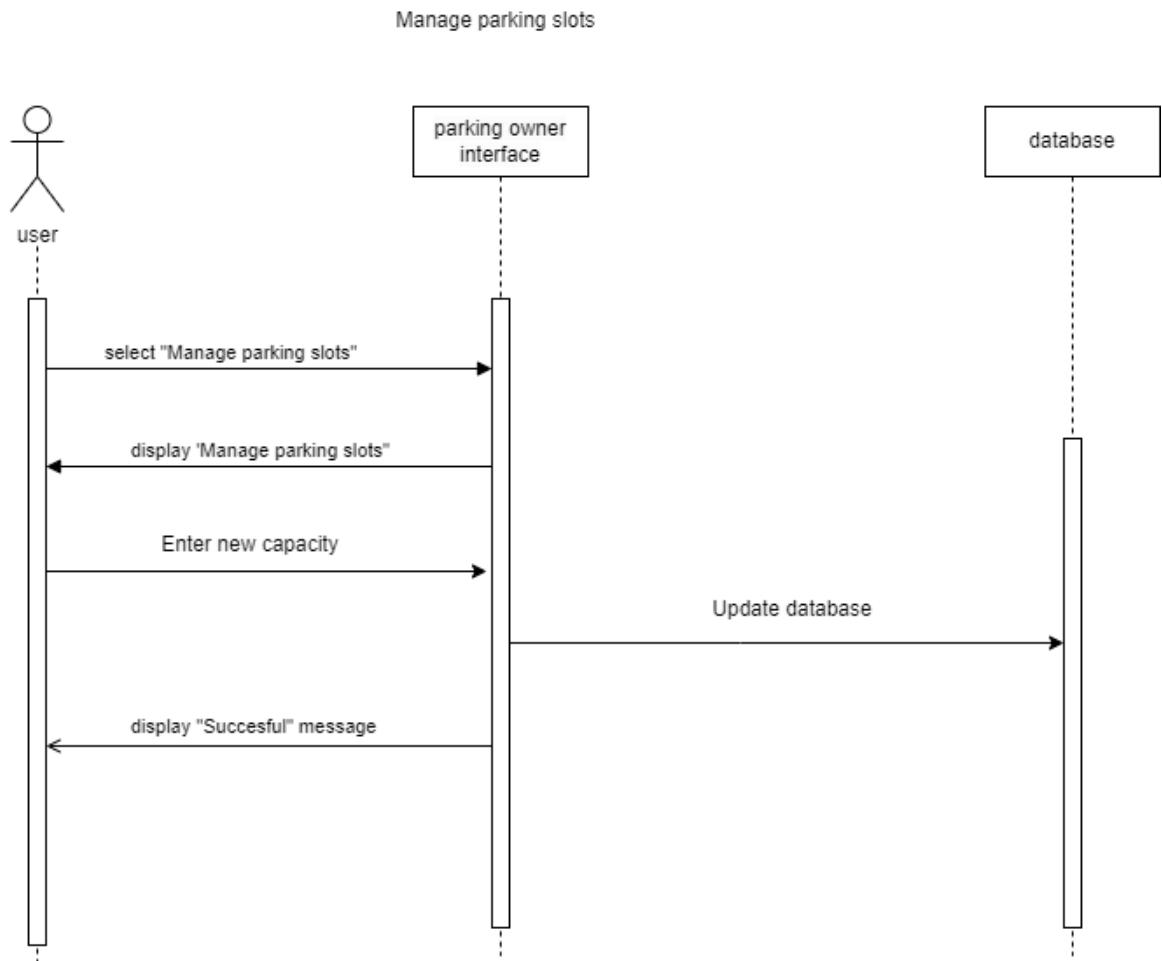


Figure 56: Manage parking slots

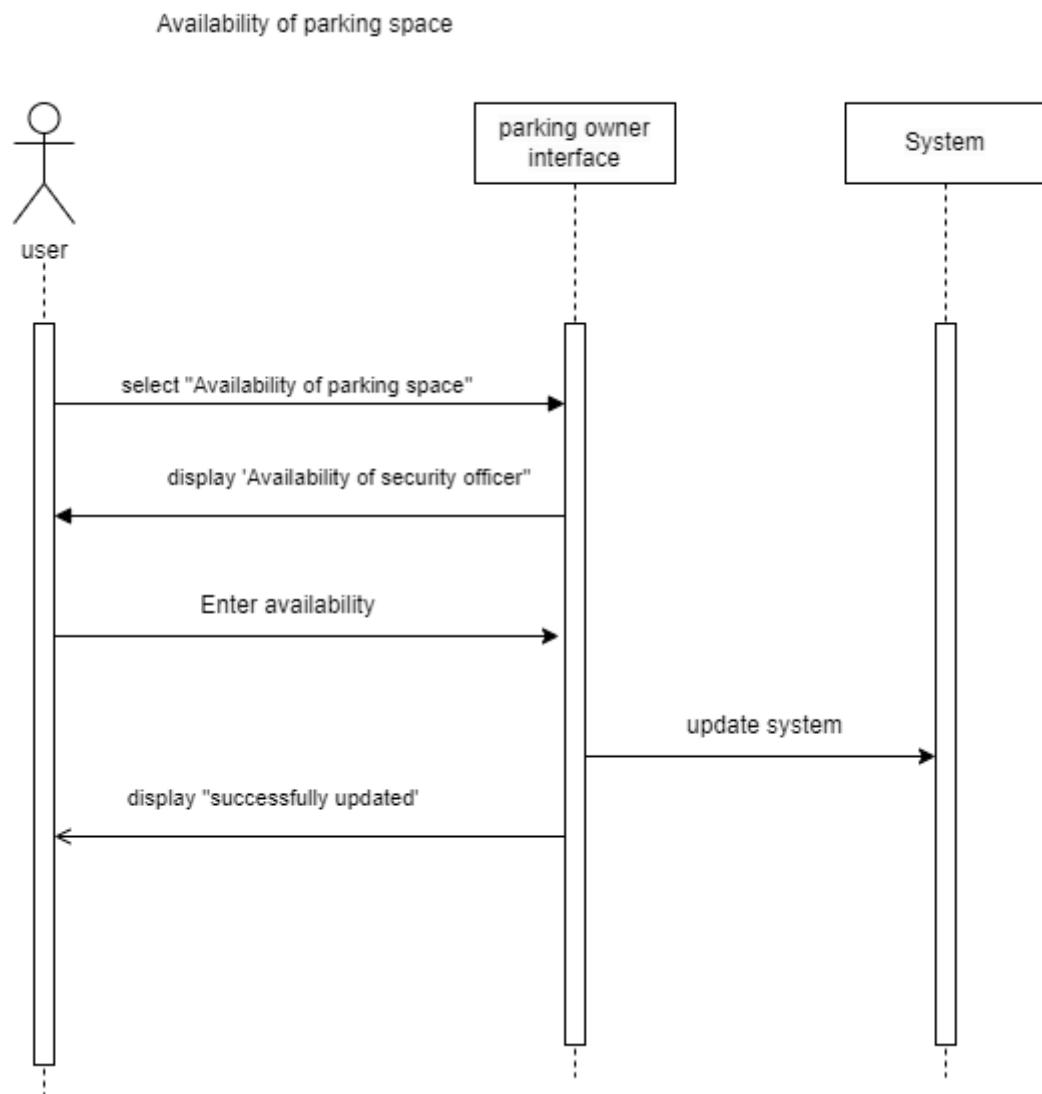


Figure 57: Availability of parking space

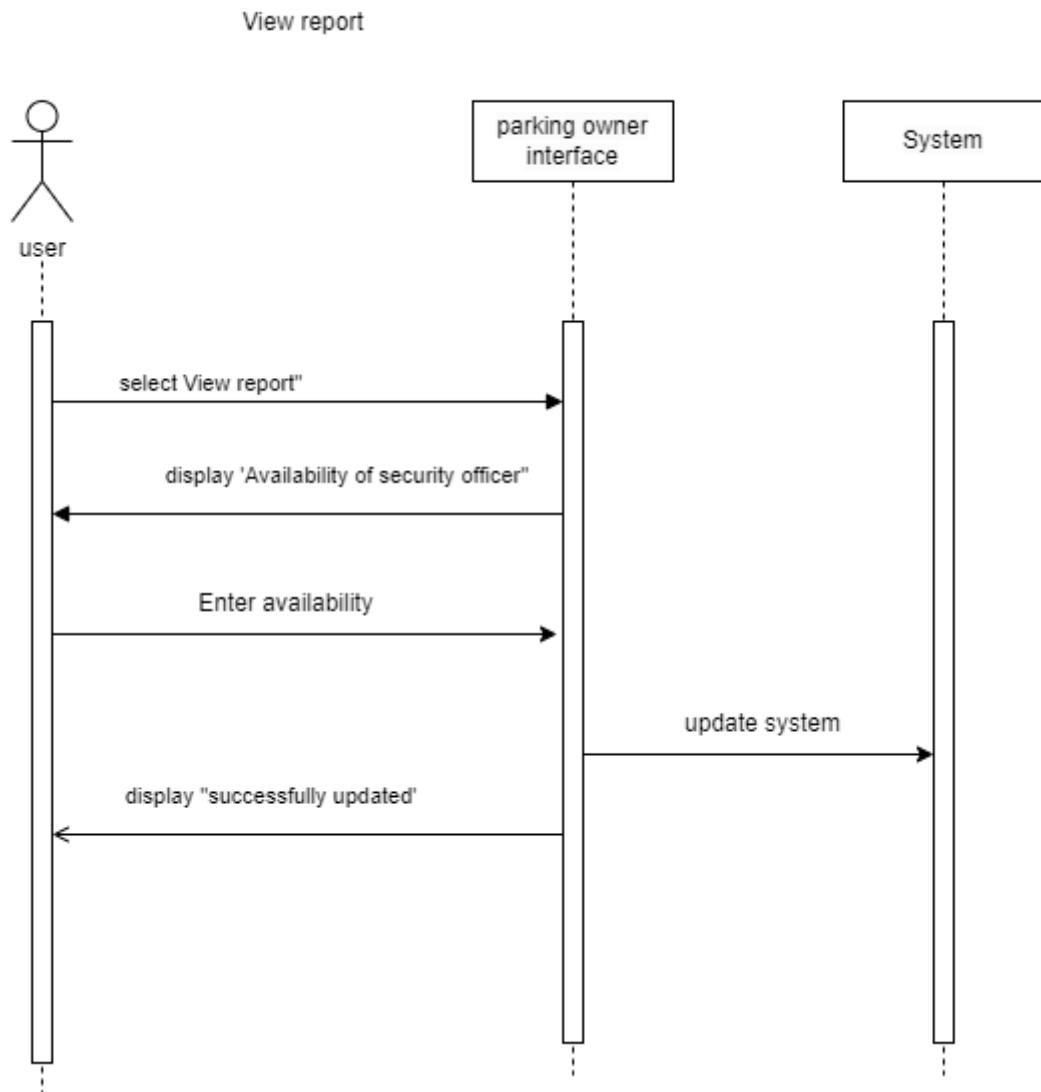


Figure 58:View report

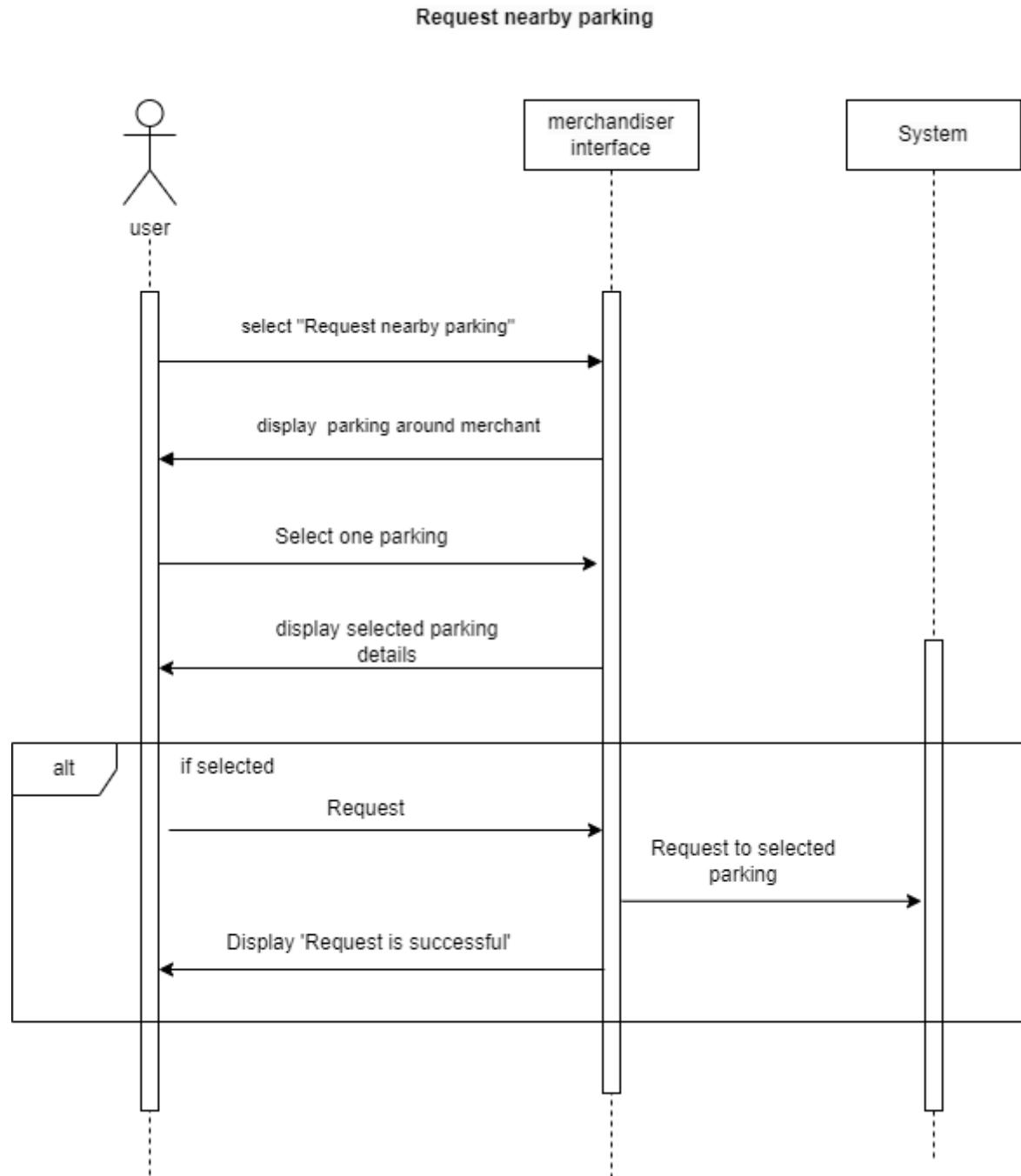


Figure 59:Request nearby parking

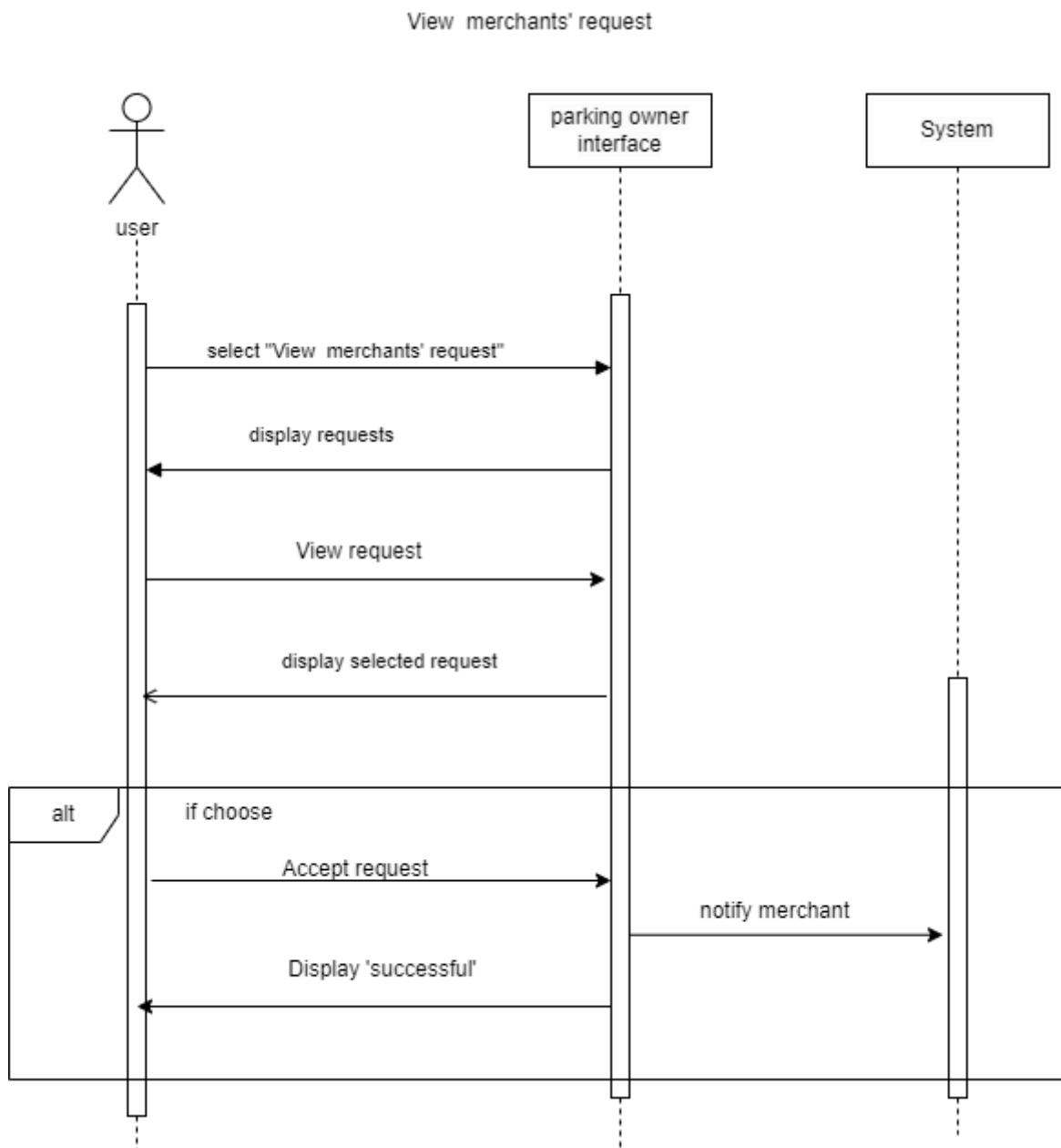


Figure 60:View merchant's request

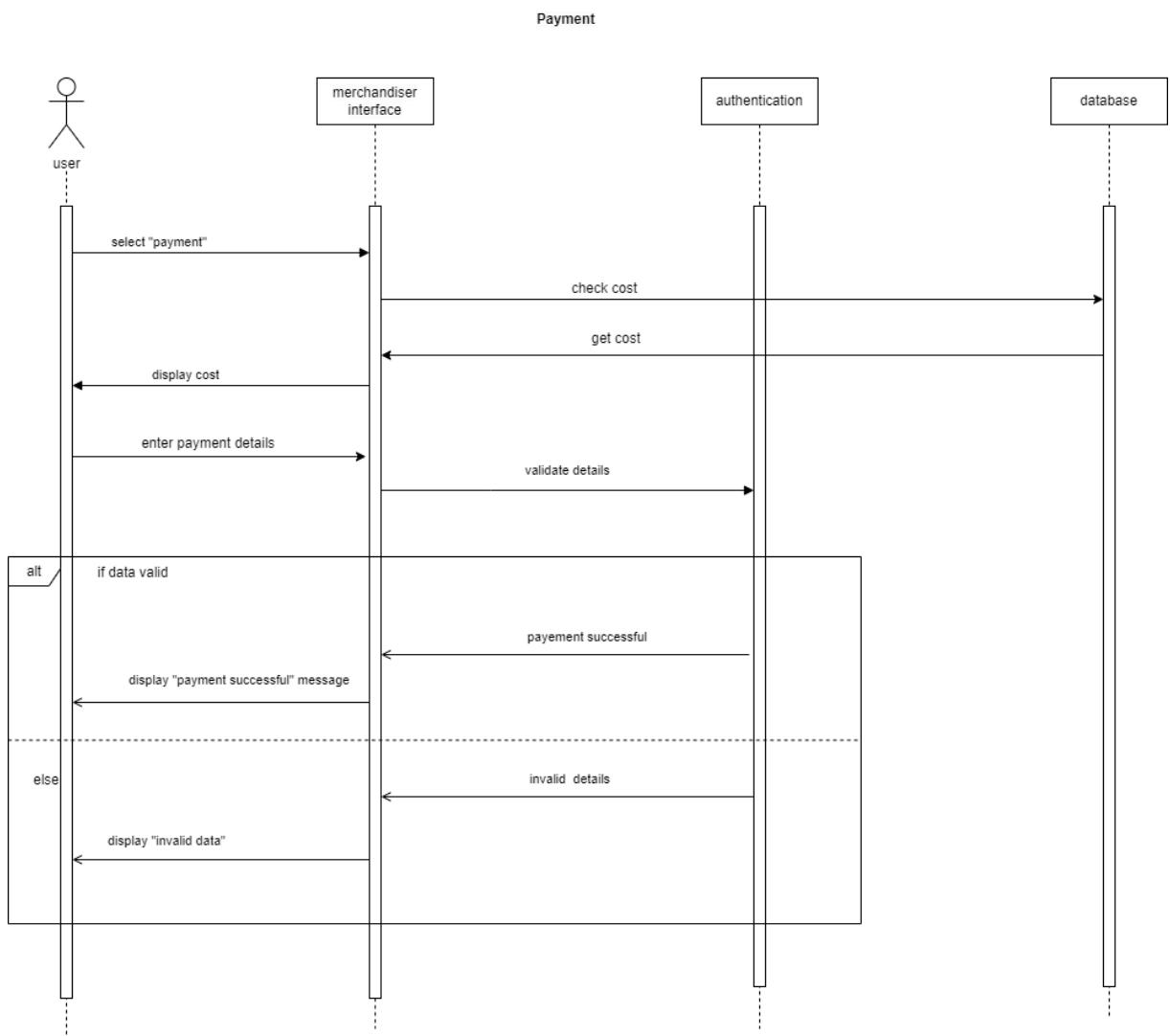


Figure 61:Payment

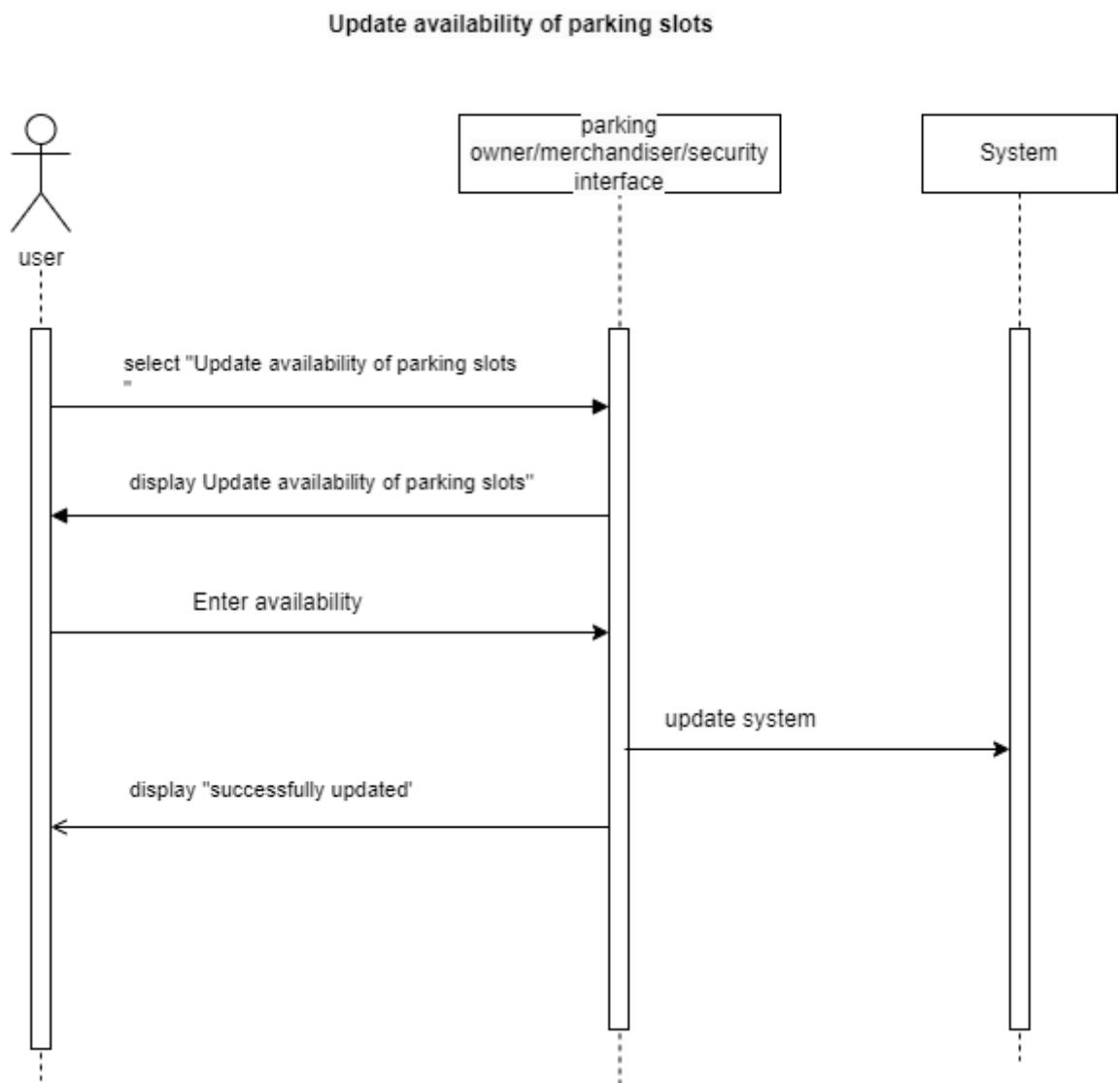


Figure 62:Update availability of parking slots

