

Chapter 1: Introduction

1.1 Purpose

This document is the Software Requirements Specification (SRS) for "Vehicle Renting and Management System (VRMS)". It contains detailed functional, non-functional, and support requirements and establishes a requirements baseline for development of the system. The requirements contained in the SRS are independent, uniquely numbered, and organized by topic. The SRS serves as the official means of communicating user requirements to the developer and provides a common reference point for both the developer team and stakeholder community. The SRS will evolve over time as users and developers work together to validate, clarify and expand its contents.

1.2 Intended Audience

This SRS is intended for several audiences, including the customer, as well as the project managers, designers, developers, and testers.

- The customer will use this SRS to verify that the developer team has created a product that is acceptable to the customer.
- The project managers of the developer team will use this SRS to plan milestones and a delivery date, and ensure that the developing team is on track during development of the system.
- The designers will use this SRS as a basis for creating the system's design. The designers will continually refer back to this SRS to ensure that the system they are designing will fulfill the customer's needs.
- The developers will use this SRS as a basis for developing the system's functionality. The developers will link the requirements defined in this SRS to the software they create to ensure that they have created software that will fulfill all of the customer's documented requirements.

- The testers will use this SRS to derive test plans and test cases for each documented requirement. When portions of the software are complete, the testers will run their tests on that software to ensure that the software fulfills the requirements documented in this SRS.

1.3 Conclusion

This analysis of the audience helped us to focus on the users who will be using our analysis. This overall document will help each and every person related to this project to have a better idea about the project.

Chapter 2: Inception

In this chapter, the Inception part of the SRS will be discussed briefly.

2.1 Introduction

Inception is the beginning phase of requirements engineering. It defines how does a software project get started and what is the scope and nature of the problem to be solved. The goal of the inception phase is to identify concurrence needs and conflict requirements among the stakeholders of a software project.

To establish the groundwork we have worked with the following factors related to the inception phases:

- Identifying Stakeholders
- Recognizing multiple viewpoints
- Working towards collaboration
- Requirements questionnaire

2.1.1 Identifying Stakeholders

Stakeholder refers to any person or group who will be affected by the system directly or indirectly. Stakeholders include end-users who interact with the system and everyone else in an organization that may be affected by its installation. At inception, a list of people who will contribute input as requirements are elicited. The initial list will grow as stakeholders are contacted because every stakeholder will be asked: "Whom else do you think I should talk to?"

The following stakeholders were identified for the "Vehicle Renting and Management System".

- Vehicle owner:** Vehicle owner are the one who owns the vehicles.

- Vehicle Borrower/Customer:** Vehicle Borrower/customer are the individuals who need the vehicles. He/she can borrow one or more vehicles.
- Admin:** Admin are the one who control the interconnection between user and the system.

Depending on the stakeholders role we have asked the following questions:

- When a customer have to confirm a truck?
- How would you prefer to book a truck?
- What are the necessary information to book?
- How customer communicate with owner before confirming truck?
- When the booking will be confirmed?
- How can an owner and a customer communicate with driver?
- Who will be responsible for goods load/unload?
- On What basis the pricing policy depend?
- Depending on its basis what will be the rate?
- In case if there is extra delay, what amount do you prefer to charge for waiting charge?
- What type of payment system will you prefer?
- How much percent of total amount owner want as advance?

2.1.2 Recognizing multiple viewpoints

Different stakeholders demand different features from the software. To satisfy the stakeholders, most of these features should be included in the software.

Vehicle Owner's Viewpoints:

- Suggestion on vehicle selection according to pick-up place, destination place, date of journey and vehicle's weight carrying capability.
- Notification after completion of booking to owner and customer.
- Capability of tracking vehicle.
- Cost policy according to journey length and seasons.
- Payment policy according to distance and extra delay.
- May have features to add and remove vehicle and driver information.
- Should contain messaging and standard call system for communication.

- Capability of storing trip history and details for every single trip

Vehicle Borrower's/customer's Viewpoints:

- Should have filters to select pick-up location, destination and vehicle according to journey date and weight capability of truck.
- Notification after completion of booking to owner and customer.
- Must contain support to handle emergency and accidental situation through messaging and calling.
- Capability of tracking vehicle.
- May have features and technical support if a customer cancels a trip.
- Must contain features to provide feedback.
- Must contain the list of vehicles that they have booked.
- Should contain messaging and standard call system for communication.
- Capability of storing trip history and details for every single trip

2.1.3 Working towards Collaboration

While working with different stakeholders, some conflicting and common viewpoints can be noticed. For this reason, final requirements can be gotten by collaborating the viewpoints. We followed following steps to merge these requirements:

- Identify the common and conflicting requirements
- Categorize the requirements
- Take priority points for each requirements from stakeholders and on the basis of this voting prioritize the requirements
- Make final decision about the requirements

Common Requirements:

- Notification after completion of booking to owner and customer.
- Capability of tracking vehicle.
- Should contain messaging and standard call system for communication.
- Capability of storing trip history and details for every single trip.

Conflicting Requirements:

- May have features and technical support if a customer cancels a trip.

Final Requirements:

- Suggestion on vehicle selection according to pick-up place, destination place, date of journey and vehicle's weight carrying capability.
- Notification after completion of booking to owner and customer.
- Capability of tracking vehicle.
- Cost policy according to journey length and seasons.
- Payment policy according to distance and extra delay.
- May have features to add and remove vehicle and driver information.
- Should contain messaging and standard call system for communication.
- Capability of storing trip history and details for every single trip
- Should have filters to select pick-up location, destination and vehicle according to journey date and weight capability of truck.
- Must contain support to handle emergency and accidental situation through messaging and calling.
- May have features and technical support if a customer cancels a trip.
- Must contain features to provide feedback
- Must contain the list of vehicles that they have booked.

Restrict access to functionality of the system based upon user roles. For example, only Administrators of the system will be provided functionality to configure vehicle availability.

2.1.4 Requirements Questionnaire

We set our first set of context-free questions to understand the project overall performance, goal and benefits. The questions are mentioned above in section 2.1.1. These questions helped us to identify all stakeholders, measurable benefit of the successful implementation and possible alternatives to custom software development. Then we asked our next Question.

2.2 Conclusion

The Inception phase helped us to establish basic understanding about the Vehicle Renting and Management System, identify the stakeholders who will be benefited if this system becomes automated, define the nature of the system and the tasks done by the system, and establish a preliminary communication with our stakeholders. More studies and communication will help both side (developer and user) to understand the future prospect of the project. Our team believes that the full functioning document will help us to define that future prospect.

Chapter 3: Elicitation

3.1 Introduction

Elicitation is a part of requirements engineering that is the practice of gathering requirements. Many difficulties were faced, like understanding the problems, making questions for the stakeholders, limited communication with the stakeholders due to a short amount of time and volatility. Though it is not easy to gather requirements within a very short time, these problems have been surpassed in an organized and systematic manner.

3.2 Eliciting Requirements

The main task of this phase is to combine the elements of problem solving, elaboration, negotiation and specification. The collaborative working approach of the stakeholders is required to elicit the requirements. The following tasks were done for eliciting requirements:

1. Collaborative Requirements Gathering
2. Quality Function Deployment
3. Usage Scenarios

3.2.1 Collaborative Requirements Gathering

Many different approaches to collaborative requirements gathering have been proposed. Each makes use of a slightly different scenario. We completed following steps to do it.

- The meeting were conducted with Program Chair. He was questioned about their requirements and expectation from the Vehicle Renting and Management system.

- The owner was asked about the problems he/she facing with the current system.
- At last we selected our final requirements from the meetings.

3.2.2 Quality Function Deployment

Quality Function Deployment (QFD) is a technique that translates the needs of the customer into technical requirements for software. It concentrates on maximizing customer satisfaction from the Software engineering process. With respect to our project the following requirements are identified by a QFD.

Normal Requirements: The normal requirements are generally the objectives and goals that are stated for a product or system during meetings with the customer. The presence of these requirements fulfills customers' satisfaction.

These are the normal requirements for the project.

- Accessible via the Internet.
- Administrator to check availability of vehicle for user or not.
- Suggestion on vehicle selection according to pick-up place, destination place, date of journey and vehicle's weight carrying capability.
- Notification after completion of booking to owner and customer.
- Capability of tracking vehicle.
- Should contain messaging and standard call system for communication.
- Capability of storing trip history and details for every single trip

Expected Requirements: These requirements are intrinsic to the product or system and may be so elementary that the customer does not explicitly state them. Their absence will be a cause for significant dissatisfaction. Below the expected requirements for our project are briefly described.

- User friendly software
- Strong authentication system
- Cost policy according to journey length and seasons.
- Payment policy according to distance and extra delay.

- Should have filters to select pick-up location, destination and vehicle according to journey date and weight capability of truck.
- Must contain support to handle emergency and accidental situation through messaging and calling.
- May have features and technical support if a customer cancels a trip.
- Must contain the list of vehicles that they have booked.

Exciting Requirements: These requirements are for features that go beyond the customer's expectations and prove to be very satisfying when present

- User interface should follow standard android interface.
- Must contain features to provide feedback.
- Should contain messaging and standard call system for communication.
- System automatically sends notification to the borrower/customer after delivery.

3.2.3 Usage Scenario

Authentication

- **Sign Up:** In the "Vehicle Renting and Management System", there is an authentication part, where it allows the user to access the system. Administrators, vehicle owner and vehicle borrower/customer are the user of this system. To access the system, a user requires an account and for that he/she must fill up a form. Users have to give phone number, username, national ID, password, account type. All information must maintain regular expression. If all the information is correct, user will be sent a verification code to his/her mobile number.
- **Sign In:** If any user has an account, he/she sign in to the system. To sign in, user has to give his/her mobile number and password. The mobile number and password will be verified. If the verification is successful, the user can sign in to the system successfully and his/her login information (date and time of login) will be stored. If the password is wrong there is retry option. If the number of retry exceeds

its maximum chance given, the user will be blocked for 1 minute. After block time has finished user can attempt to sign in to the system again. If the user forgets the password, he has an option to change password by entering its mobile number. New verification code is sent to the mobile number and the user can sign in again using the new verification code.

- **Sign out:** User has an option to sign out from the system.

Search

- **Search by pick up location:** If the borrower/customer want to search the vehicle by pick up location, he/she need to select location from search bar.
- **Search by vehicle capacity:** If the borrower/customer want to search the vehicle by its carrying capacity, he/she need to select capacity from list.
- **Search by pick up date:** If the borrower/customer want to search the vehicle by pick up date, he/she need to select date from calendar.
- **Search by goods weight:** If the borrower/customer want to search the vehicle depending on its goods weight, he/she need to enter weight of goods.

Maintenance

- **Add Driver Information:** In "Vehicle Renting and Management System" Owner can add driver information. Owner have to give driver's name, permanent address, NID, mobile number, driving license number. All information maintains regular expression.
- **Add Vehicle Information:** In "Vehicle Renting and Management System" Owner can add vehicle information. Owner have to give vehicle number, Blue book, vehicle type, vehicle capacity.

- **Remove Driver:** Owner can remove driver from the system when they are out of contact. If any driver don't work, owner can remove them.
- **Remove Vehicle:** Owner can remove vehicles from the system when they are out of service. Non-functional vehicles can be removed from the list.
- **Tracking:** Owner and borrower/customer can track the vehicle which are on the way to serve. The owner select the vehicles and position of vehicle will be showed up in new window.

Booking

- **Schedule a trip:** customer can schedule a trip.
- **Trip Confirmation:** Customer can confirm their trip.
- **Trip cancellation:** Customer can cancel their trip before specified time.

Transaction

- **Advance Online Payment:** If any borrower/customer wants to pay online, he/she can pay via mobile banking. If he/she wants to pay via mobile banking he/she can pay via bKash or DBBL mobile banking. To pay he/she has to give the information and complete the payment.
- **Cash on Delivery:** If any borrower/customer wants to pay offline, he/she can pay cash on delivery. It gives borrower/customer opportunity to pay after goods delivery. But in this case borrower/customer must pay some advance to the vehicle owner.

Notification

- **Confirmation notification:** Both vehicle owner and borrower/customer receives confirmation notification from the system
- **Delivery notification:** After goods delivery, both vehicle owner and borrower/customer receives the Delivery notification from the system.

- **Delay notification:** Both vehicle owner and borrower/customer receives delay notification when trip delay occurs.
- **Cancellation notification:** Both vehicle owner and borrower/customer receives cancel notification when borrower/customer cancel the trip.

Communication

- **Standard Call system:** It gives information on how vehicle owner and borrower/customer communicate through. The standard call system connect vehicle owner and borrower/customer through standard call system. Customer can call to the owner.
- **Standard message system:** The standard message system connect vehicle owner and borrower/customer through messaging. Customer can message to the owner.

Chapter 4: Scenario Based Model

4.1 Introduction

In this model the system is described from the user's point of view. As this is the first model, it serves as input for creation of other modeling elements.

4.2 Use Case Scenario

Table 4.2.1: Usage case scenario

Vehicle Renting And Management System	Authentication	Sign In
		Sign up
		Sign out
	Search	By pick up location
		By vehicle capacity
		By pick up Date
		By Vehicle type
	Maintenance	Add Driver Information
		Add Vehicle Information
		Remove driver
		Remove vehicle

		Tracking
Booking		Schedule Trip
		Confirm a trip
		Cancel a trip
Payment		Advance Online Booking
		Cash on delivery
Notification		Confirmation
		Delivery
		Delay
		Cancelation
Communication		Standard call system
		Standard message system

4.3 Use Case Descriptions

4.3.1 Authentication

4.3.1.1 Sign up

Use Case: Sign up

Primary Actor: Admin, Vehicle owner, Borrower/Customer

Goal in context: to create an account

Precondition:

1. System has been designed for Sign up.
2. System has interface for Sign up.

Triggers: Admin, Vehicle owner, Borrower/Customer has to sign up.

Scenario:

- Visit page and sign up
- Provided the required information
- Check availability of username, mobile number and password
- Verify information
- Send confirmation code

Exception:

- Same mobile number
- Ambiguous Input
- Verification failed

Priority: Essential, must be implemented.

When Available: First increment

4.3.1.2 Sign In

Use Case: Sign In

Primary Actor: Admin, Vehicle owner, Borrower/Customer

Goal in context: to enter the system

Precondition:

- System has been designed for Sign In.
- System has interface for Sign In.

Triggers: Admin, Vehicle owner, Borrower/Customer has to sign in.

Scenario:

- Visit page and sign in
- Provided the required information
- Proceed for booking
- Forget password
- Send verification code to mobile number

Exception:

- Unrecognized mobile number
- Wrong password
- Verification failed

Priority: Essential, must be implemented.

When Available: First increment

4.3.1.3 Sign Out

Use Case: Sign Out

Primary Actor: Admin, Vehicle owner, Borrower/Customer

Goal in context: exit from the system

Precondition:

- System has been designed for Sign Out.
- System has interface for Sign Out.

Triggers: Admin, Vehicle owner, Borrower/Customer has to sign out.

Scenario:

- Visit page
- Click sign out button

Exception:

- User not logged in

Priority: Essential, must be implemented.

When Available: First increment

4.3.2 Search

4.3.2.1 By Pick up location

Use Case: Search by pick up location

Primary Actor: Borrower/Customer

Goal in context: to search a vehicle by its pick up location

Precondition:

- System has been designed for search by pick up location.
- System has interface for search by pick up location.

Triggers: Borrower/Customer has to search by pick up location.

Scenario:

- Visit page
- Click search button
- Click search by pick up location
- Select pick up location

Exception:

- Location not found
- Vehicles not available

Priority: Essential, must be implemented.

When Available: First increment

4.3.2.2 By Vehicle Capacity

Use Case: Search by vehicle carrying capacity

Primary Actor: Borrower/Customer

Goal in context: to search a vehicle by its carrying capacity

Precondition:

- System has been designed for search by vehicle carrying capacity.
- System has interface for search by vehicle carrying capacity.

Triggers: Borrower/Customer has to search by vehicle carrying capacity.

Scenario:

- Visit page
- Click search button
- Click search by vehicle carrying capacity
- Select vehicle carrying capacity

Exception:

- capacity not in list
- Vehicles not available

Priority: Essential, must be implemented.

When Available: First increment

4.3.2.3 By Pick up date

Use Case: Search by pick up date

Primary Actor: Borrower/Customer

Goal in context: to search a vehicle by its pick up date

Precondition:

- System has been designed for search by pick up date.
- System has interface for search by pick up date.

Triggers: Borrower/Customer has to search by pick up date.

Scenario:

- Visit page
- Click search button
- Click search by pick up date
- Select pick up date from calendar

Exception:

- Vehicles not available

Priority: Essential, must be implemented.

When Available: First increment

4.3.2.4 By Vehicle type

Use Case: Search by vehicle type

Primary Actor: Borrower/Customer

Goal in context: to search a vehicle by its type

Precondition:

- System has been designed for search by vehicle type.
- System has interface for search by vehicle type.

Triggers: Borrower/Customer has to search vehicle by its type.

Scenario:

- Visit page
- Click search button
- Click search by vehicle type
- Select vehicle type

Exception:

- Vehicles not available
- Type not available

Priority: Essential, must be implemented.

When Available: First increment

4.3.3 Maintenance

4.3.3.1 Add Driver Information

Use Case: Add Driver Information

Primary Actor: Vehicle owner

Goal in context: to add driver information

Precondition:

- System has been designed for adding driver information.
- System has interface for adding driver information.

Triggers: vehicle owner has to add driver information.

Scenario:

- Visit page
- Click add driver information
- Enter required information of driver
- Check validity of information

Exception:

- Ambiguous Input
- Verification failed

Priority: Essential, must be implemented.

When Available: First increment

4.3.3.2 Add Vehicle Information

Use Case: Add Vehicle Information

Primary Actor: Vehicle owner

Goal in context: to add Vehicle information

Precondition:

- System has been designed for adding vehicle information.
- System has interface for adding vehicle information.

Triggers: vehicle owner has to add vehicle information.

Scenario:

- Visit page
- Click add vehicle information
- Enter required information of vehicle
- Check validity of information

Exception:

- Ambiguous Input
- Verification failed

Priority: Essential, must be implemented.

When Available: First increment

4.3.3.3 Remove Vehicle

Use Case: Remove Vehicle

Primary Actor: Vehicle owner

Goal in context: to remove Vehicle

Precondition:

- System has been designed for removing vehicle.
- System has interface for removing vehicle.

Triggers: vehicle owner has to remove vehicle.

Scenario:

- Visit page
- Select vehicle to remove
- Confirm remove

Exception:

- Vehicle not selected
- remove failed

Priority: Essential, must be implemented.

When Available: First increment

4.3.3.4 Remove Driver

Use Case: Remove driver

Primary Actor: Vehicle owner

Goal in context: to remove Vehicle

Precondition:

- System has been designed for removing driver.
- System has interface for removing driver.

Triggers: vehicle owner has to remove driver.

Scenario:

- Visit page
- Select driver to remove
- Confirm remove

Exception:

- Driver not selected
- remove failed

Priority: Essential, must be implemented.

When Available: First increment

4.3.3.5 Tracking

Use Case: Tracking

Primary Actor: Vehicle owner, Borrower/customer

Goal in context: to track Vehicle

Precondition:

- System has been designed for tracking vehicle.
- System has interface for tracking vehicle.

Triggers: vehicle owner and borrower/customer has to track vehicle.

Scenario:

- Visit page
- Select a vehicle to track
- Turn on GPS
- See the location of vehicle

Exception:

- Vehicle not selected
- Internet access failed

Priority: Essential, must be implemented.

When Available: First increment

4.3.4 Booking

4.3.4.1 Schedule Trip

Use Case: Schedule trip

Primary Actor: Borrower/customer

Goal in context: to schedule trip

Precondition:

- System has been designed for scheduling trip.
- System has interface for scheduling trip.

Triggers: borrower/customer has to schedule a trip.

Scenario:

- Visit page
- Select vehicles
- Select pick-up location and destination
- Select a date

Exception:

- Vehicle not available
- location not found
- Internet access failed

Priority: Essential, must be implemented.

When Available: First increment

4.3.4.2 Trip Confirmation

Use Case: trip confirmation

Primary Actor: Borrower/customer

Goal in context: to confirm a trip

Precondition:

- System has been designed for confirming trip.
- System has interface for confirming trip.

Triggers: borrower/customer has to confirm a trip.

Scenario:

- Visit page
- Select vehicles
- Select pick-up location and destination
- Select a date
- confirm a trip

Exception:

- Vehicle not available
- location not found
- Internet access failed
- confirmation denied

Priority: Essential, must be implemented.

When Available: First increment

4.3.4.3 Trip Cancellation

Use Case: trip cancellation

Primary Actor: Borrower/customer

Goal in context: to cancel a trip

Precondition:

- System has been designed for cancelling trip.
- System has interface for cancelling trip.

Triggers: borrower/customer has to cancel a trip.

Scenario:

- Visit page
- Select trip that have been confirmed
- cancel a trip

Exception:

- Internet access failed
- cancellation denied

Priority: Essential, must be implemented.

When Available: First increment

4.3.5 Payment

4.3.5.1 Advance Online Payment

Use Case: Advance Online payment

Primary Actor: Borrower/customer

Goal in context: to pay through advance online payment

Precondition:

- System has been designed for advance online payment.
- System has interface for advance online payment.
- System has list of online payment method.

Triggers: borrower/customer pay cash online.

Scenario:

- Visit page
- Need to select payment method
- Enter Required information for payment
- Confirm payment

Exception:

- system not working properly

Priority: Essential, must be implemented.

When Available: First increment

4.3.5.2 Cash on Delivery

Use Case: cash on delivery

Primary Actor: Borrower/customer

Goal in context: to pay cash on delivery

Precondition:

- System has been designed for cash on delivery.
- System has interface for cash on delivery.

Triggers: borrower/customer pay cash on delivery.

Scenario:

- Visit page
- Need to select payment method
- Pay cash

Priority: Essential, must be implemented.

When Available: First increment

4.3.6 Notification

4.3.6.1 Confirmation Notification

Use Case: confirmation notification

Primary Actor: vehicle owner and Borrower/customer

Goal in context: to notify users after confirmation

Precondition:

- System has been designed for confirm notification.
- System has interface for confirm notification.

Triggers: System has to notify users after confirmation.

Scenario:

- Is a system
- System should notify after confirmation

Exception:

- system not working properly

Priority: Essential, must be implemented.

When Available: First increment

4.3.6.2 Delivery Notification

Use Case: delivery notification

Primary Actor: vehicle owner and Borrower/customer

Goal in context: to notify users after delivery

Precondition:

- System has been designed for delivery notification.
- System has interface for delivery notification.

Triggers: System has to notify users after delivery.

Scenario:

- Is a system
- System should notify after delivery

Exception:

- system not working properly

Priority: Essential, must be implemented.

When Available: First increment

4.3.6.3 Delay Notification

Use Case: delay notification

Primary Actor: vehicle owner and Borrower/customer

Goal in context: to notify users when there is delay

Precondition:

- System has been designed for delay notification.
- System has interface for delay notification.

Triggers: System has to notify users when there is delay.

Scenario:

- Is a system
- System should notify when there is delay

Exception:

- system not working properly

Priority: Essential, must be implemented.

When Available: First increment

4.3.6.4 Cancellation Notification

Use Case: cancellation notification

Primary Actor: vehicle owner and Borrower/customer

Goal in context: to notify users after cancellation

Precondition:

- System has been designed for cancel notification.
- System has interface for cancel notification.

Triggers: System has to notify users after cancellation.

Scenario:

- Is a system
- System should notify after cancellation

Exception:

- system not working properly

Priority: Essential, must be implemented.

When Available: First increment

4.3.7 Communication

4.3.7.1 Standard Call System

Use Case: standard call system

Primary Actor: vehicle owner and Borrower/customer

Goal in context: to manage vehicle owner and Borrower/customer
Communication through call system

Precondition:

- System has been designed for communication through standard call system.
- System has interface for communication through standard call system.

Triggers: System has to manage vehicle owner and Borrower/customer communication through call system.

Scenario:

- Is a system
- System should have required information i.e mobile number of users.
- System should direct owner call to borrower/customer or vice versa.

Exception:

- system not working properly
- Invalid input

Priority: Essential, must be implemented.

When Available: First increment

4.3.7.1 Standard Message System

Use Case: standard message system

Primary Actor: vehicle owner and Borrower/customer

Goal in context: to manage vehicle owner and Borrower/customer
Communication through message system

Precondition:

- System has been designed for communication through standard message system.
- System has interface for communication through standard message system.

Triggers: System has to manage vehicle owner and Borrower/customer communication through message system.

Scenario:

- Is a system
- System should have required information i.e mobile number of users or user name
- System should direct owner message to borrower/customer or vice versa.

Exception:

- system not working properly
- Invalid input

Priority: Essential, must be implemented.