### 5.3.3 Driver

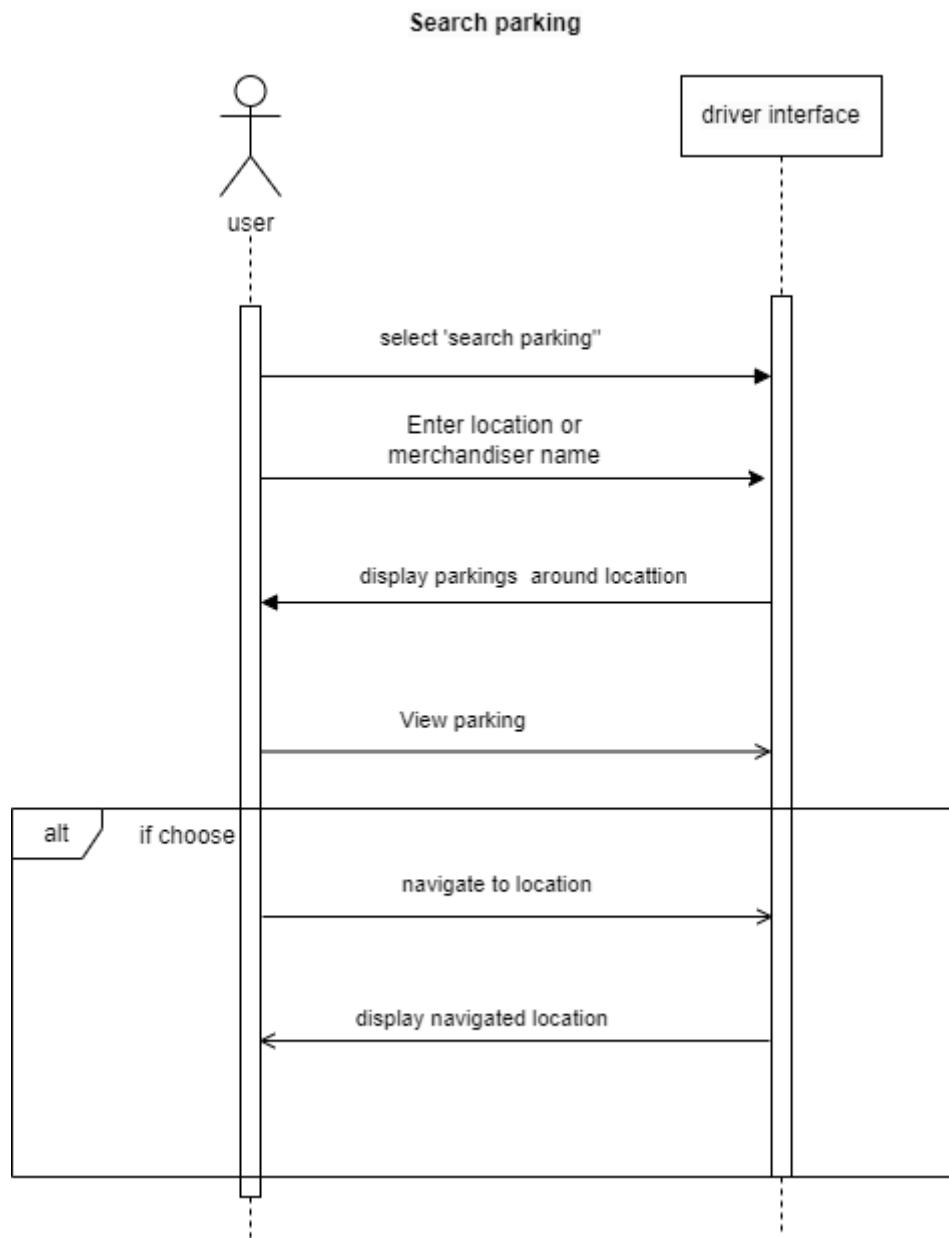


Figure 62:Search parking

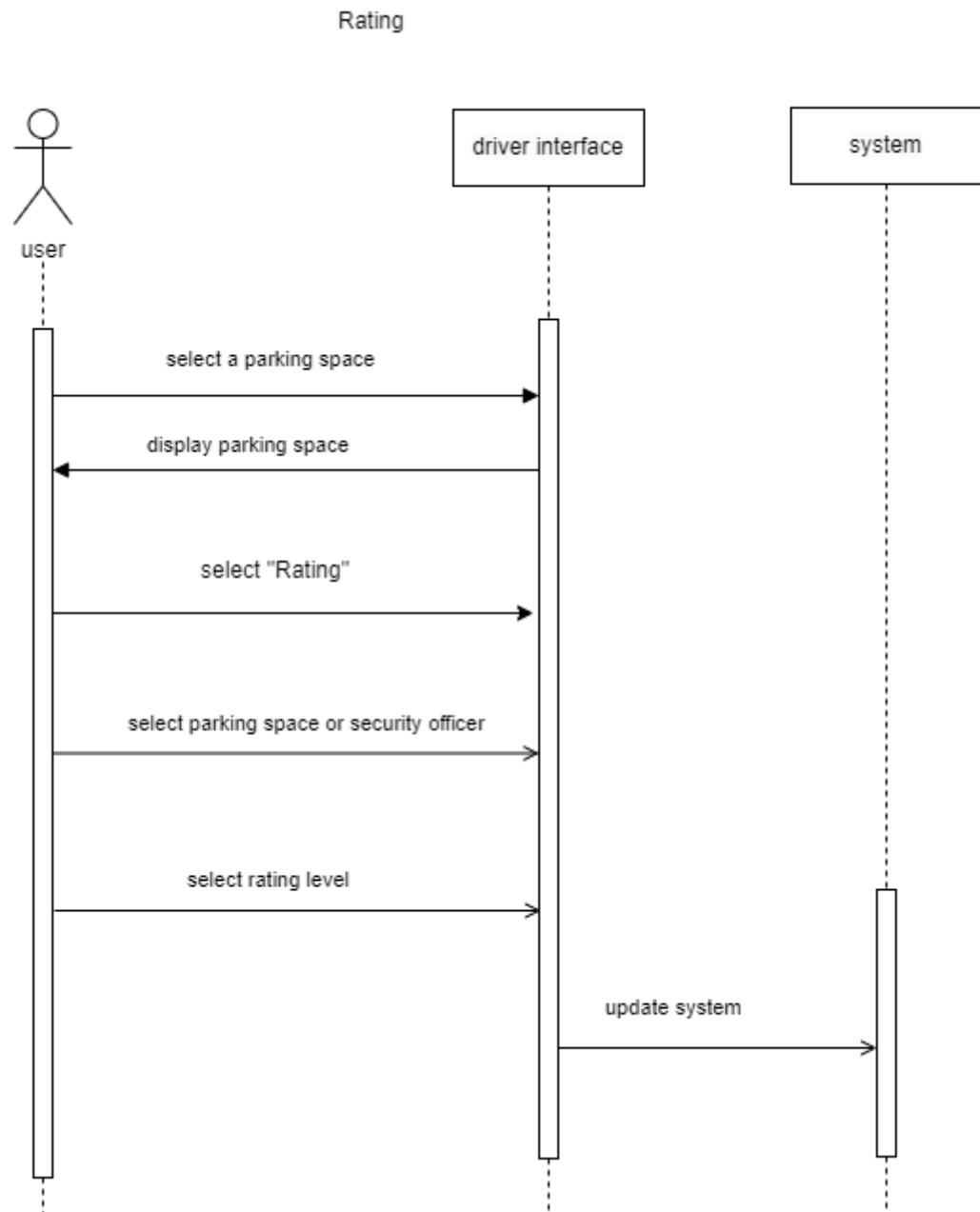


Figure 63: Rating

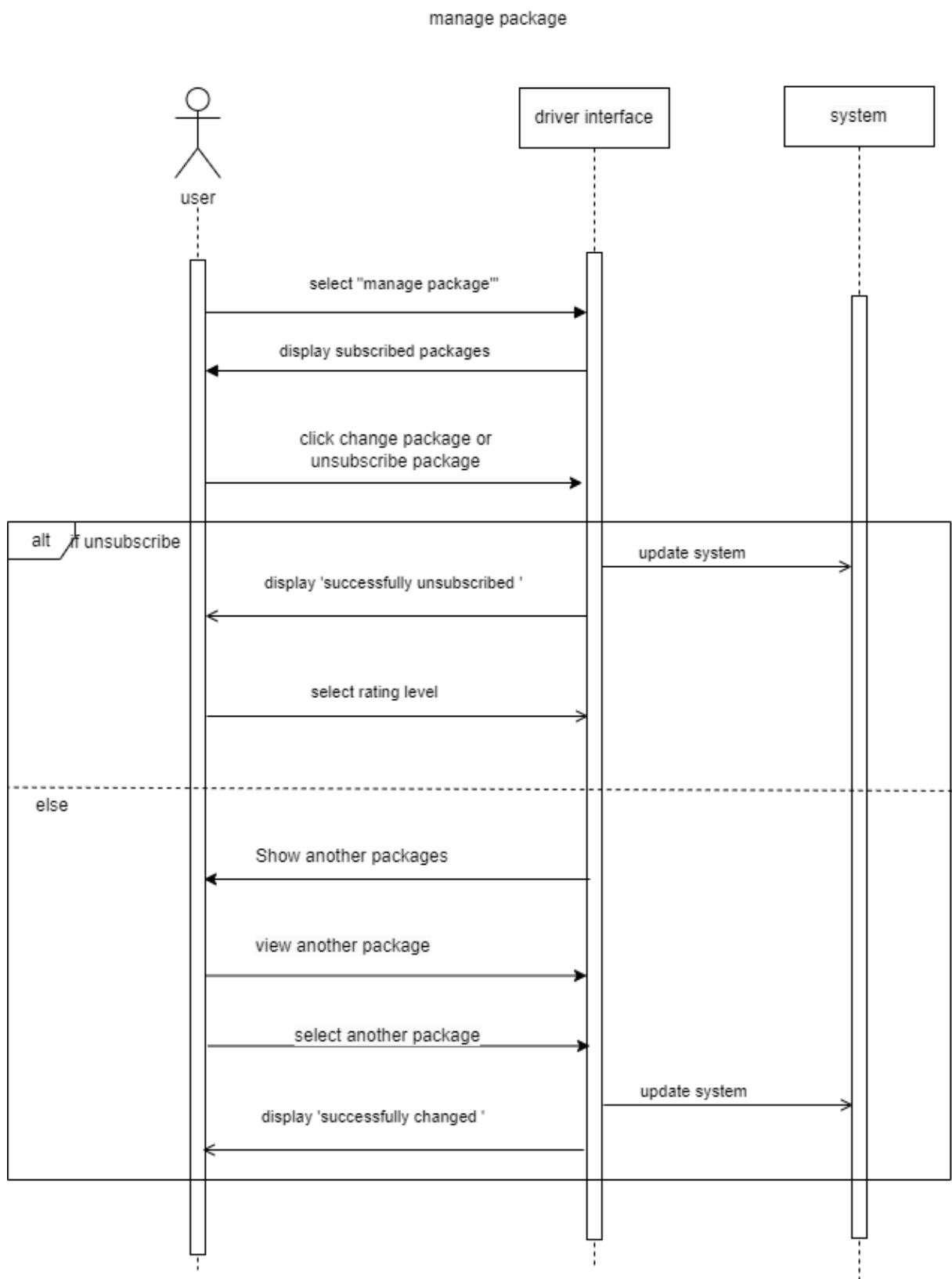


Figure 64:Manage package

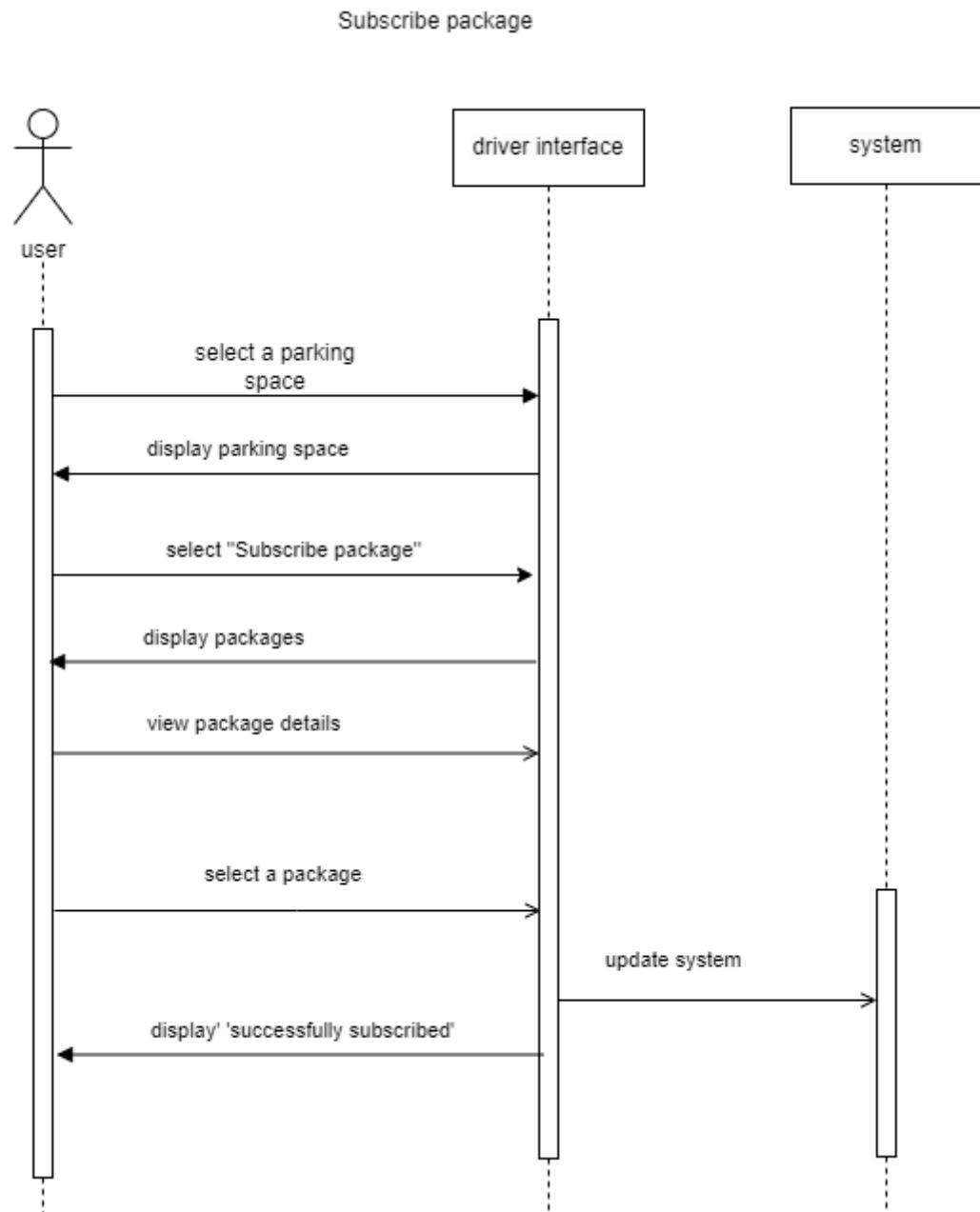


Figure 65:Subscribe package

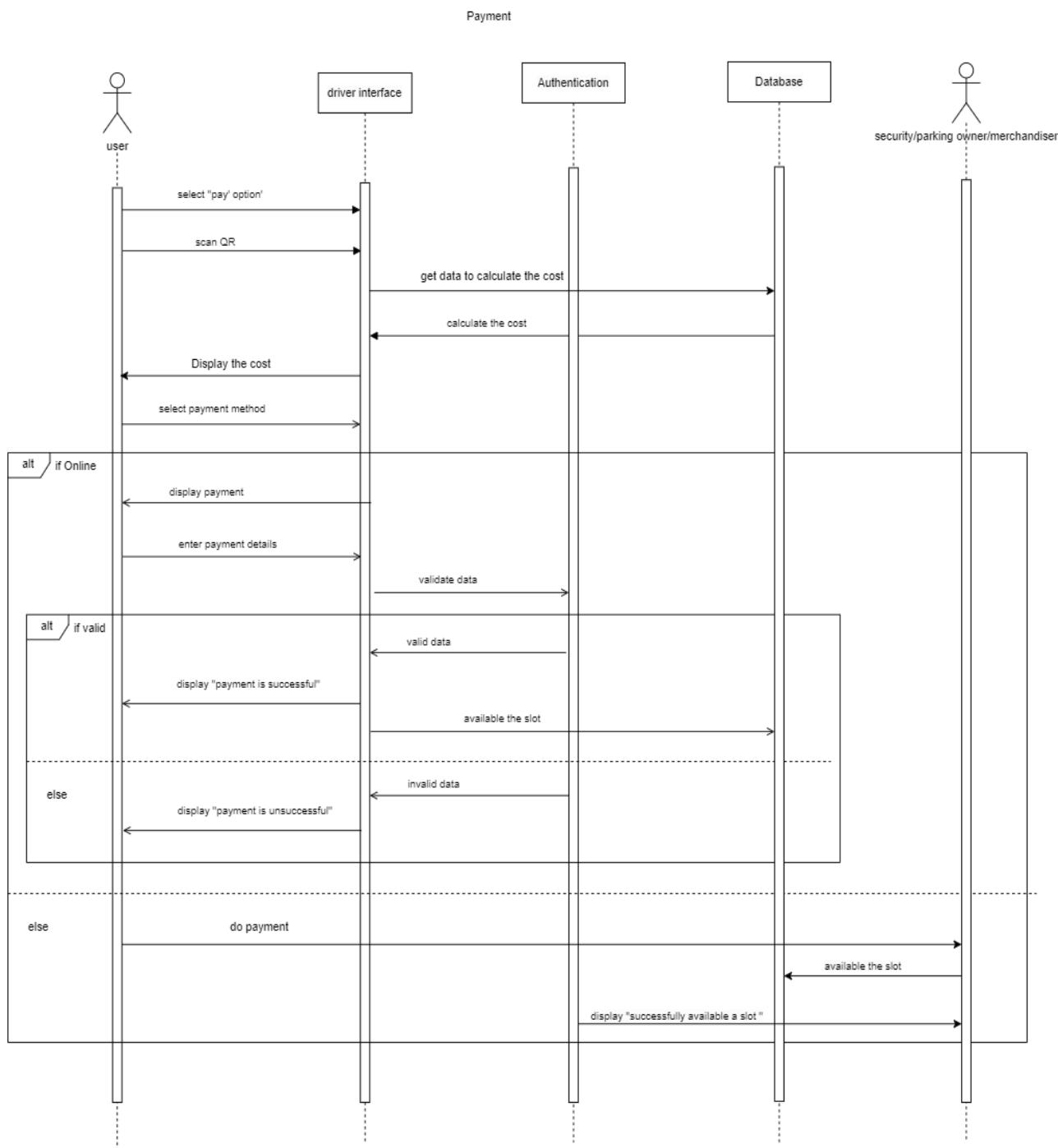


Figure 66:Payment

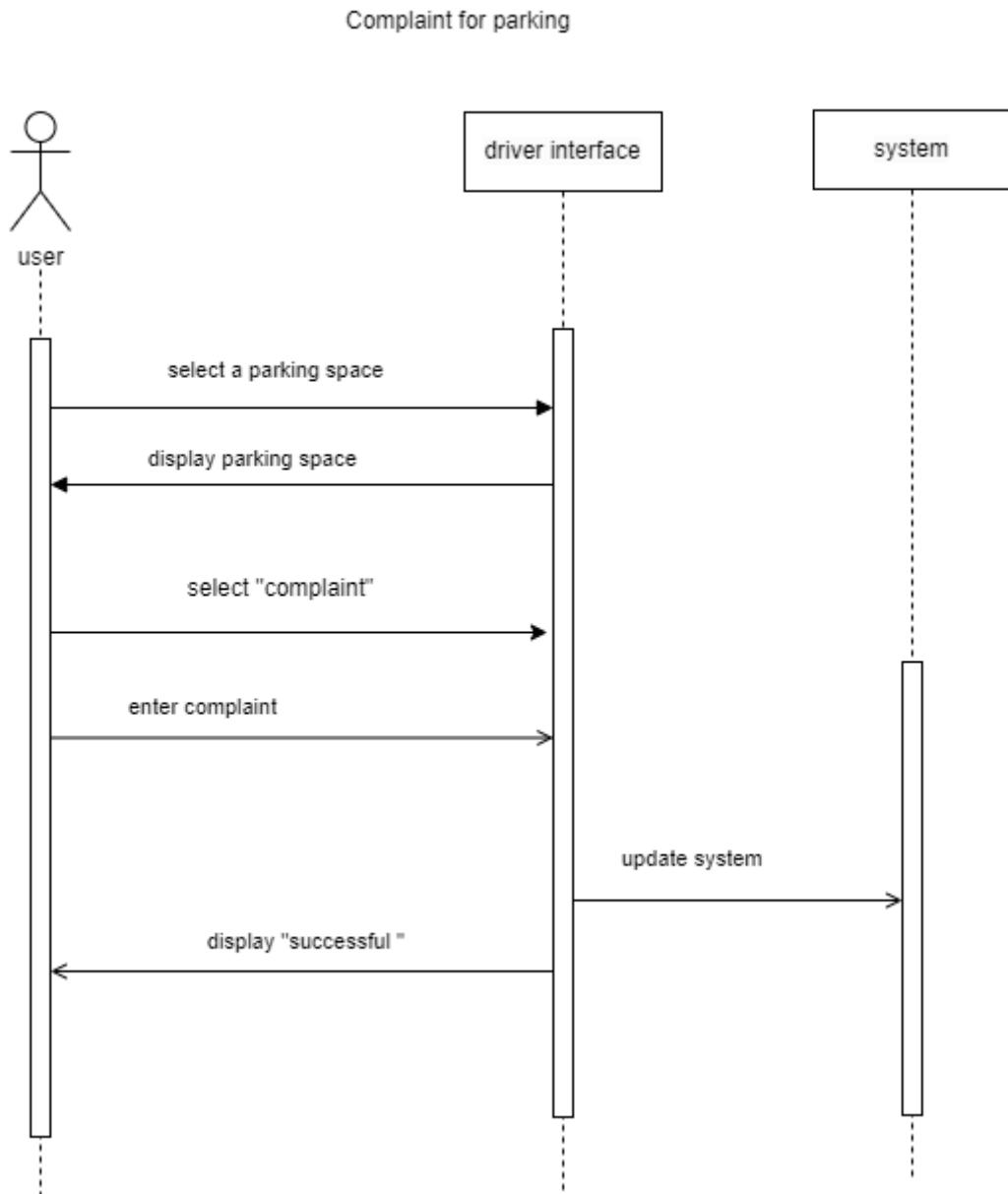


Figure 67:Complaint for a parking

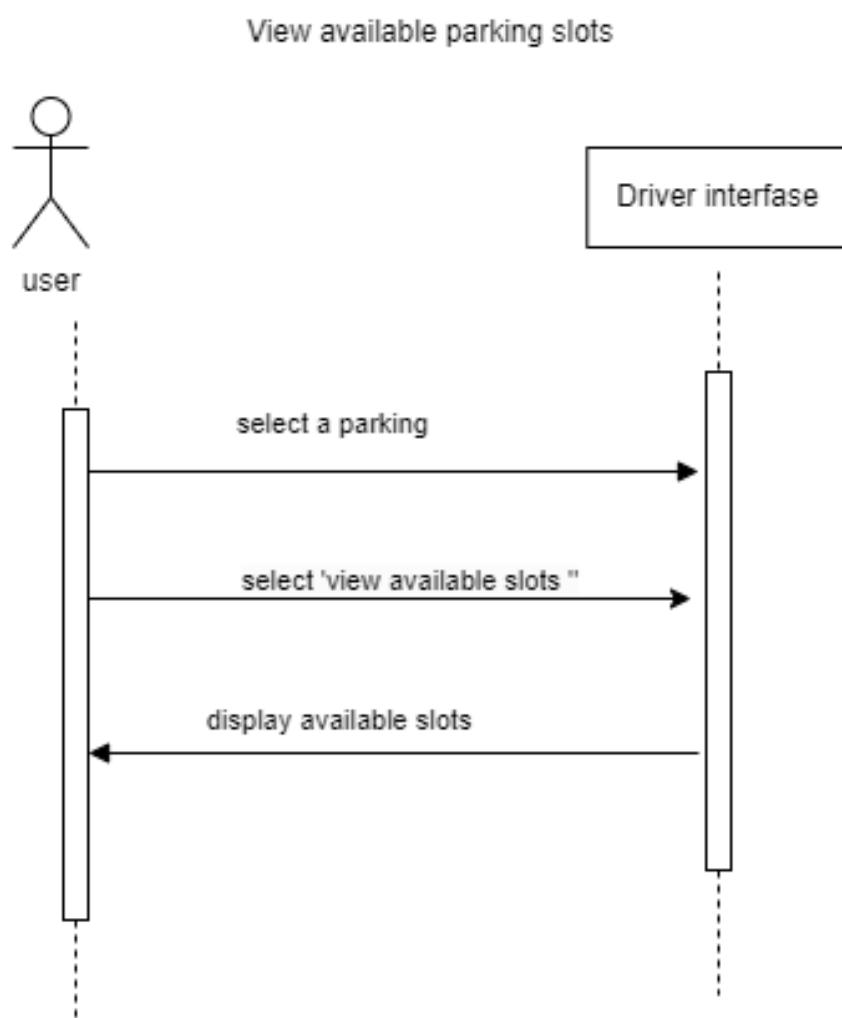


Figure 68:View available parking slots

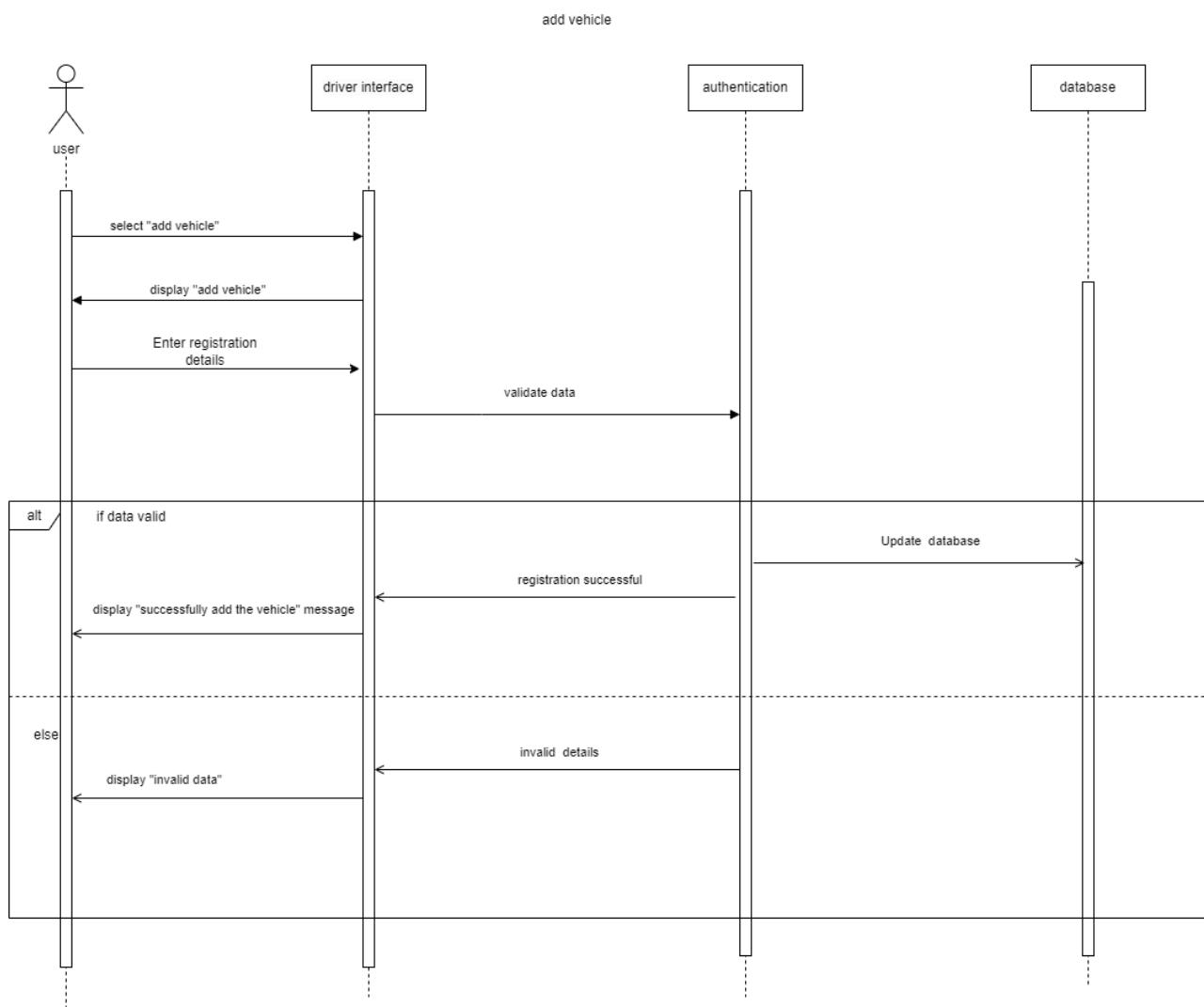


Figure 69: Add vehicle

### 5.3.4 Security

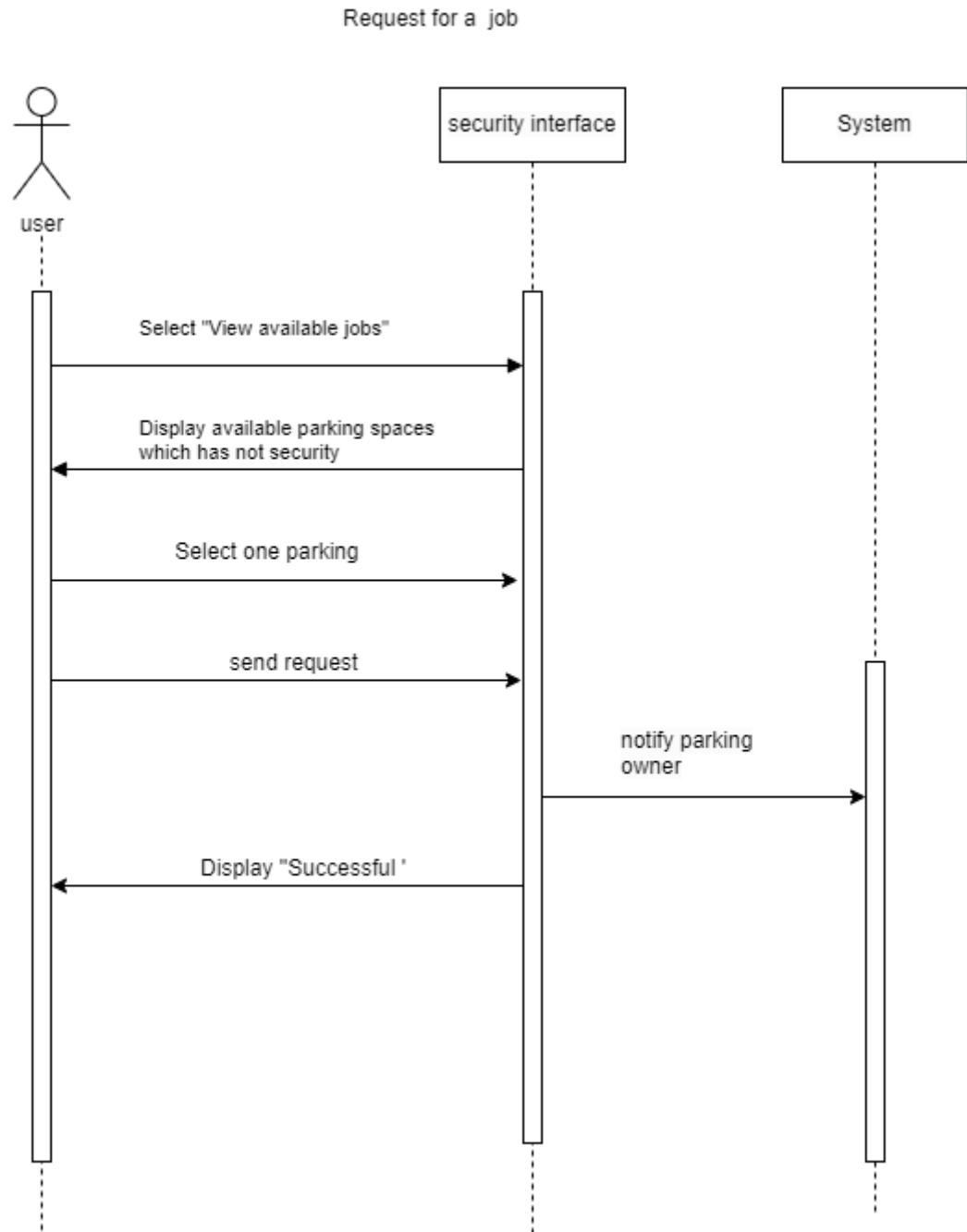


Figure 70:Request for a job

### 5.3.5 Admin

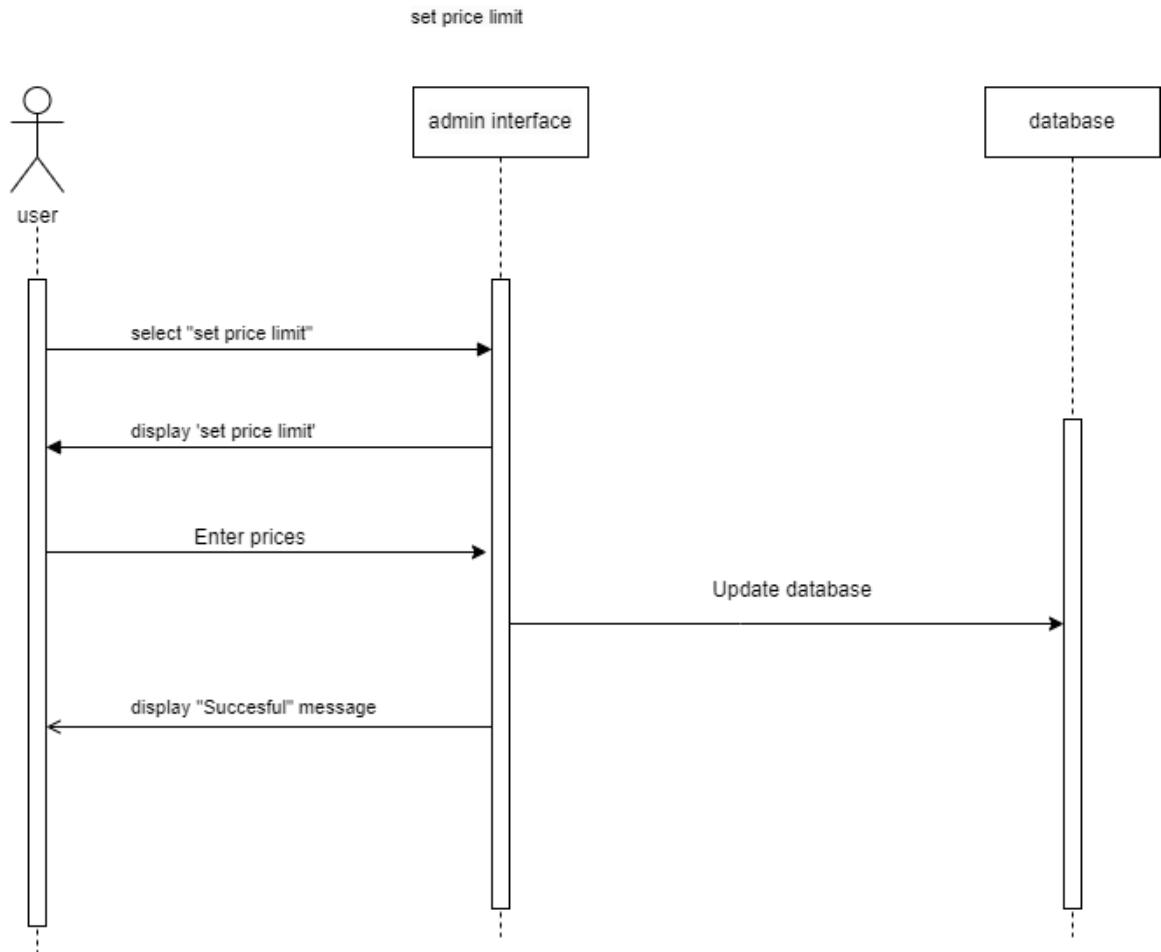


Figure 71: Set price limit

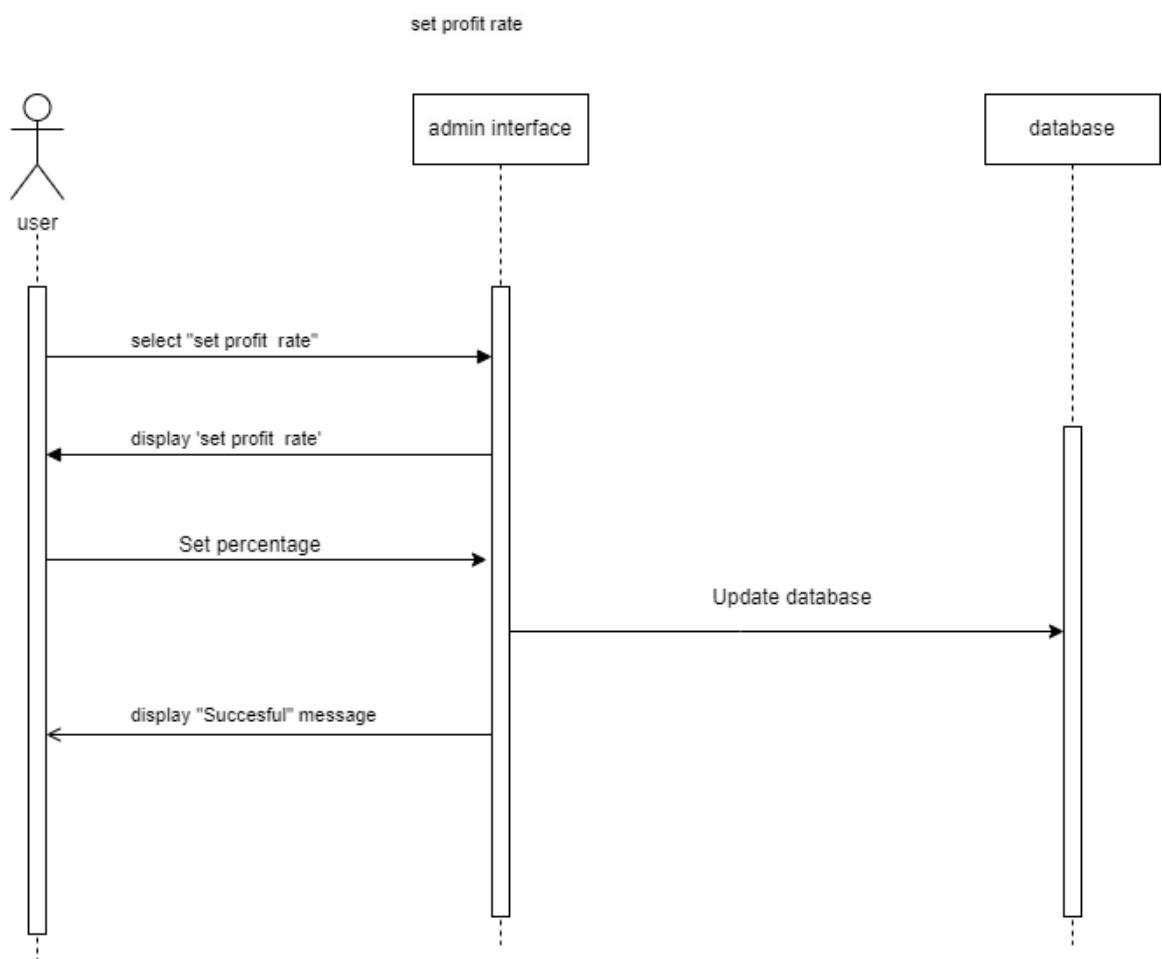


Figure 72: Set profit rate

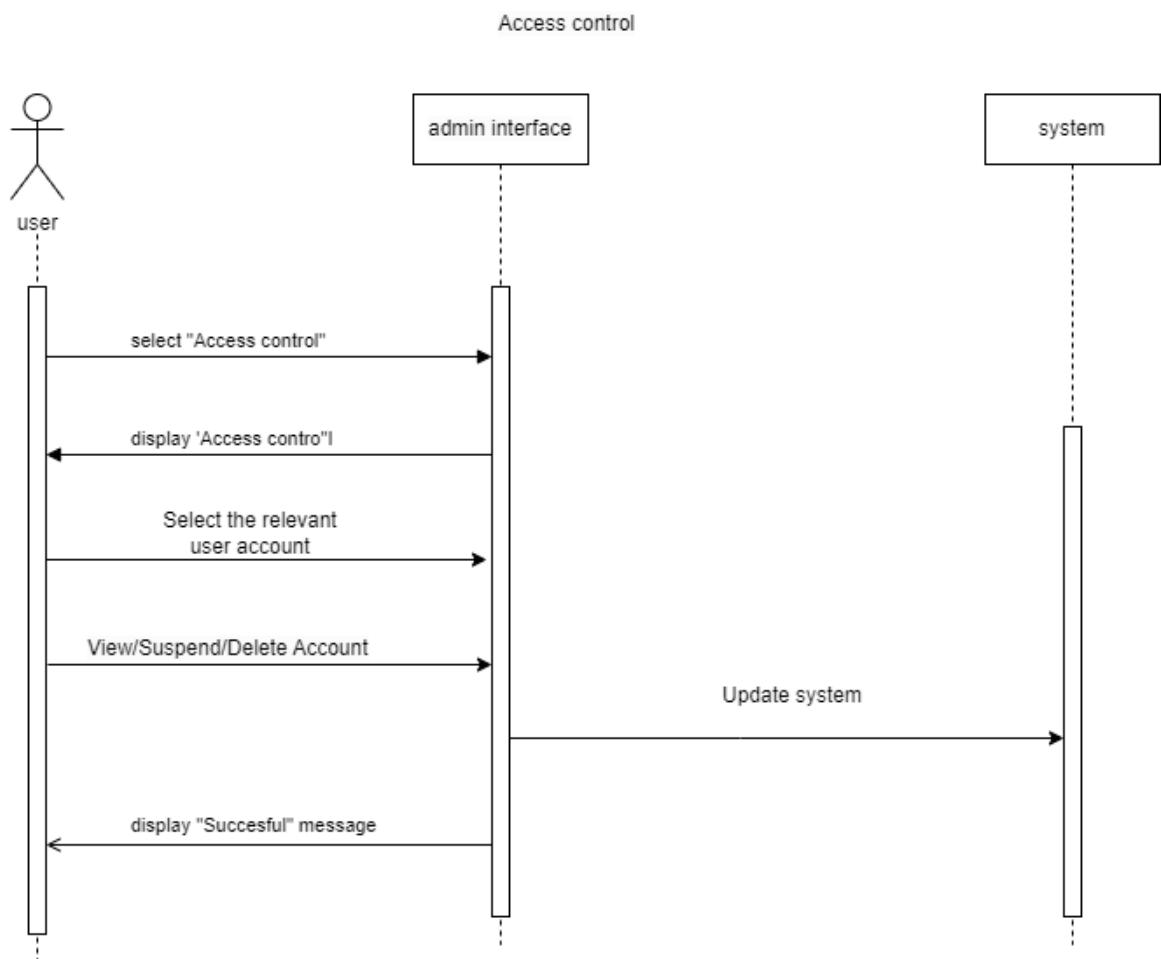


Figure 73: Access control

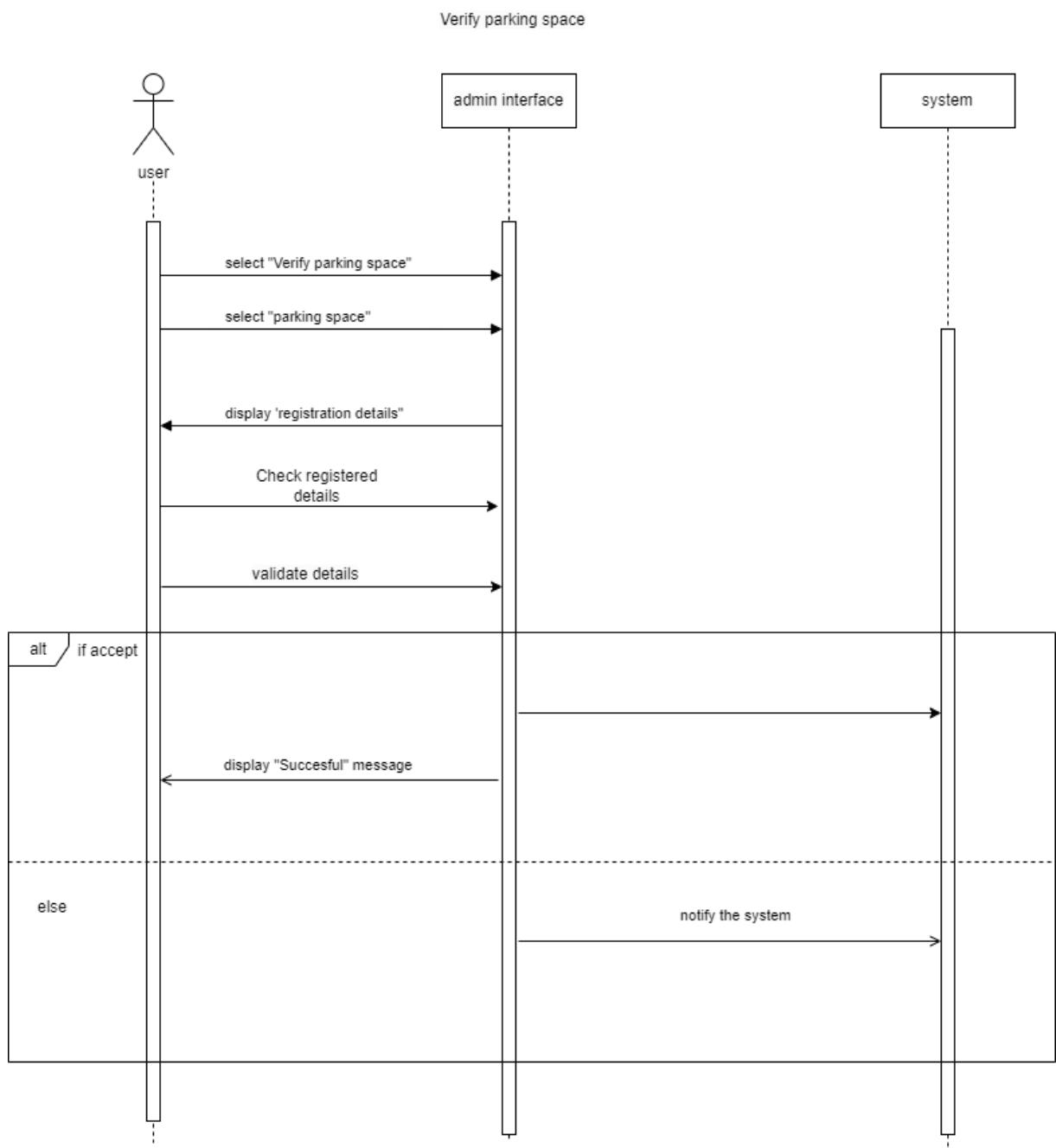


Figure 74:verify parking space

Sequence diagrams link :[sequence diagrams](#)