When Available: First increment

4.4 Use Case Diagram

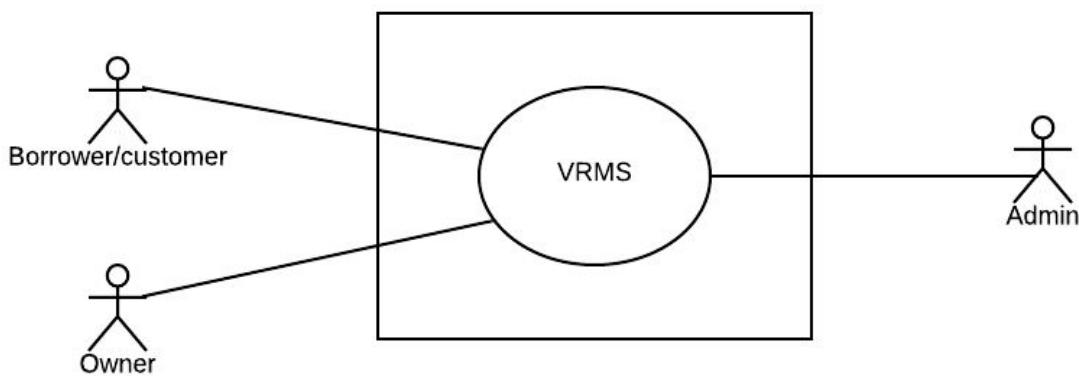


Figure 1: Level 0 use case diagram Vehicle Renting and Management System

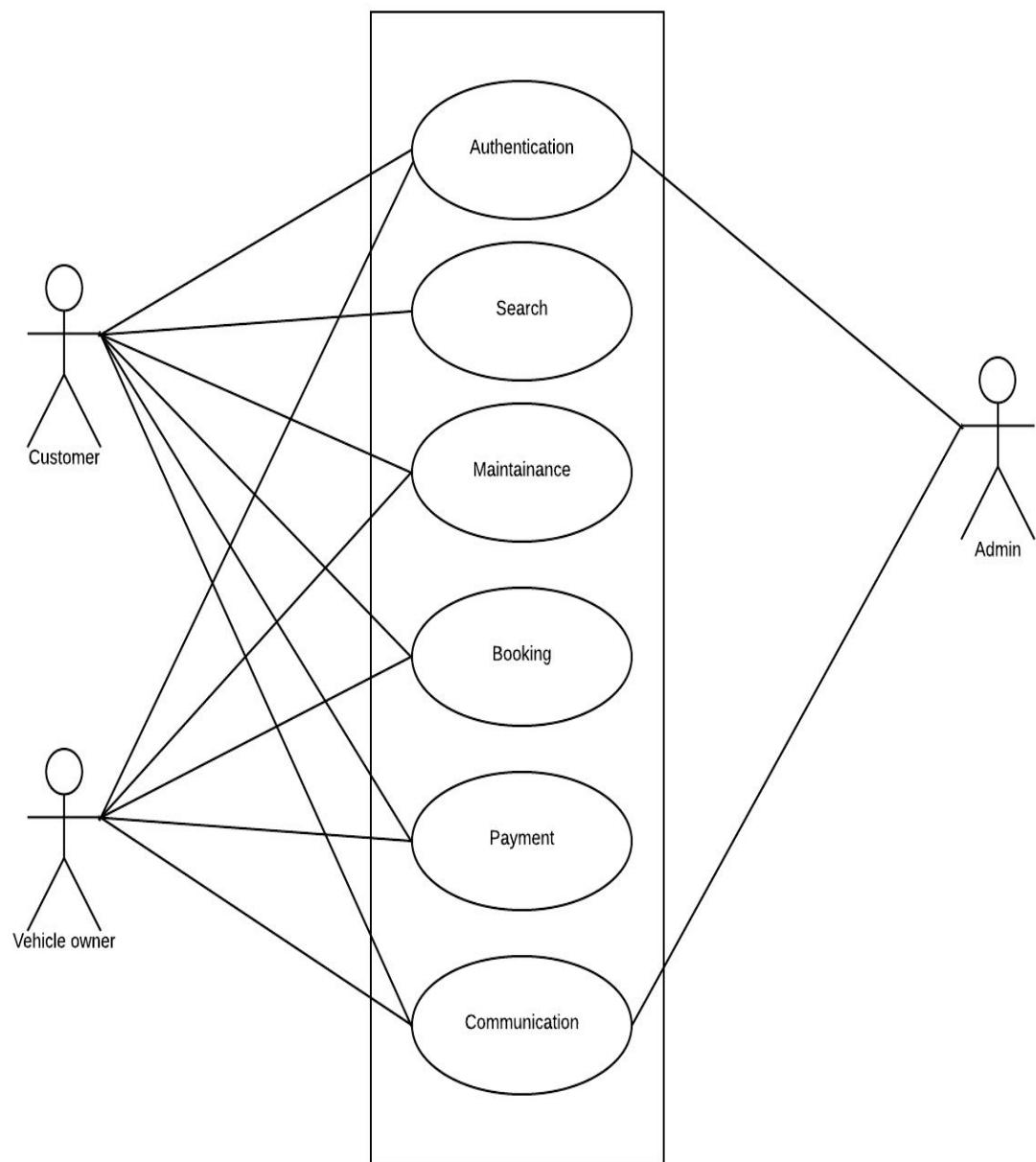


Figure 2: Level 1 use case diagram Vehicle Renting and Management System

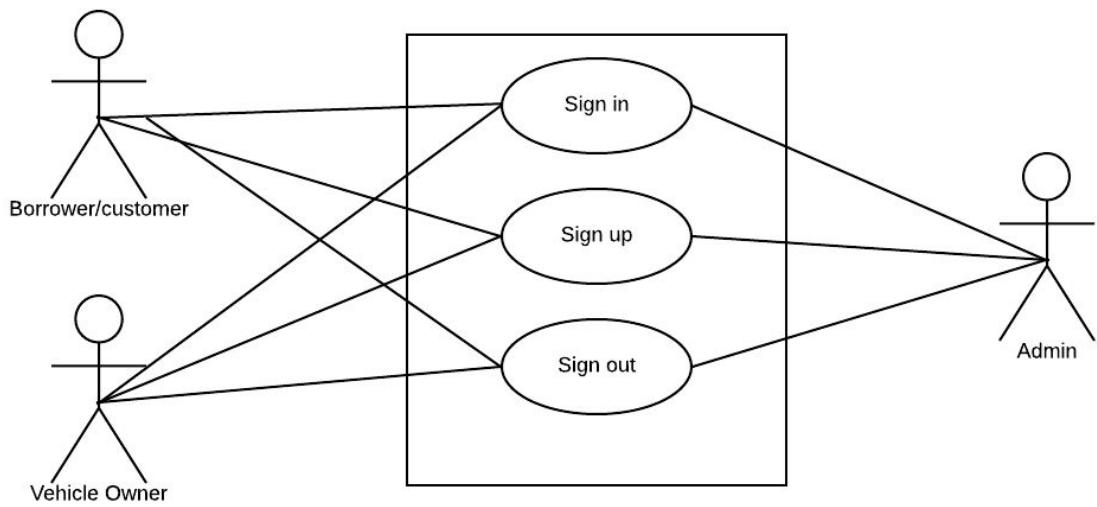


Figure 3: Level 2.1 use case diagram Vehicle Renting and Management System

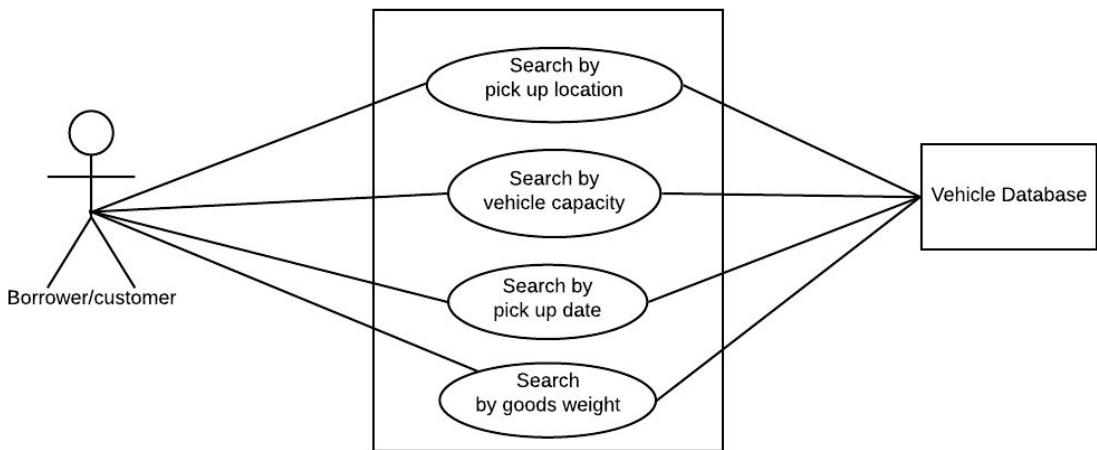


Figure 4: Level 2.2 use case diagram Vehicle Renting and Management System

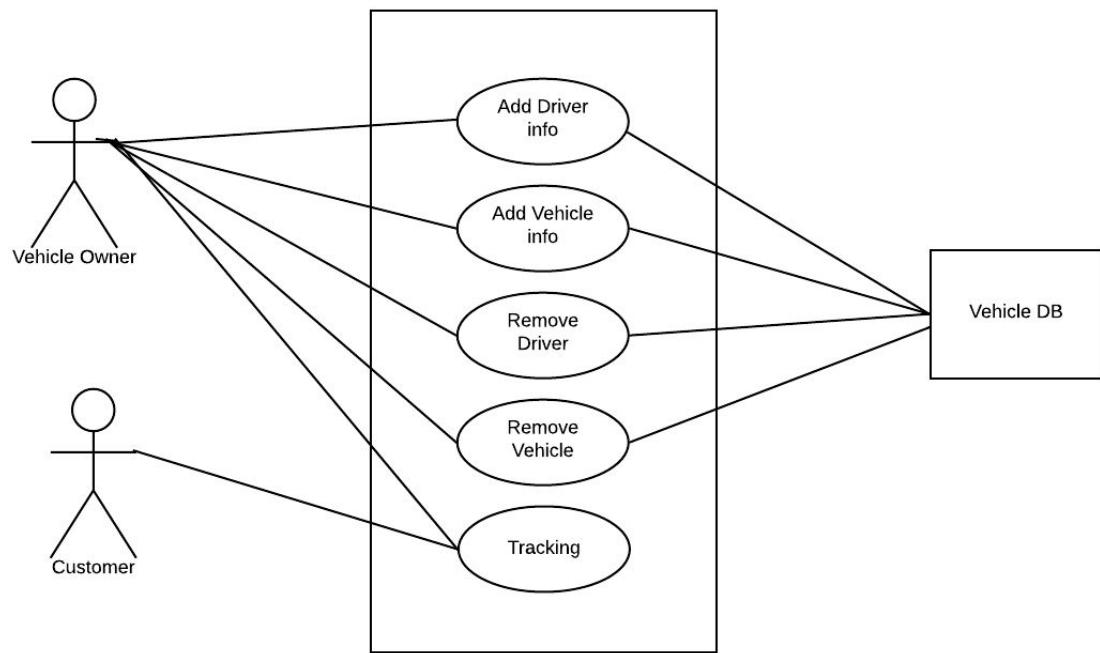


Figure 5: Level 2.3 use case diagram Vehicle Renting and Management System

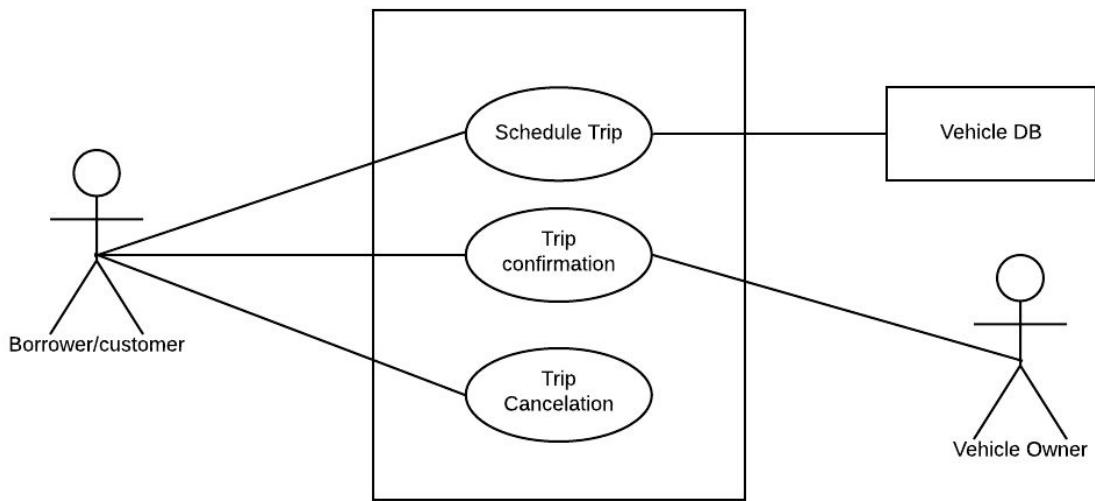


Figure 6: Level 2.4 use case diagram Vehicle Renting and Management System

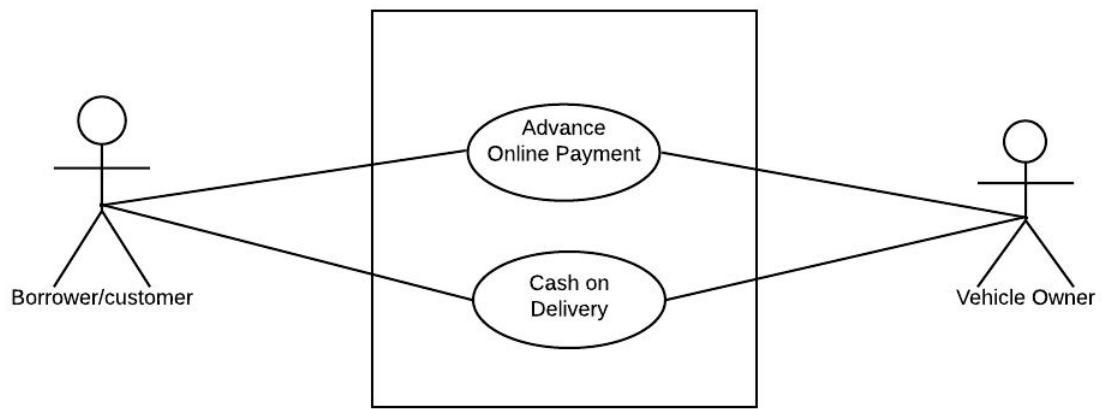


Figure 7: Level 2.5 use case diagram Vehicle Renting and Management System

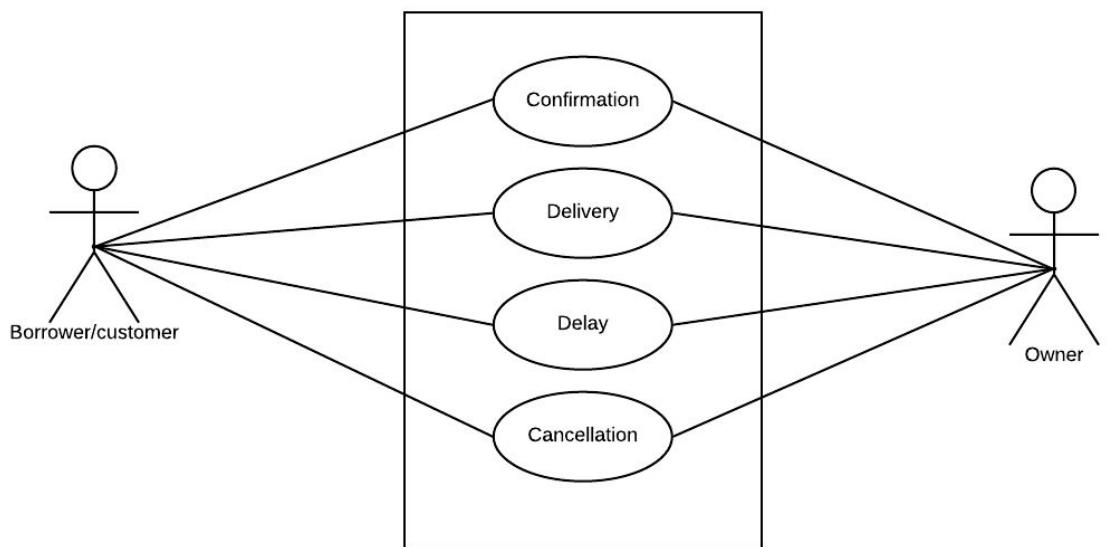


Figure 8: Level 2.6 use case diagram Vehicle Renting and Management System

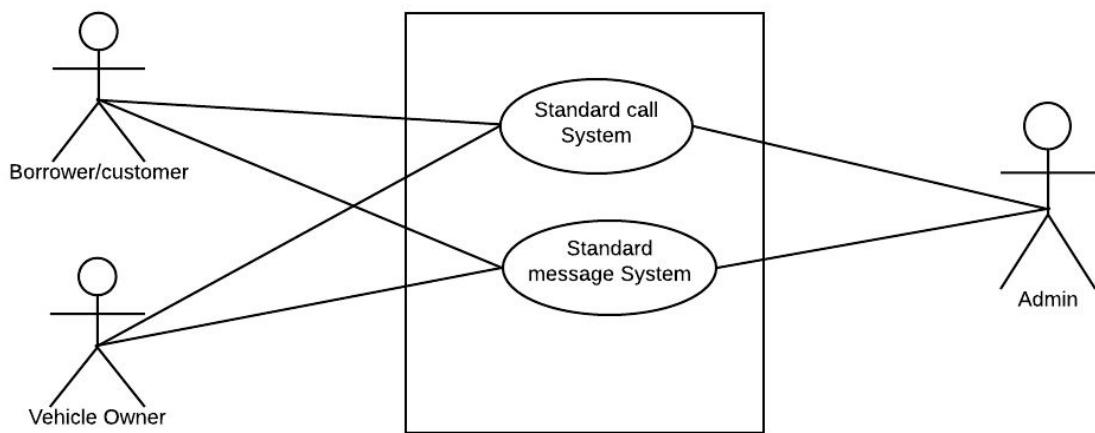


Figure 9: Level 2.7 use case diagram Vehicle Renting and Management System

4.5 Activity Diagram and swimlane Diagram of Use Cases

Use case 1: Sign Up

Activity Diagram

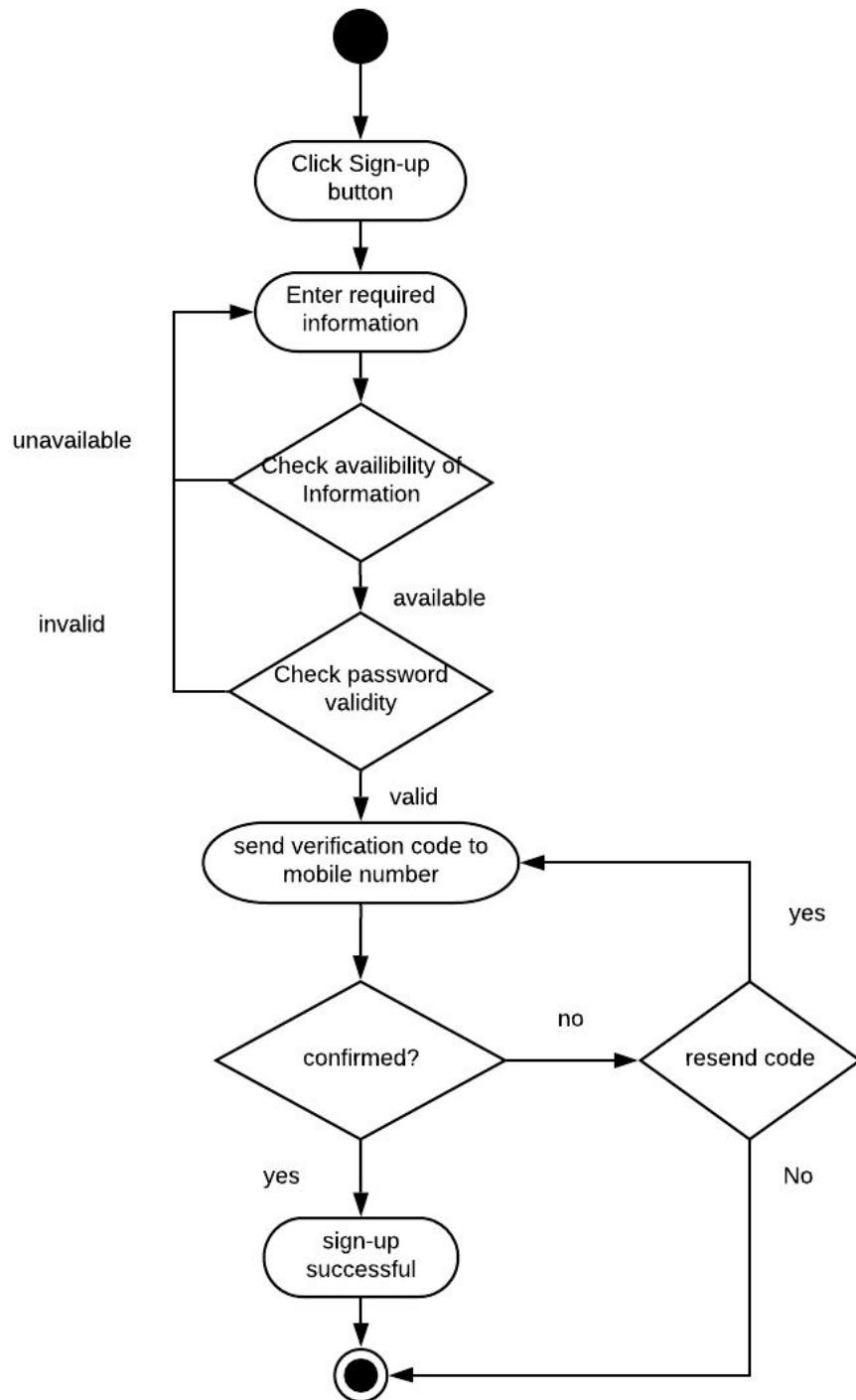


Figure 10: Activity diagram for sign up

Swimlane Diagram

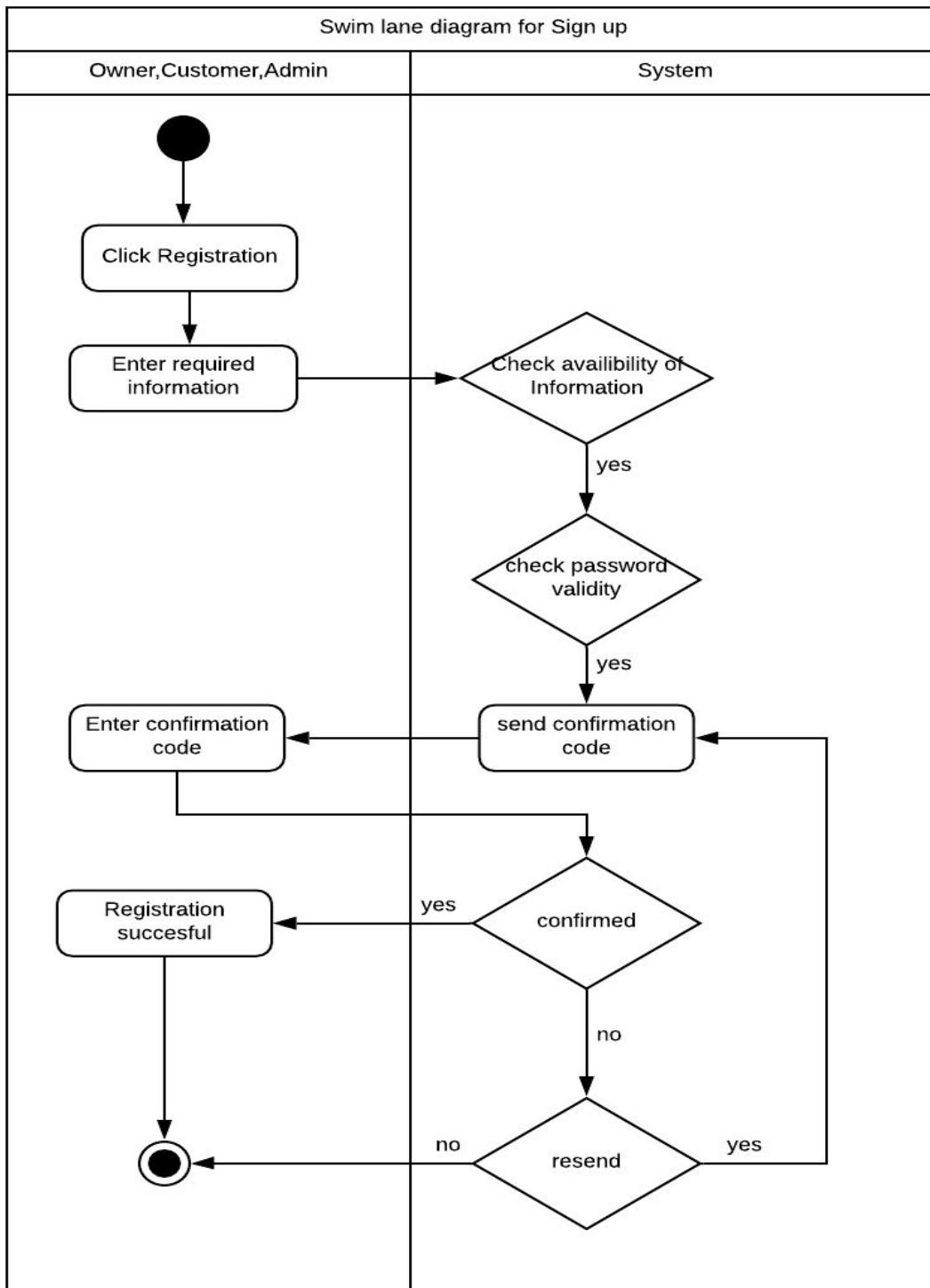


Figure 11: Swimlane diagram for sign up

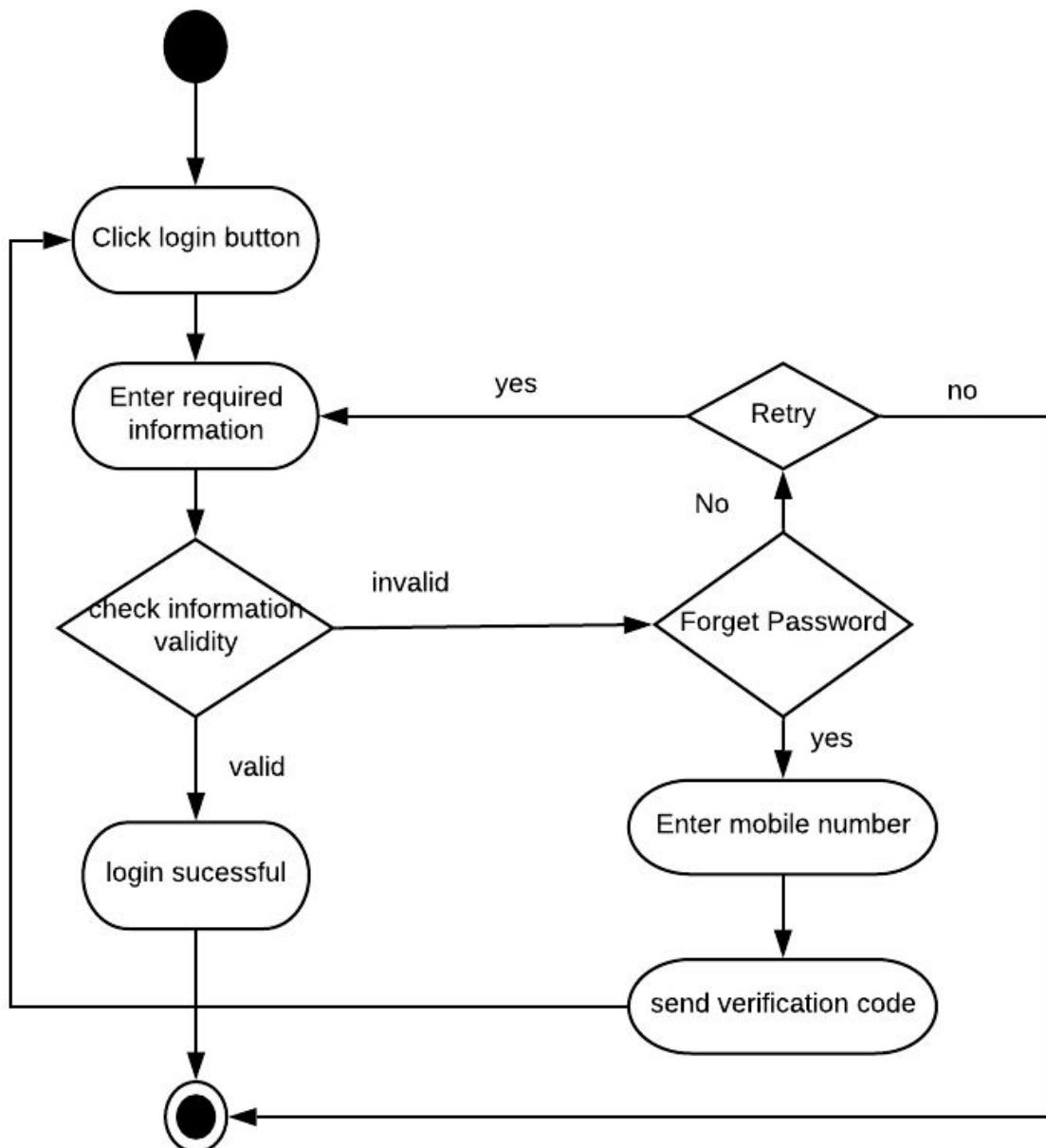
Use case 2: Sign In**Activity Diagram**

Figure 12: Activity diagram for sign in