## 5.4 Activity Diagrams

### 5.4.1 All users

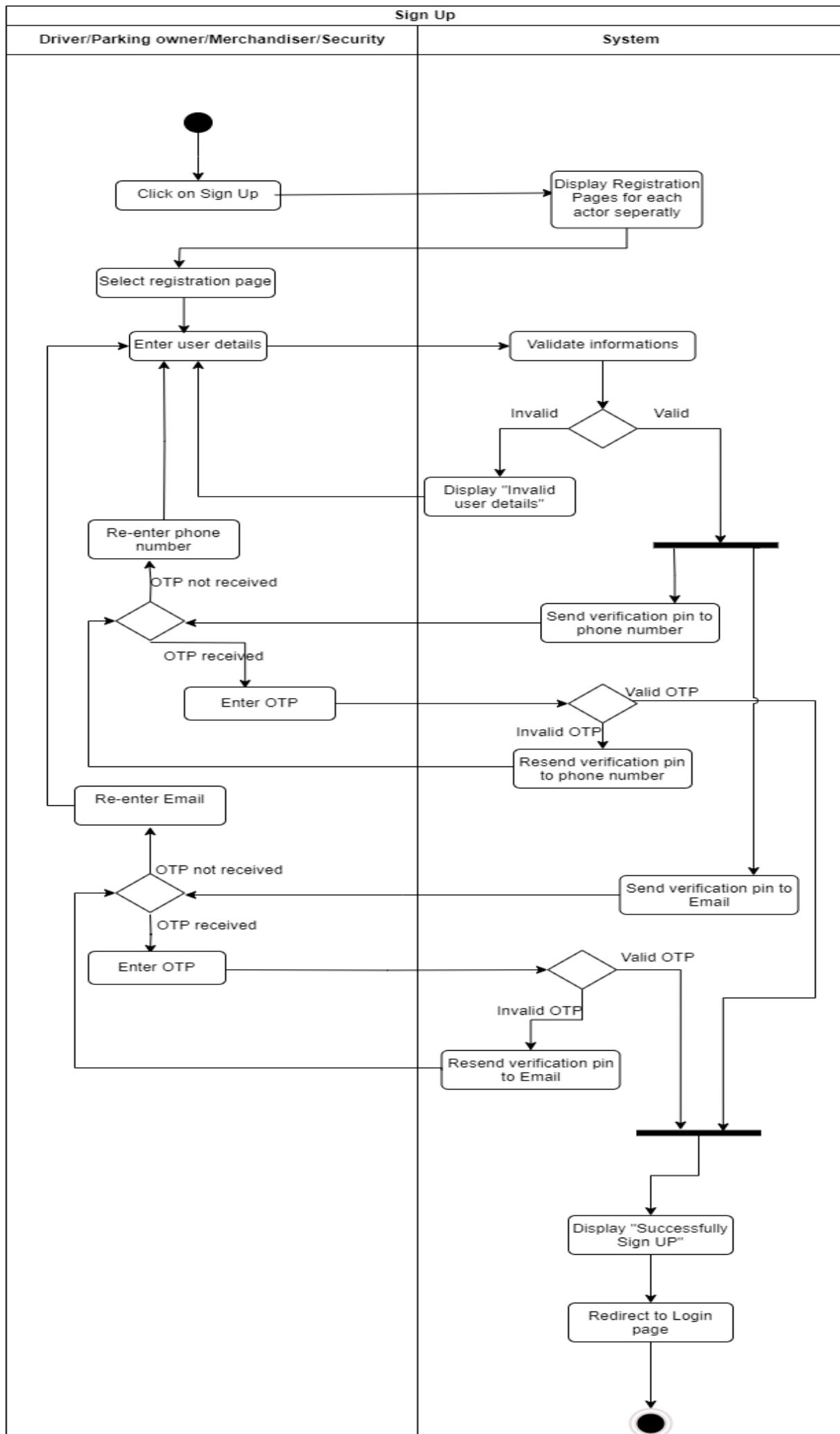


Figure 75:Sign Up

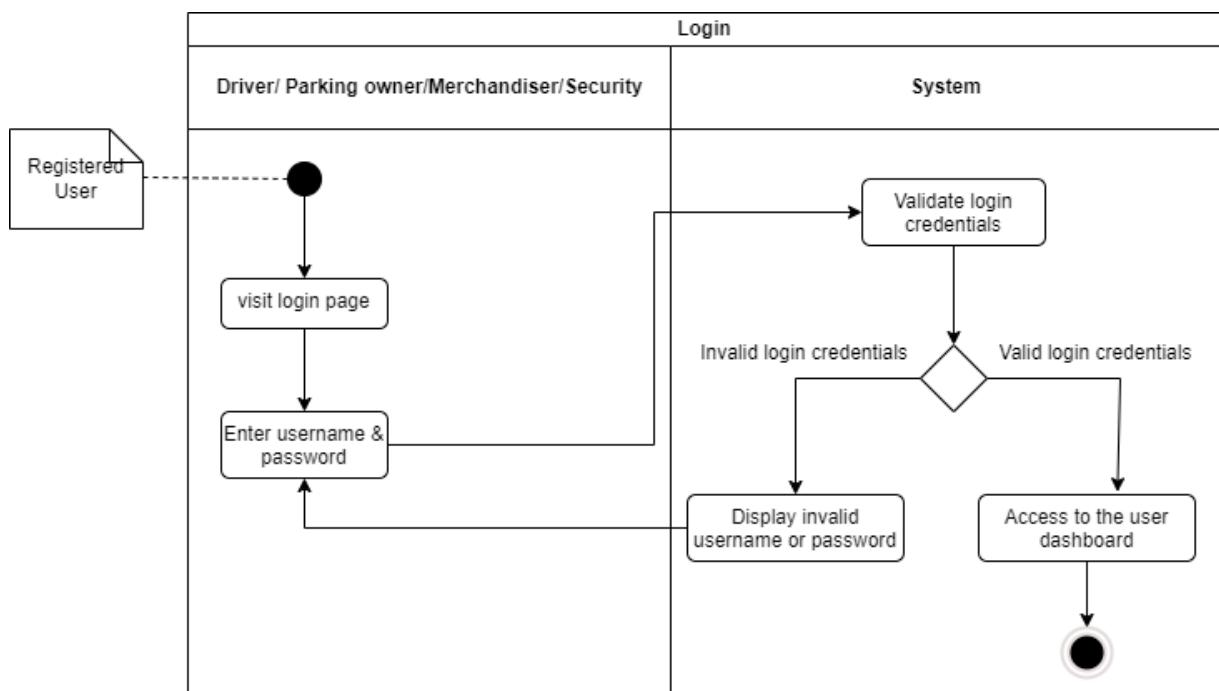


Figure 76:Log In

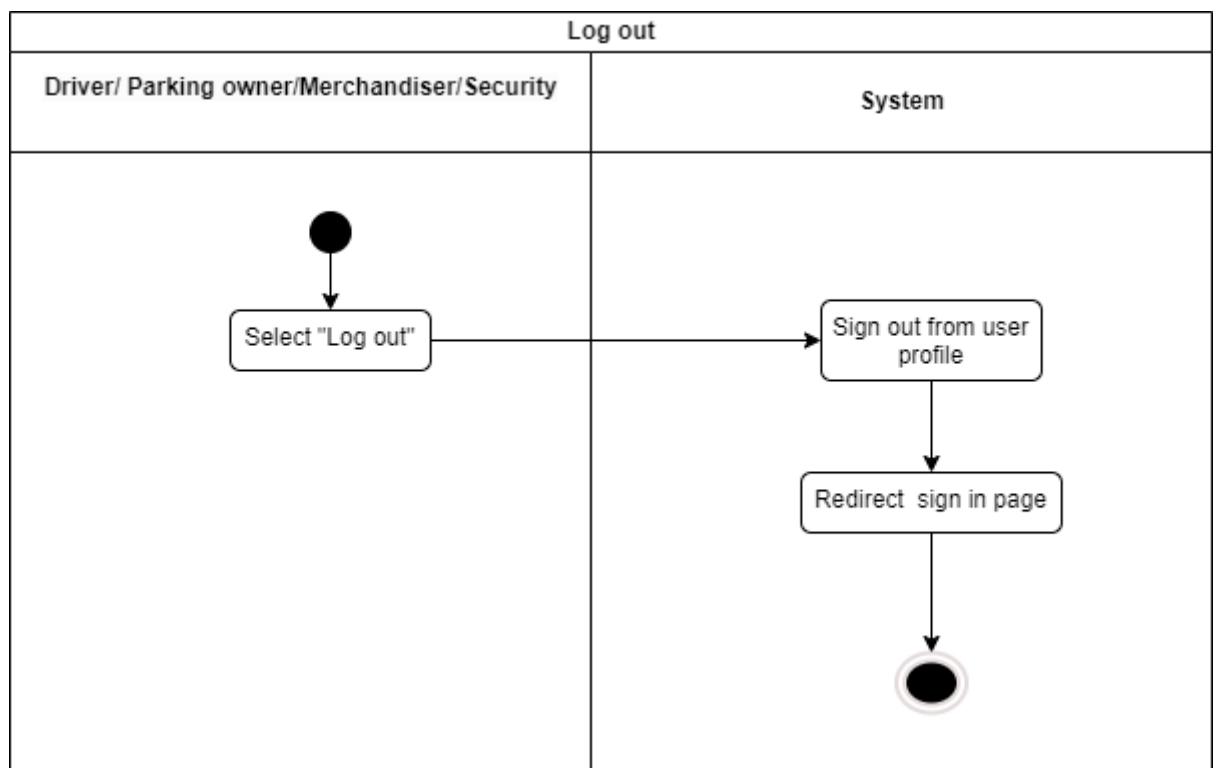


Figure 77:Log out

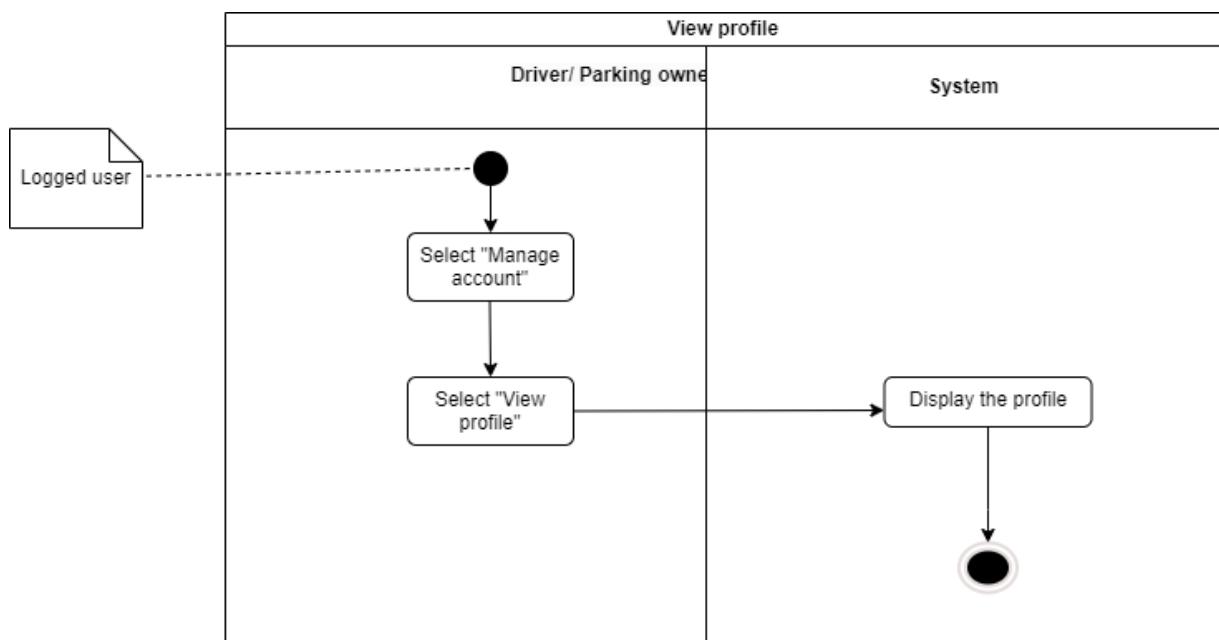


Figure 78:View profile

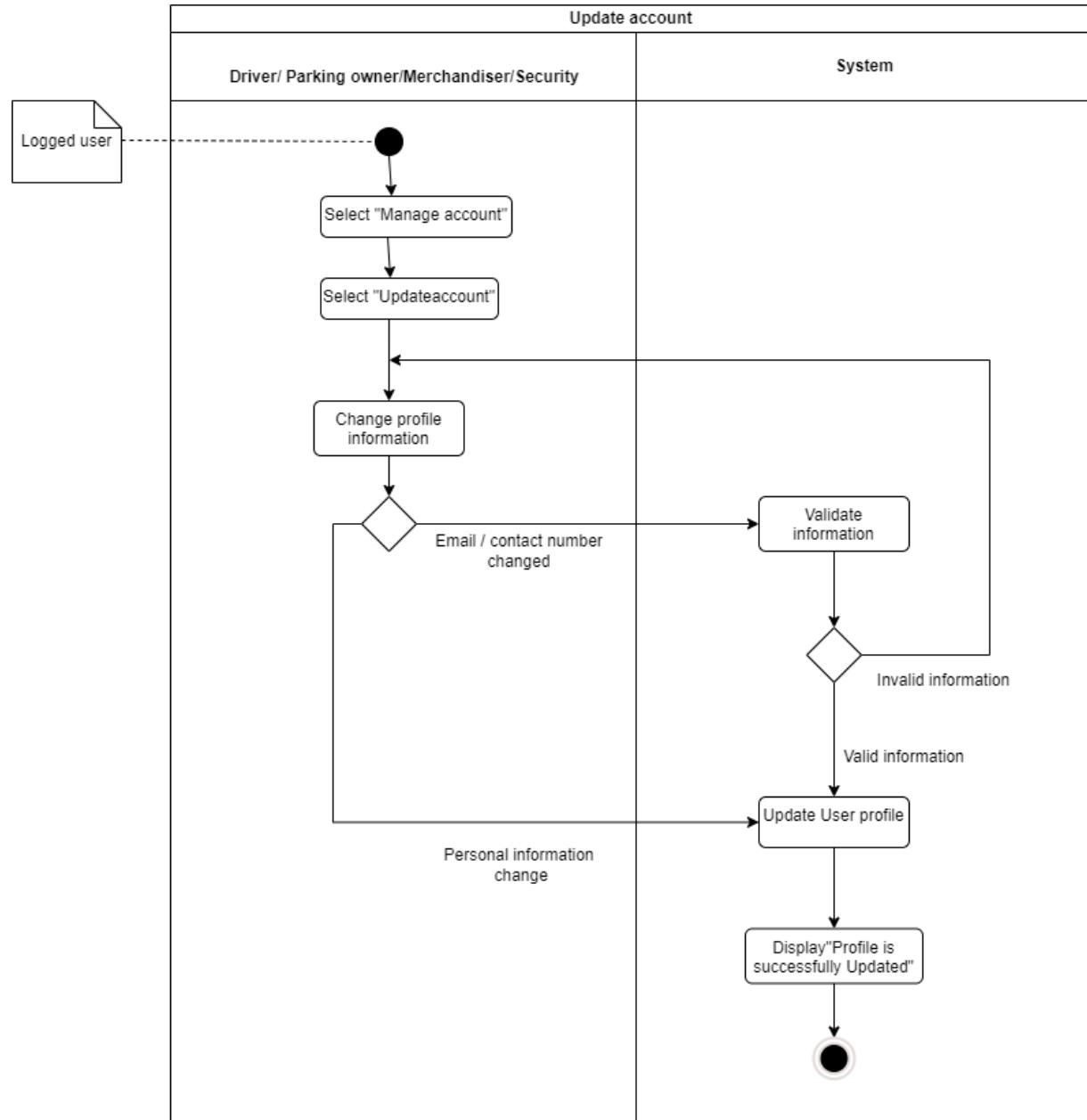


Figure 79:Update account

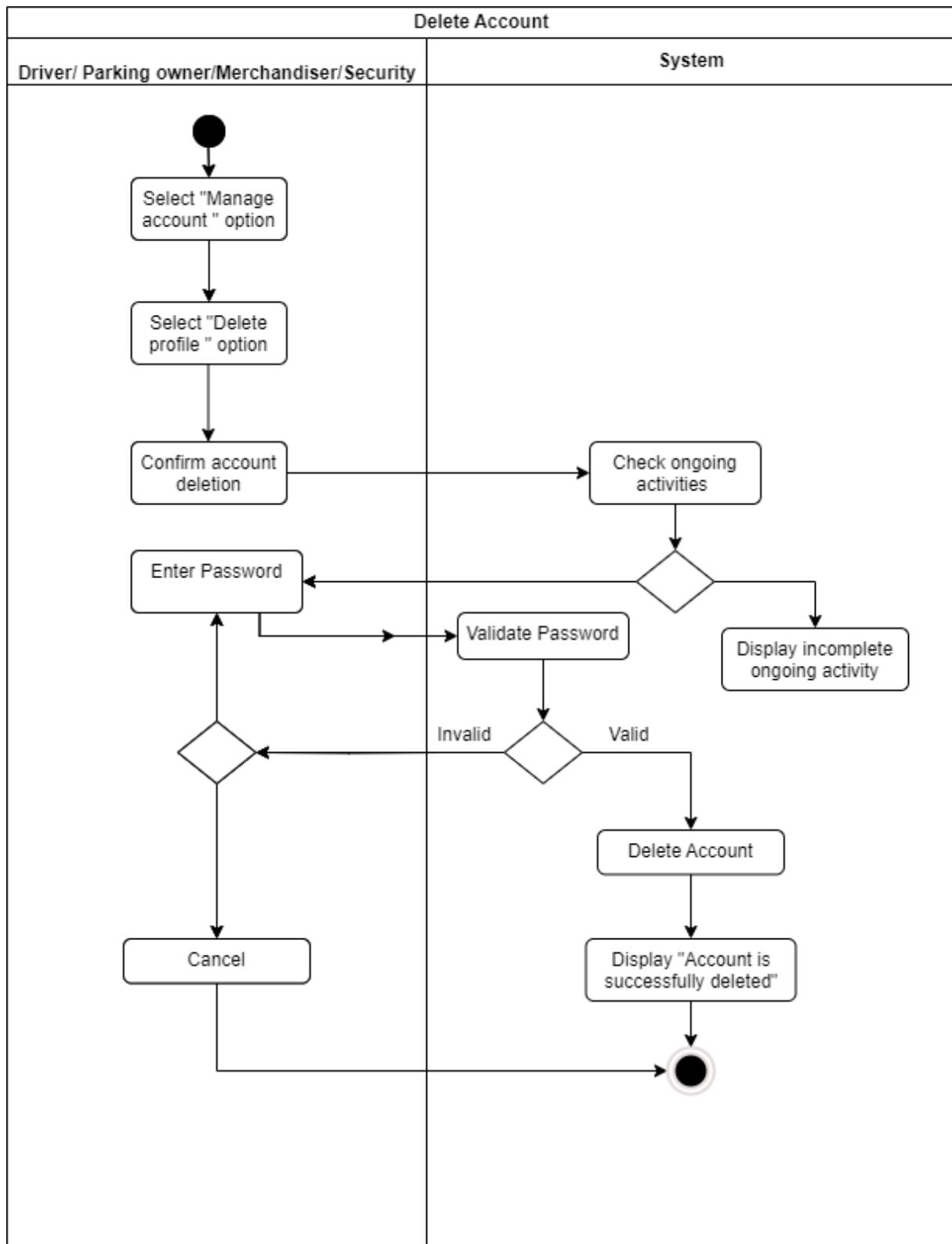


Figure 80:Delete account

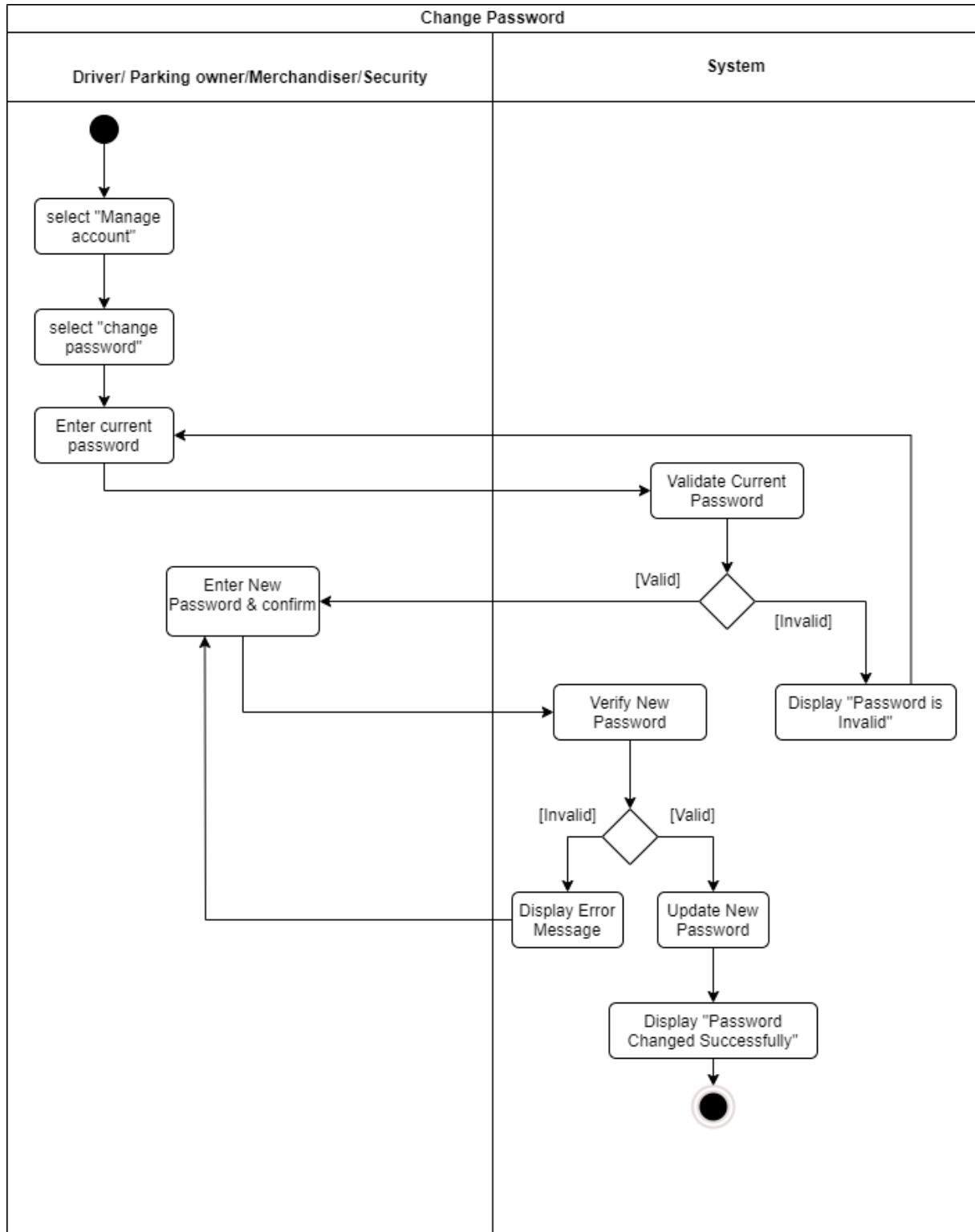


Figure 81:Change password

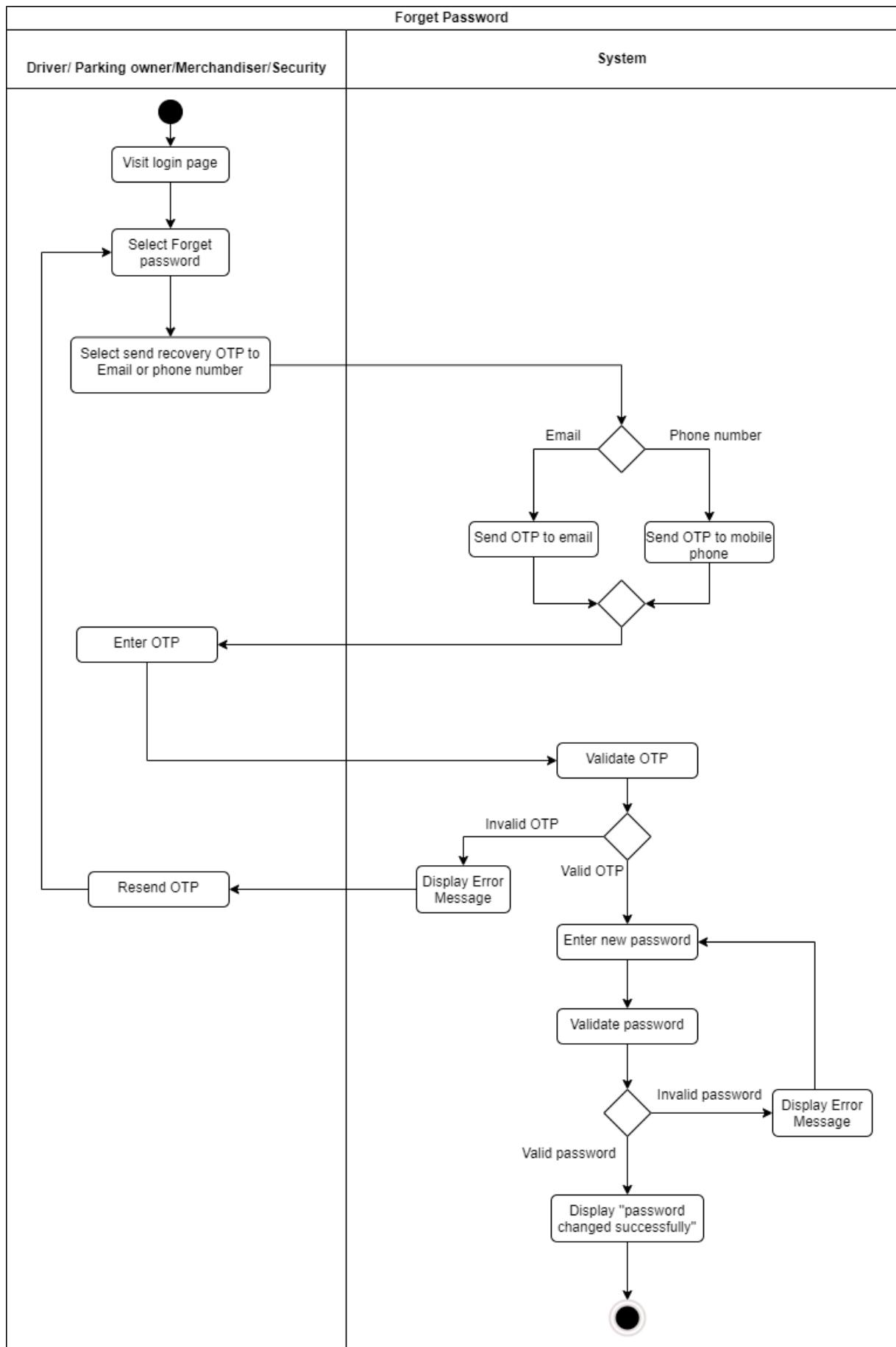


Figure 82:Forgot password

#### 5.4.2 Parking owner and Merchandiser

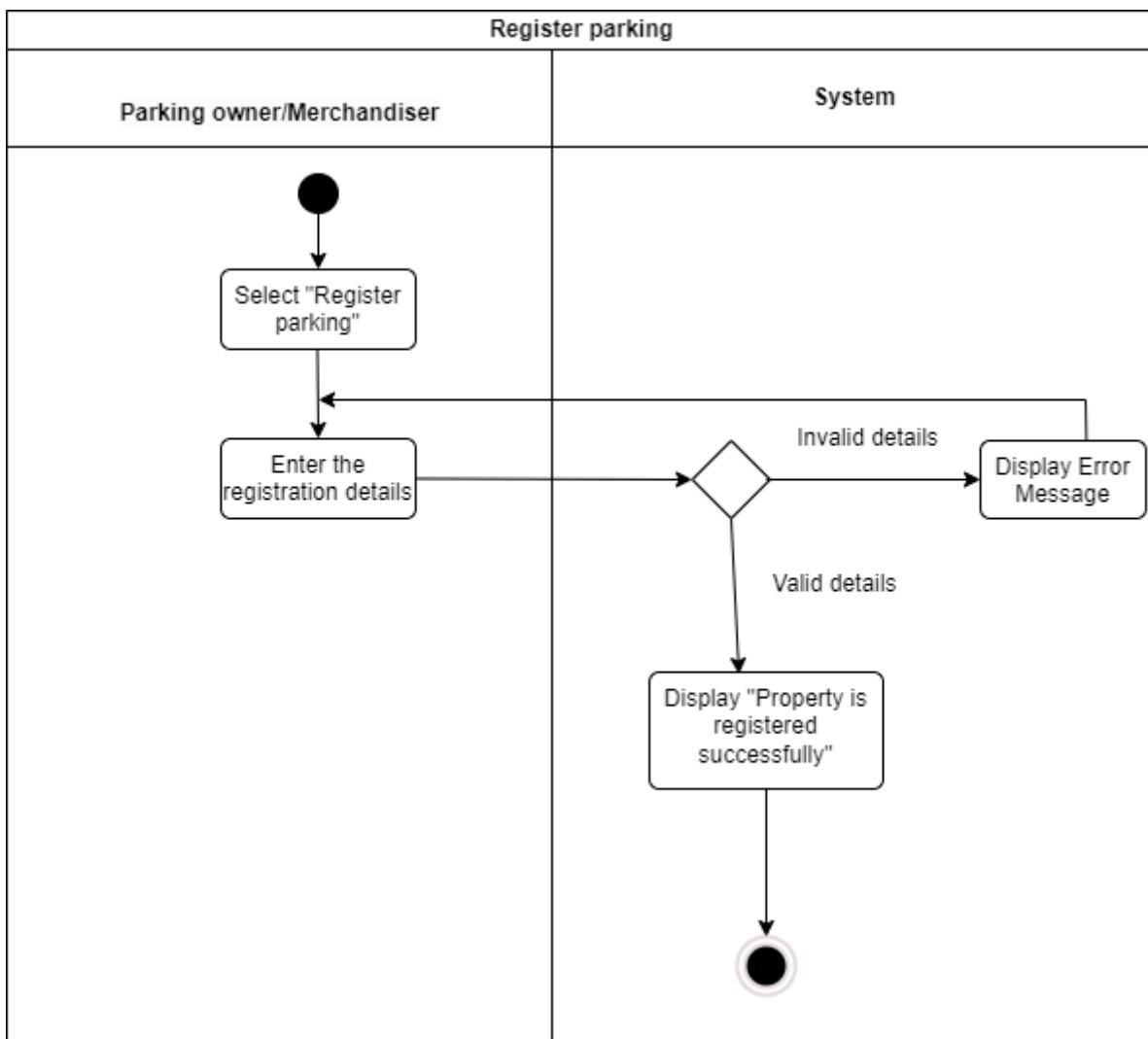


Figure 83:Register parking

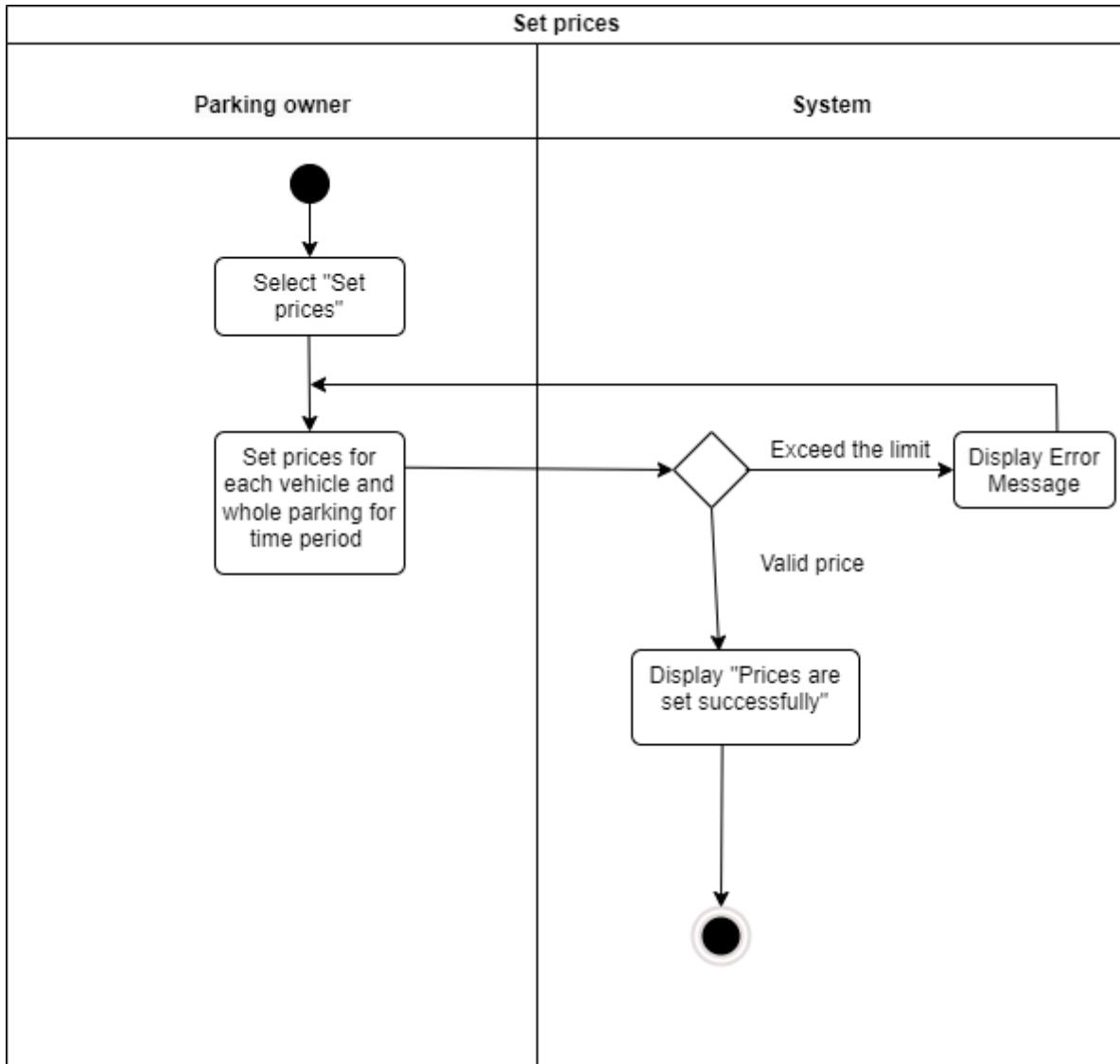


Figure 84: Set prices

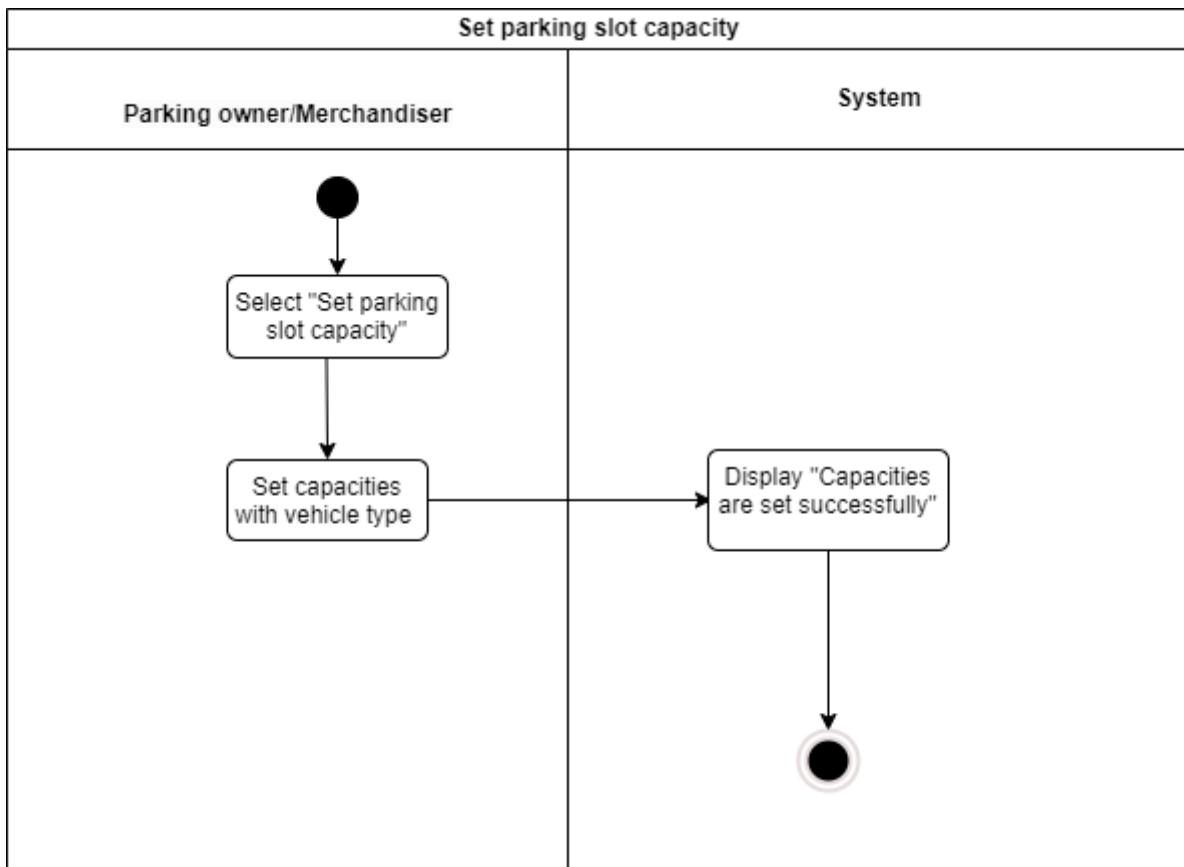


Figure 85: Set parking slot capacity

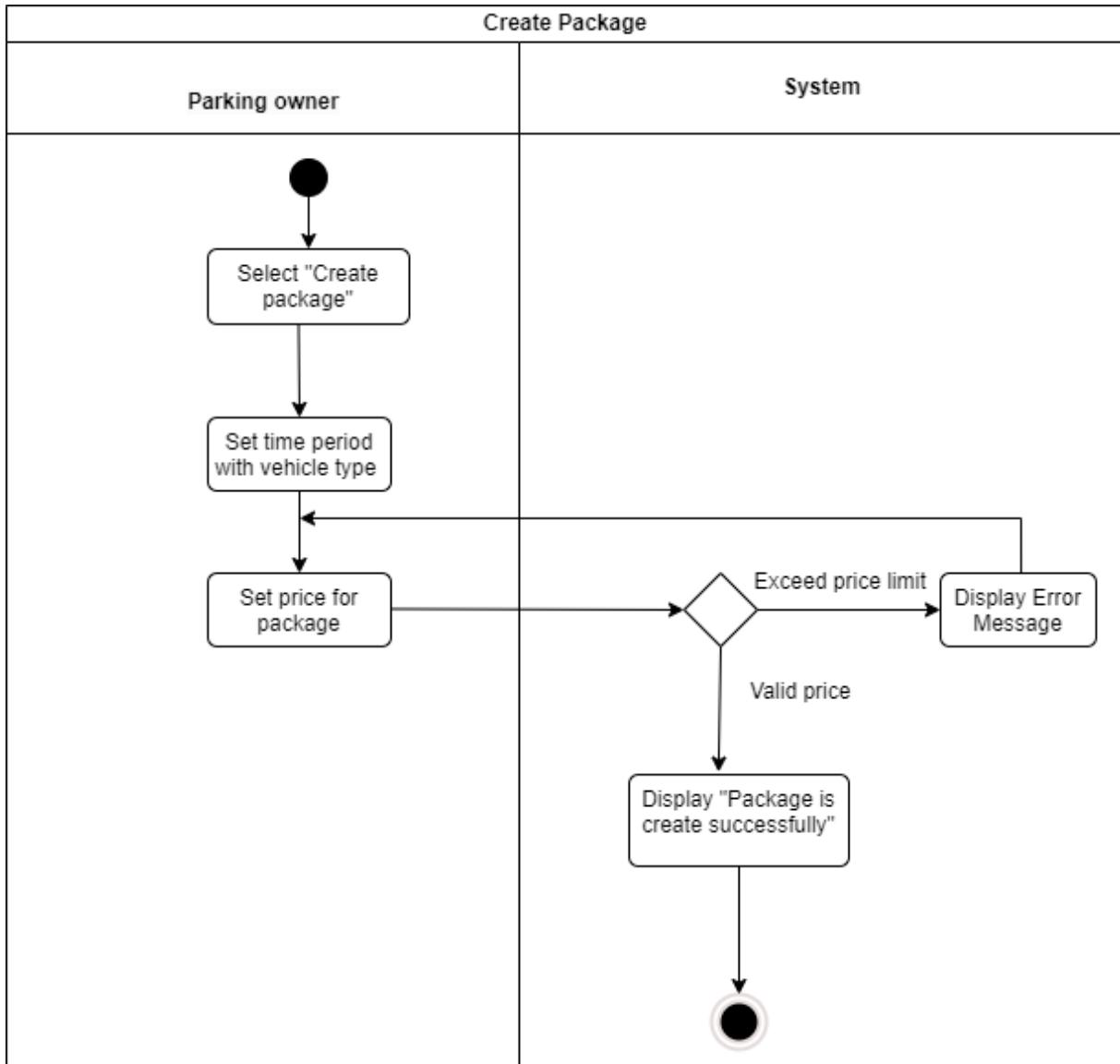


Figure 86:Create packages

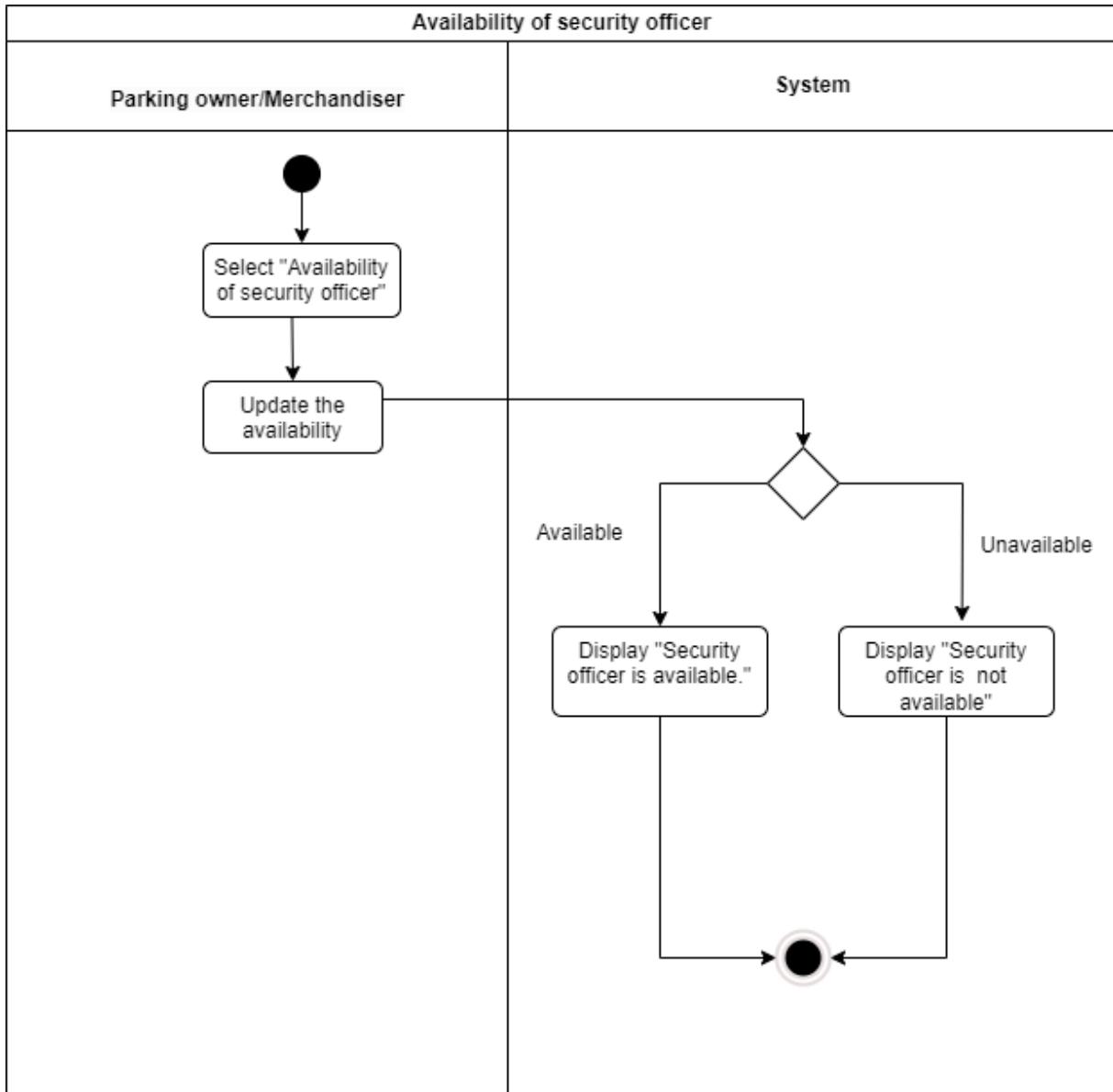


Figure 87: Availability of security officer

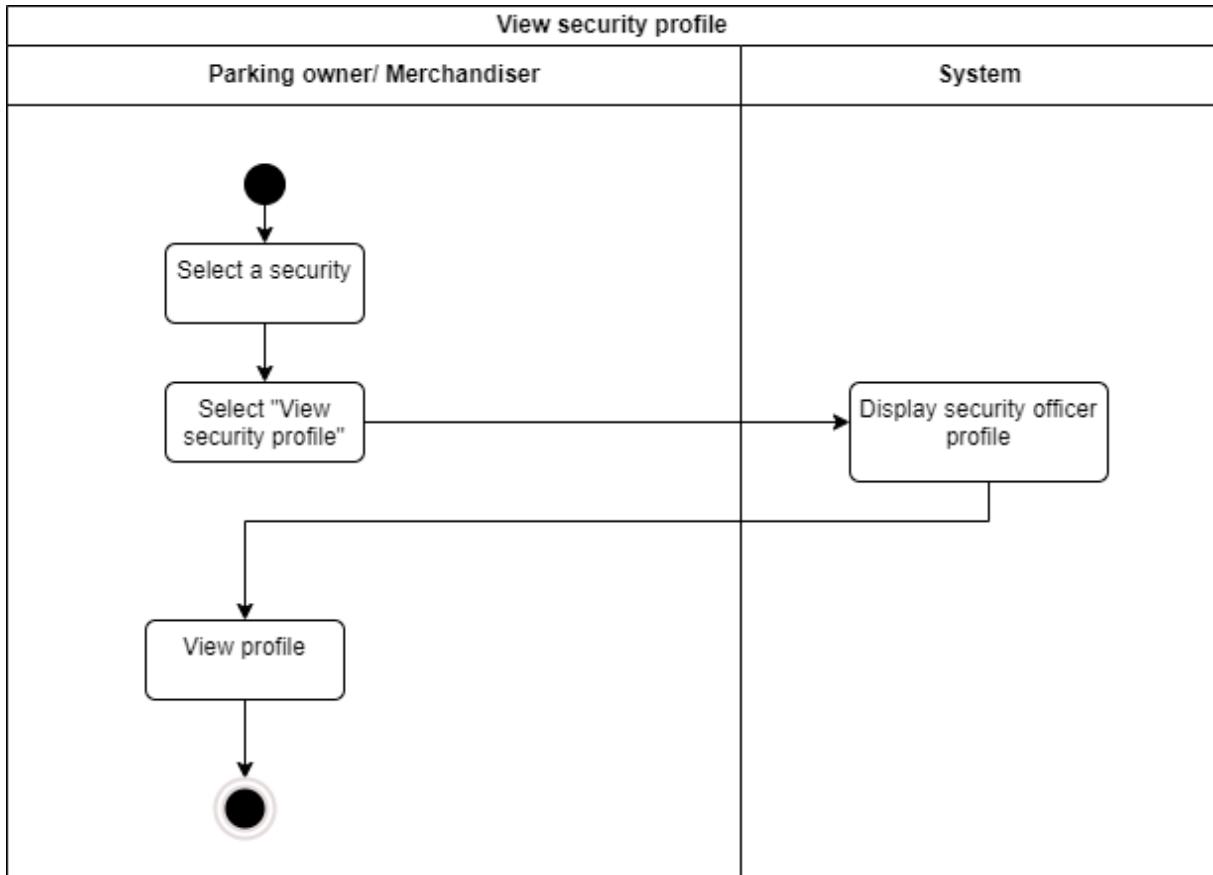


Figure 88:View security profile

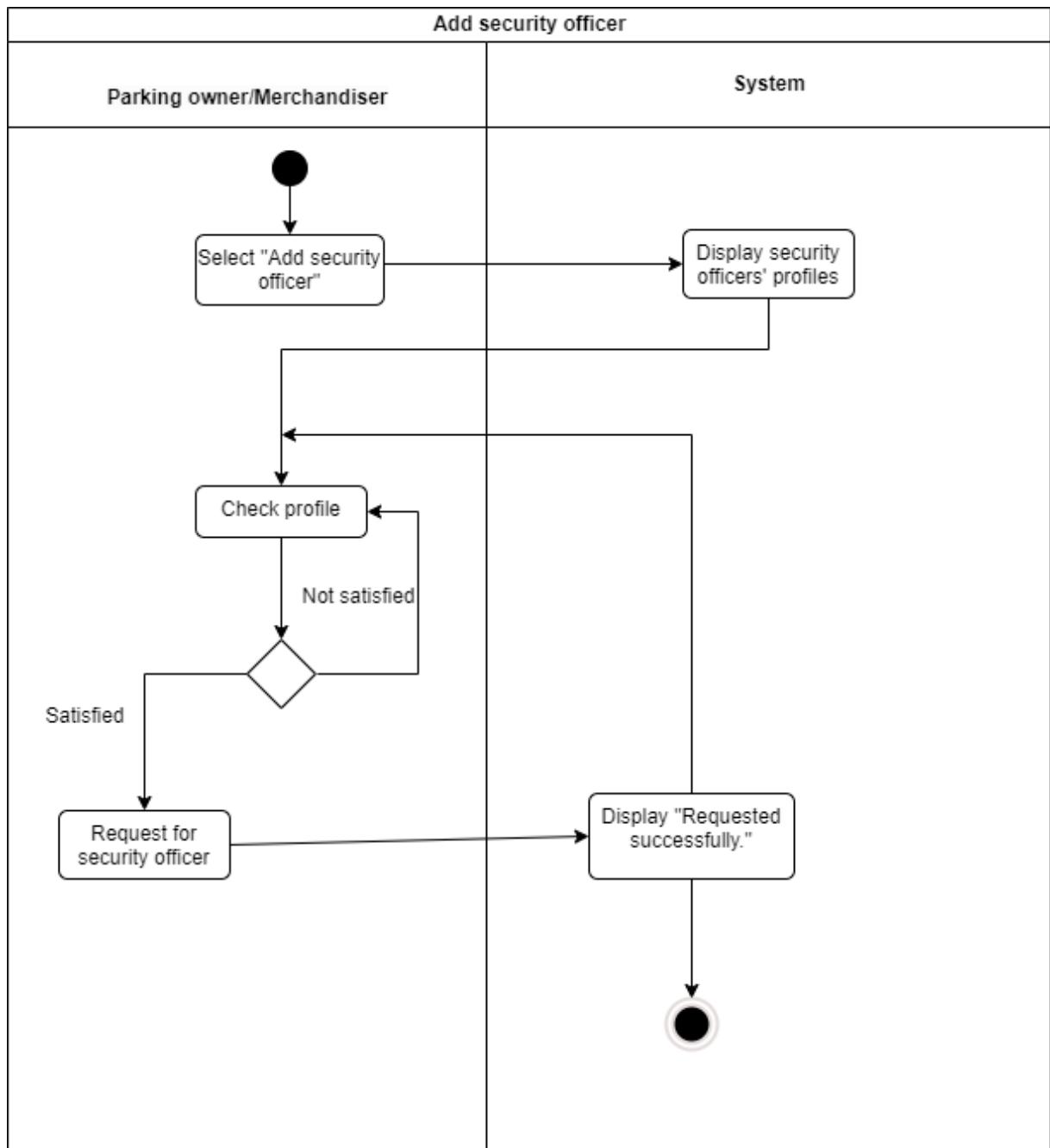


Figure 89: Add security officer

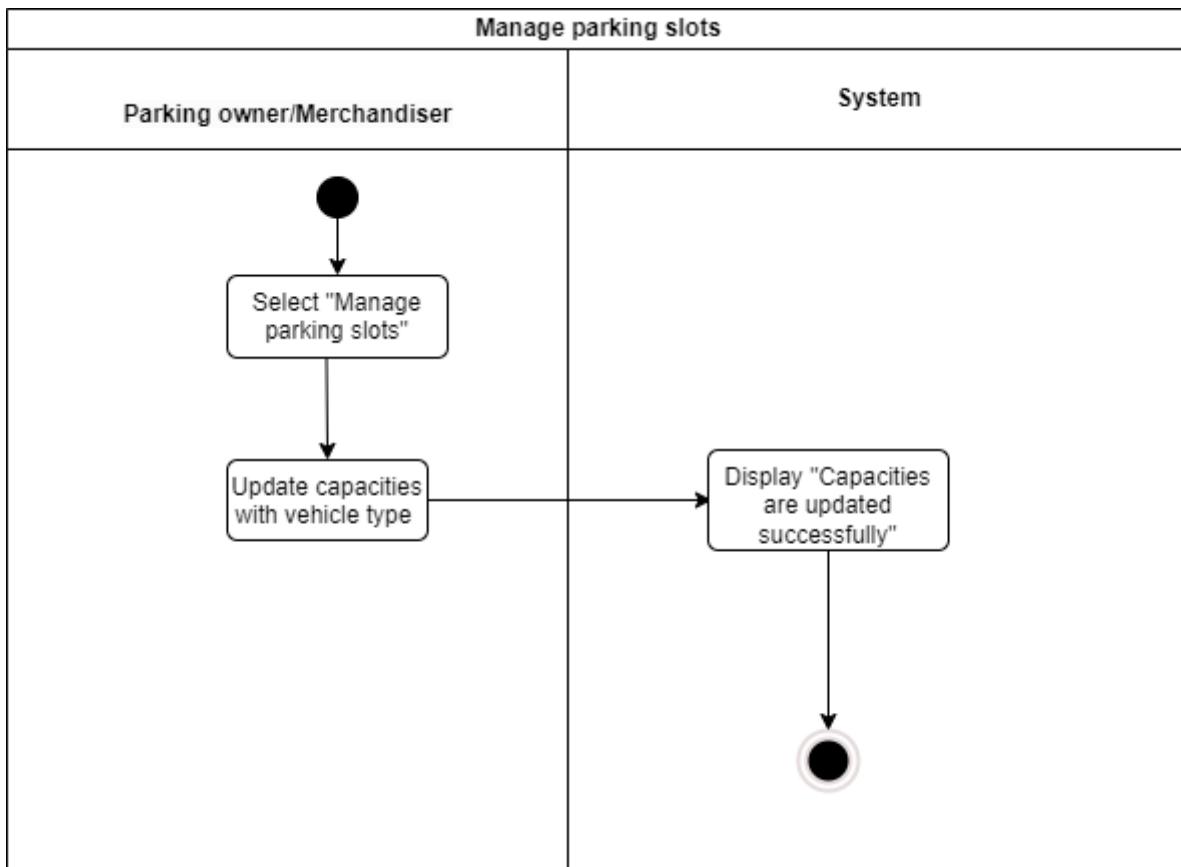


Figure 90:Manage parking slots

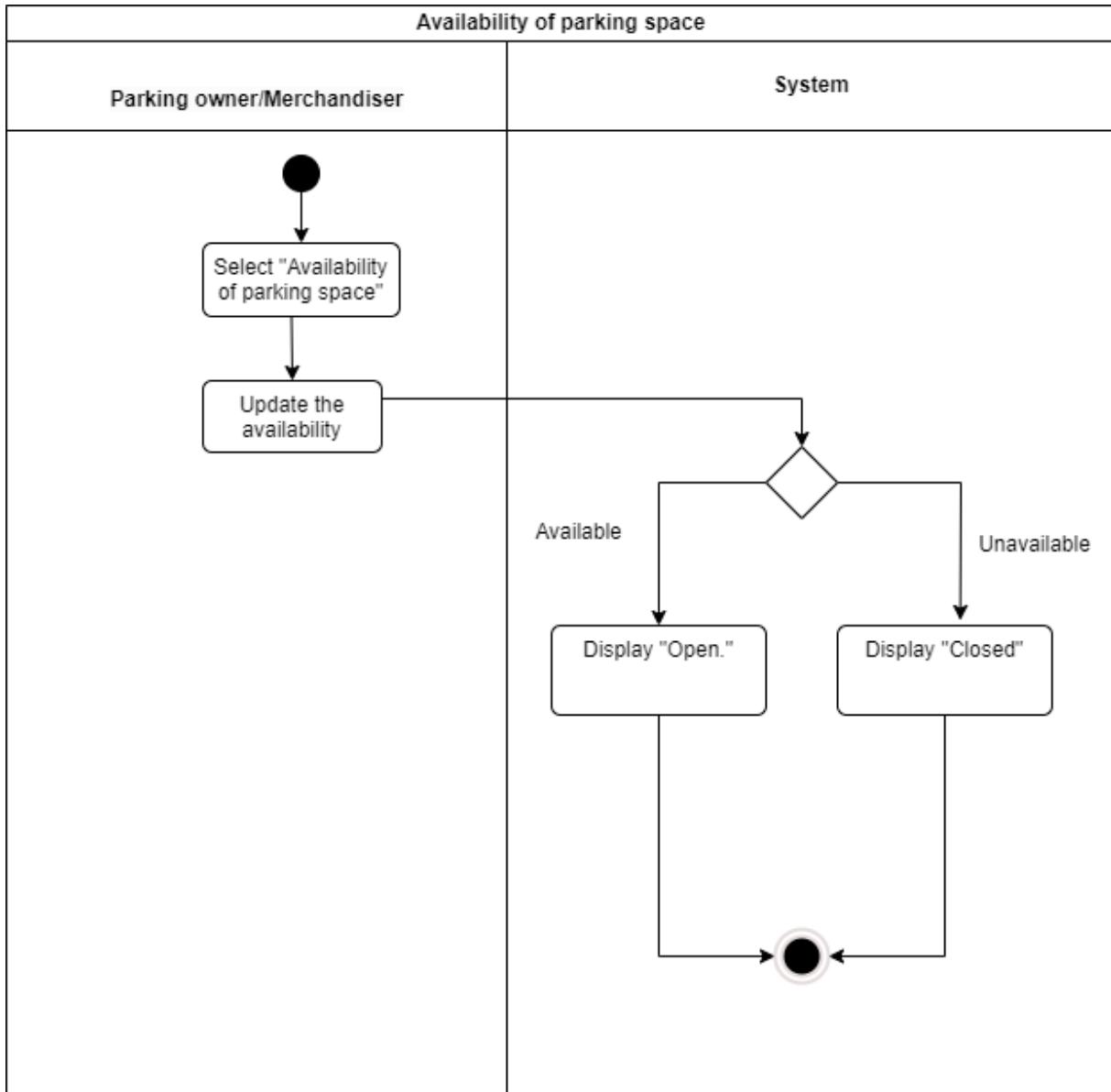


Figure 91:Availability of parking space

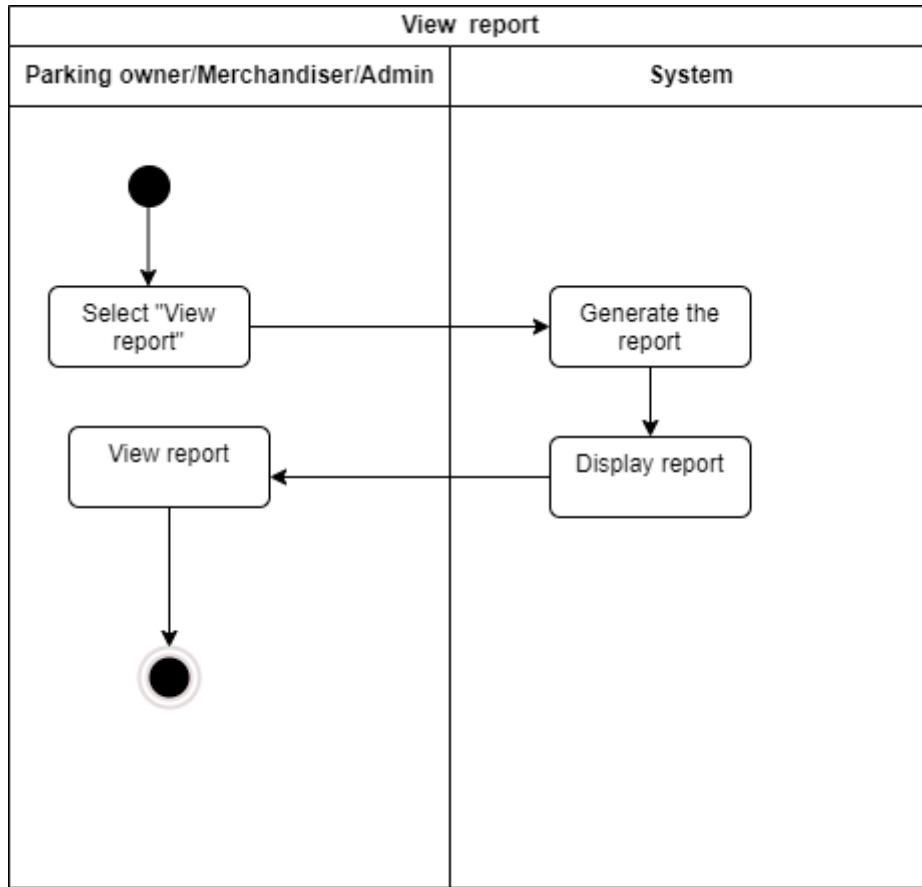


Figure 92:View report

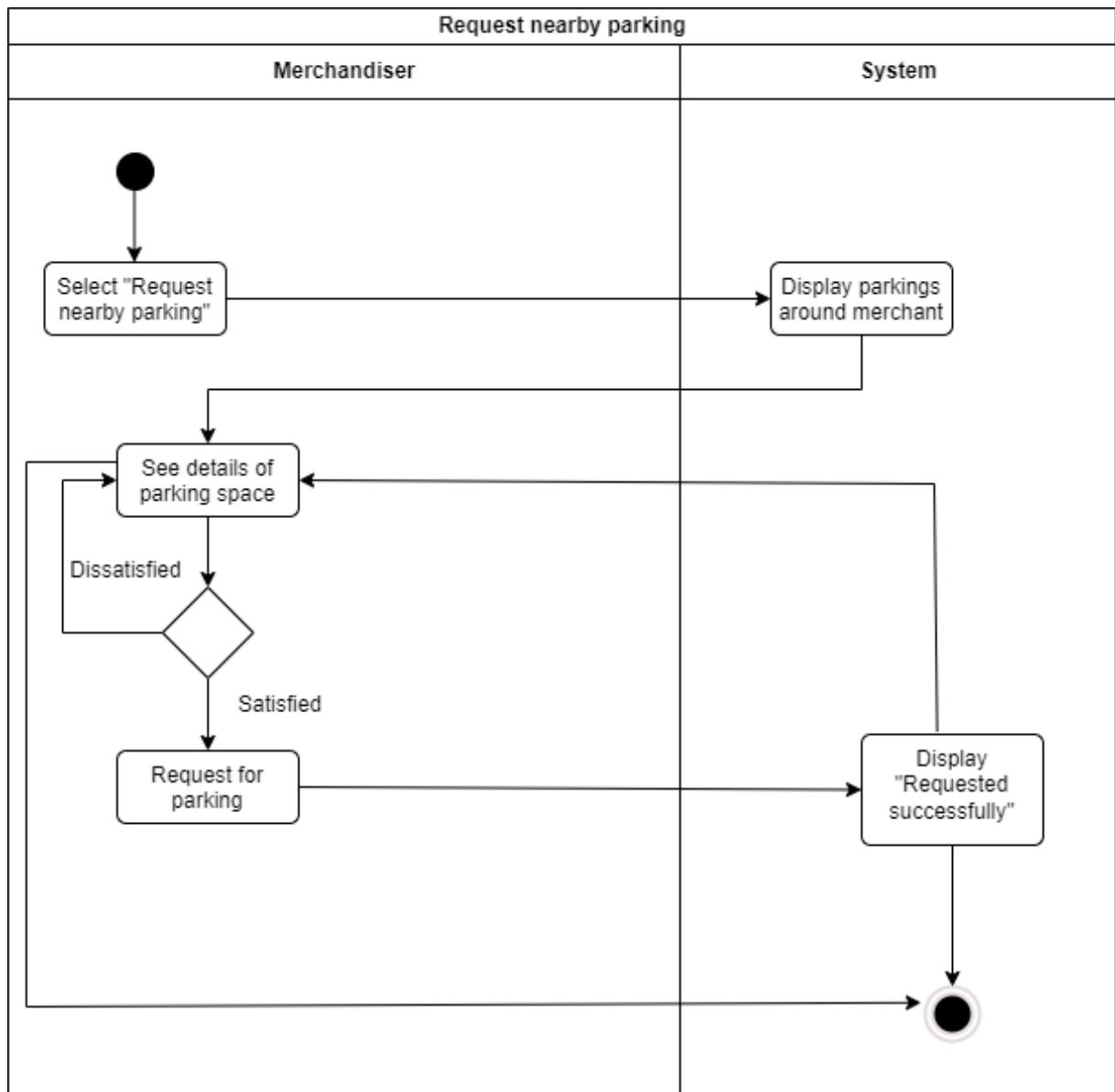


Figure 93:Request nearby parking

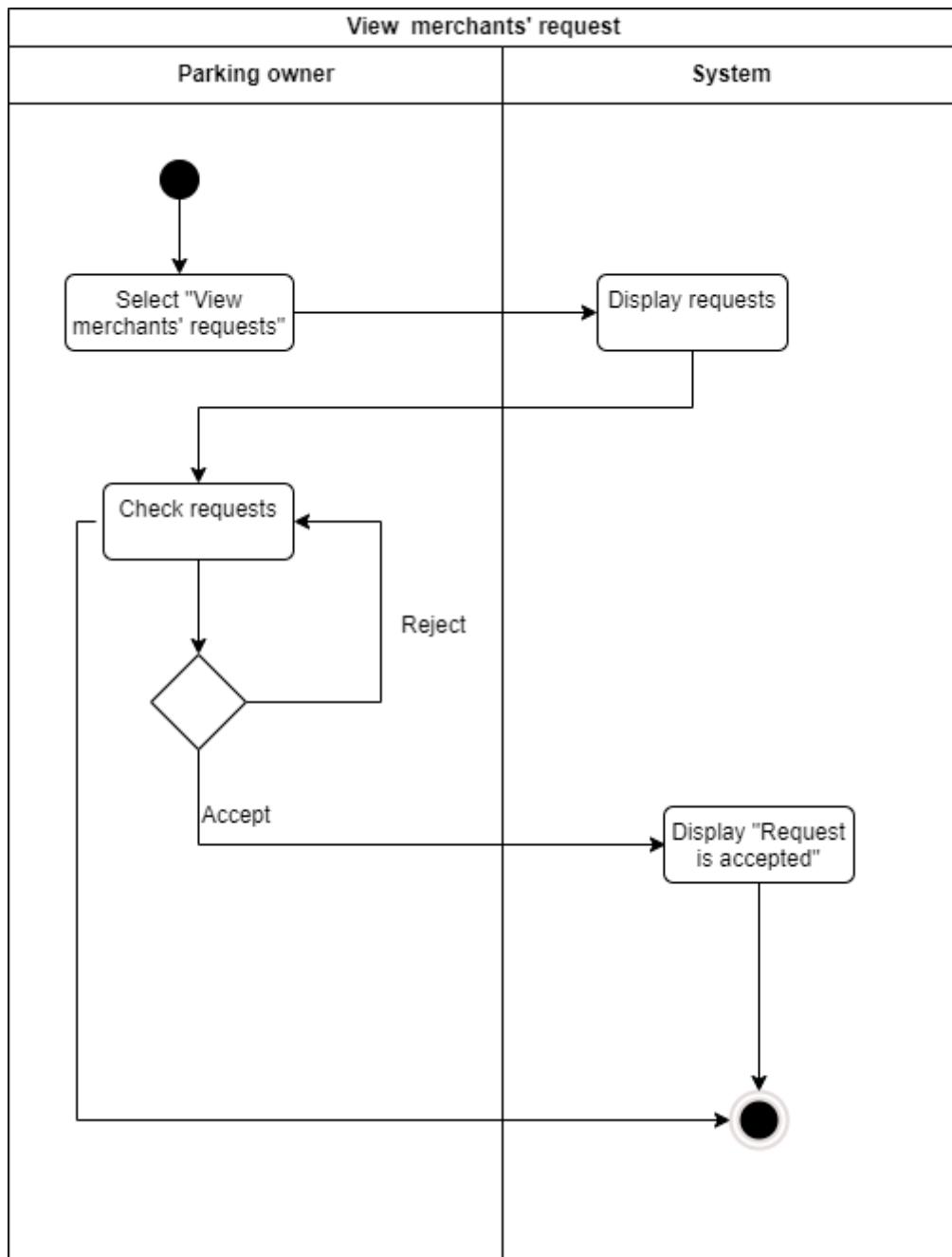


Figure 94:View merchant's request

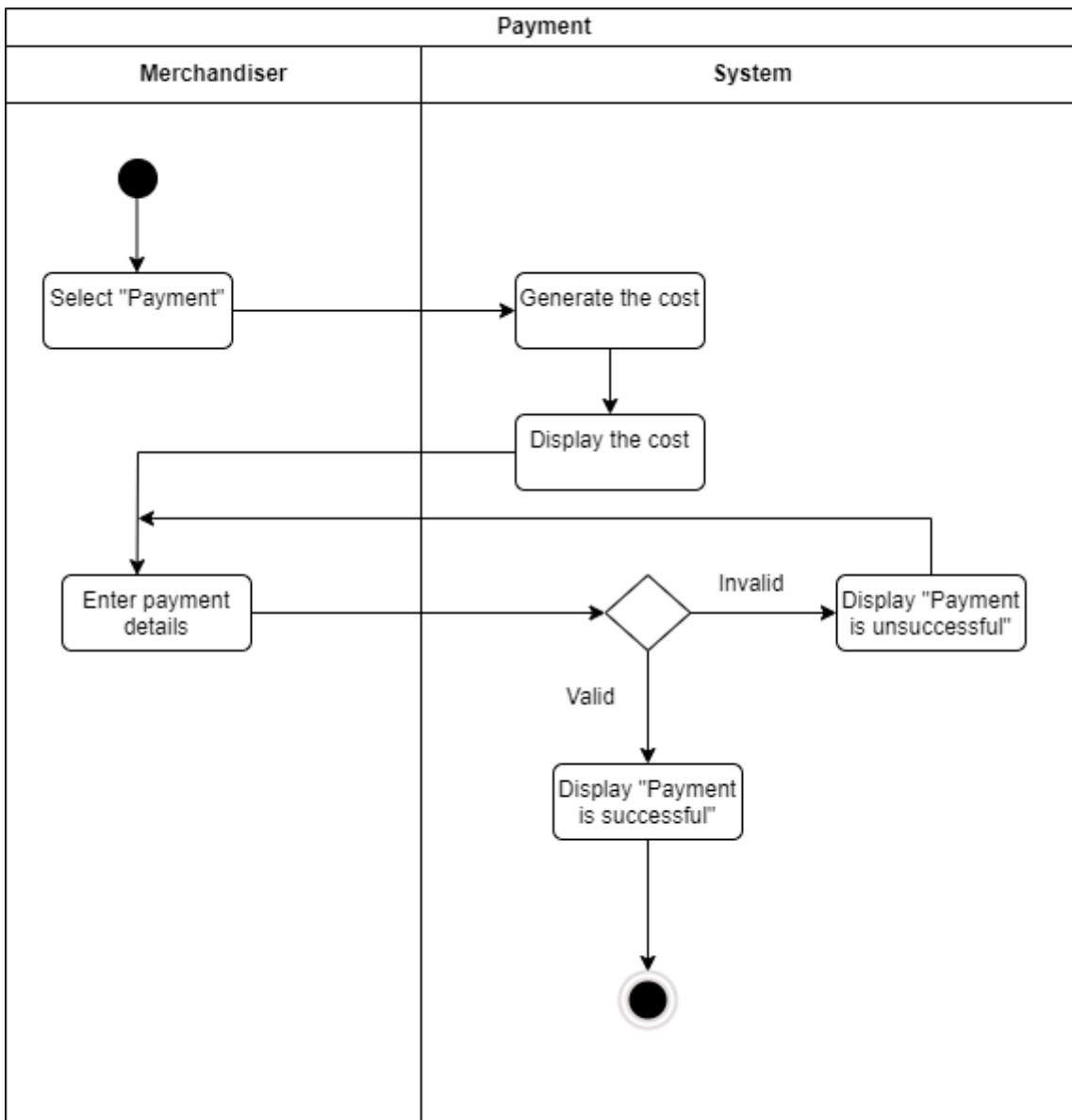


Figure 95:Payment

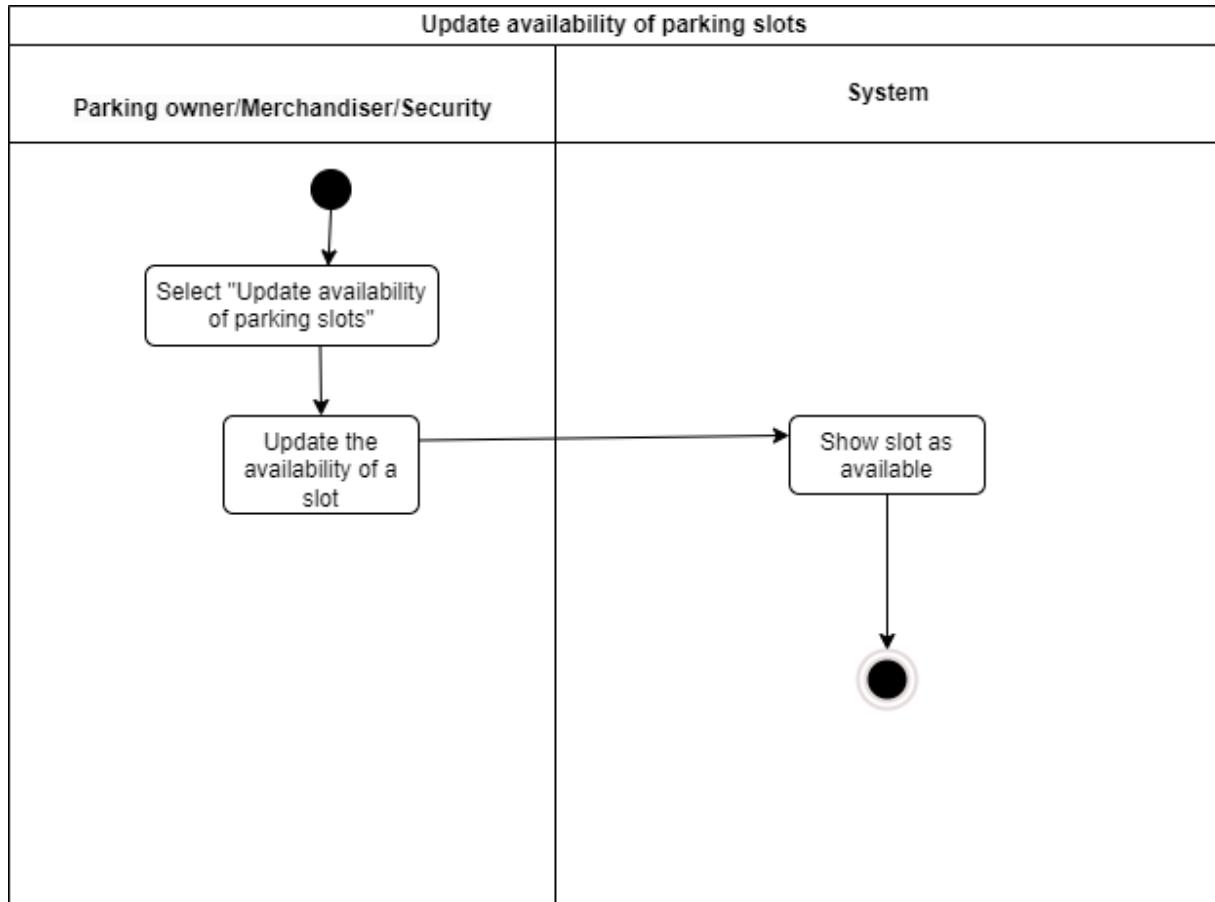


Figure 96:Update availability of parking slots

### 5.4.3 Driver

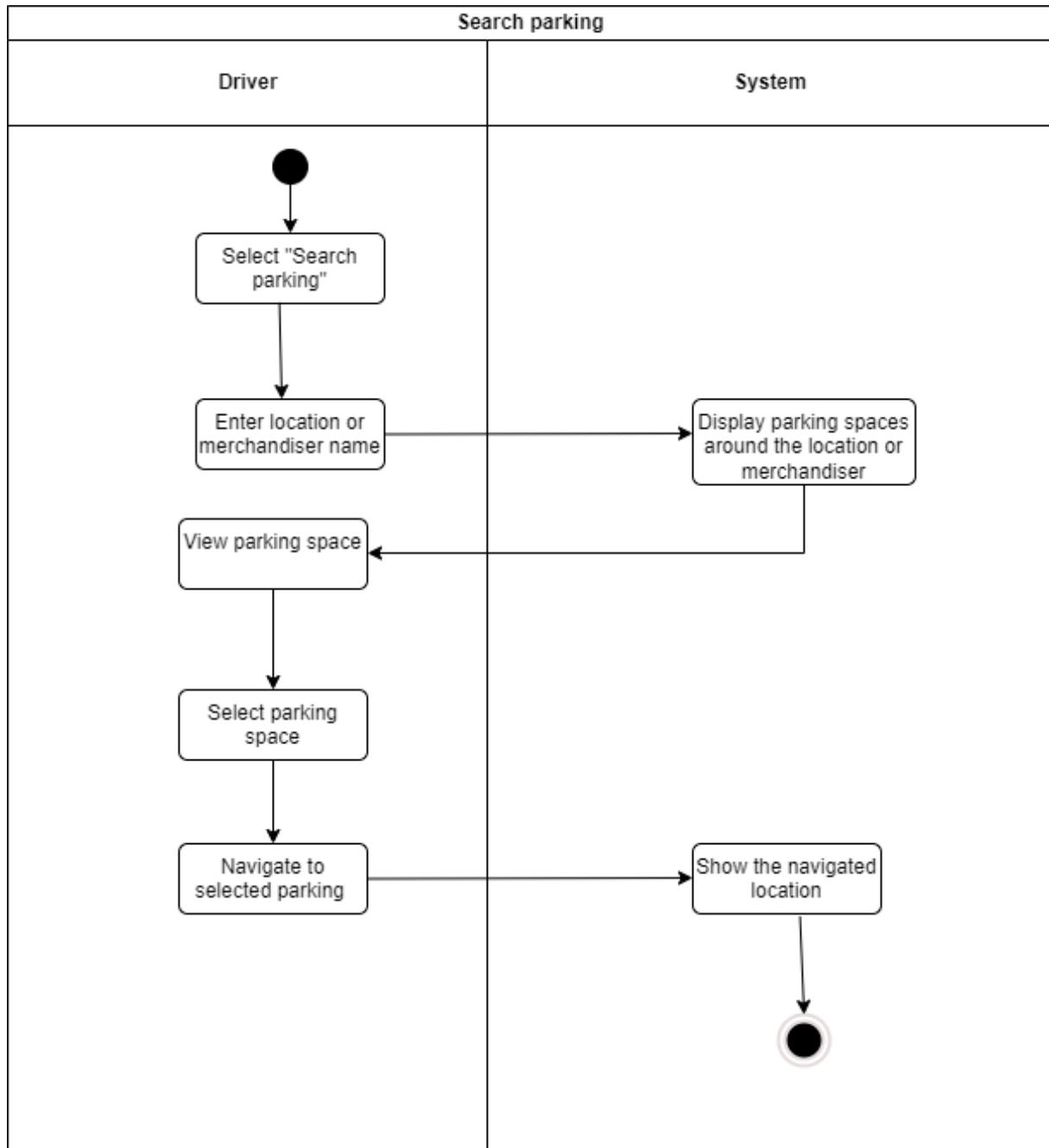


Figure 97:Search parking

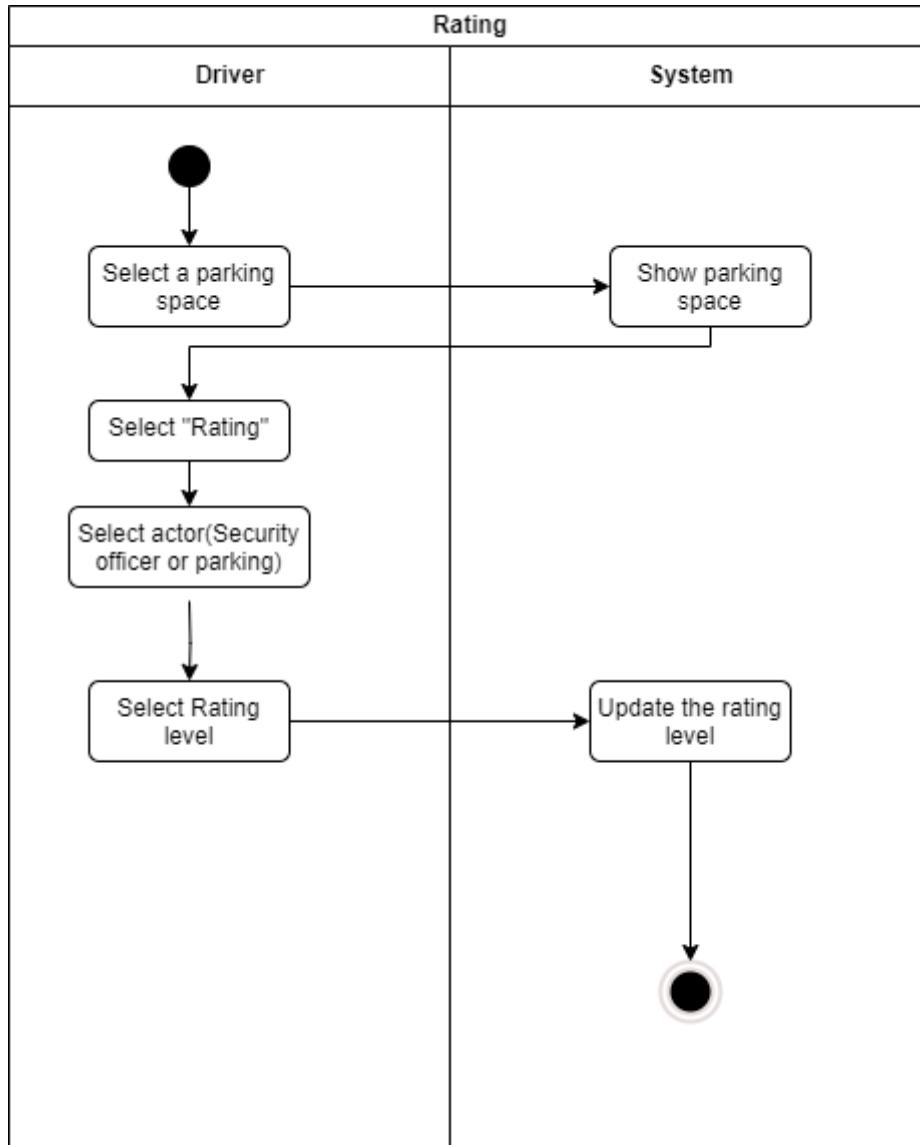


Figure 98: Rating

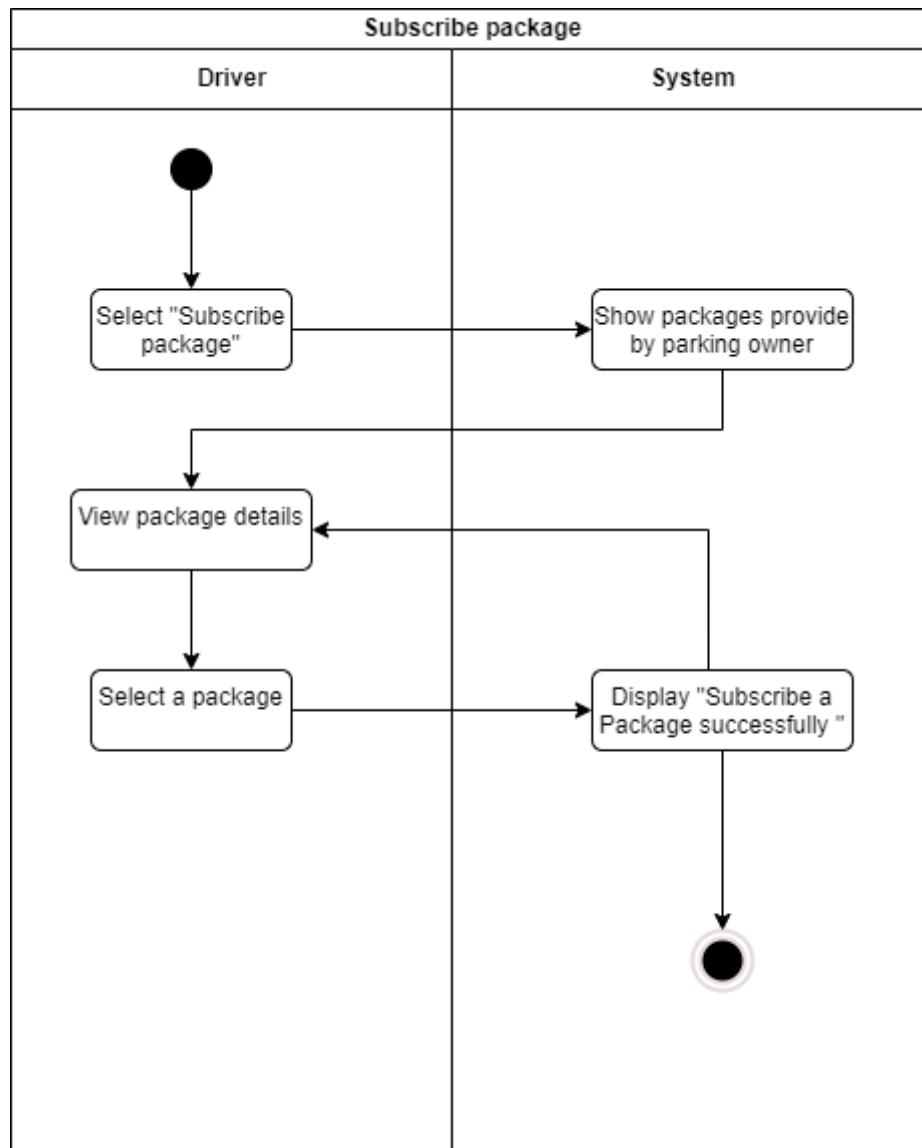


Figure 99:Subscribe package