Swimlane Diagram

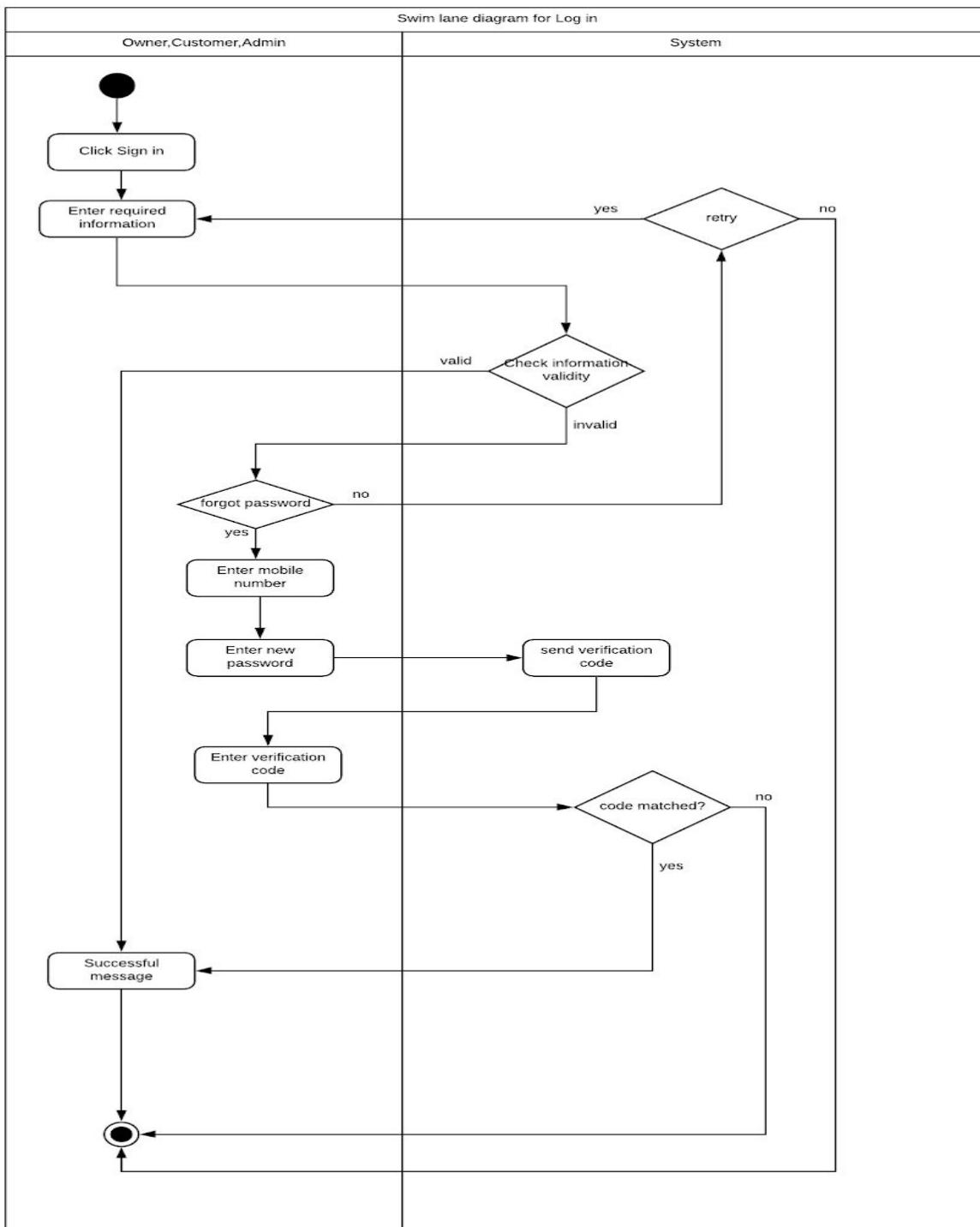


Figure 13: Swimlane diagram for sign in

Use case 3: Sign Out

Activity Diagram

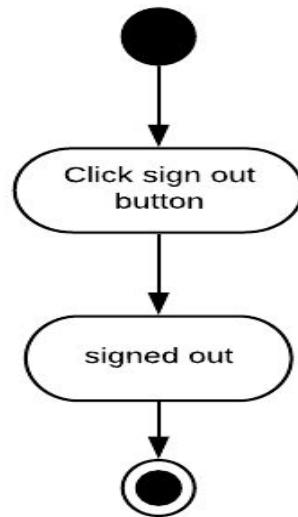


Figure 14: Activity diagram for sign out

Swimlane Diagram

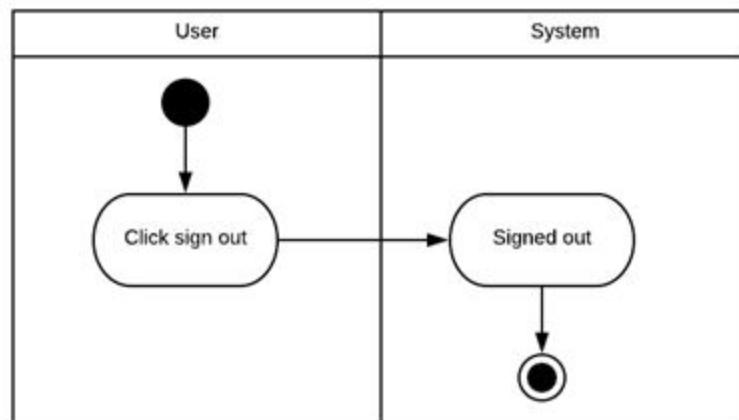


Figure 15: Swimlane diagram for sign out

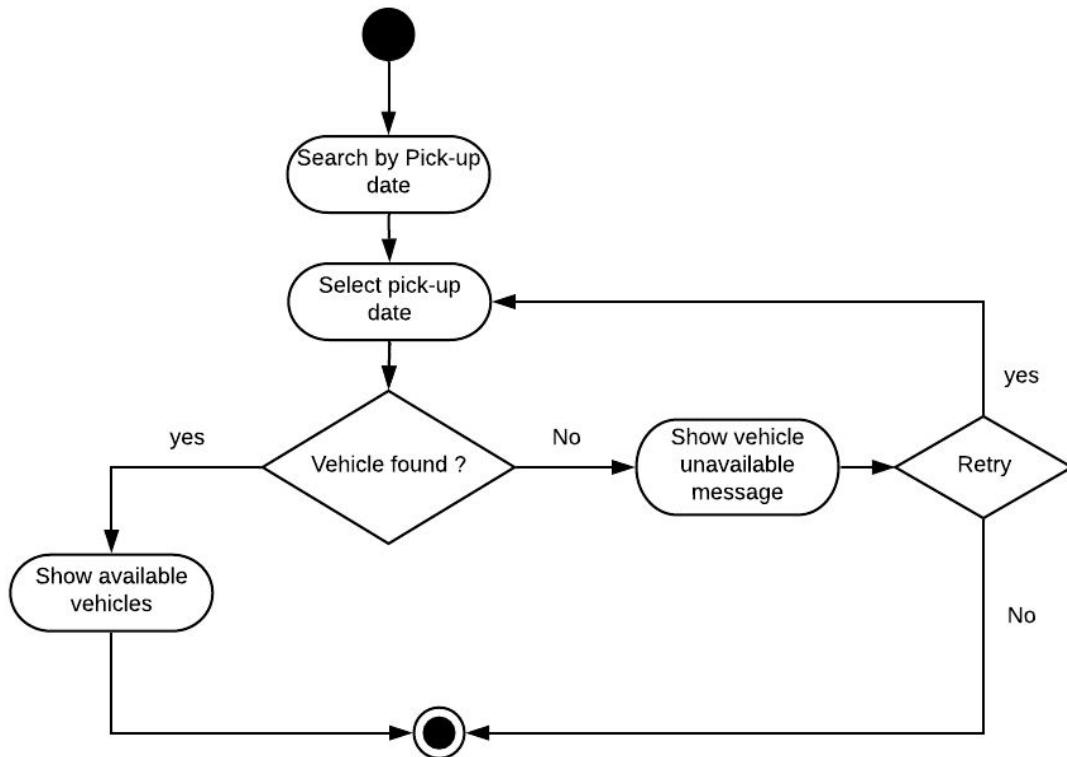
Use case 4: Search by Pickup date**Activity Diagram**

Figure 16: Activity diagram for search by pickup date

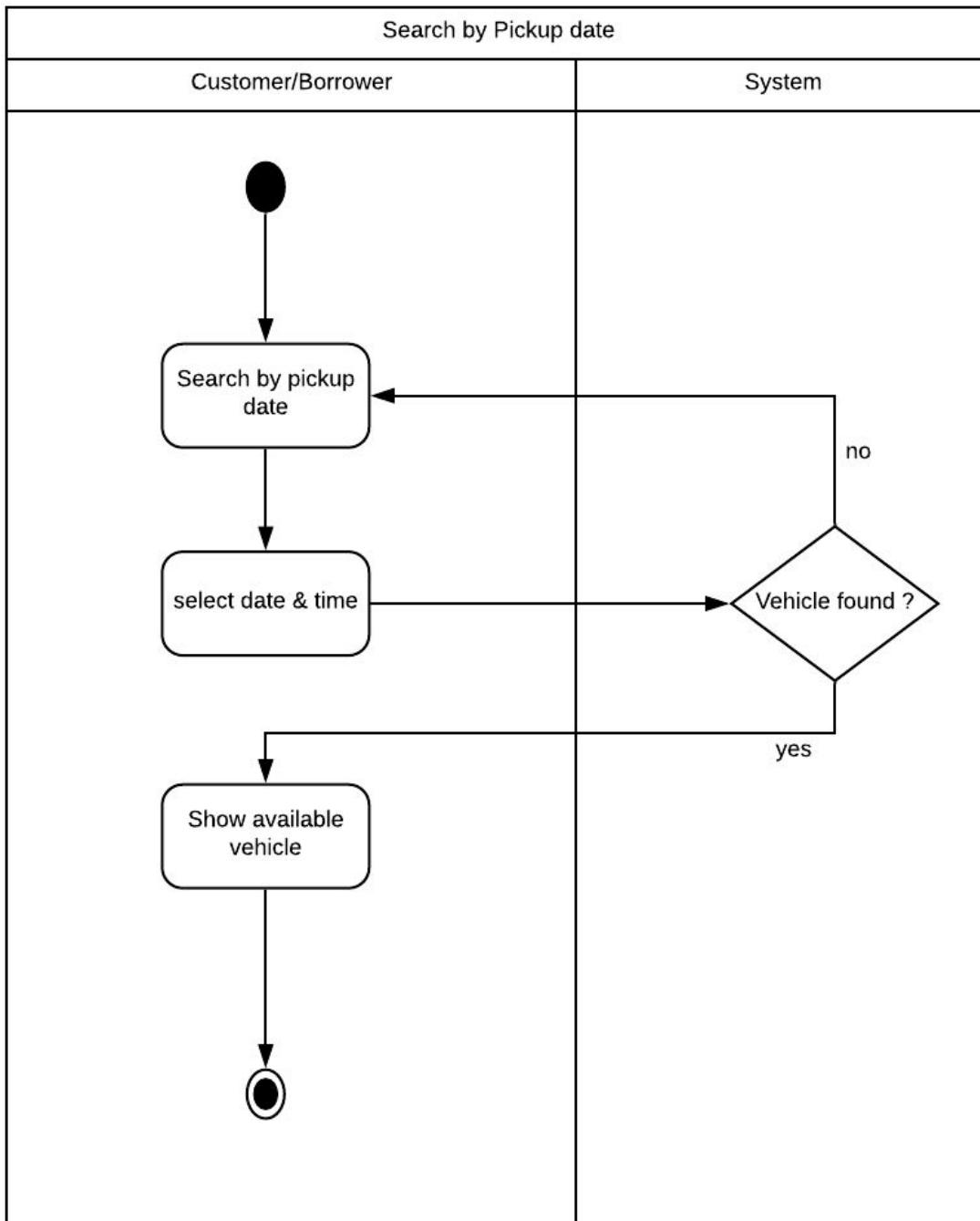
Swimlane Diagram

Figure 17: Swimlane diagram for search by pickup date

Use case 5: Search by Vehicle Capacity

Activity Diagram

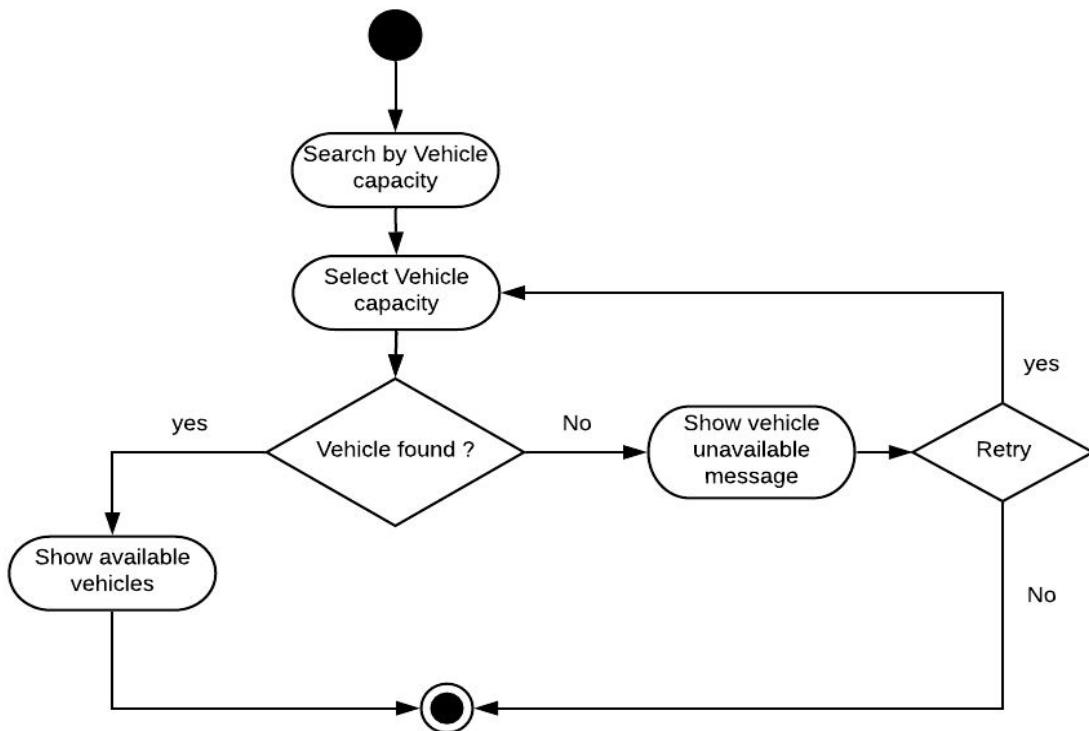


Figure 18: Activity diagram for search by Vehicle Capacity

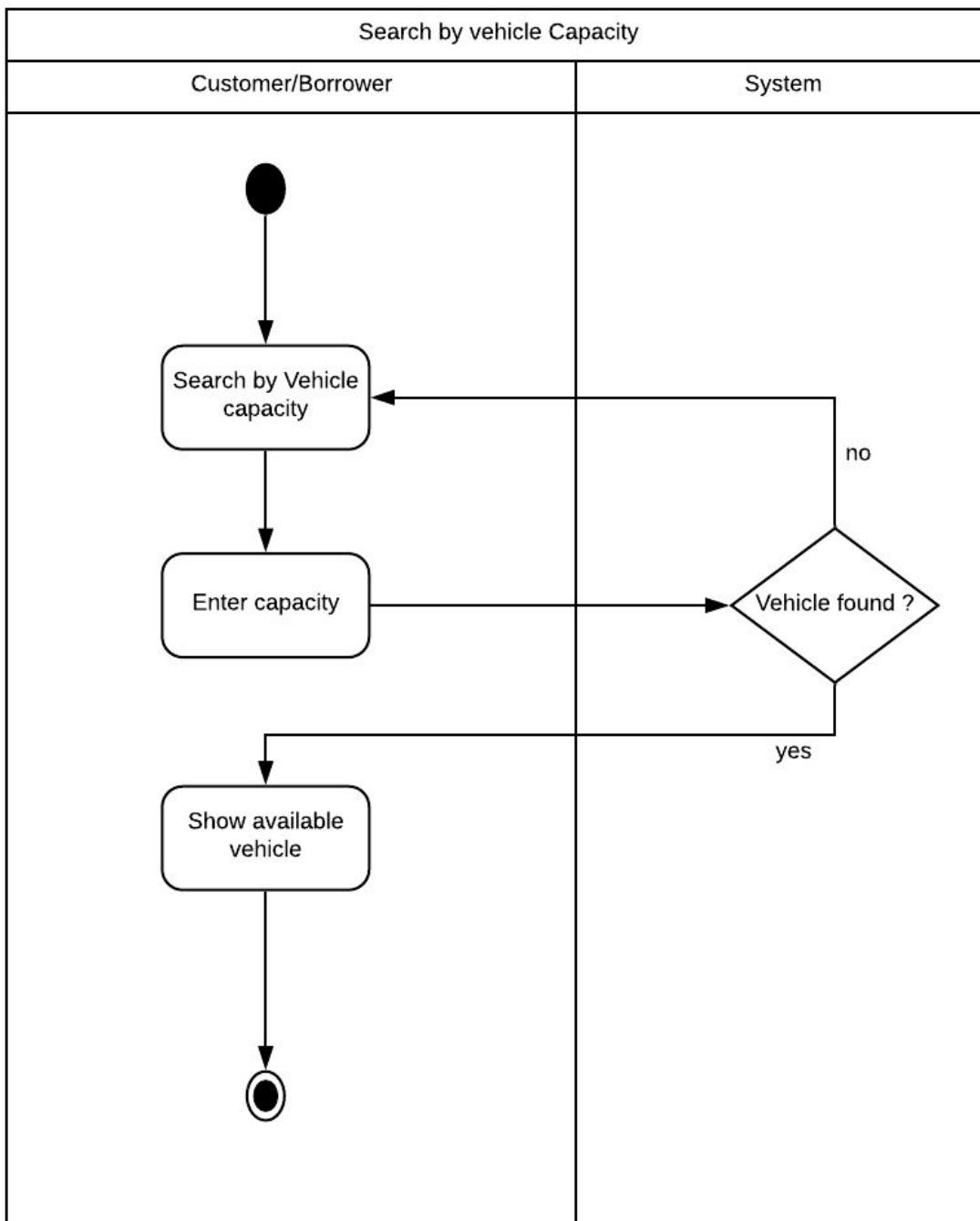
Swimlane Diagram

Figure 19: Swimlane diagram for search by vehicle Capacity

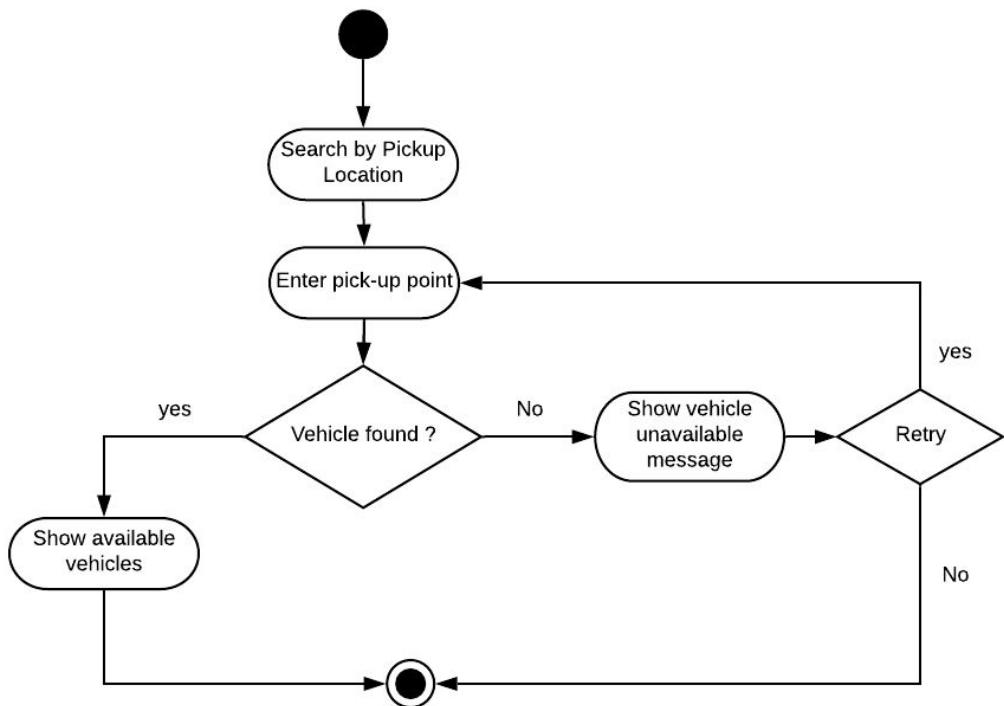
Use case 6: Search by Pickup Location**Activity Diagram**

Figure 20: Activity diagram for search by pickup location

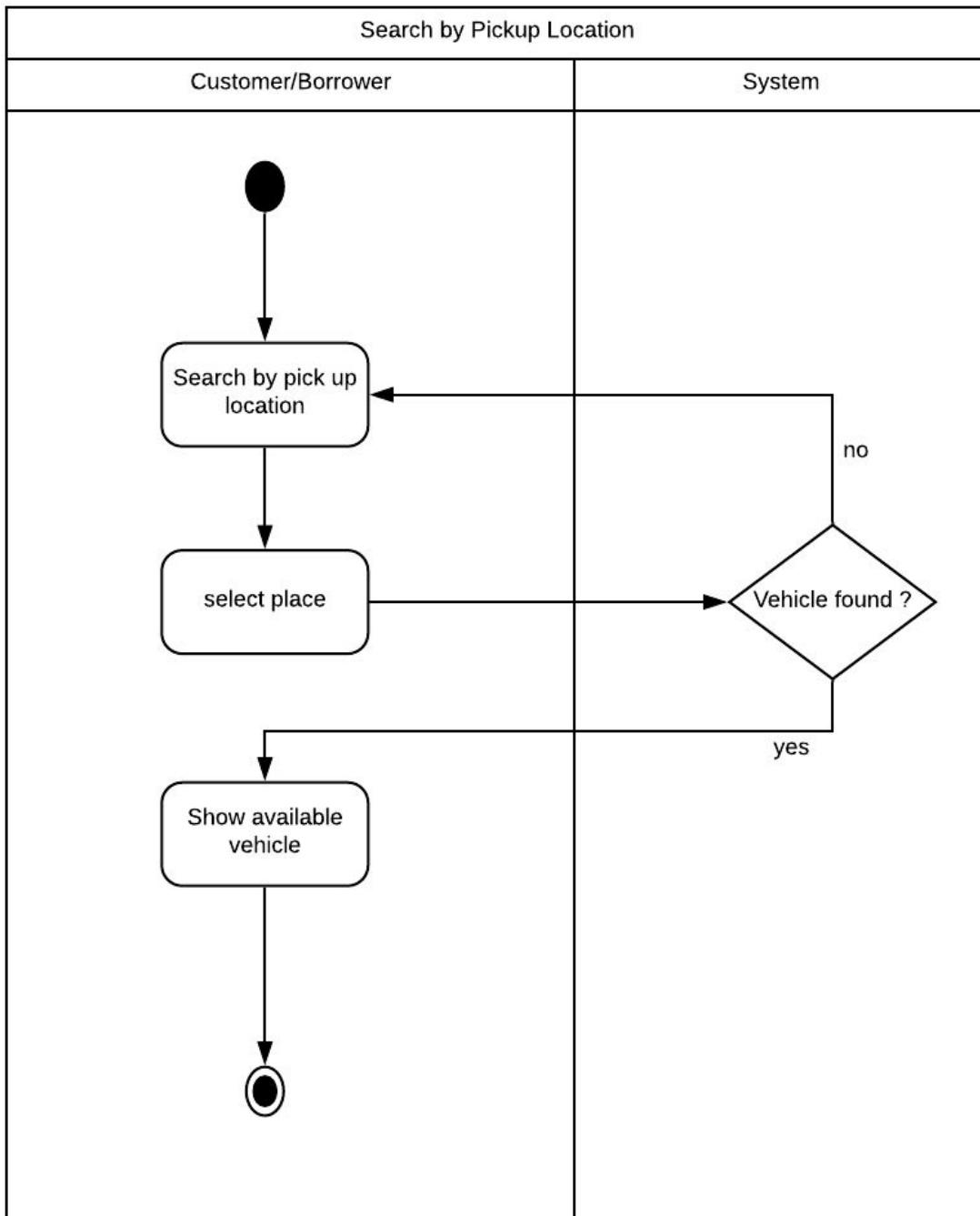
Swimlane Diagram

Figure 21: Swimlane diagram for search by Pickup location

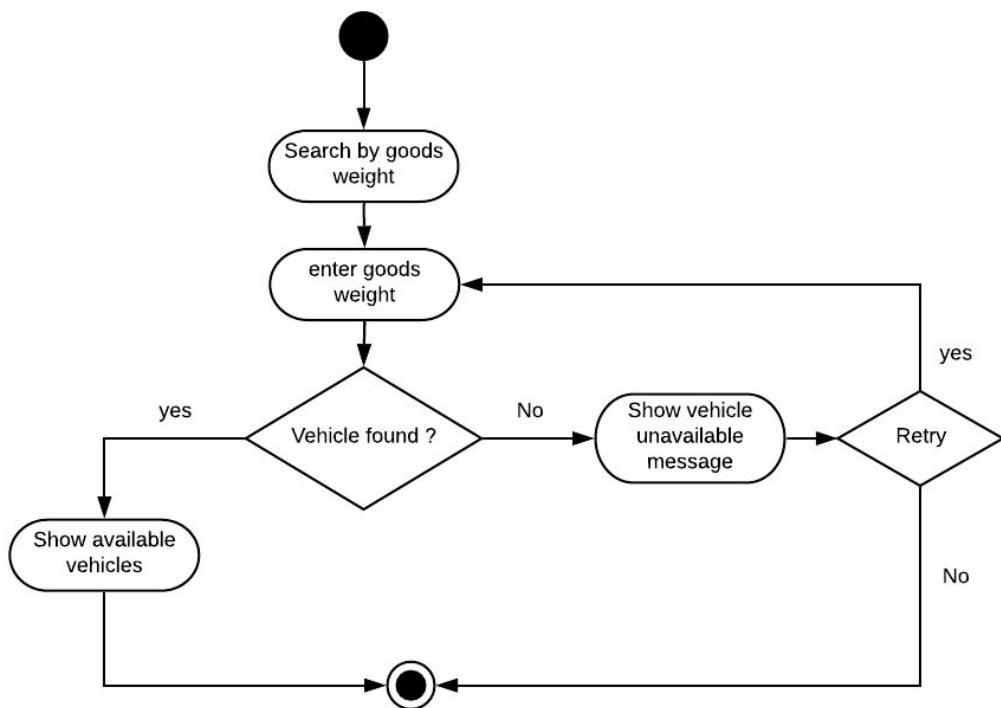
Use case 7: Search by goods weight**Activity Diagram**

Figure 22: Activity diagram for search by goods weight

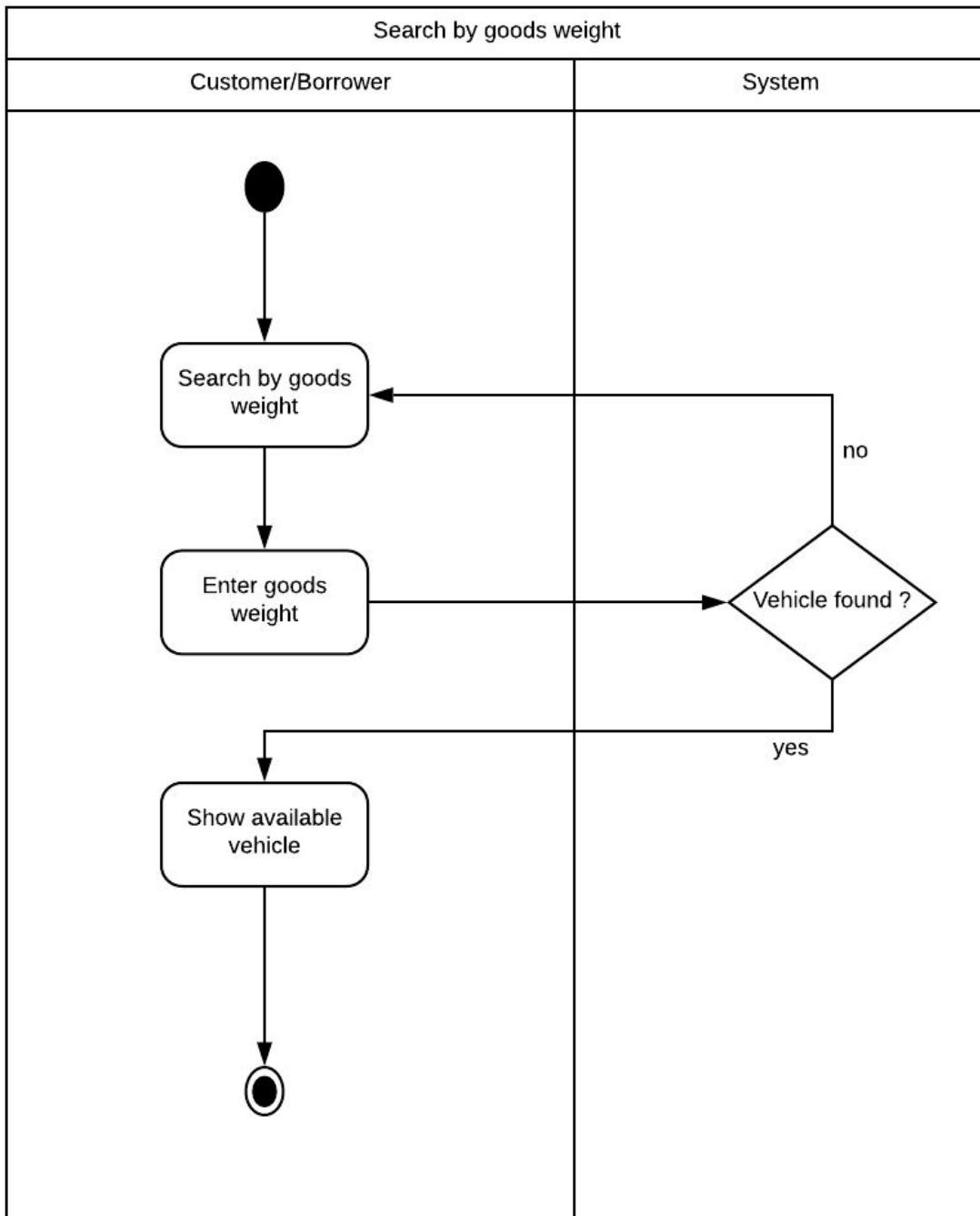
Swimlane Diagram

Figure 23: Swimlane diagram for search by goods weight

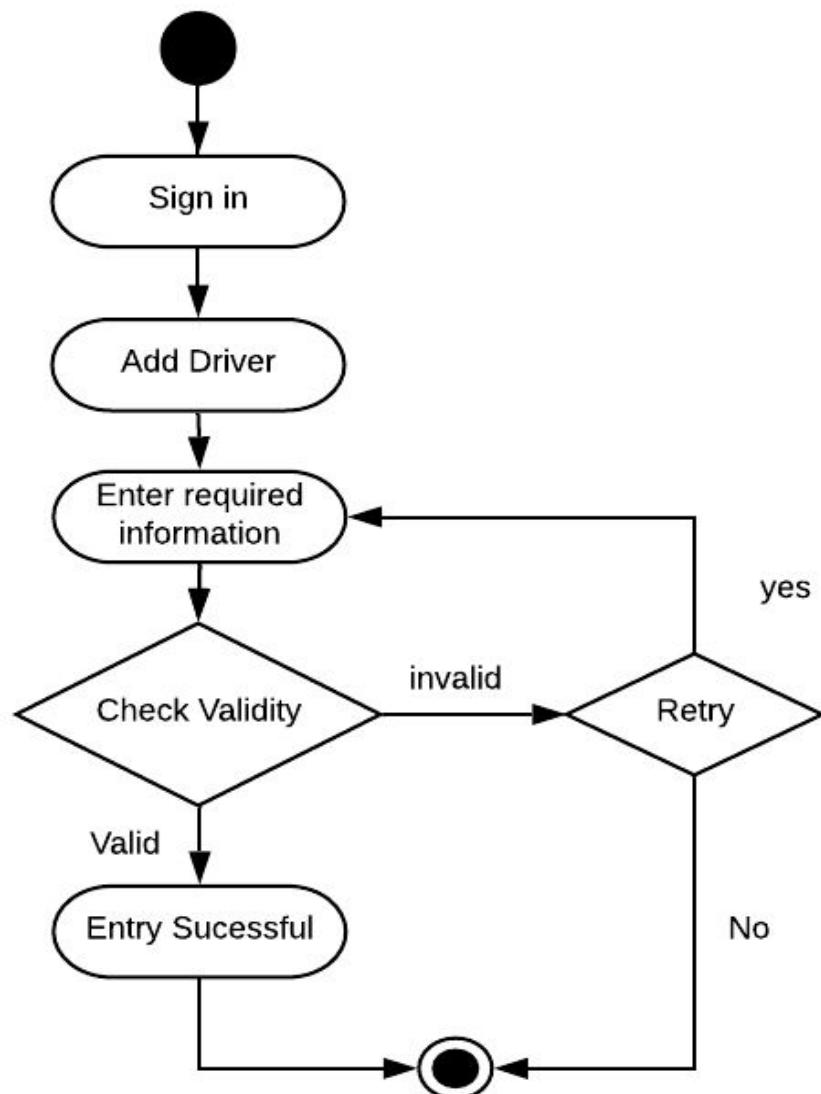
Use case 8: Add Driver Information**Activity Diagram**

Figure 24: Activity diagram for add driver information

Swimlane Diagram

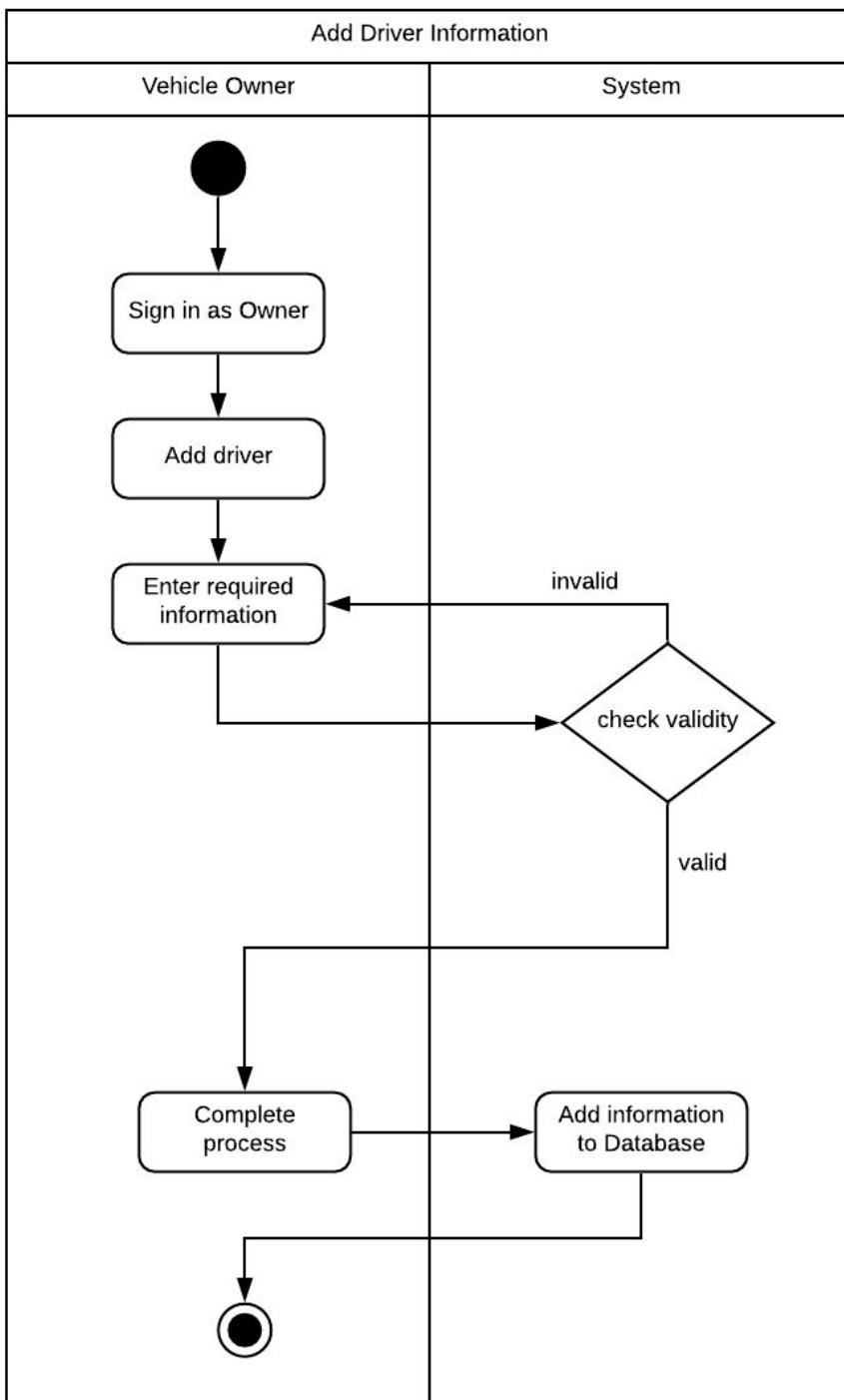


Figure 25: Swimlane diagram for add driver information

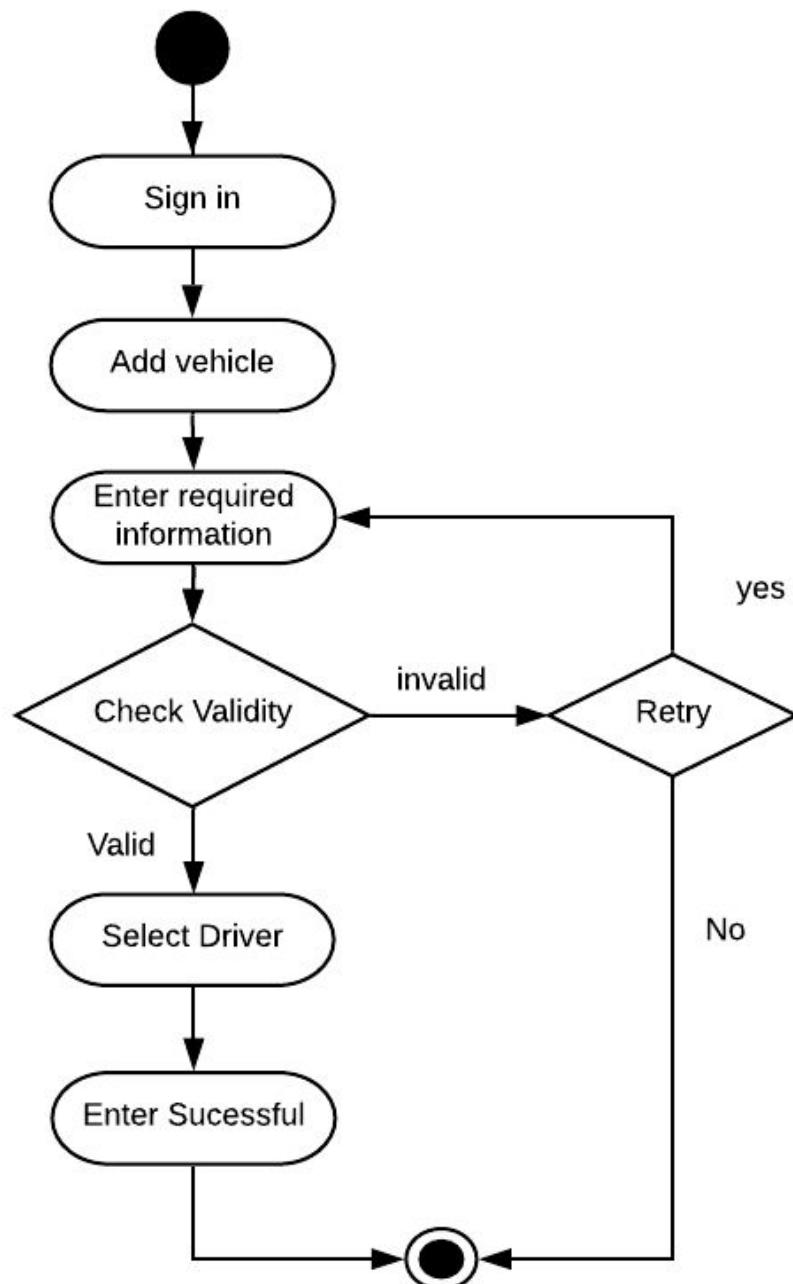
Use case 9: Add Vehicle Information**Activity Diagram**