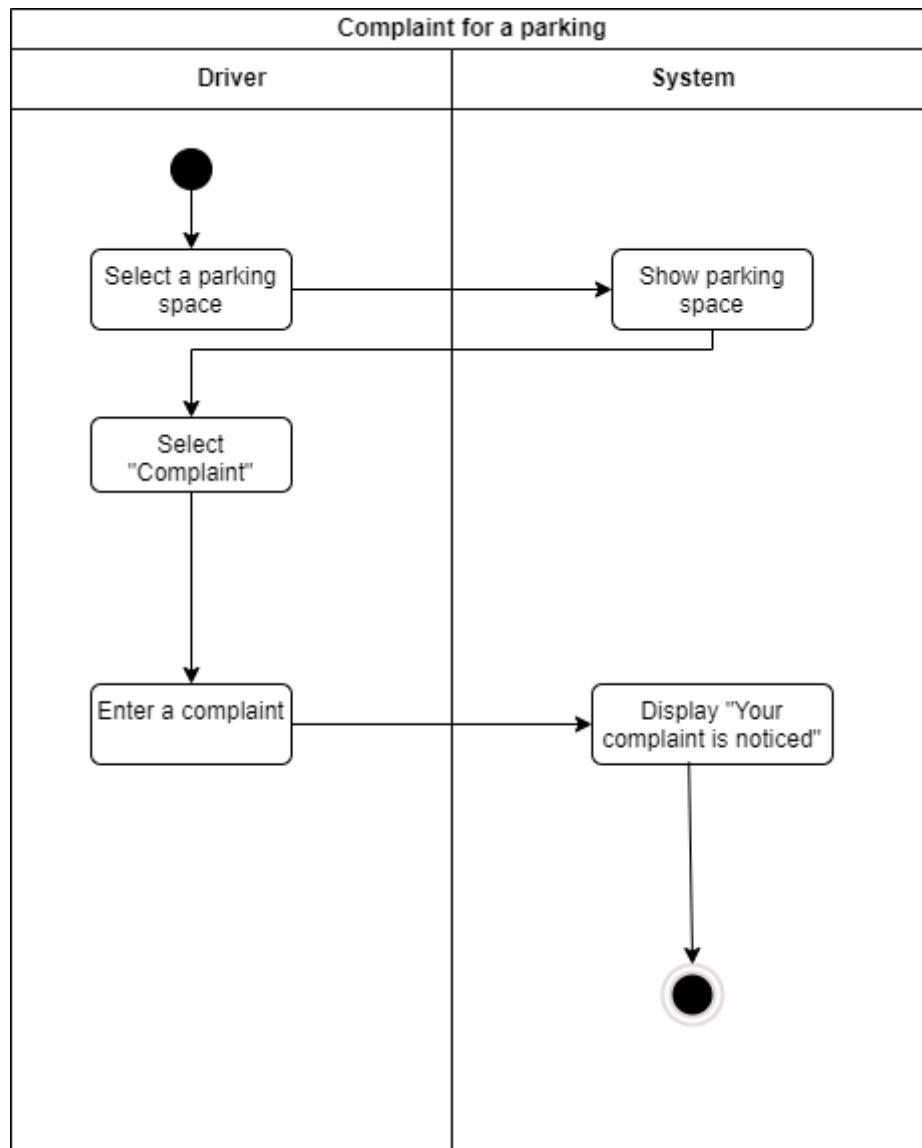


Figure 100:Complaint for a parking

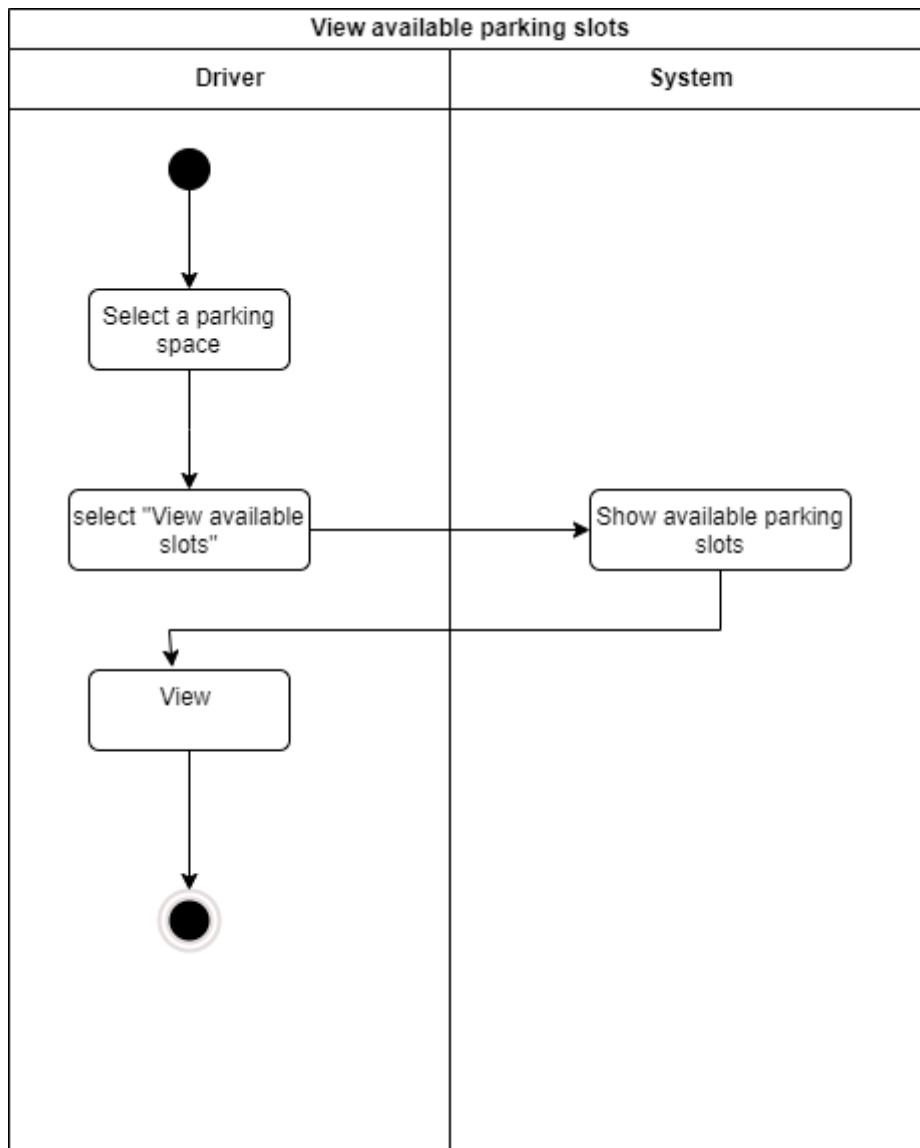


Figure 101:View available parking slots

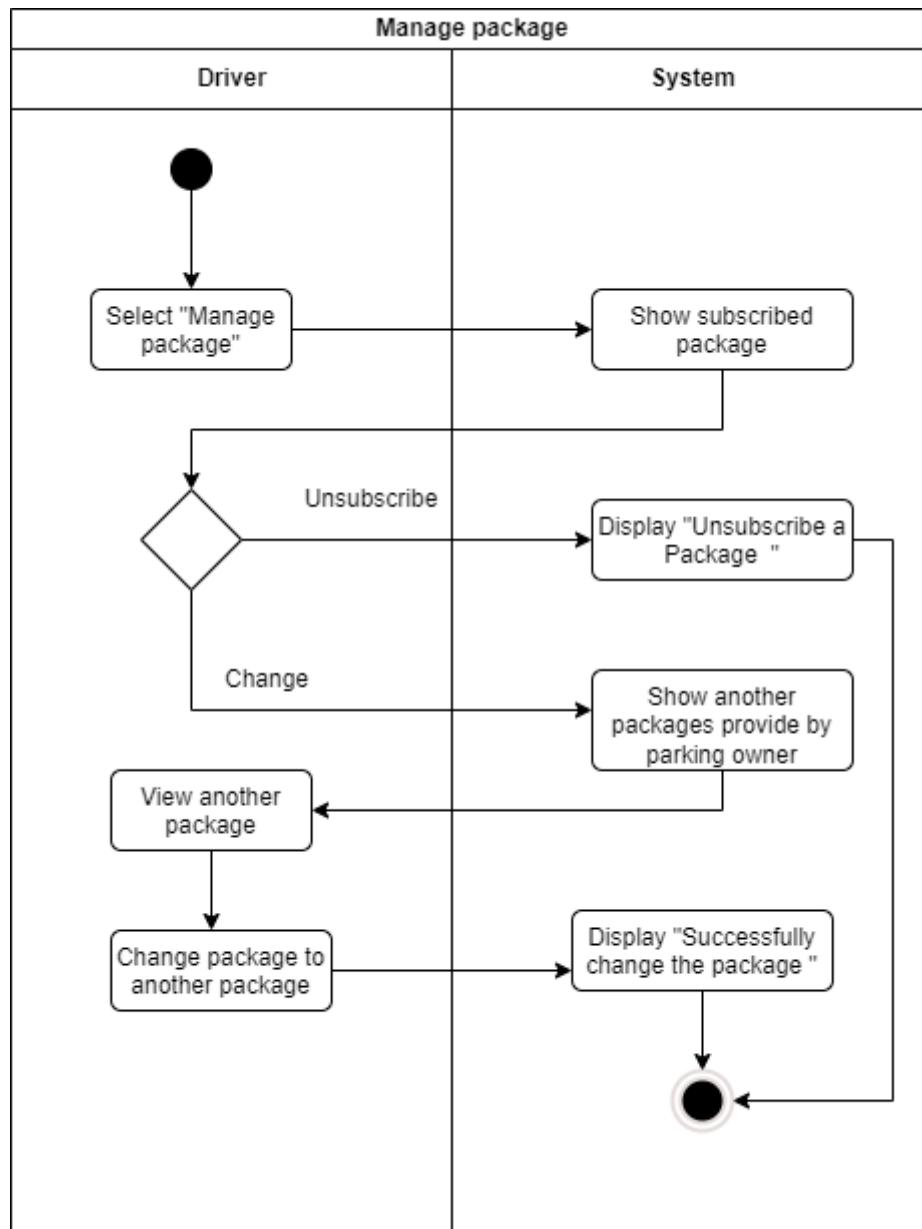


Figure 102:Manage package

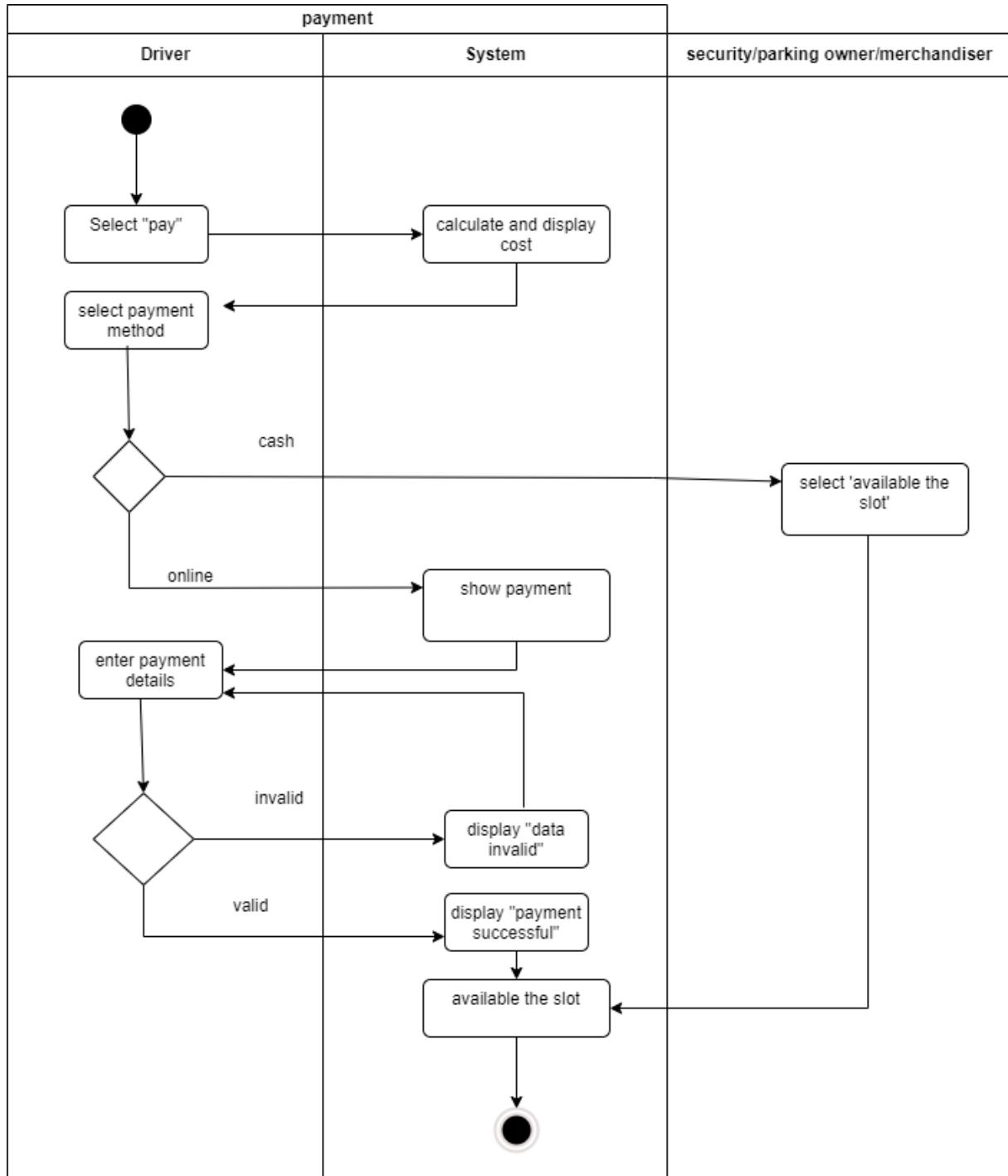


Figure 103:Payment

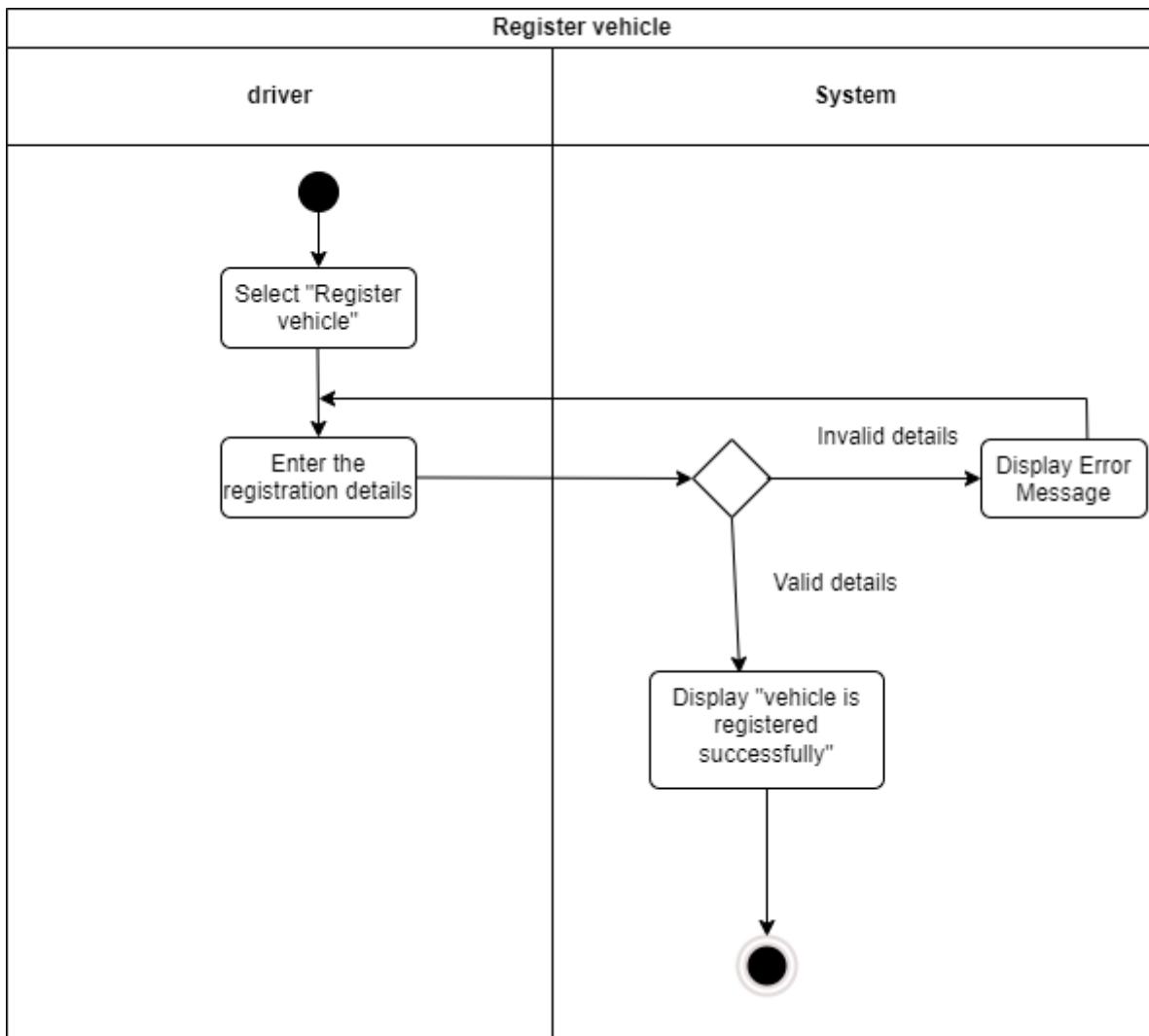


Figure 104:Register vehicle

#### 5.4.4 Security

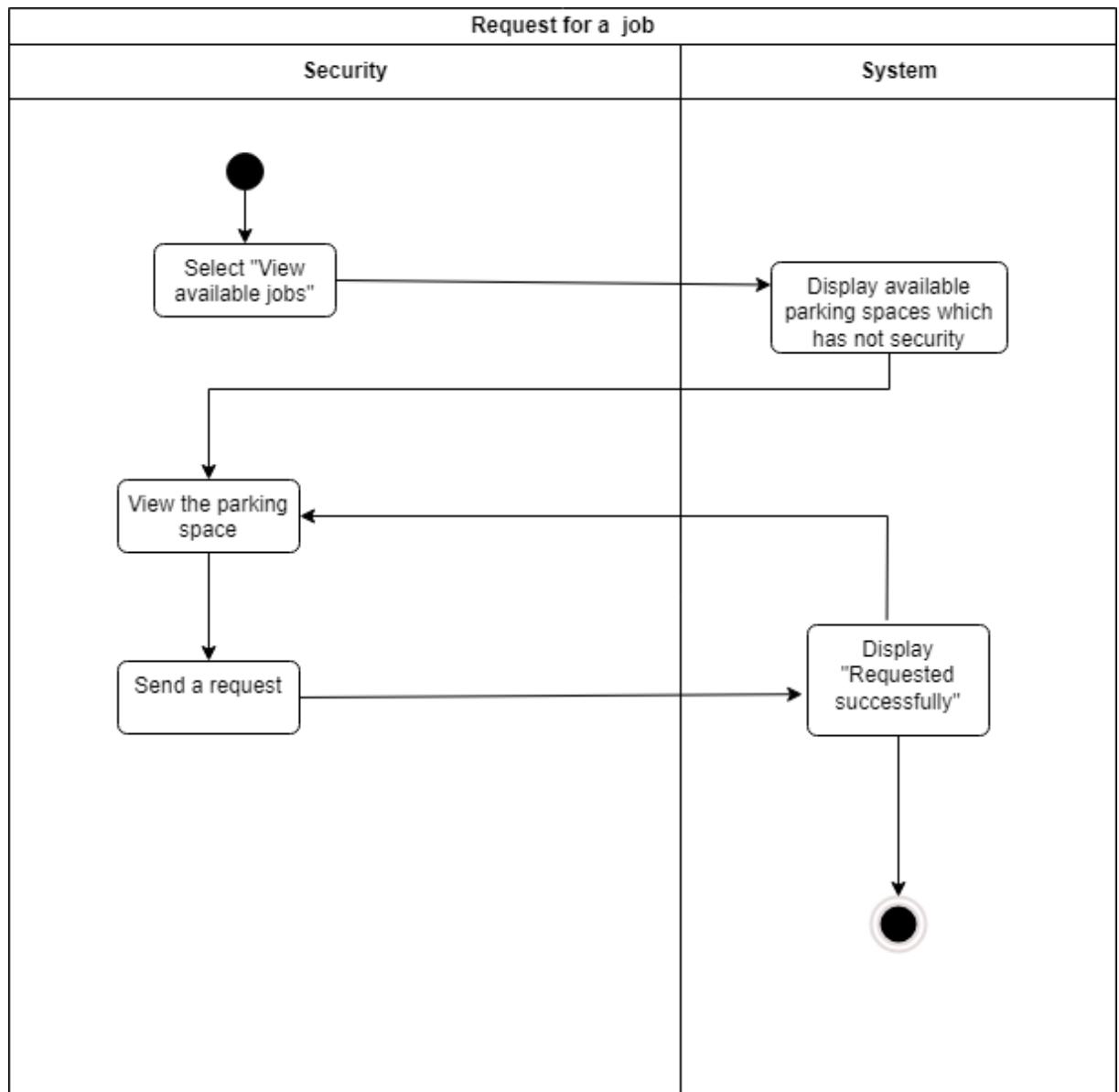


Figure 105:Request for a job

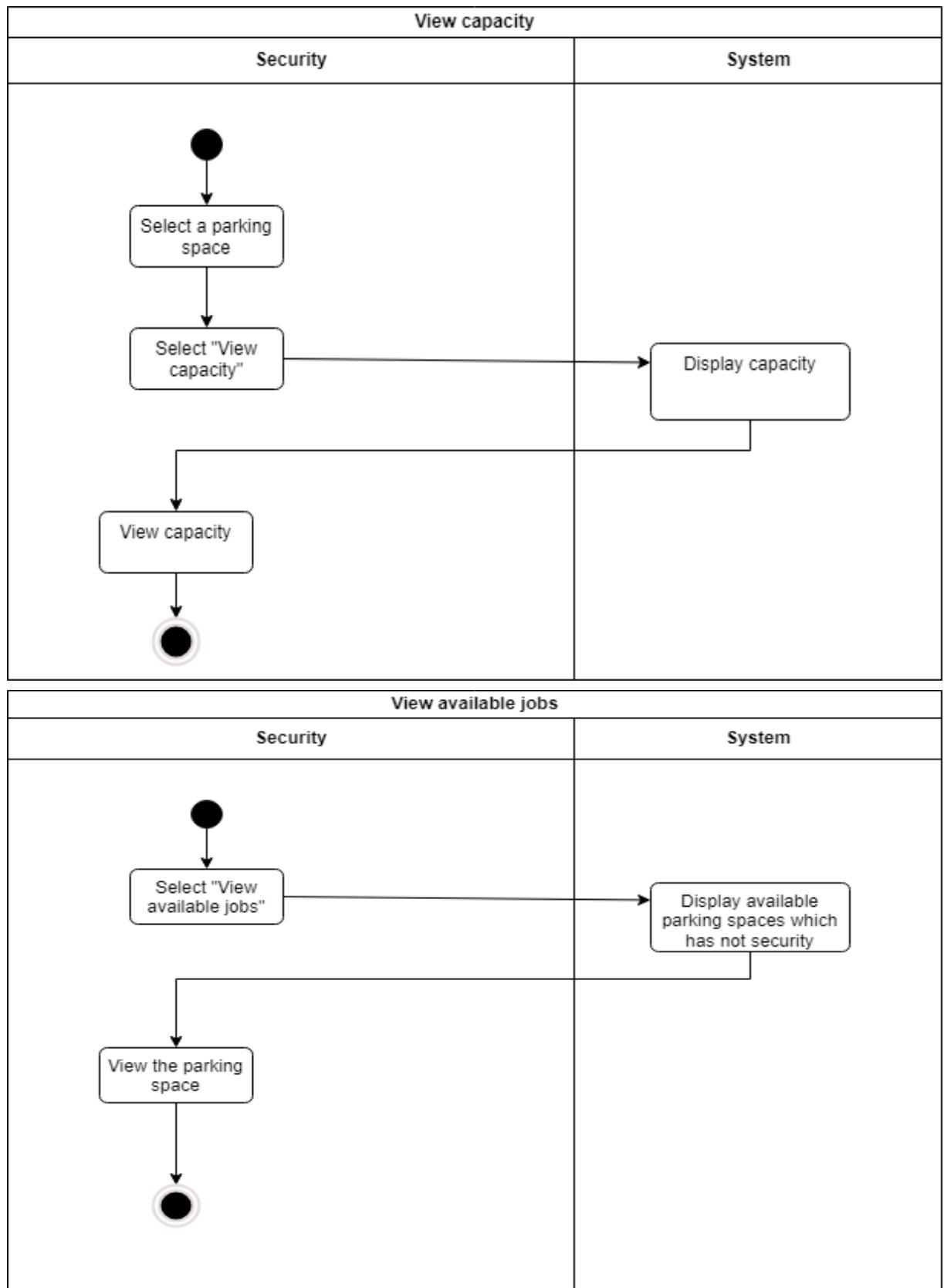


Figure 106:View capacity

Figure 107:View available jobs

#### 5.4.5 Admin

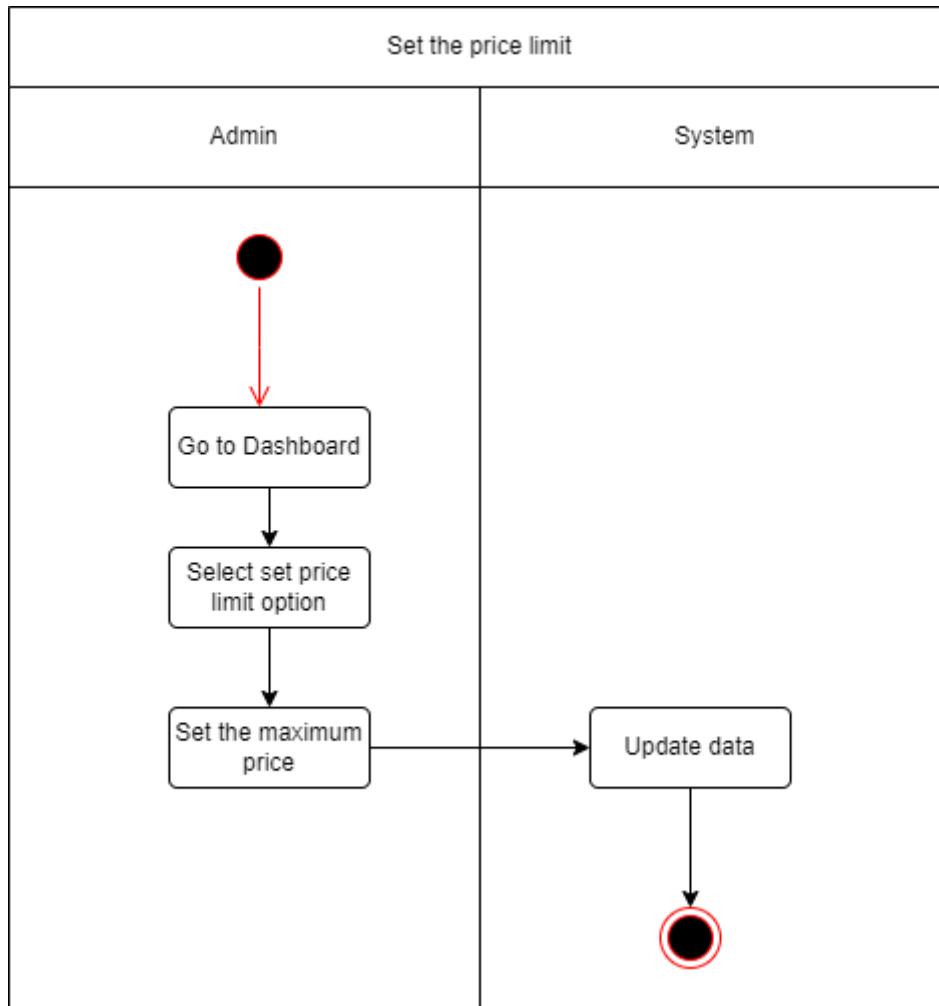


Figure 108: Set the price limit

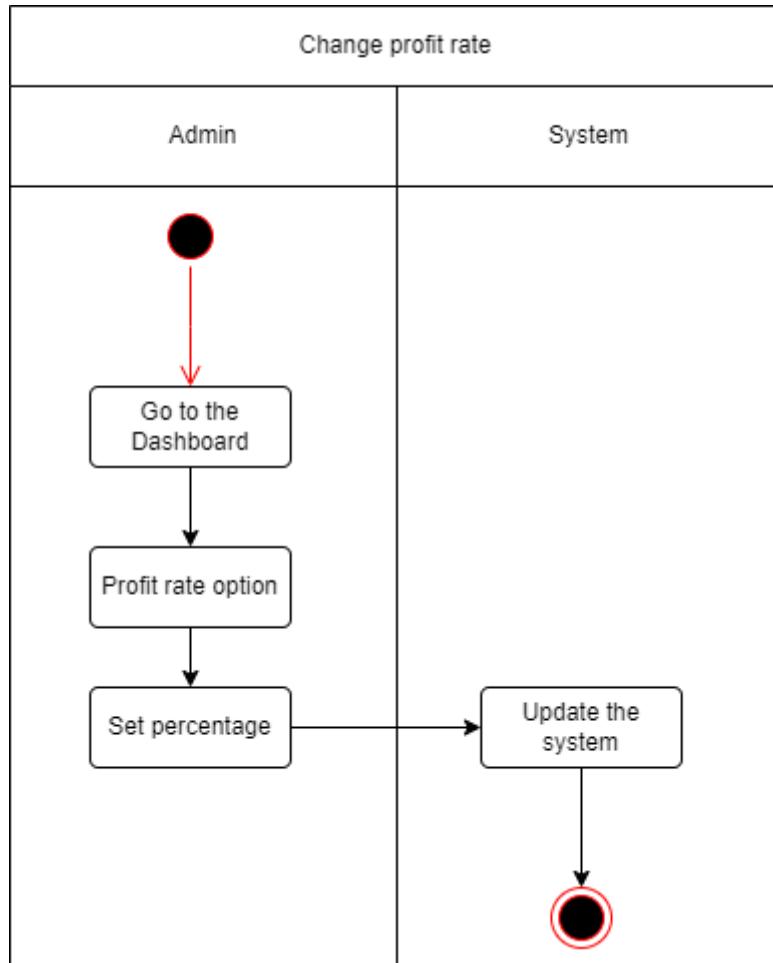


Figure 109:Change profit rate

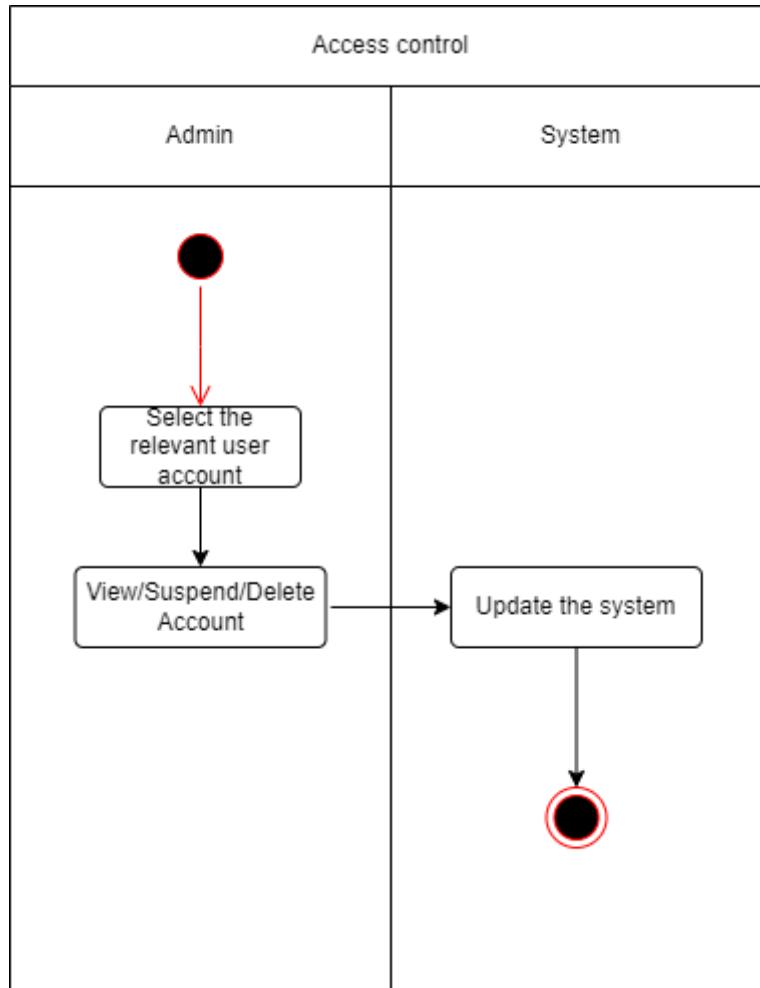


Figure 110:Access control

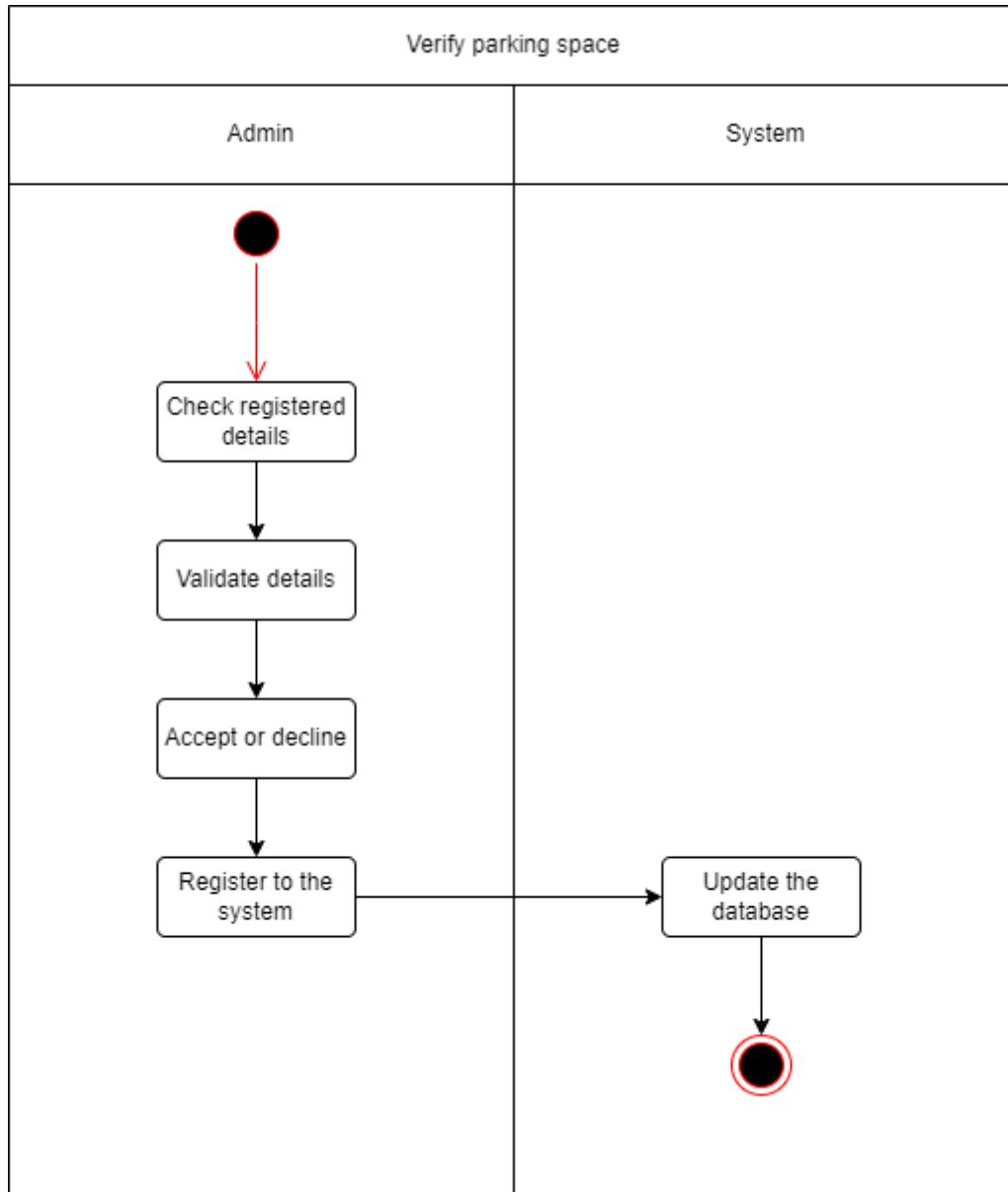


Figure 111:Verify parking space

Activity Diagram link :[activity diagrams](#)