Figure 26: Activity diagram for add vehicle information

Swimlane Diagram

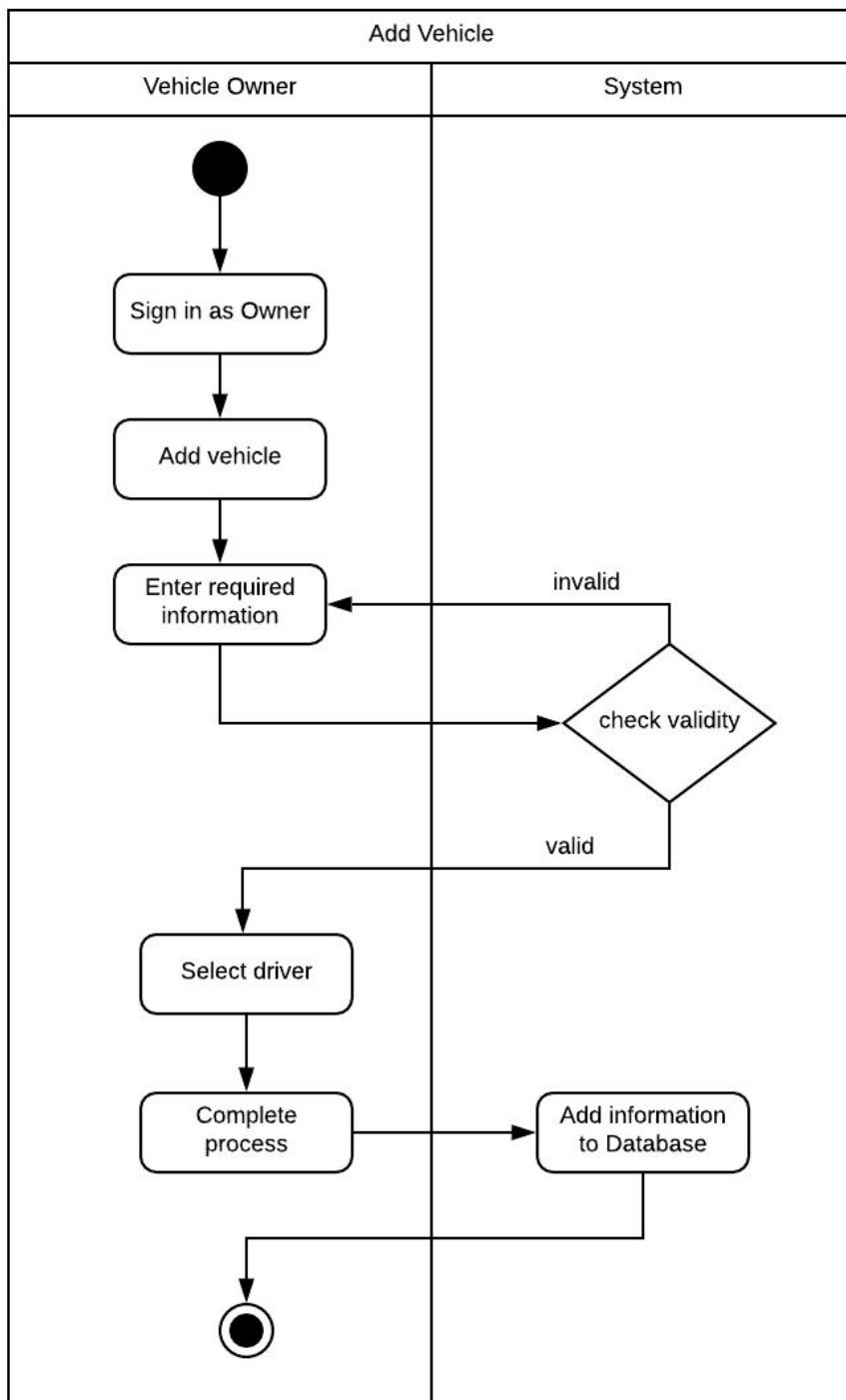


Figure 27: Swimlane diagram for add vehicle information

Use case 10: Remove Driver

Activity Diagram

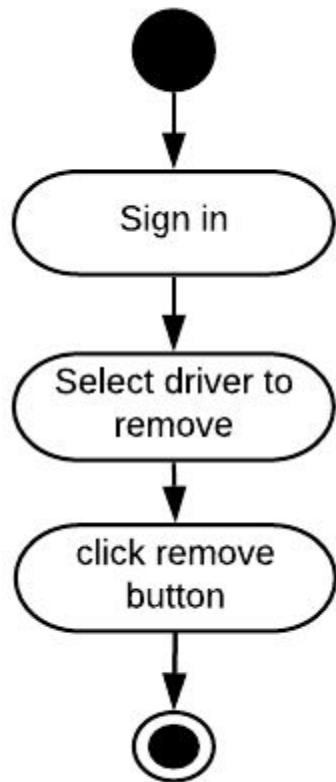


Figure 28: Activity diagram for remove driver

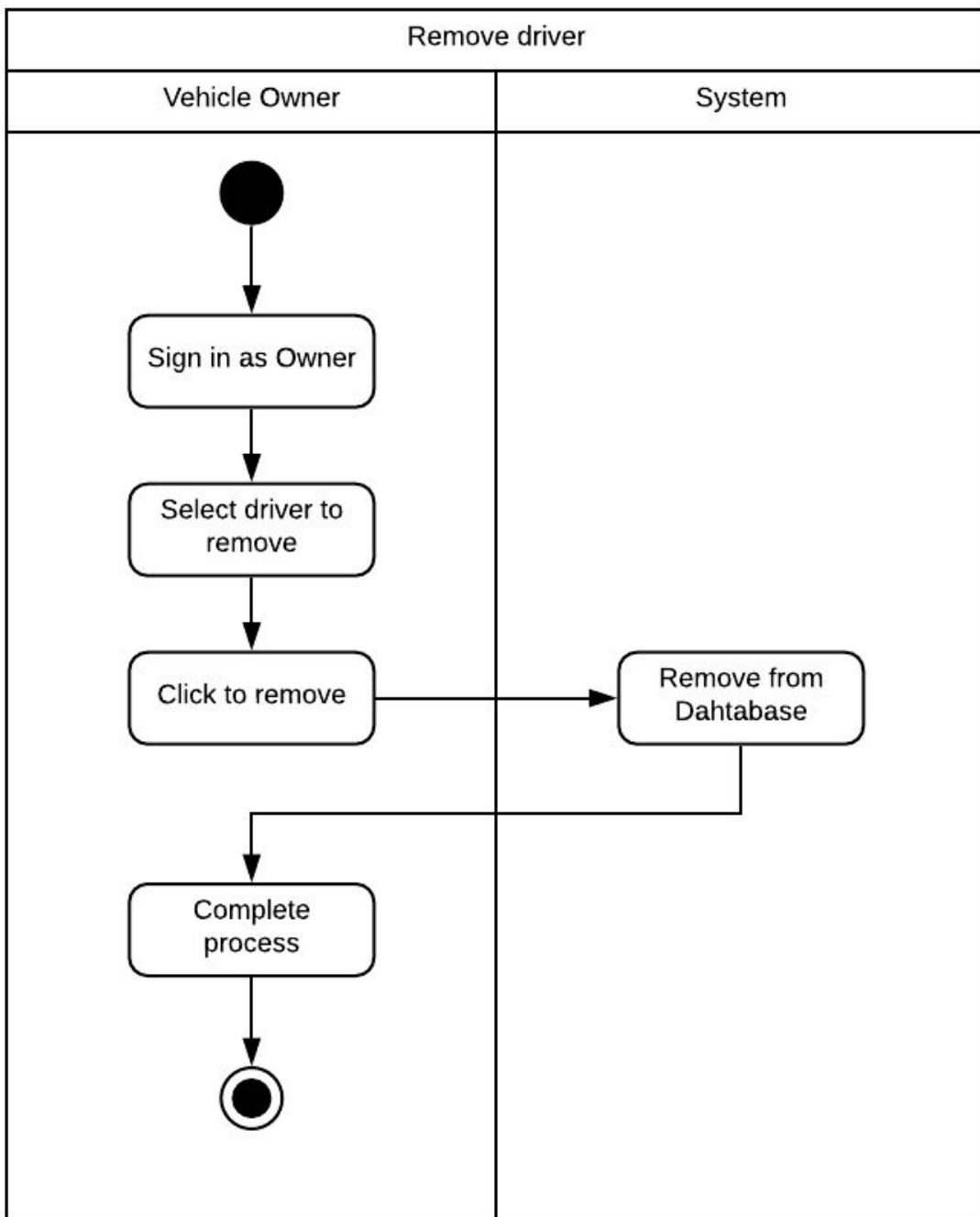
Swimlane Diagram

Figure 29: Swimlane diagram for remove driver

Use case 11: Remove Vehicle

Activity Diagram

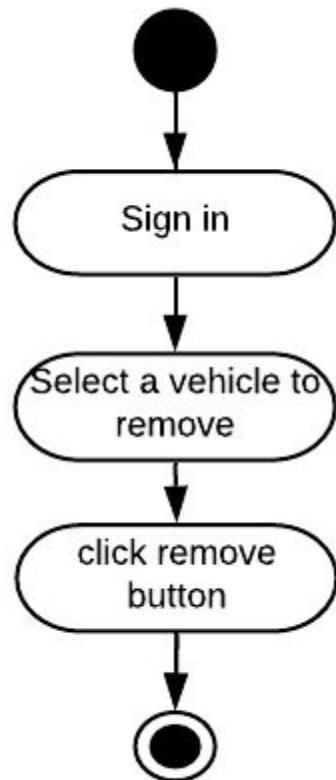


Figure 30: Activity diagram for remove vehicle

Swimlane Diagram

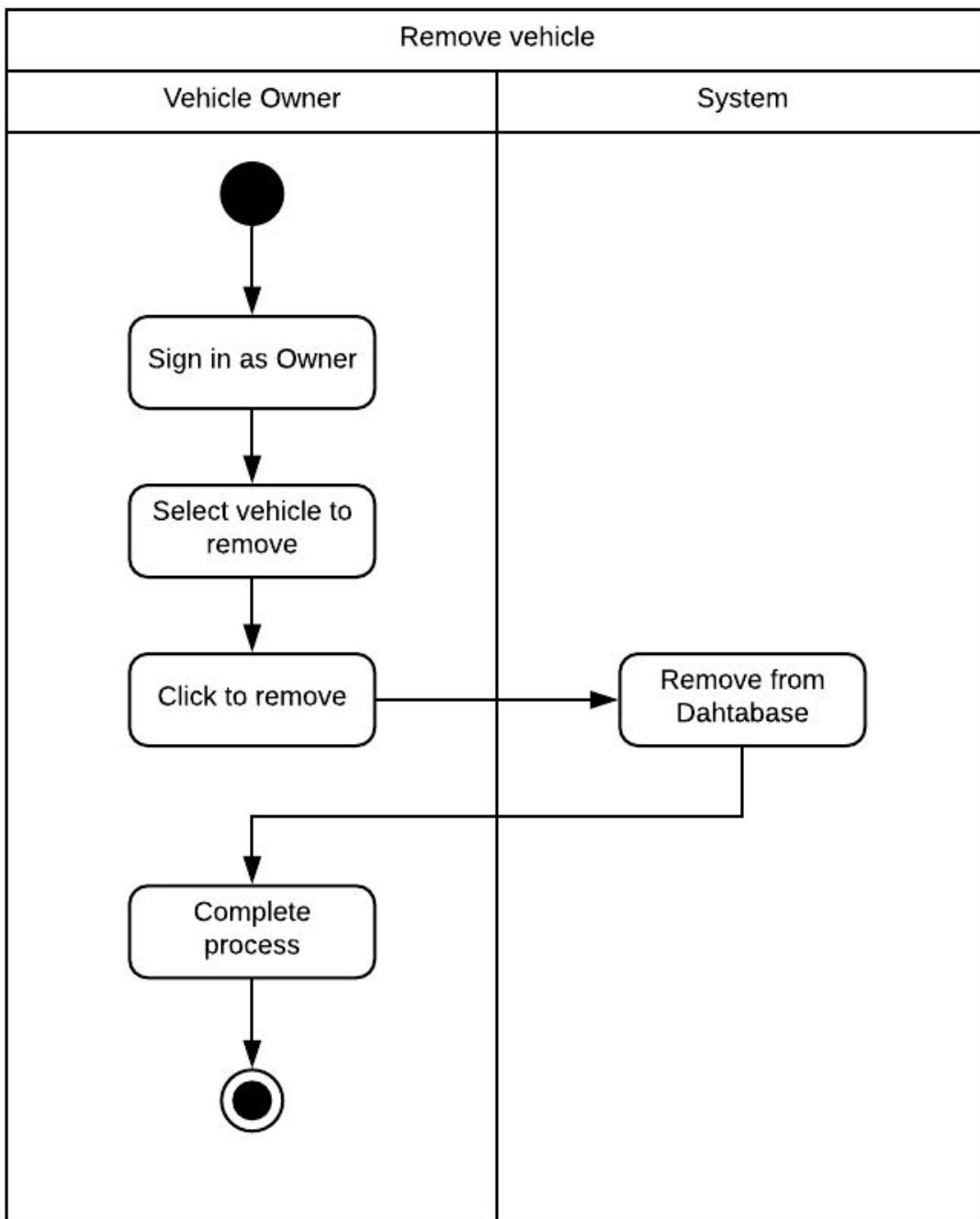


Figure 31: Swimlane diagram for remove vehicle

Use case 12: Tracking

Activity Diagram

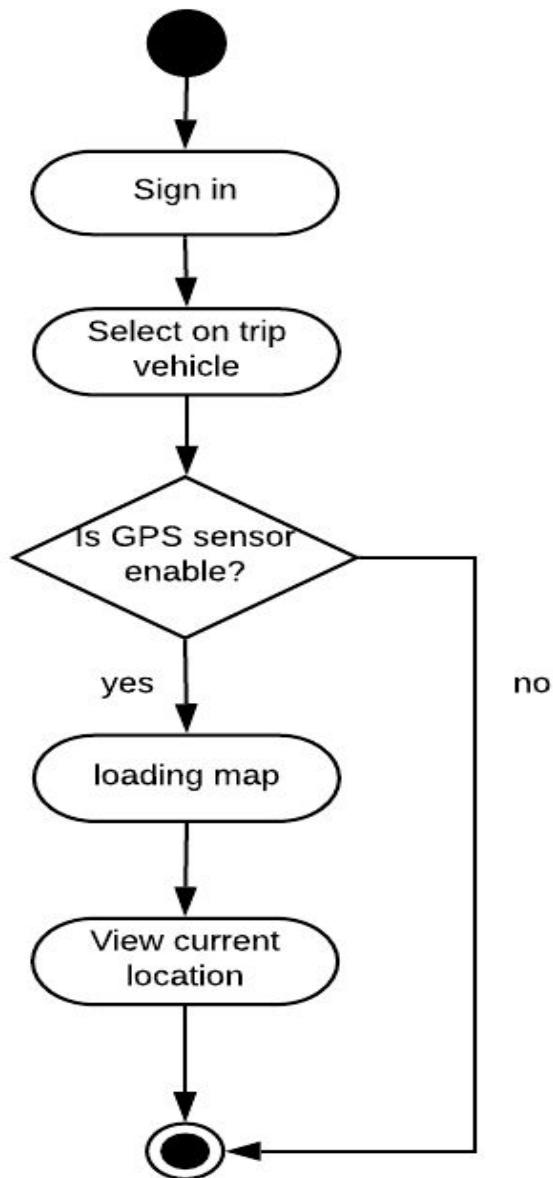


Figure 32: Activity diagram for tracking

Swimlane Diagram

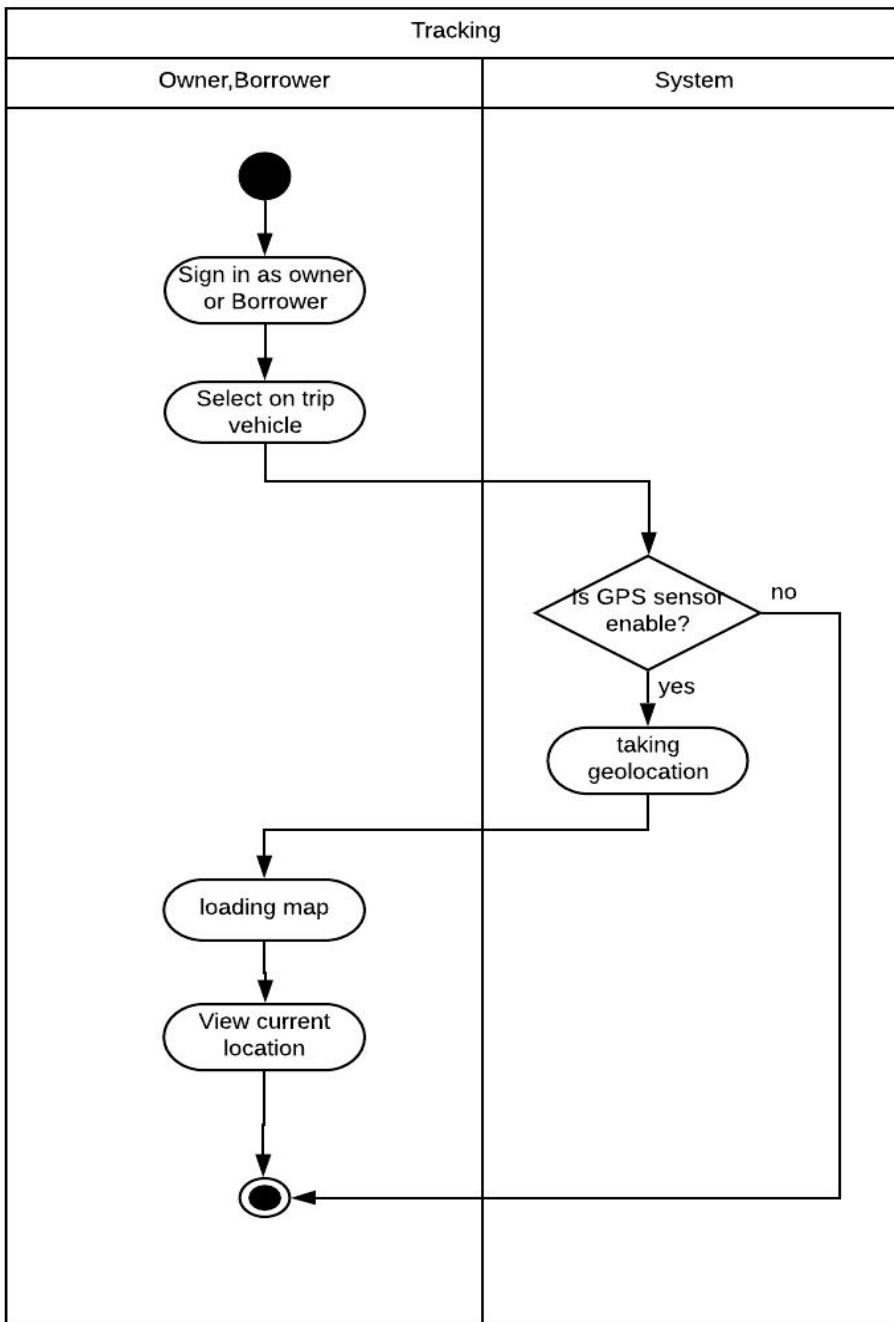


Figure 33: Swimlane diagram for tracking

Use case 13: ReSchedule

Activity Diagram

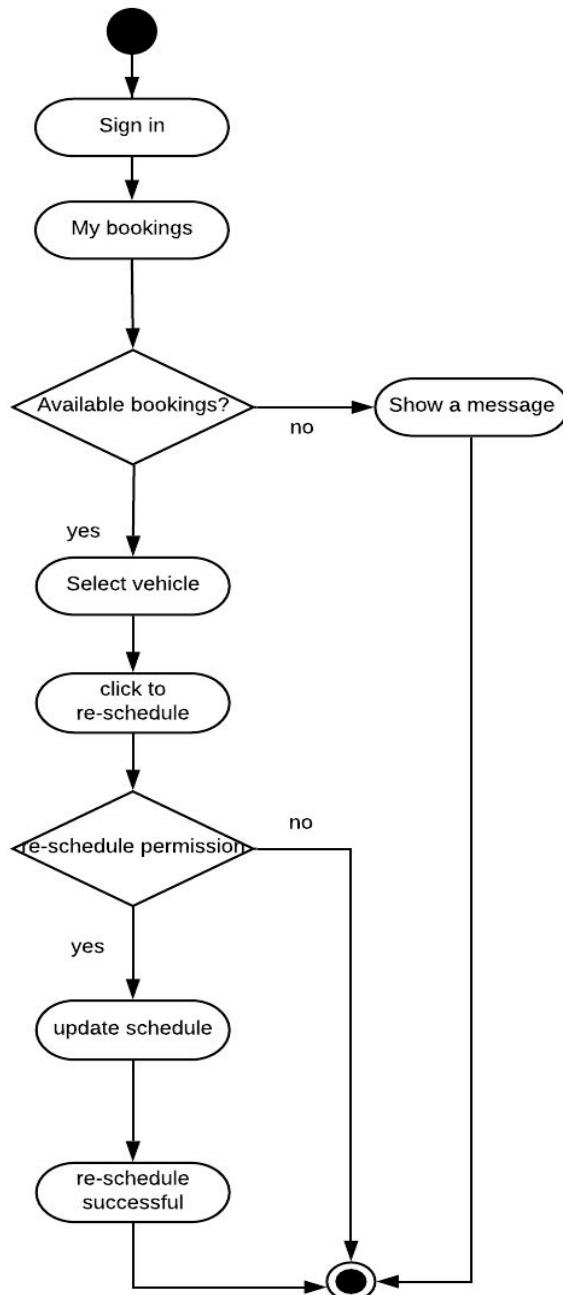


Figure 34: Activity diagram for Reschedule

Swimlane Diagram

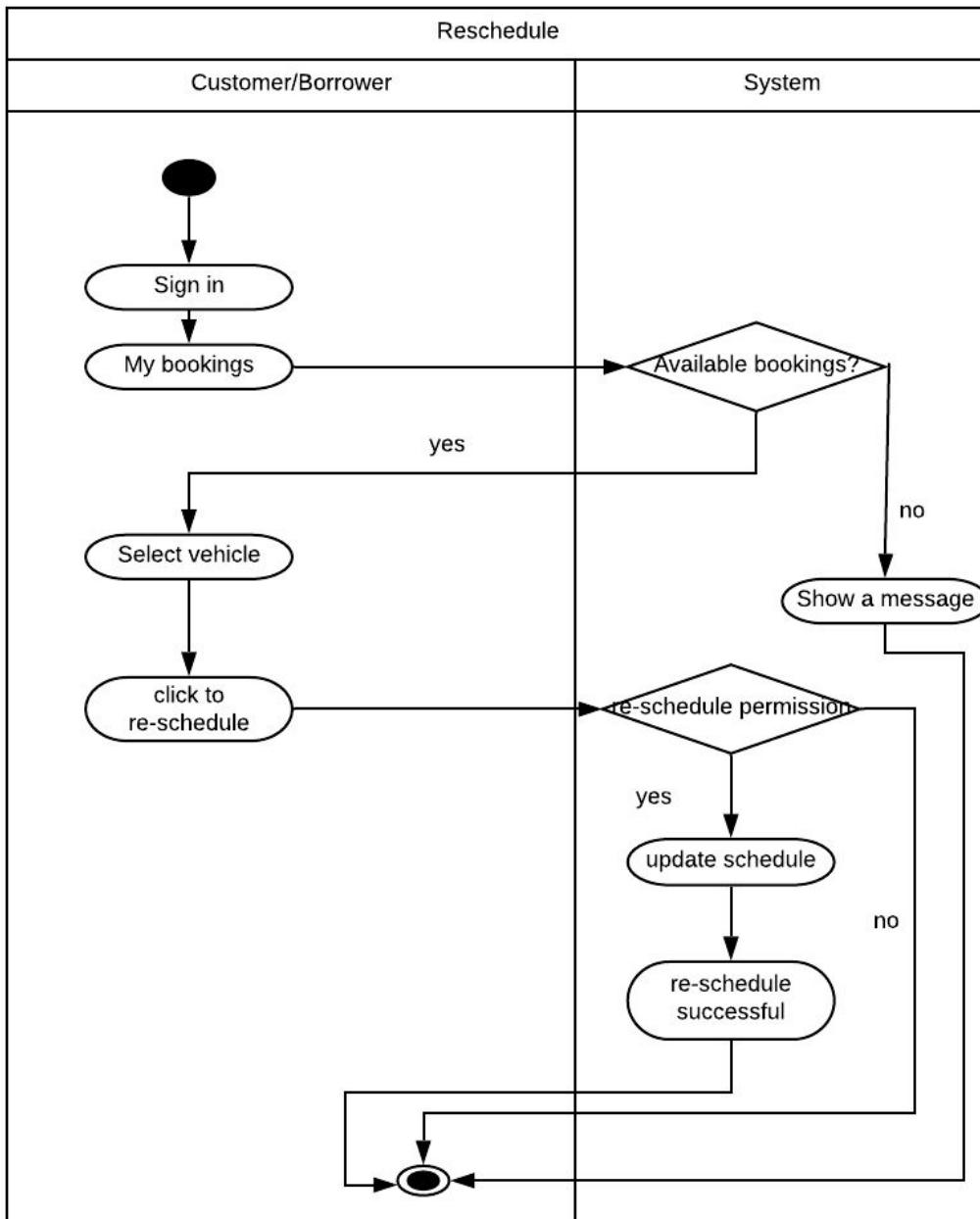


Figure 35: Swimlane diagram for reschedule

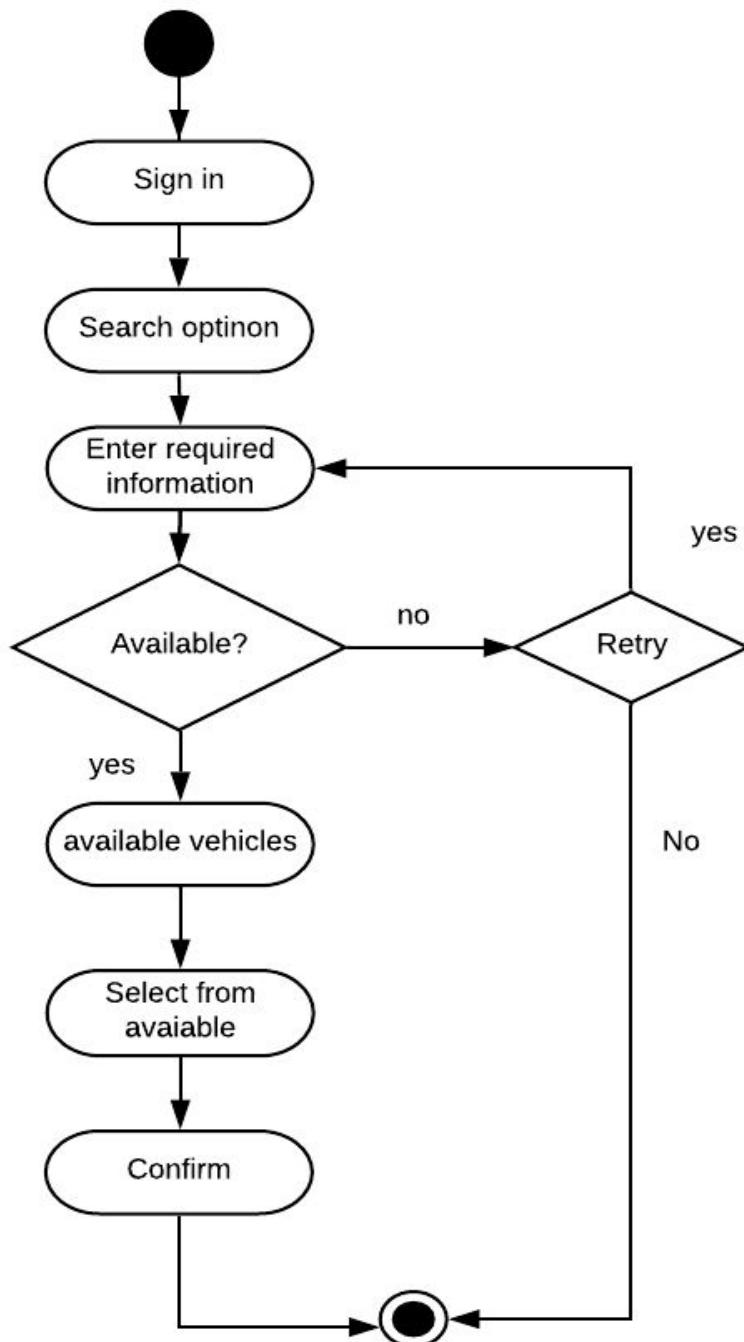
Use case 14: Trip Confirmation
Activity Diagram

Figure 36: Activity diagram for trip confirmation

Swimlane Diagram

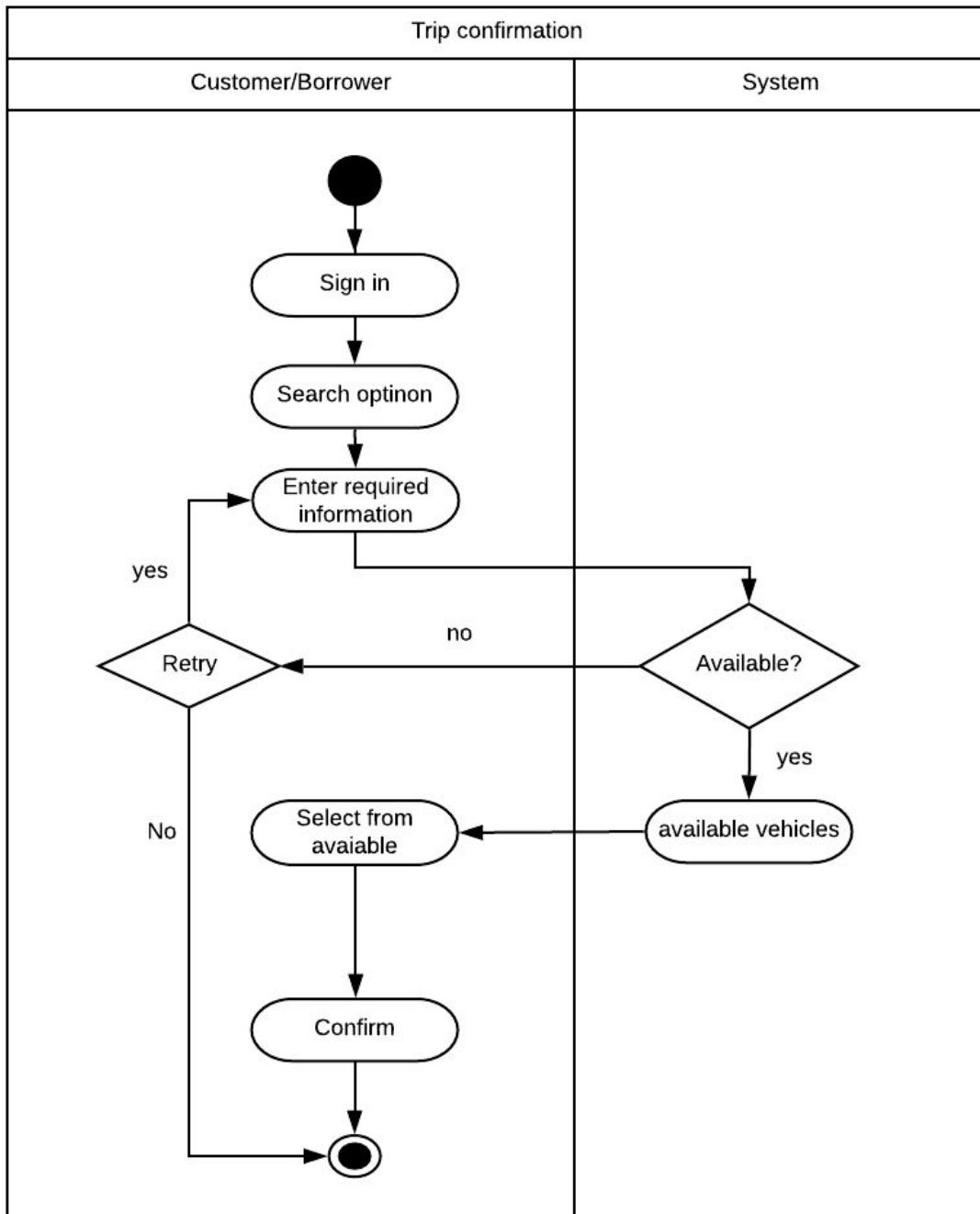


Figure 37: Swimlane diagram for trip confirmation

Use case 15: Trip Cancellation

Activity Diagram

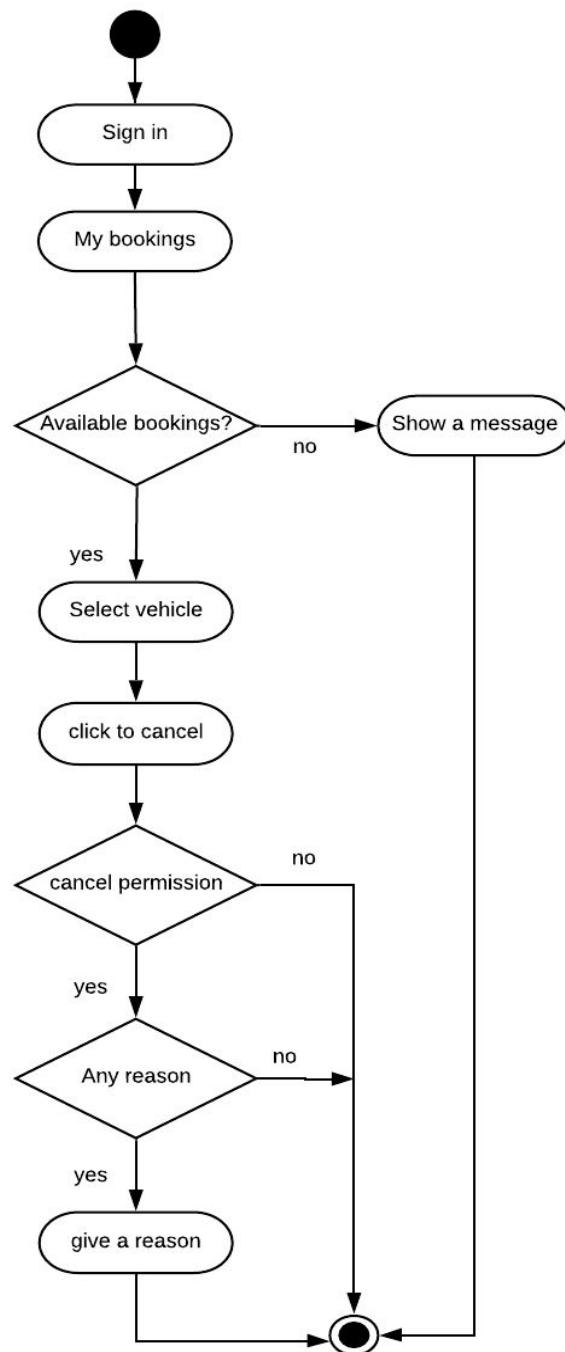


Figure 38: Activity diagram for trip cancellation

Swimlane Diagram

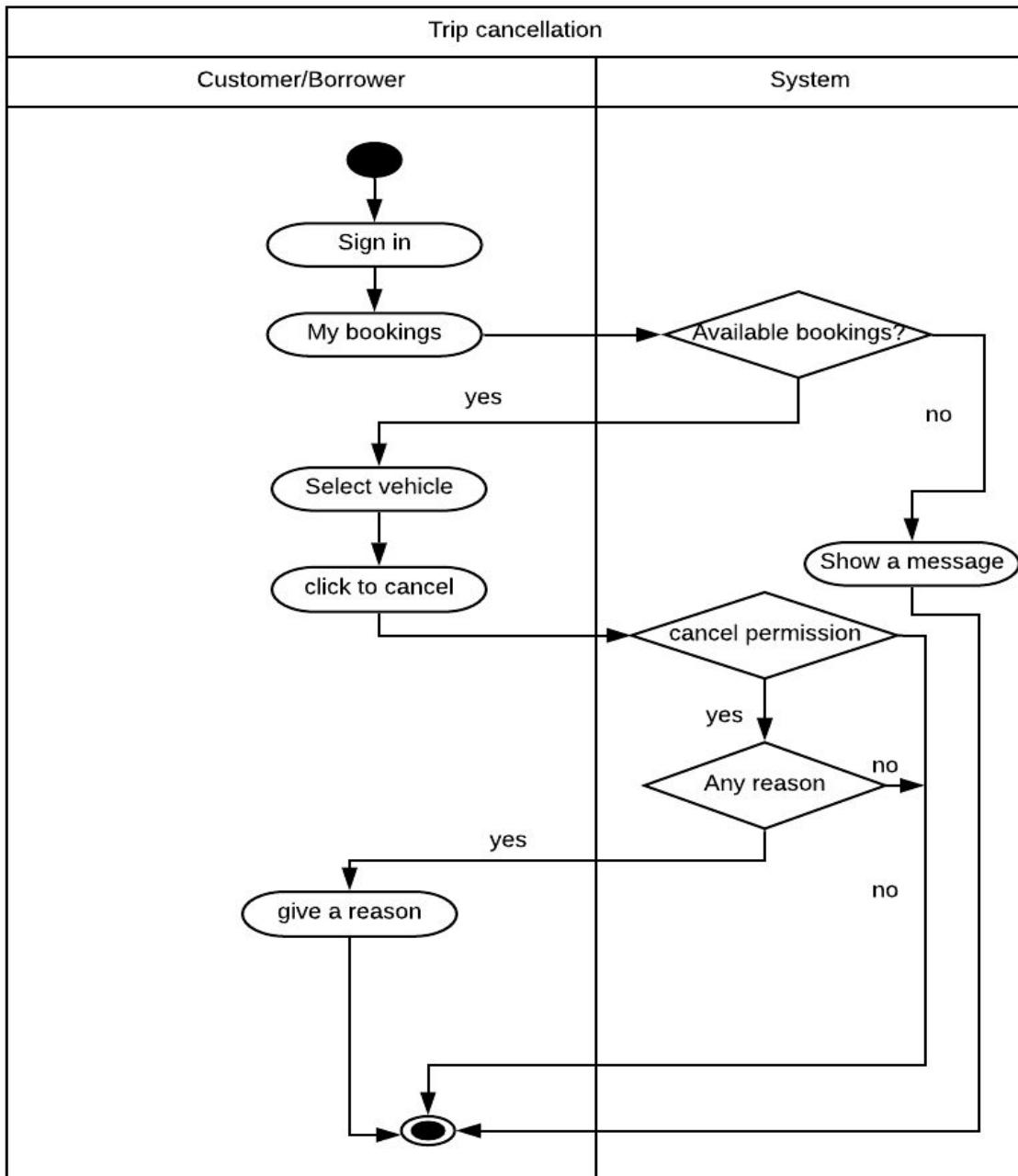


Figure 39: Swimlane diagram for trip cancellation

Use case 16: Payment Activity Diagram

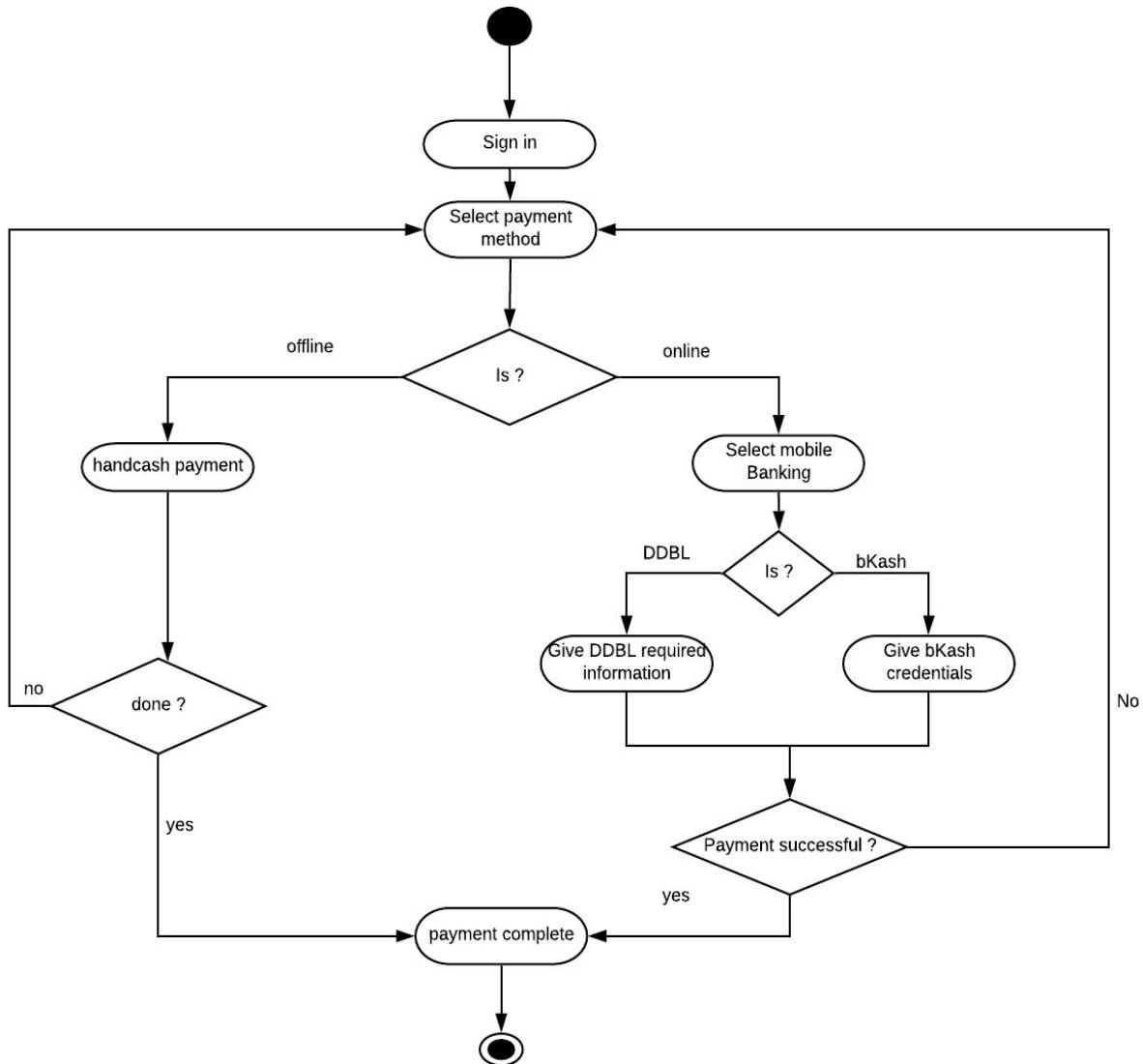


Figure 40: Activity diagram for payment

Swimlane Diagram

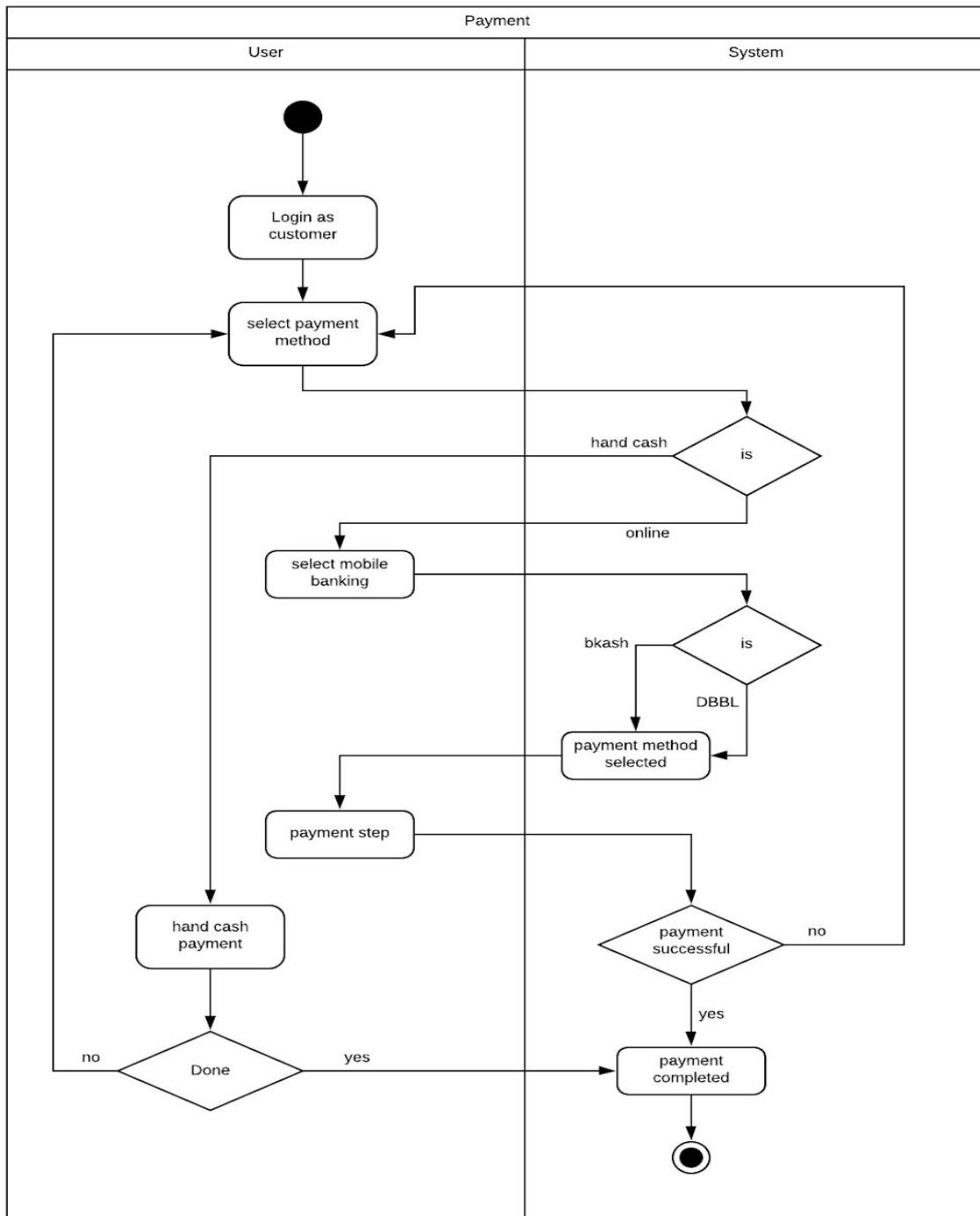


Figure 41: Swimlane diagram for payment

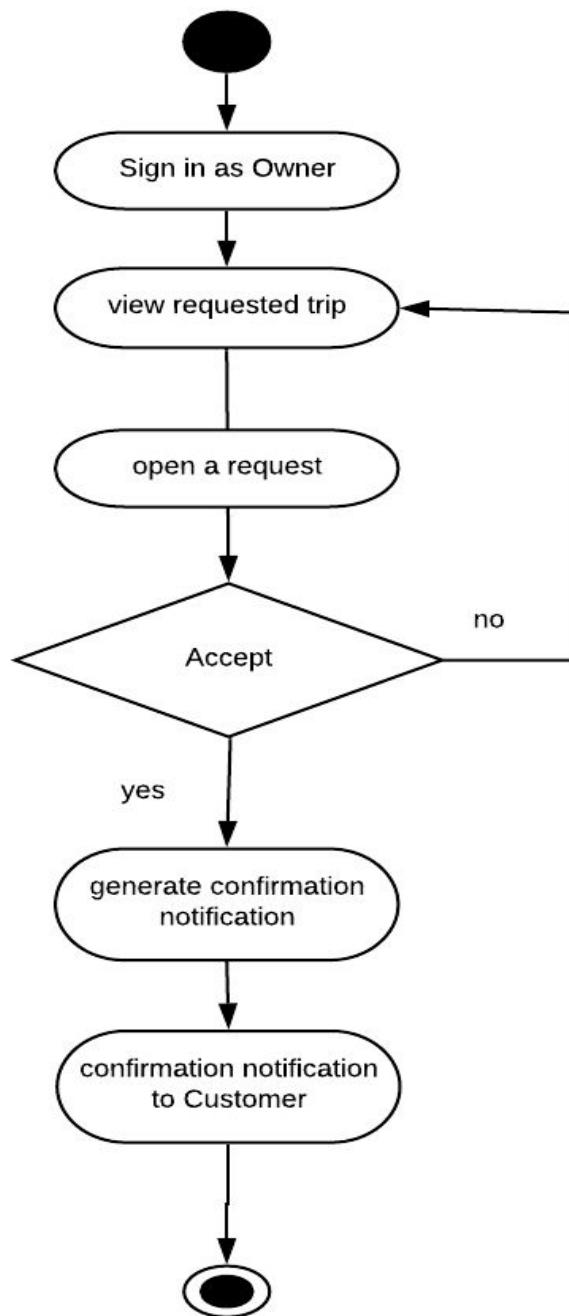
**Use case 17: Confirmation Notification
Activity Diagram**