## 5.5 State transition diagrams

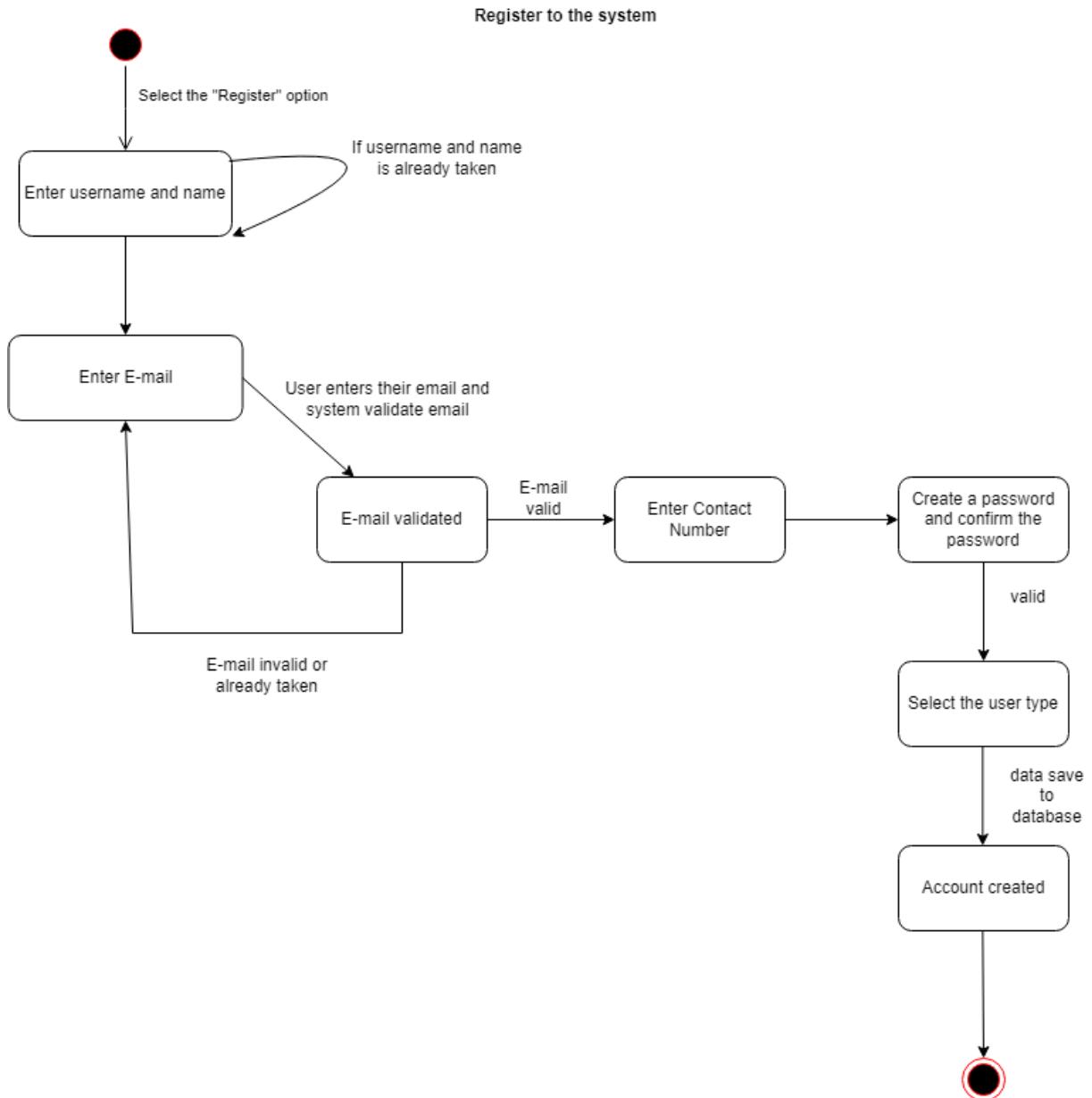


Figure 112: Register to the system

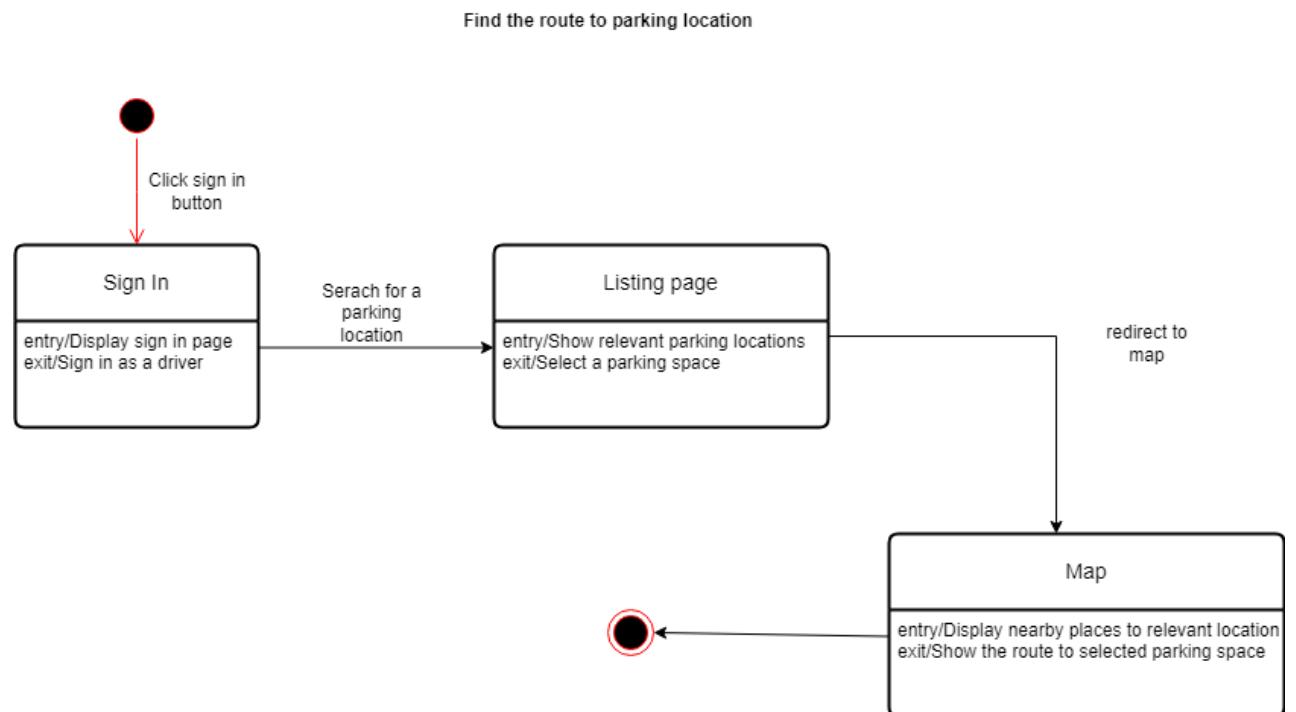


Figure 113:find the route to parking location

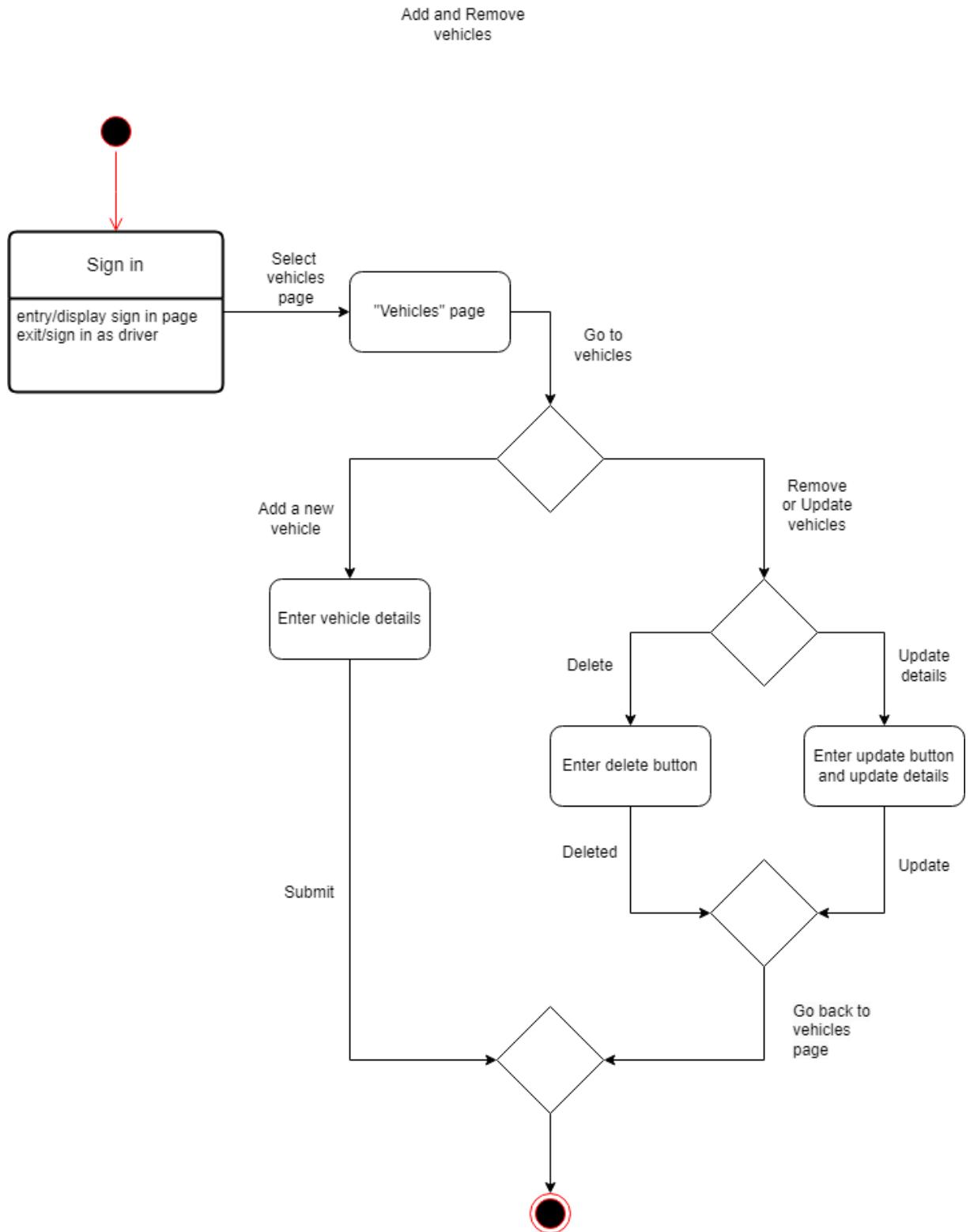


Figure 114: Add and remove vehicles

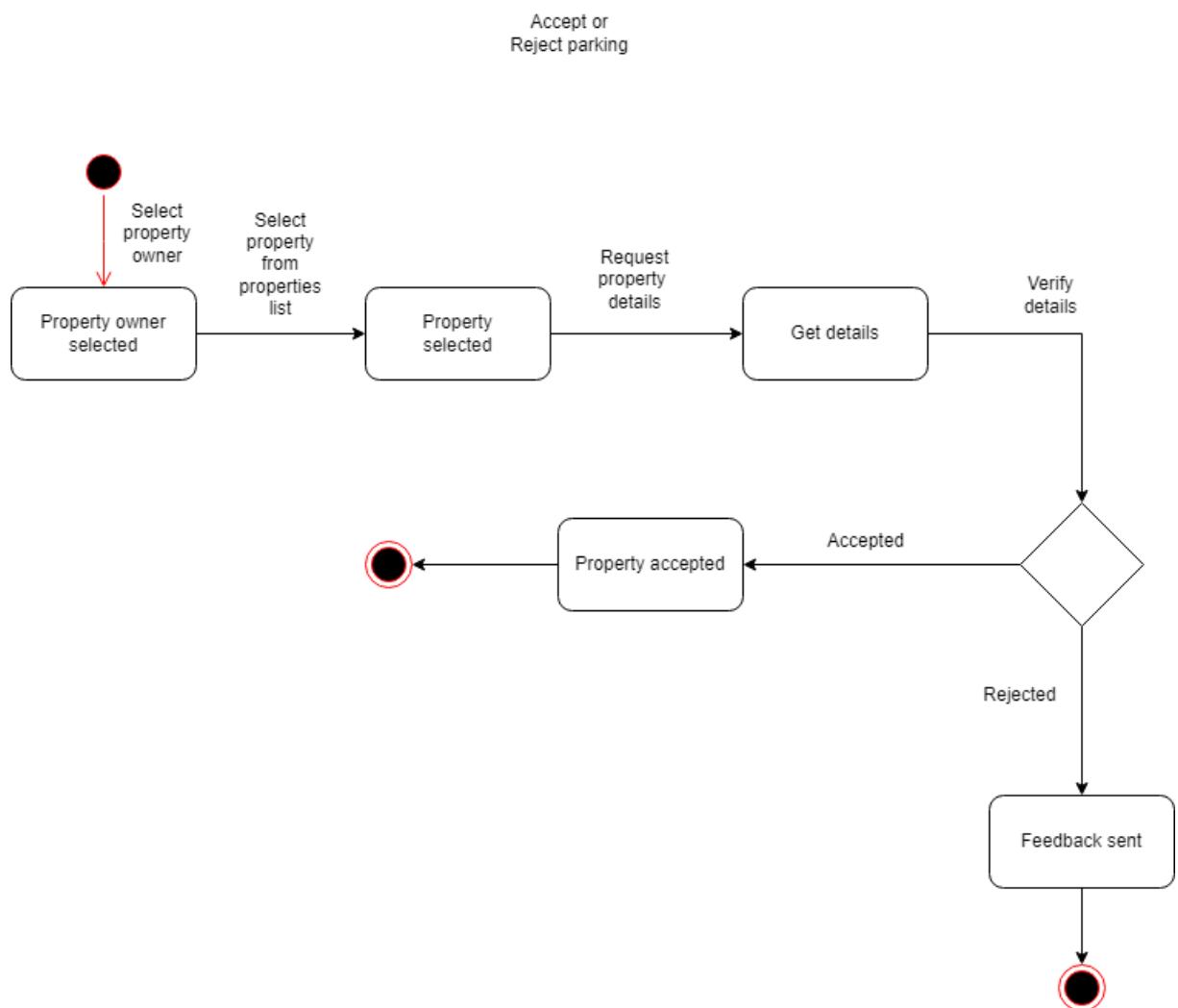


Figure 115:Accept or reject parking

## 6 User Interface Flow Diagram using wireframes

### 6.1 Admin



## 6.2 Parking Owner

**Signup**

**Login**

**LAND OWNER**

**register a parking**

**Set Capacity of your parking**

**Frame**

**Succ-property registration**

**LAND OWNER**

**Succ-property registration**

**property Details**

**MacBook Pro 16" - 8**

**MacBook Pro 16" - 9**

**Packages - dashboard**

**MacBook Pro 16" - 10**

**MacBook Pro 16" - 10**

**MacBook Pro 16" - 11**

**MacBook Pro 16" - 12**

**MacBook Pro 16" - 13**

**Manage packages**

**MacBook Pro 16" - 14**

**MacBook Pro 16" - 15**

**MacBook Pro 16" - 16**

**Update parking space