Figure 42: Activity diagram for confirmation notification

Swimlane Diagram

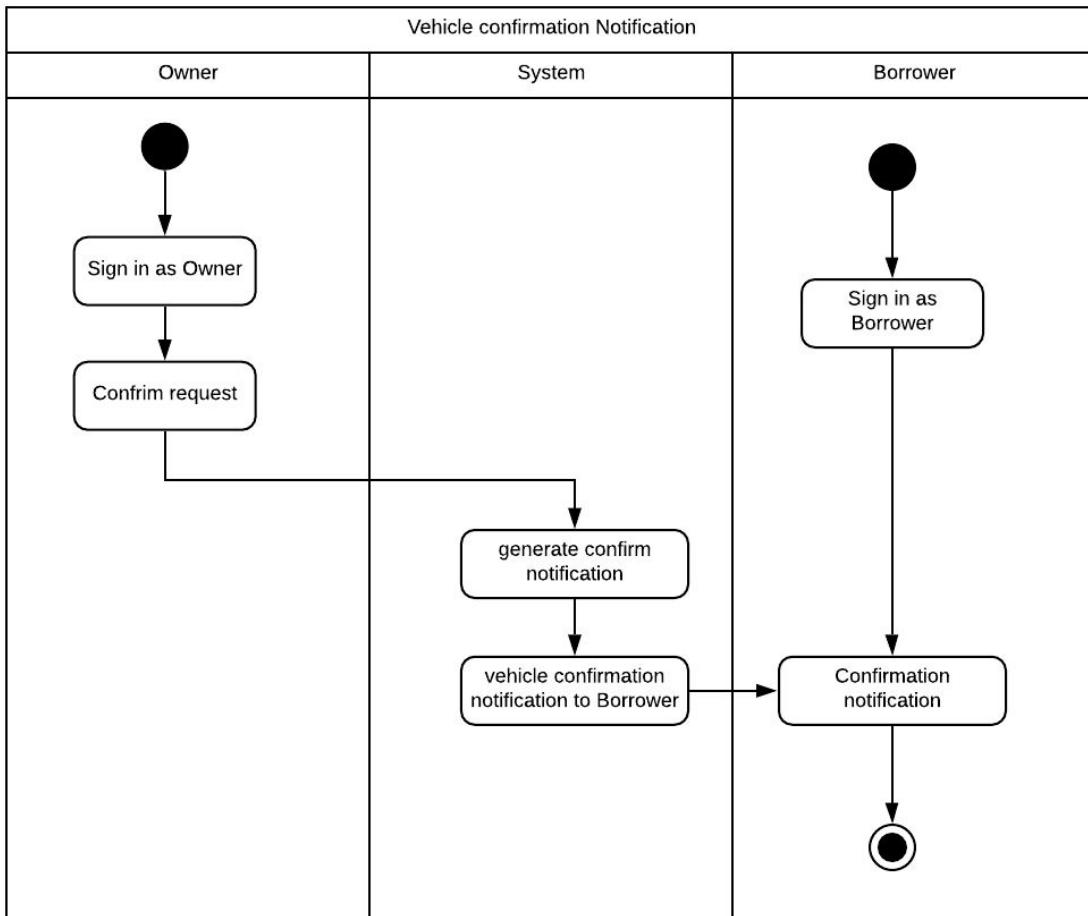


Figure 43: Swimlane diagram for confirmation notification

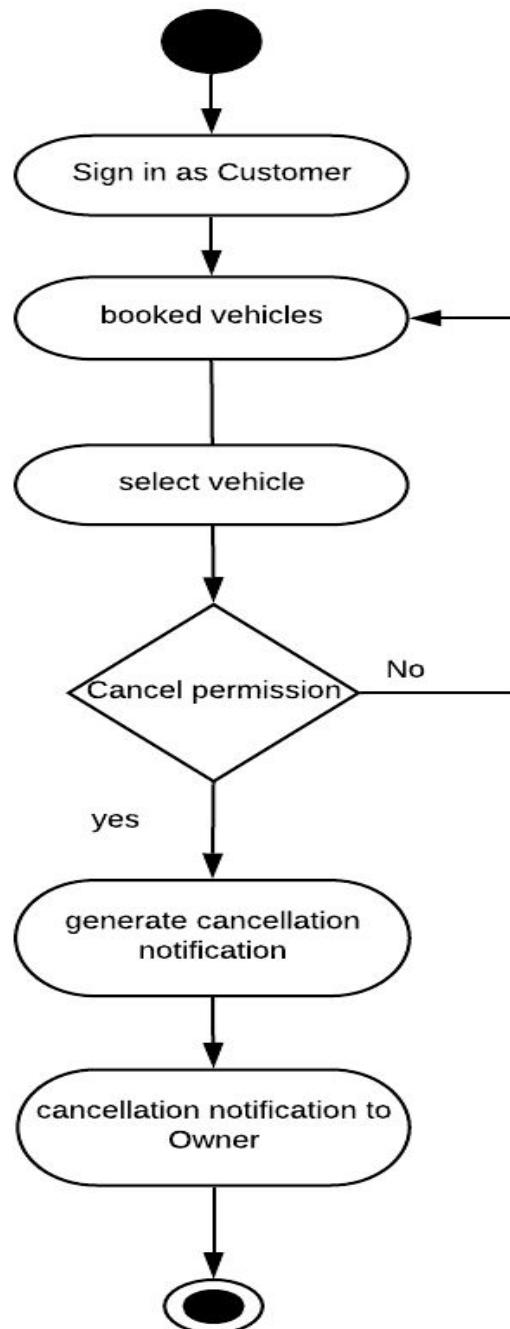
**Use case 18: Cancellation Notification
Activity Diagram**

Figure 44: Activity diagram for cancellation notification

Swimlane Diagram

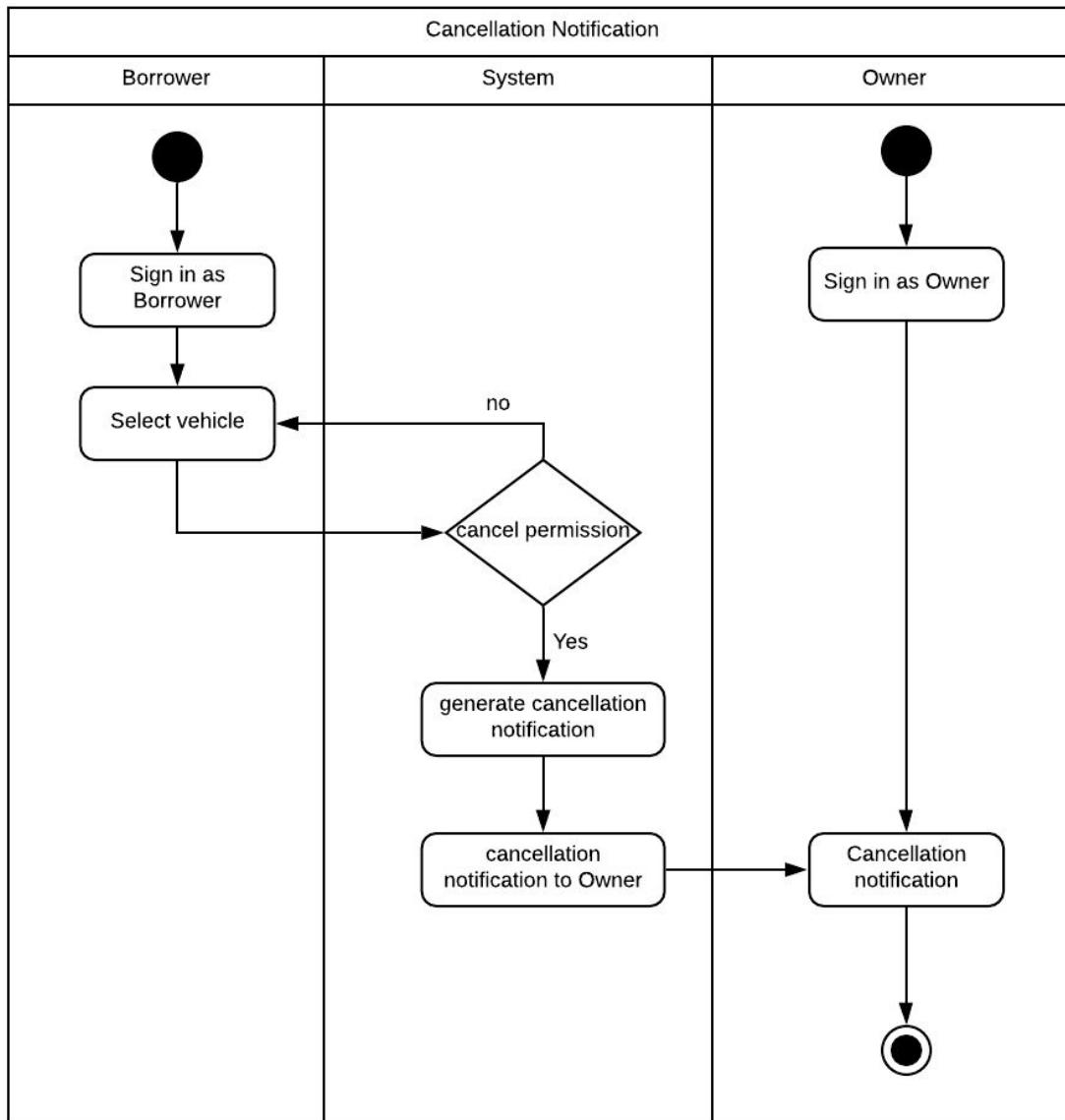


Figure 45: Swimlane diagram for cancellation notification

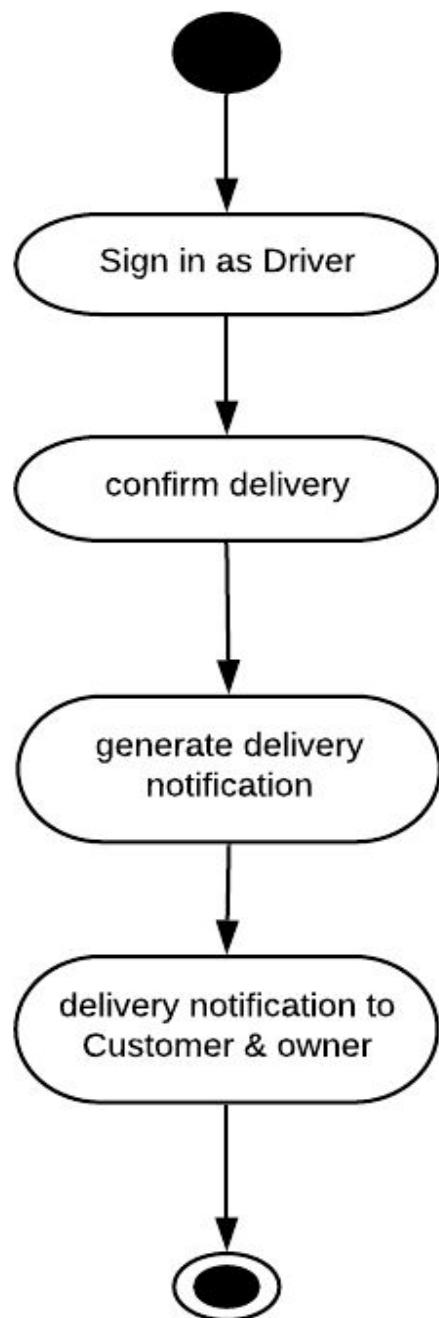
**Use case 19: Delivery Notification
Activity Diagram**

Figure 46: Activity diagram for delivery notification

Swimlane Diagram

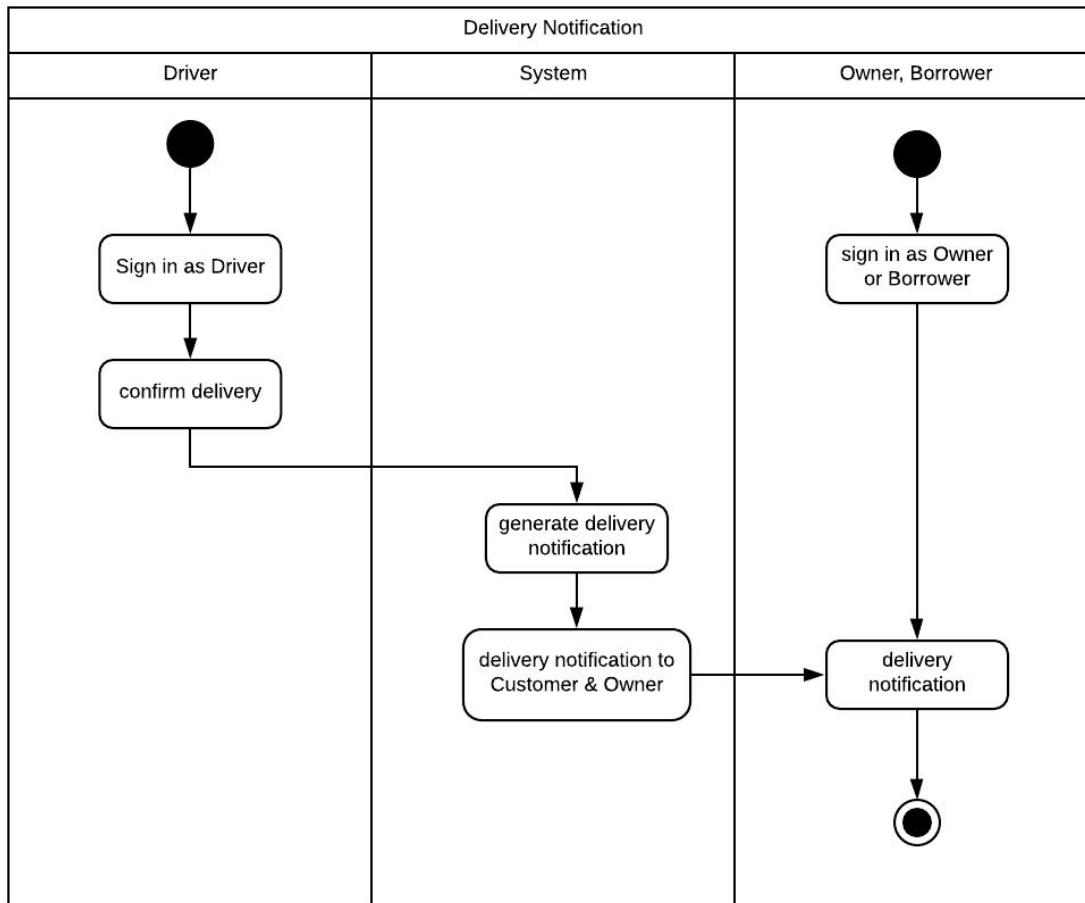


Figure 47: Swimlane diagram for delivery notification

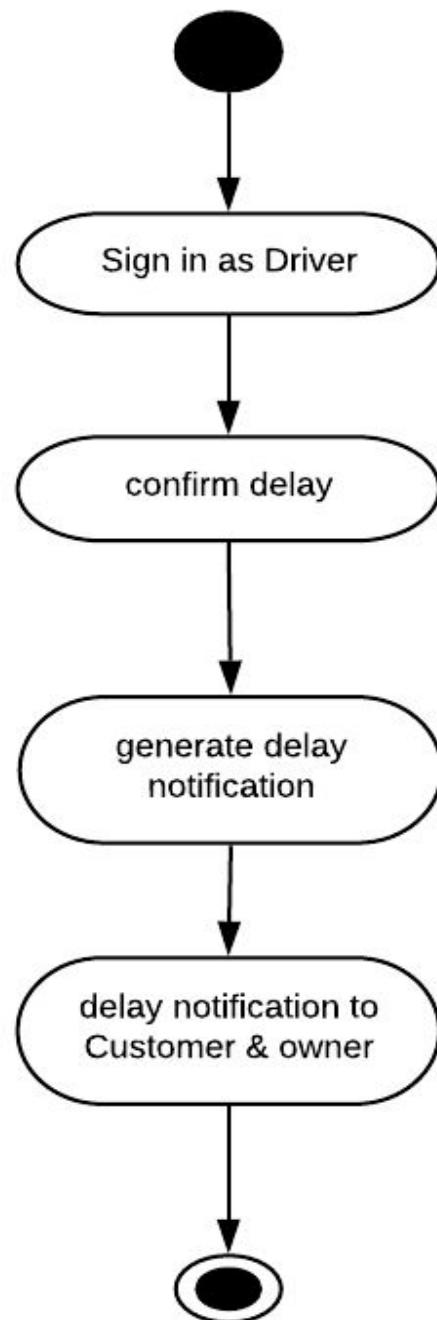
Use case 20: Delay Notification
Activity Diagram

Figure 48: Activity diagram for delay notification

Swimlane Diagram

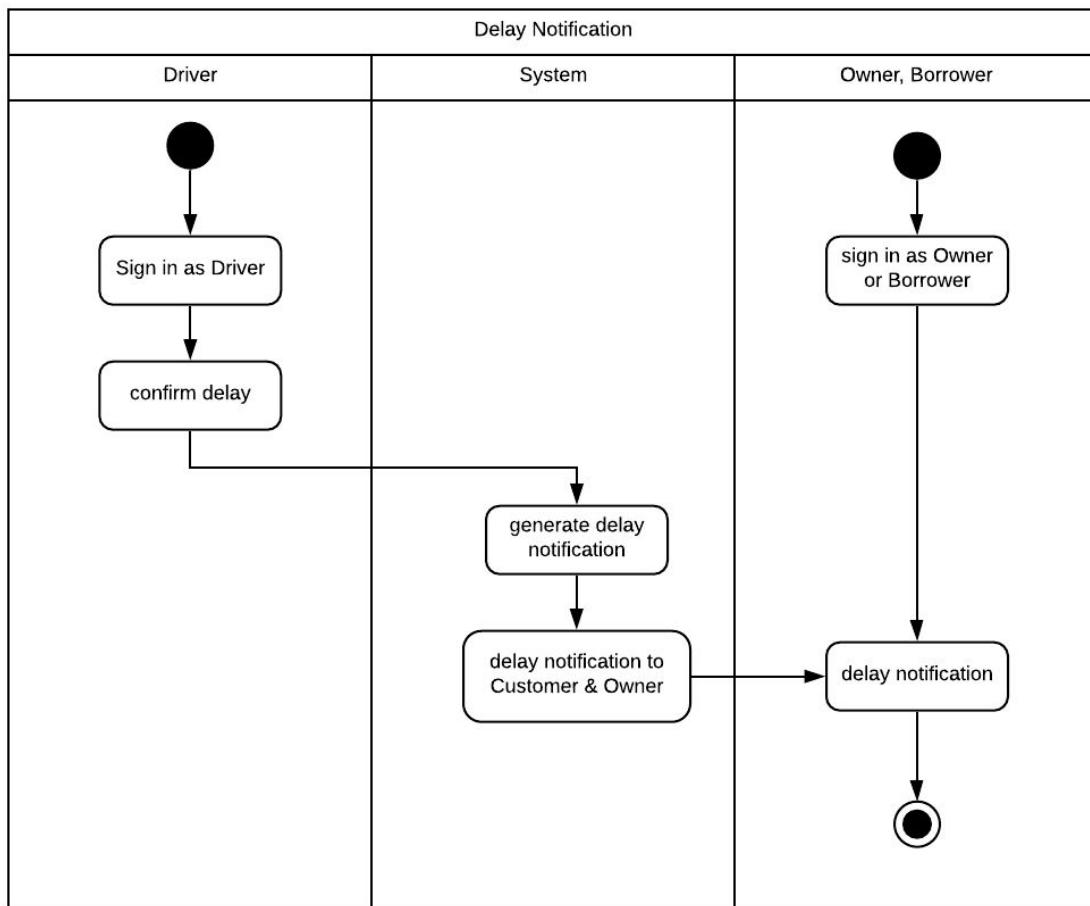


Figure 49: Swimlane diagram for delay notification

Use case 21: Communication Activity Diagram

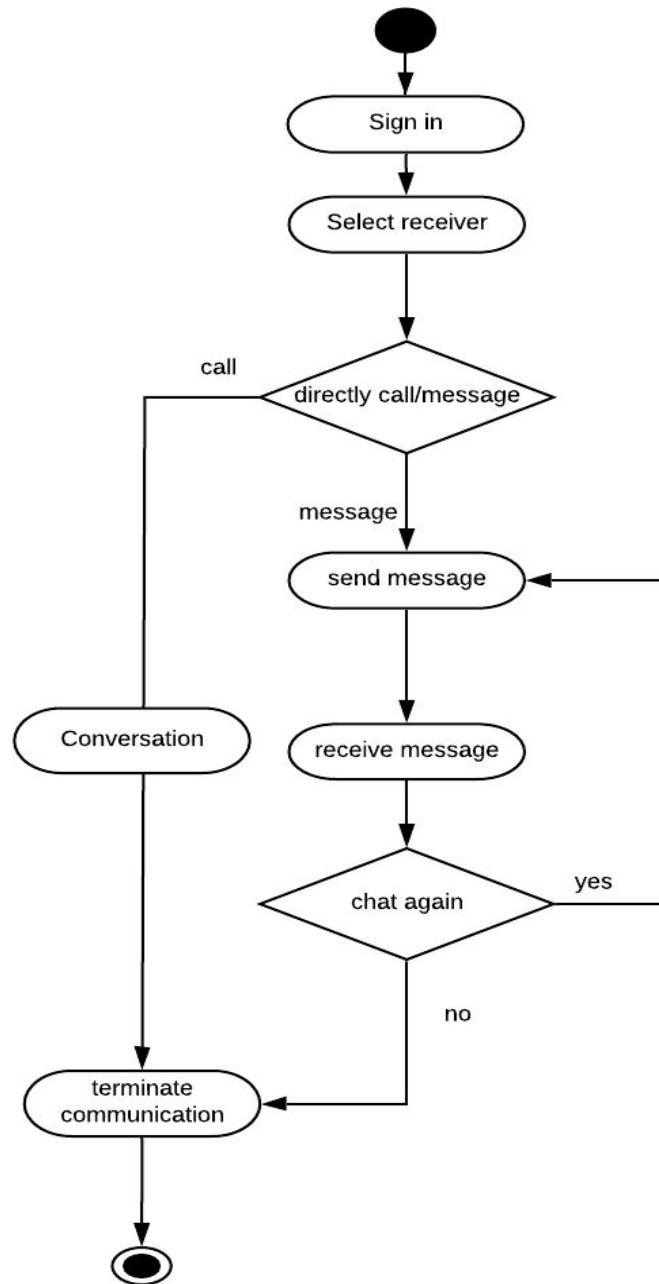


Figure 50: Activity diagram for communication

Swimlane Diagram

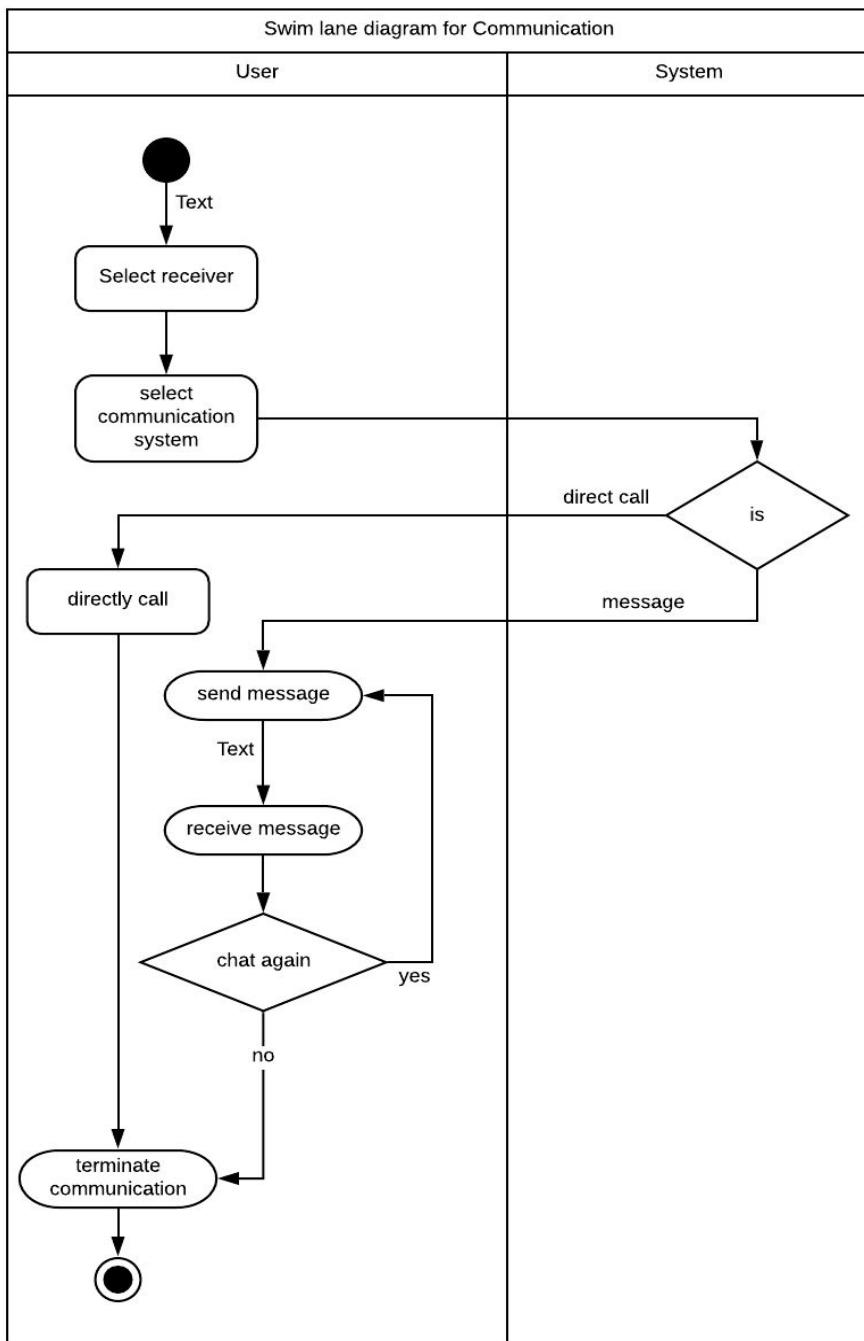


Figure 51: Swimlane diagram for communication

Chapter 5: Data Model

5.1 Introduction

Software requirements include the need to create, extend or interface with a database or if complex data structures must be constructed and manipulated, the software team may choose to create a data model as part of overall requirements modeling.

5.1.1 Data Object Selection

A data object is a representation of information which has different properties or attributes that must be understood by software. Here is the table of potential data objects.

5.1.1.1 Noun Identification

Table 5.1: Noun Identification

Serial no	Noun	P/S	Attributes
1	Confirmation	S	
2	Authority	P	
3	Administrator	S	9,29,57,73
4	Destination	P	
5	Maintenance	P	
6	Vehicle Number	P	
7	Notification	S	8,53,61,67,81
8	Date	S	
9	Password	P	
10	Way	P	

11	Position	P	
12	driver	S	29,34,43,57,49,82,125
13	Authentication	S	
14	shipping	P	
15	Completion	P	
16	Owner	S	29,31,43,57,32,57,73, 116,125
17	Overview	P	
18	Chat	S	
19	Booking	S	
20	delay	S	
21	Requests	S	
22	Bidding	S	
23	Blue book number	P	
24	Total Cost	S	
25	Login	S	
26	Searching	S	
27	trade license	S	
28	Number	P	
29	Mobile number	S	
30	Addition	S	87,117
31	Email	S	
32	Bidder	S	
33	Option	P	
34	driver id	S	

35	Profile	S	
36	Calendar	P	
37	Customers	S	
38	goods weight	S	
39	Distance	P	
40	Types	P	
41	Extra cost	S	
42	Verification	P	
43	NID	s	
44	Truck	S	
45	Vehicle	S	12,16,23,40,57,73,79
46	History	S	4,6,41,52,59,67,79,126
47	Birth	P	
48	Journey	P	
49	driving license	s	
50	Procedure	P	
51	Validity	P	
52	Time	P	
53	Message	S	
54	Terms	P	
55	Storage	P	
56	User	S	
57	Names	S	
58	Business	P	

59	cost	S
60	License	S
61	Borrower	S 9,29,43,31,57
62	price	S
63	sms	s
64	Company	P
65	Place	S
66	Notification	P
67	Id	S
68	Reason	P
69	Page	P
70	Existence	P
71	Communication	S 18,53,89,63
72	Characters	P
73	Capacity	S
74	period	P
75	Booking	S
76	Management	S
77	Company name	S
78	System	S
79	Details	S
80	Minutes	P
81	User Type	S
82	Ratings	S

83	Conditions	P
84	Database	P
85	Vehicle	P
86	Customers	S
87	Expire date	S
88	Model	S
89	Call	P
90	Trips	S
91	Level	P
92	Money	P
93	Capability	P
94	Parts	P
95	Payment	S
		38,39,59, 62,92,104,111
96	Code	S
97	Task	P
98	Government	P
99	Information	P
100	Chassis no	S
101	Completion	P
102	goods	P
103	Capability	S
104	Due	S
105	Demand	P
106	Current position	P

107	Suggestion	P	
108	User Account	S	125
109	Content	P	
110	Screen	P	
111	Unit	S	
112	Memory	P	
113	Technique	P	
114	Image	P	
115	Manufacturing date	S	
116	Phone	S	
117	Start date	S	
118	Borrowing	S	8,119,120
119	initial price	S	
120	Track	S	10,106
121	Current bid	S	
122	District	S	
123	Sub District	S	
124	City	S	
125	address	S	
126	source	S	

5.1.1.2 Potential Data Objects

- User Account: 12,122,123,127
- Administrator: 8,9,57,73
- Vehicle Owner: 29,31,43,57,32,57,73, 116,125
- Borrower: 9,12,29,43,31,57,73,125
- Driver: 29,34,43,57,49,82,125
- Vehicle: 12,16,23,40,57,73,79
- Payment: 38,39,59, 62,92,104,111
- History: 4,6,41,52,59,67,79,126
- Notification: 8,53,61,67,81

5.1.1.3 Final Data Object

Table 5.2: Final Data Object

User Account	<u>Phone</u> , <u>password</u> , <u>name</u> , <u>address</u> , <u>email</u> , NID
Administrator	<u>Admin id</u> , phone, password
Vehicle Owner	<u>Owner id</u> , trade_license, phone,email, password,name,address,NID
Borrower	<u>Borrower id</u> , phone_number, name, password, address, NID
Driver	<u>Phone number</u> , login_code, name, driving_license, NID
Vehicle	<u>Vehicle number</u> , bluebook, capacity, manufacturer_name, type
Payment	<u>Payment id</u> , advance, due, status, date
History	<u>History id</u> , source, destination, vehicle_number, driver_name, vehicle_details, date
Notification	<u>Notification id</u> , message, receiver

5.2 Entity Relationship Diagram

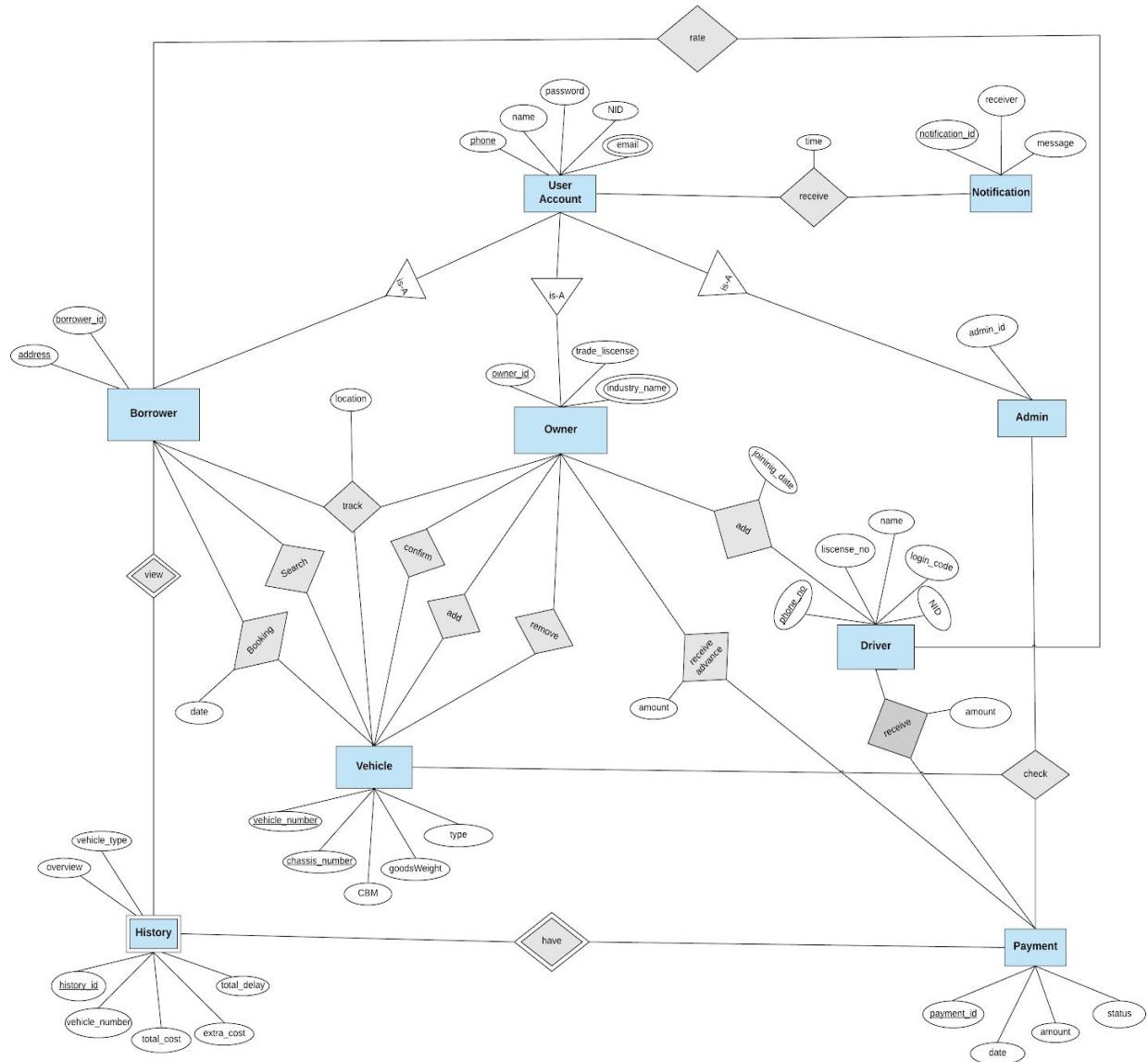


Figure 52: ER diagram of VRMS

5.3 SCHEMA DIAGRAM

Table 5.3: Schema for User Account

User Account		
Attributes	Type	Size
Name	VARCHAR2	80
Password	VARCHAR2	80
Mobile no	NUMBER2	14
Address	VARCHAR2	80
NID	VARCHAR2	80

Table 5.4: Schema for Vehicle Owner

Vehicle Owner		
Attributes	Type	Size
Owner Id	VARCHAR2	80
Contact no	VARCHAR2	14
Email	VARCHAR2	80
Password	VARCHAR2	40
Trade license	VARCHAR2	40
NID	VARCHAR2	20

Table 5.5: Schema for Vehicle

Vehicle		
Attributes	Type	Size
Vehicle Number	VARCHAR2	80
Capacity	VARCHAR2	14
Vehicle Type	VARCHAR2	80
Blue book Number	VARCHAR2	80
Manufacturer name	VARCHAR2	80
Type	VARCHAR2	40

Table 5.6: Schema for User Administrator

Admin		
Attributes	Type	Size
Admin ID	VARCHAR2	80
Contact no	NUMBER	14
Password	VARCHAR2	40

Table 5.7: Schema for Driver

Driver		
Attributes	Type	Size
Name	VARCHAR2	80
Phone number	NUMBER	14
Login_code	VARCHAR2	40
Driving license	VARCHAR2	80
NID	NUMBER	20

Table 5.8: Schema for Trip history

History		
Attributes	Type	Size
Vehicle Type	VARCHAR2	80
Pickup	VARCHAR2	14
Unload	VARCHAR2	80
Vehicle number	NUMBER	80
Driver name	NUMBER	14
Goods details	NUMBER	80
Vehicle details	VARCHAR2	80
Date	VARCHAR2	40
Delay	NUMBER	40
Cost	NUMBER	40
Total payment	NUMBER	40

Table 5.9: Schema for Transaction

Payment		
Attributes	Type	Size
Payment id	VARCHAR2	80
Date	VARCHAR2	20
Advance	NUMBER	40
Due	NUMBER	40
Status	VARCHAR	20

Table 5.10: Schema for Notification

Notification		
Attributes	Type	Size
Notify ID	VARCHAR2	40
Message	VARCHAR2	200
Receiver	VARCHAR2	40

Chapter 6: Class Based Model

This Chapter is intended to describe class-based modeling of Vehicle Renting and Management System.

6.1 Class Based Modeling Concept

Class-based modeling represents the objects that the system will manipulate, the operations that will be applied to the objects, relationships between the objects and the collaborations that occur between the classes that are defined.

6.2 General Classification

To identify the potential classes, nouns were selected from the solution space of the story. These were then characterized in seven general classifications. The seven general characteristics are as follows:

1. External Entities
2. Things
3. Events
4. Roles
5. Organizational units
6. Places
7. Structures

Following are the Specifications of the nouns according to the general classifications.

Table 6.2: general classification

Serial no.	Noun	GC
1	Confirmation	3
2	Authority	1,5

3	Administrator	4,5,7
4	Maintenance	3
5	Purpose	
6	Notification	2,3,7
7	Date	
8	Password	
9	Way	
10	Position	
11	driver	4,5,7
12	Authentication	1,3
13	shipping	3
14	Completion	3
15	Owner	4,5,7
16	Documents	2
17	Chat	3
18	Booking	3
19	delay	2
20	Requests	3
21	Bidding	3
22	Basis	
23	Window	
24	SignUp	3
25	Login	3

26	Searching	3
27	trade liscense	2
28	Number	
29	Mobile number	
30	Addition	
31	Email	
32	Bidder	4,5
33	Option	
34	driver id	
35	Profile	
36	Calendar	2
37	Customers	4,5,7
38	goods weight	2
39	Person	
40	Types	
41	Transaction	3
42	Verification	3
43	NID	
44	Truck	2
45	Vehicle	2
46	History	2
47	Birth	
48	Journey	3

49	driving liscense	
50	Validity	
51	Time	
52	Message	2
53	Terms	
54	Storage	
55	User	4,5,7
56	Names	
57	Business	
58	cost	
59	License	
60	Borrower	4,5,7
61	price	
62	Company	5
63	Id	
64	Reason	
65	Page	
66	Communication	3
67	Availability	
68	period	
69	Management	3
70	System	4,5,7
71	Details	

72	Minutes	
73	Opinions	
74	Conditions	
75	Database	1,6
76	Expire date	
77	Model	
78	Call	
79	Trips	3
80	Level	
81	Money	
82	Capability	
83	Parts	
84	Payment	3
85	Code	
86	Task	
87	Government	
88	Information	2
89	Chassis no	
90	goods	2
91	Due	
92	Demand	
93	Current position	
94	Suggestion	

95	User Account	3
96	Content	2
97	Distance	
98	Unit	
99	Memory	
100	Technique	
101	Image	
102	Manufacturing date	
103	Phone	
104	Start date	
105	Borrowing	3
106	initial price	
107	Track	3
108	Current bid	
109	District	
110	Sub District	
111	City	
112	address	
113	source	2
114	Password	

6.3 Selection Criteria

The potential classes were then selected as classes by six selection criteria. A potential class becomes a class when it fulfills all six characteristics.

1. Retained Information
2. Needed Services
3. Multiple Attributes
4. Common attributes
5. Common operations
6. Essential requirements

Table 6.3: Selection criteria

Serial no.	Noun	AC
1	Confirmation	
2	Authority	
3	Administrator	1,2,3,4,5
4	User Account	1,2,3,4,5
5	Maintenance	1,2
6	Notification	2,3,4,5
7	Authentication	1,2,3,4,5
8	Shipping	
9	system	6
10	Owner	1,2,3,4,5
11	Document	
12	Booking	1,2,3,4

13	Bidding	1,2,3,4
14	Searching	1,2,3,4
15	Profile	
16	Borrower	1,2,3,4,5
17	Verification	1,2,3
18	Vehicle	1,2,3,4,5
19	History	1,2,4
20	Driver	1,2,3,4,5
21	Communication	1,2,3,4
22	Management	
23	Payment	1,2,4,5
24	Tracking	1,2,4,5
25	Database	6

6.4 Finalizing Classes

To identify final classes we need to first check that if there can be any hierarchies or merges. After checking, we got nine final classes. These are given below:

1. User Account
2. Admin
3. Owner
4. Borrower
5. Driver
6. Vehicle
7. System

8. Database
9. Searching
10. Booking
11. Communication
12. Notification
13. Transaction
14. Tracking
15. History

6.5 Class Card

Table 6.5.1: User Account class card

User_Accounts	
Attributes	Method
name phone password address nID	<ul style="list-style-type: none"> • signIn() • signOut() • modifyAccount() • setPasswords() • setNID() • getNID() • setAddress() • getAddress() • setEmail() • getEmail() • setMobileNo() • getMobileNo()
Responsibilities	Collaboration
→ store user info → show user info → modify user info → taking input for sign in → finding user from Database	Owner Admin Borrower

Table 6.5.2: Borrower class card

Borrower	
Attributes	Method
borrowerID(BID)	<ul style="list-style-type: none"> • viewBorrowerInterface() • searchVehicle() • viewPriceRate() • Communication() • confirmTrip() • checkNotification() • getVehicleLocation() • viewHistory() • cancelTrip()
Responsibilities	Collaboration
→ account management → vehicles booking → manage trip → check price rate → check history	UserAccount Payment Vehicle Notification Communication History

Table 6.5.3: Owner class card

Owner	
Attributes	Method
ownerID tradeLicense	<ul style="list-style-type: none"> • addDriver • removeDriver • addVehicle() • removeVehicle() • getVehicleLocation() • Communication • confirmRequest() • getNotification() • viewHistory() • setTradeLicense() • getTradeLicense()
Responsibilities	Collaboration
<ul style="list-style-type: none"> → Tracking vehicle → Maintain Driver → Vehicle Management → Transaction → Communication → Check History 	UserAccount Borrower Driver Transaction

Table 6.5.4: Admin class card

Admin	
Attributes	Method
adminID	<ul style="list-style-type: none"> • modifyAccount() • setPasswords() • setMobileNo() • getMobileNo() • checkVehicleValidity() • checkTransaction()
Responsibilities	Collaboration
→ Checking vehicle validity → Update price	Vehicle Transaction

Table 6.5.5: Driver class card

Driver	
Attributes	Method
driver_ID drivingLicense mobileNo Password NID	<ul style="list-style-type: none"> • sendCurrentLocation() • setPasswords() • setMobileNo() • getMobileNo() • checkVehicleValidity() • checkTransaction()
Responsibilities	Collaboration
<ul style="list-style-type: none"> • Tracking vehicle • Communication with owner and borrower 	Owner Borrower Communication

Table 6.5.6: vehicle class card

Vehicle	
Attributes	Method
vehicle_ID vehicleNumber Capacity vehicleType blueBookNumber manufacturerName	<ul style="list-style-type: none"> • showVehicleInfo() • UpdateVehicleInfo() • getVehicleID() • setVehicleID()
Responsibilities	Collaboration
• Getting vehicle information	Owner Borrower

Table 6.5.7: system class card

System	
Attributes	Method
database	<ul style="list-style-type: none"> • connectDatabase() • accessDatabase() • disconnectDatabase() • calculateCost() • showInterface() • authentication() • showLocation()
Responsibilities	Collaboration
→ Checking vehicle validity → Checking Transaction	Database Account

Table 6.5.8: Database class card

Database	
Attributes	Method
connectionStatus customerTable vehicleTable adminTable driverTable transactionTable	<ul style="list-style-type: none"> • search() • insert() • delete() • update() • isConnected()
Responsibilities	Collaboration
→ Checking vehicle validity → Checking Transaction	Vehicle Transaction

Table 6.5.9: Search class card

Search	
Attributes	Method
pickupLocation unloadLocation date time vehicleType vehicleCapacity	<ul style="list-style-type: none"> • searchByPickupLocation() • searchByVehicleType() • searchByCapacity() • searchByDateTime() • getAvailableVehicle()
Responsibilities	Collaboration
→ Search for availability → View available Vehicle	Vehicle Borrower

Table 6.5.10: Booking class card

Booking	
Attributes	Method
transaction_ID owner_ID borrower_ID vehicle_NO amount date	<ul style="list-style-type: none"> • InstantBooking() • BookingAfterBid() • GenerateBookingId() • GenerateTransactionDetails() • SendBookingdetails() • StoreBookingHistory
Responsibilities	Collaboration
→ BookingVehicle	Vehicle Database

Table 6.5.11: Communication class card

Communication	
Attributes	Method
ConnectionStatus owner_ID borrower_ID driver_ID messageContent	<ul style="list-style-type: none"> • establishConnection() • connectThroughCall() • connectThroughMessage() • sendMessage() • receiveMessage()

Responsibilities	Collaboration
→ Establish connection for communication	Owner Borrower Driver

Table 6.5.12: Transaction class card

Transaction	
Attributes	Method
paymentMethod accountNumber borrower_ID owner_ID distance totalDelay totalCost	<ul style="list-style-type: none"> • calculateTotalPrice() • payViaBkash() • payViaDBBL() • getAccountNumber() • setAccountNumber() • generateTID() • getBorrowerID() • setBorrowerID() • storeCashOnDeliveryInfo()
Responsibilities	Collaboration
→ Calculate total cost → Payment method	Owner Borrower Driver Database

Table 6.5.13: Notification class card

Notification	
Attributes	Method
notify_ID Type receiver	<ul style="list-style-type: none"> • setNotification() • getNotification() • setReceiver() • getReceiver() • getDescription() • setDescription() • generateNofification() • sendNofication()
Responsibilities	Collaboration
→ Notify End user	Owner Borrower

Table 6.5.14:History class card

History	
Attributes	Method
transaction_ID Vehicle Type pickupLocation UnloadLocation Vehicle number Driver name Goods details Vehicle details Date Delay Cost Total payment	<ul style="list-style-type: none"> • setTransactionIDToDatabase() • setPicupLocationToDatabase() • setUnloadLocationToDatabase() • setPaymenToDatabase() • setVehicleNumberToDatabase() • setPaymentStatusToDatabse() • setDriverNameToDatabase() • setDelayDatabase() • setGoodsDetailToDatabase() • setVehicleTypeToDatabase() • getTransactionID() • getPicupLocation() • getUnloadLocation() • getPayment() • getVehicleNumber()

	<ul style="list-style-type: none"> • getPaymentStatus() • getDriverName() • getDelay() • getGoodsDetail() • getVehicleType()
Responsibilities	Collaboration
<ul style="list-style-type: none"> → Set transaction history to database → get trip details 	Transaction Database

Table 6.5.15: Tracking class card

Tracking	
Attributes	Method
traking_Id latitude longitude	<ul style="list-style-type: none"> • generateTrackingID() • setVehicleID() • getVehicleID() • setCurrentLocation() • getCurrentLocation() • sendCurrentLocation()
Responsibilities	Collaboration
→ Track Vehicle	Owner Borrower Driver

6.6 CLASS RESPONSIBILITY COLLABORATION (CRC)

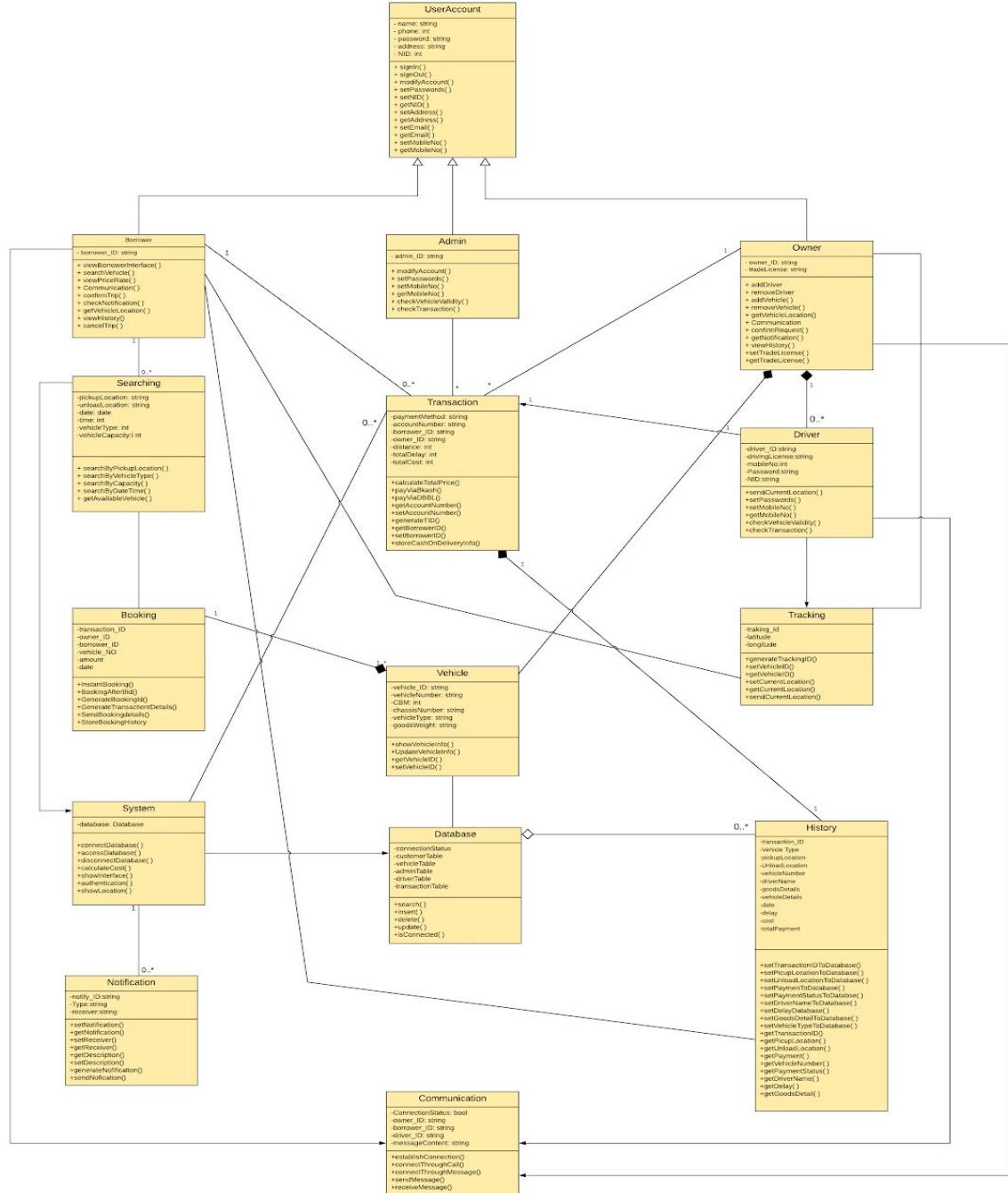


Figure 53: UML class diagram

Chapter 7: Flow-Oriented Modeling

This chapter focuses on the flow oriented modeling.

7.1 Introduction

Although data flow-oriented modeling is perceived as an outdated technique by some software engineers, it continues to be one of the most widely used requirements analysis notations in use today. It provides additional insight into system requirements and flow.

7.2 Data Flow Diagram (DFD)

The DFD takes an input-process-output view of a system. In the figures, data objects are represented by labeled arrows and transformations are represented by circles.