availability**

**Update security officer availability**

**RathaGala**

**Create weekly package**

**Create monthly package**

**Create yearly package**

**Create daily package**

**RathaGala**

**View monthly package**

**Edit monthly package**

**Do you want to delete current weekly package?**

**Update Availability of parking slots**

**Edit capacity of the parking space**

**Change current Price per hour(s)**

## 6.3 Driver

**Driver - Vehicle**

**Driver - Empty vehicle**

**Driver - Update vehicle**

**Driver - Add vehicle**

**Driver - Empty vehicle**

**Driver - Dasboard**

**Driver - Location search**

**Driver - Location result**

**Driver - Location selection**

**Driver - Location direction**

**QR Scan - start**

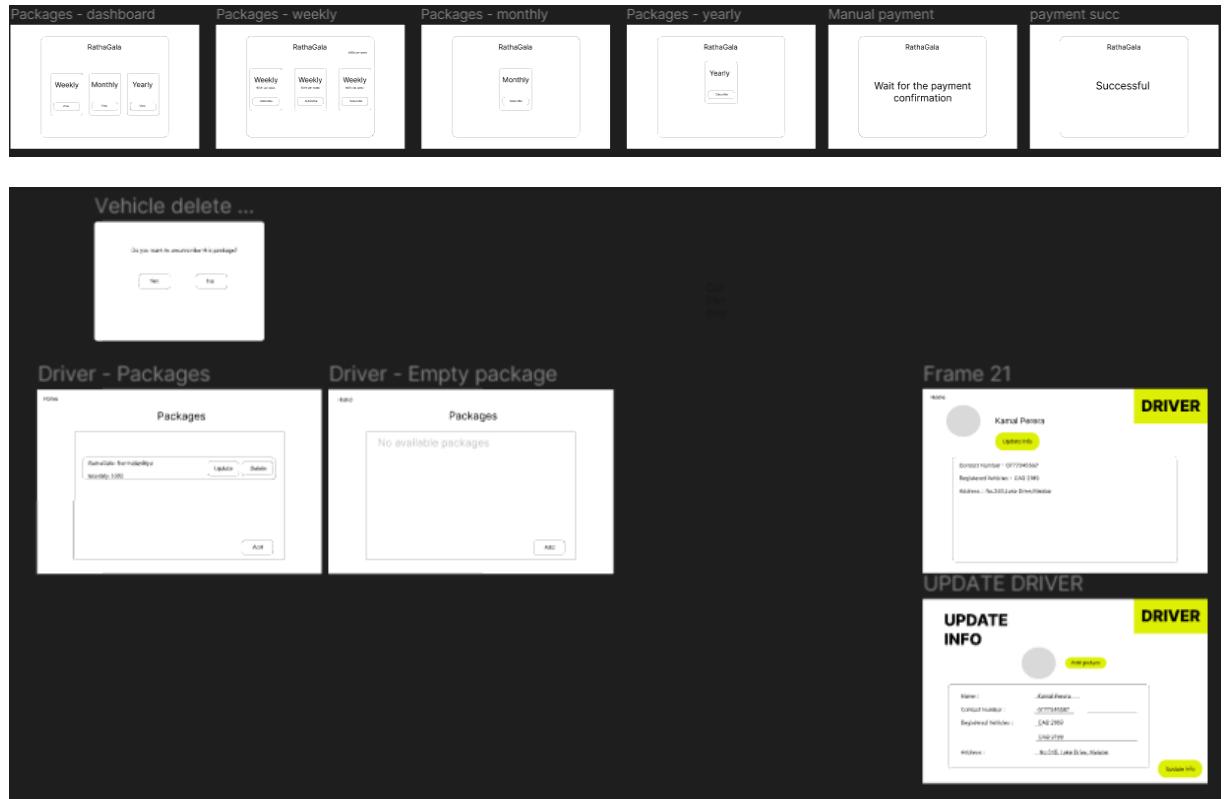
**Confirm entrance**

**Time start**

**Cost for parking**

**Online payment**

**payment succ**



## 6.4 Security



To Access the Wire Frames : [Figma Wire Frame Link](#)