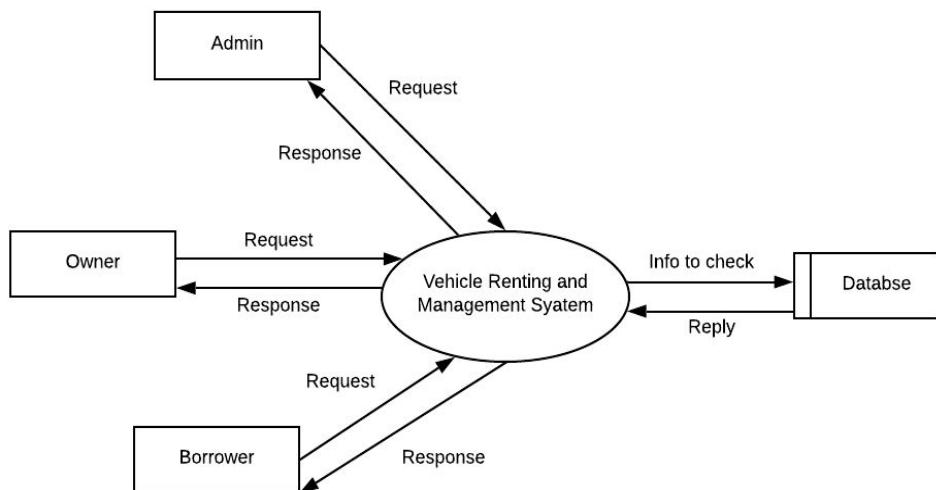


Figure 54: Level 0 DFD VRMS

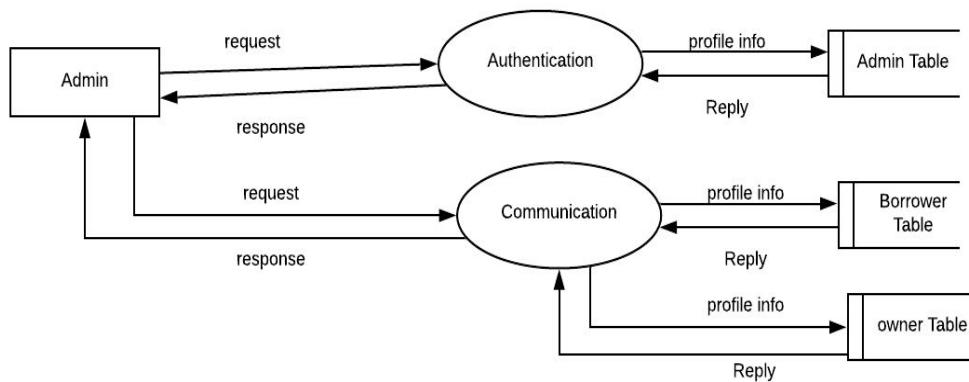


Figure 55: Level 1.1 DFD VRMS

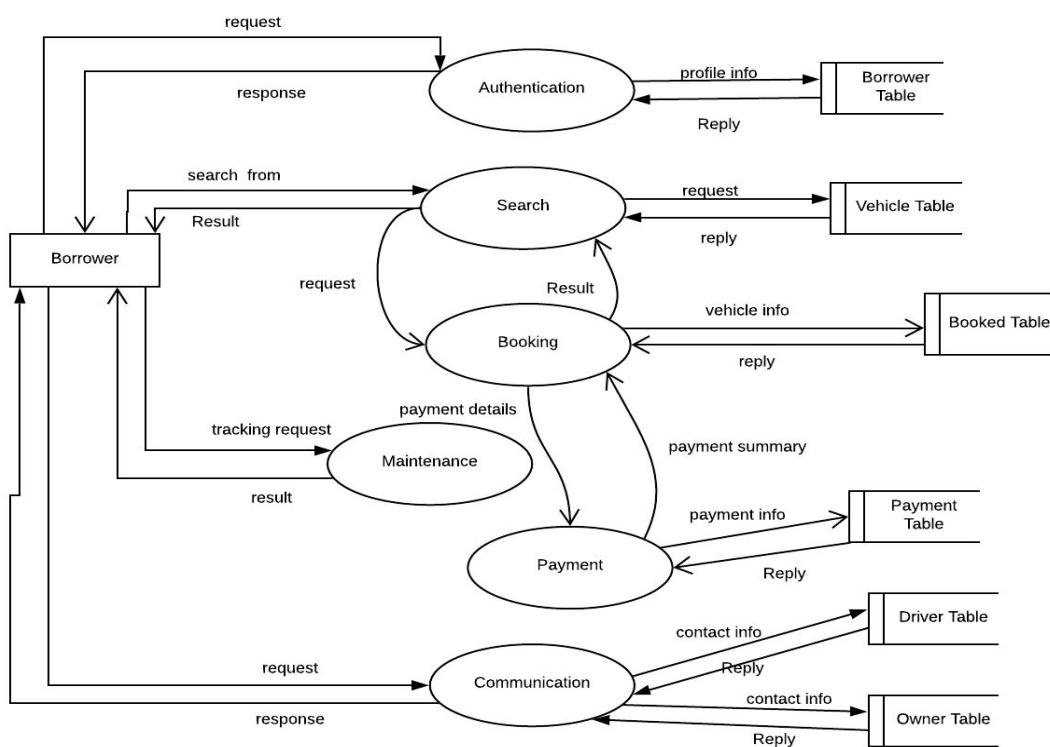


Figure 56: Level 1.2 DFD VRMS

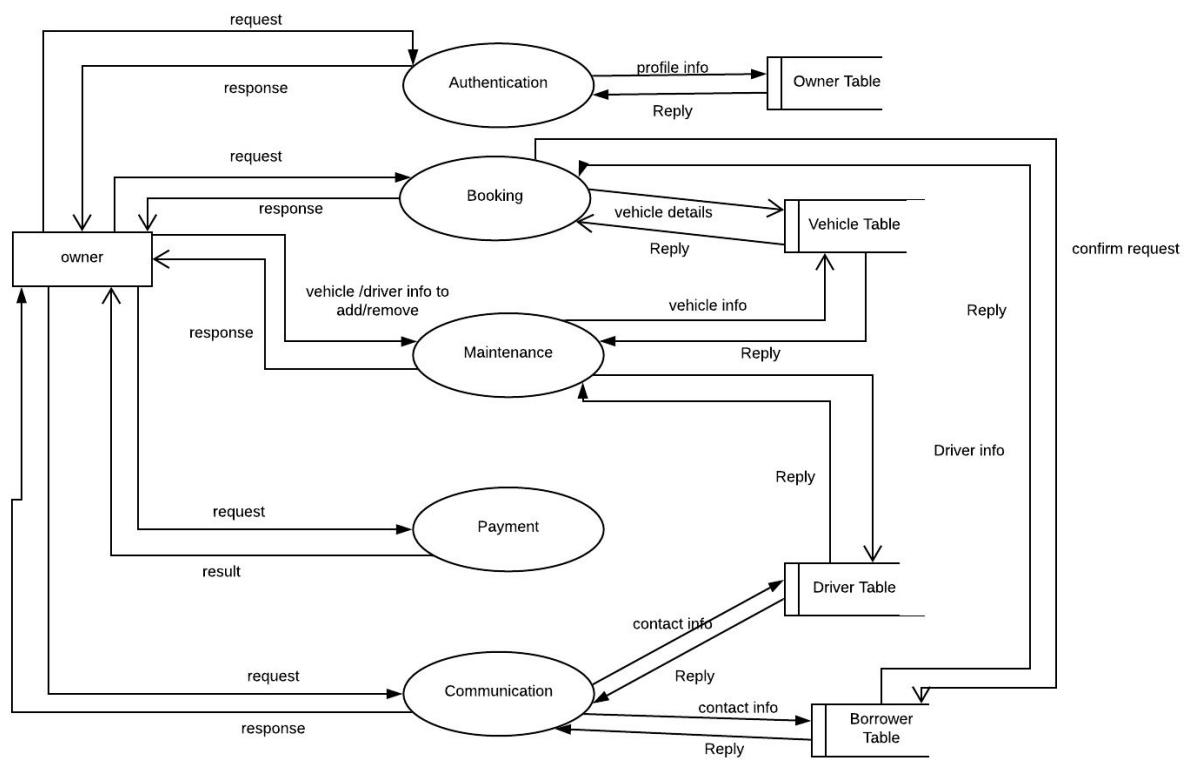


Figure 57: Level 1.3 DFD VRMS

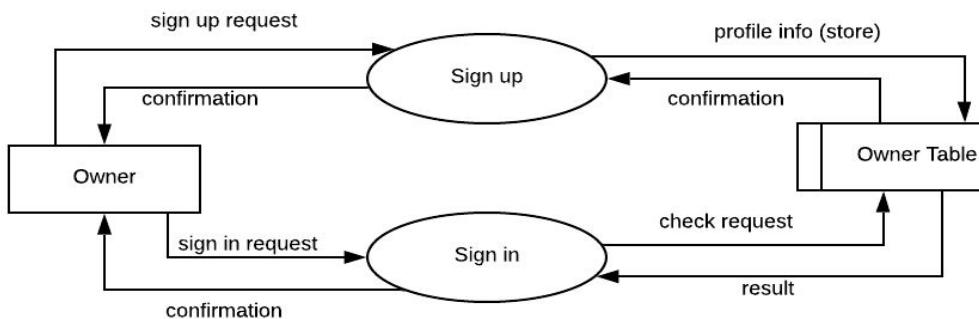


Figure 58: Level 2.1 DFD VRMS

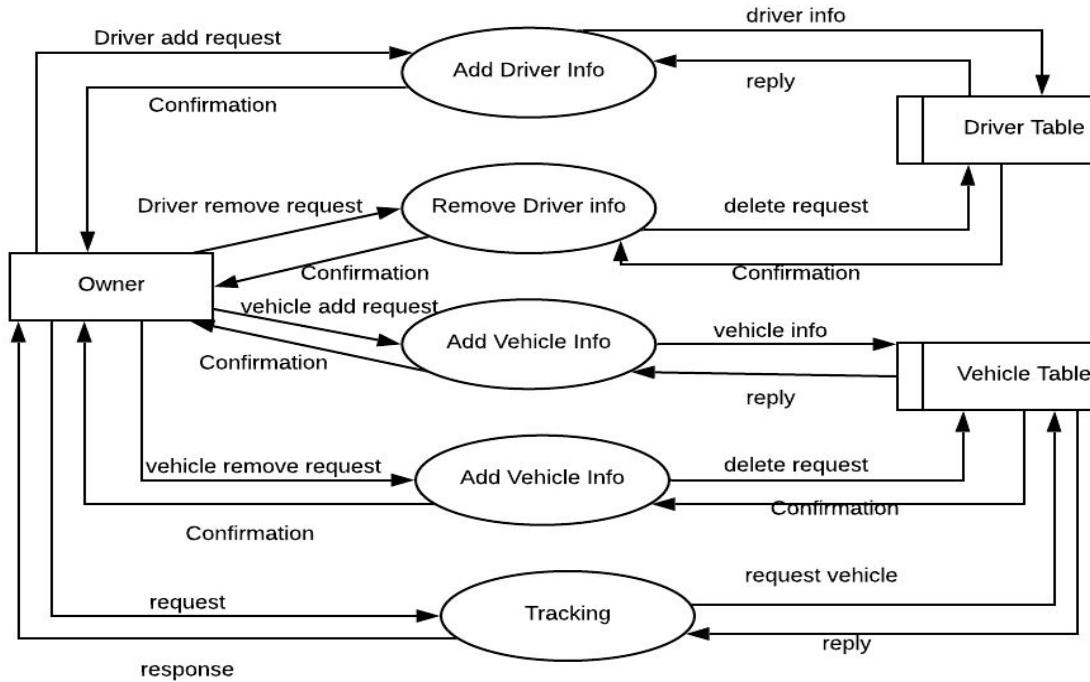


Figure 59: Level 2.2 DFD VRMS

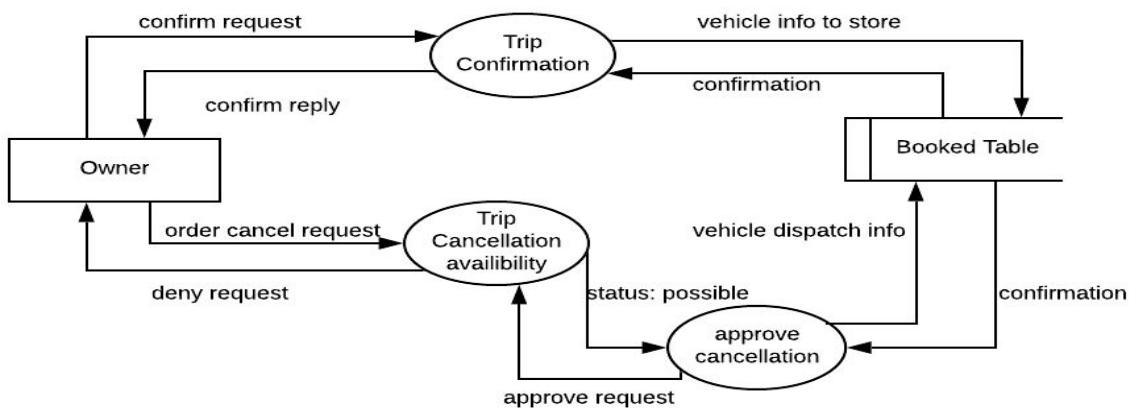


Figure 60: Level 2.3 DFD VRMS

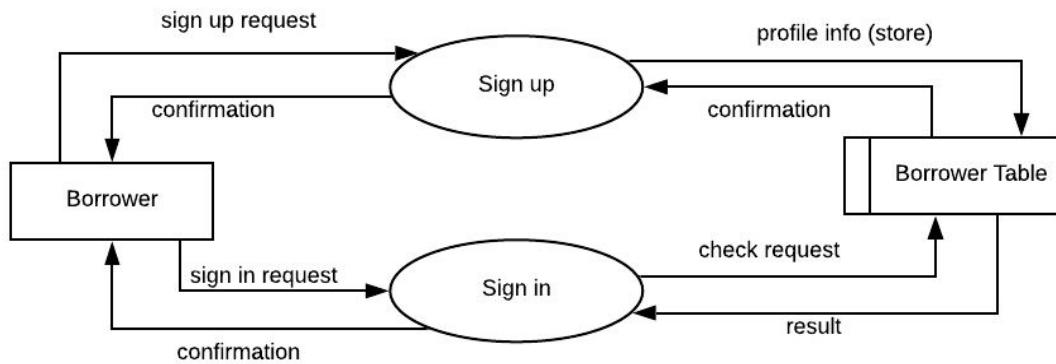


Figure 61: Level 2.4 DFD VRMS

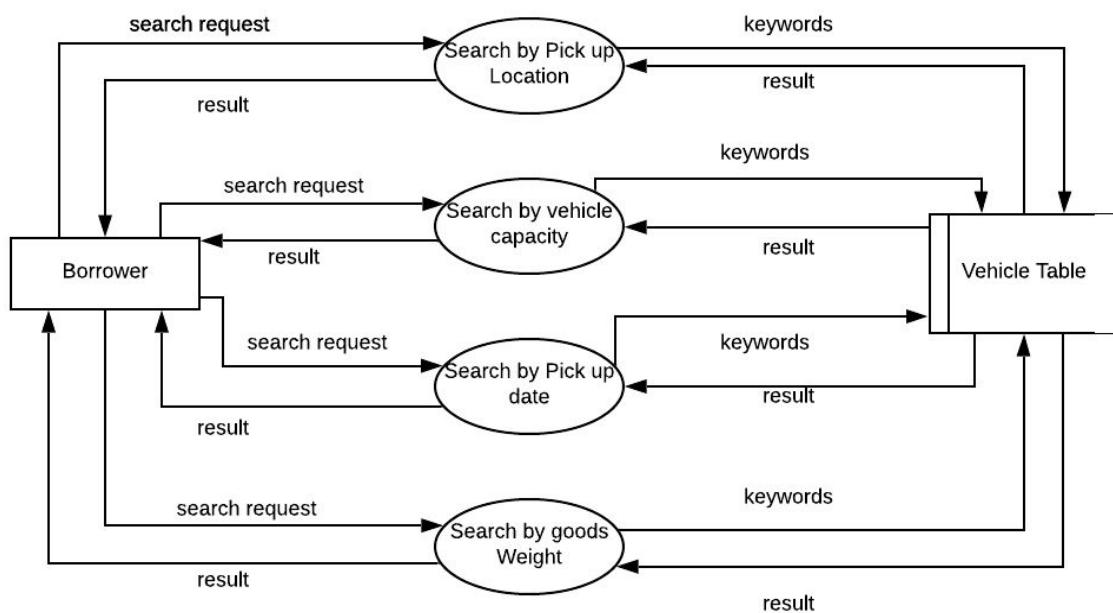


Figure 62: Level 2.5 DFD VRMS

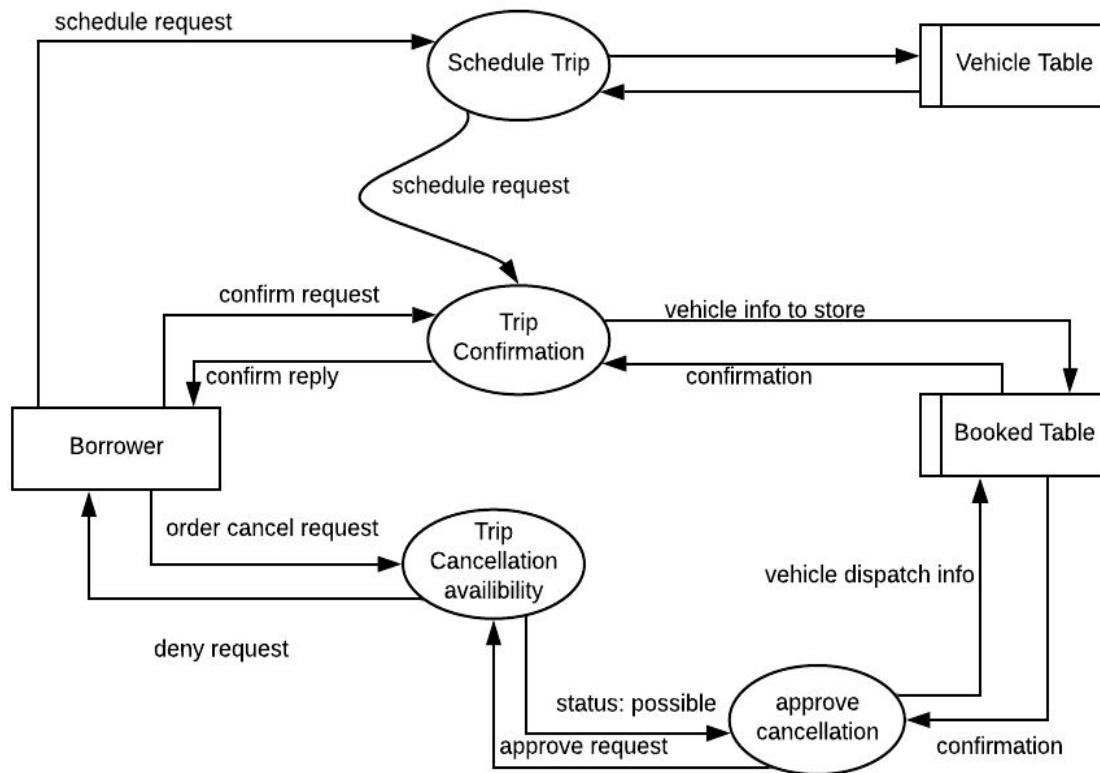


Figure 63: Level 2.6 DFD VRMS

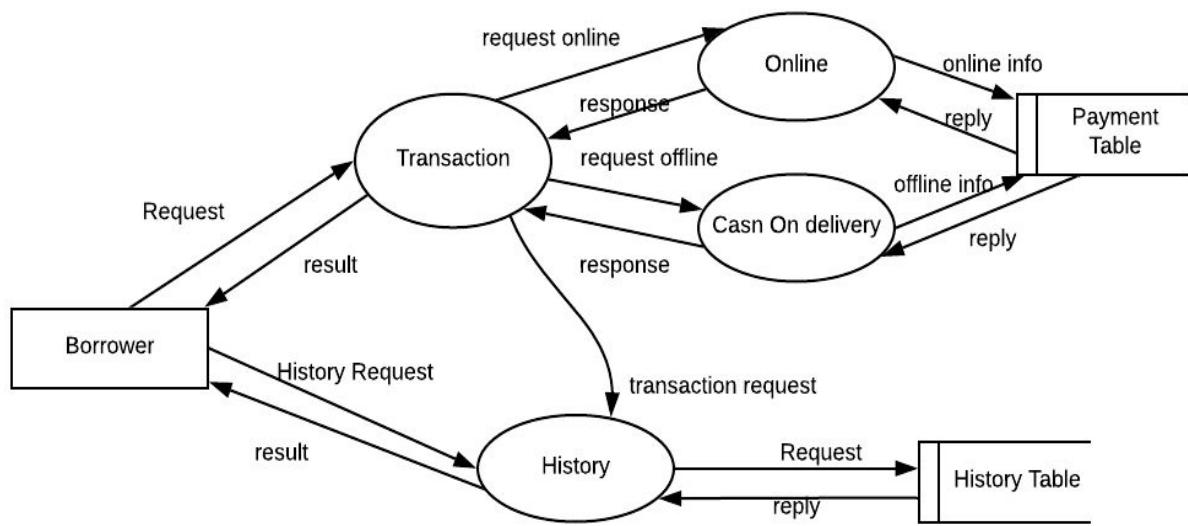


Figure 64: Level 2.7 DFD VRMS

Chapter 8: Behavioral Model

The behavioral model indicates how software will respond to external events.

8.1 State Diagram

State diagram represents active states for each class the events (triggers).

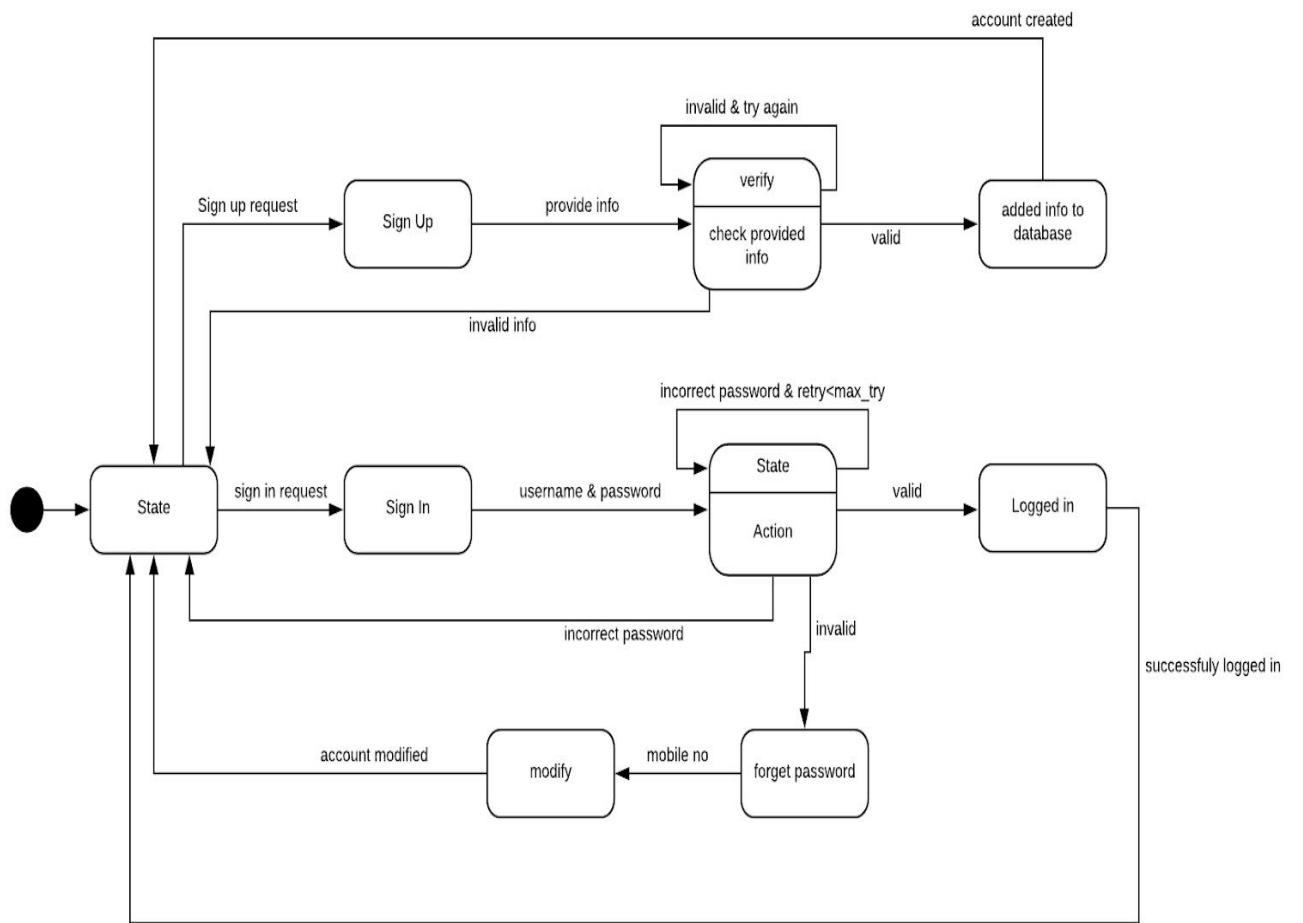


Figure 65: State Diagram(Authentication)

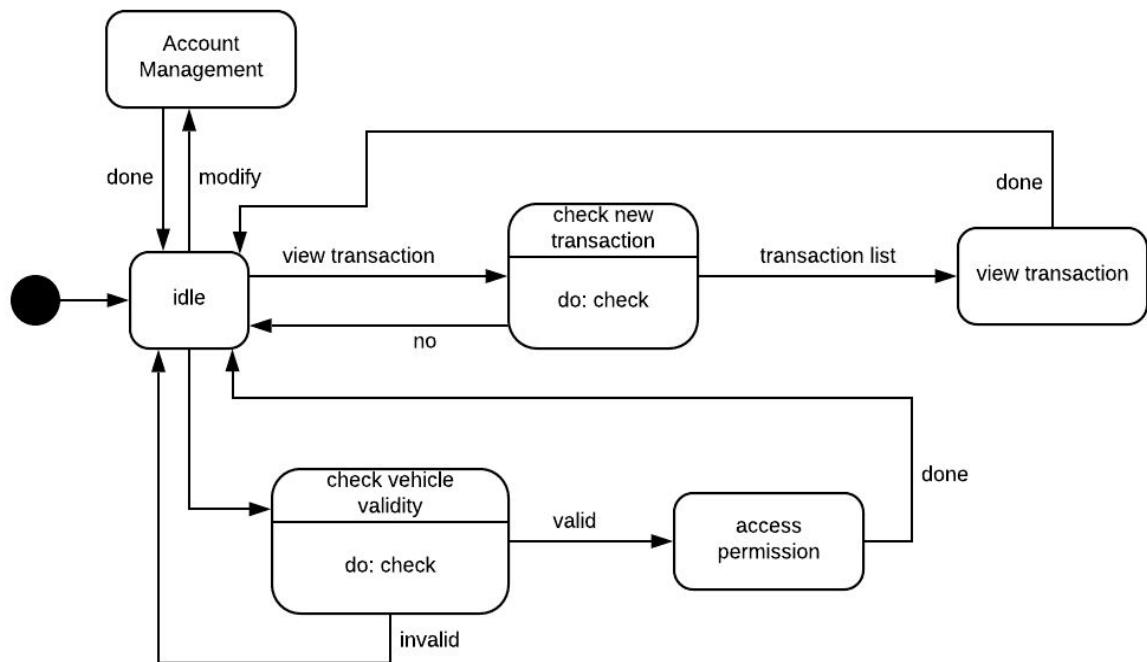


Figure 66: State diagram(Admin)

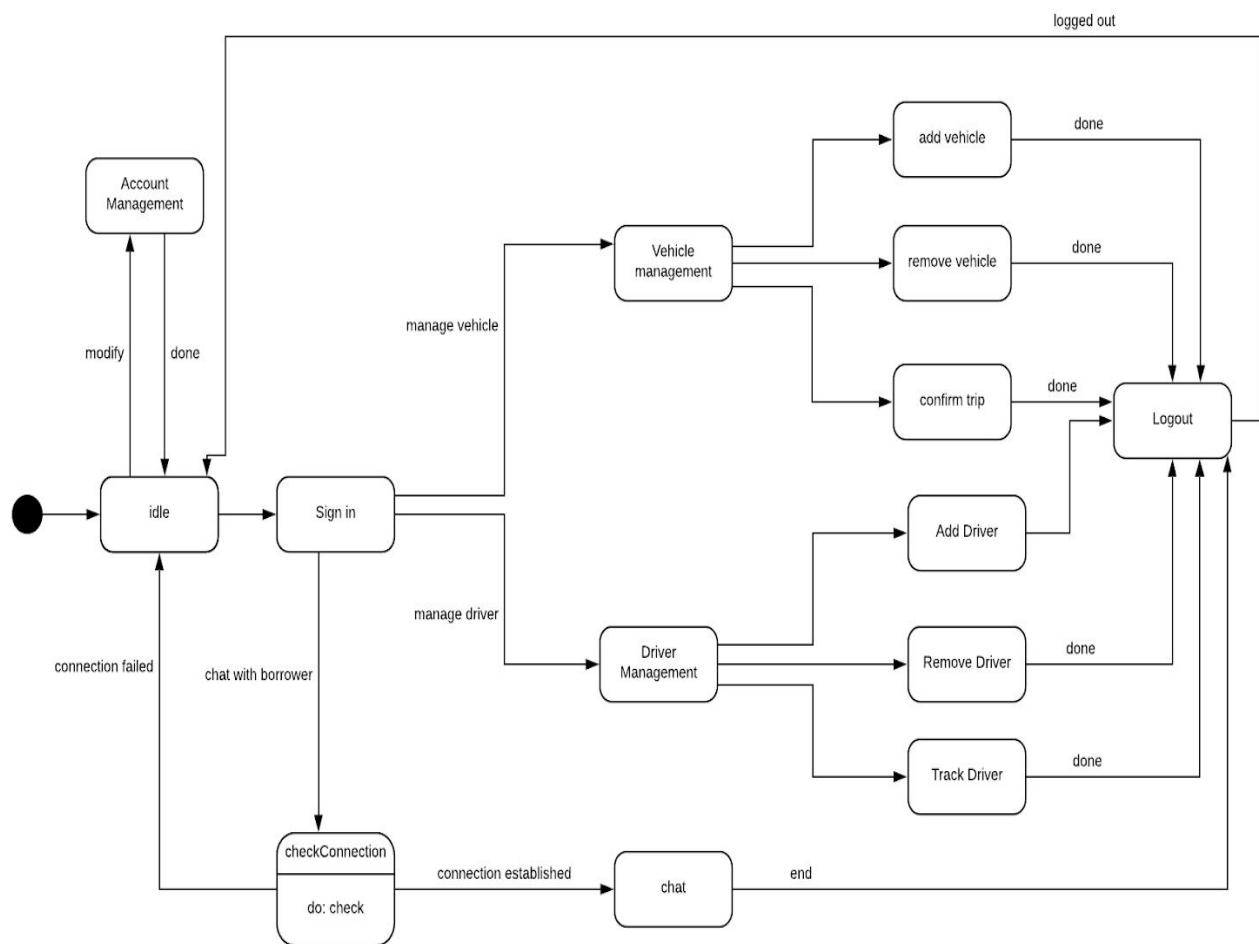


Figure 67: State diagram(Vehicle owner)

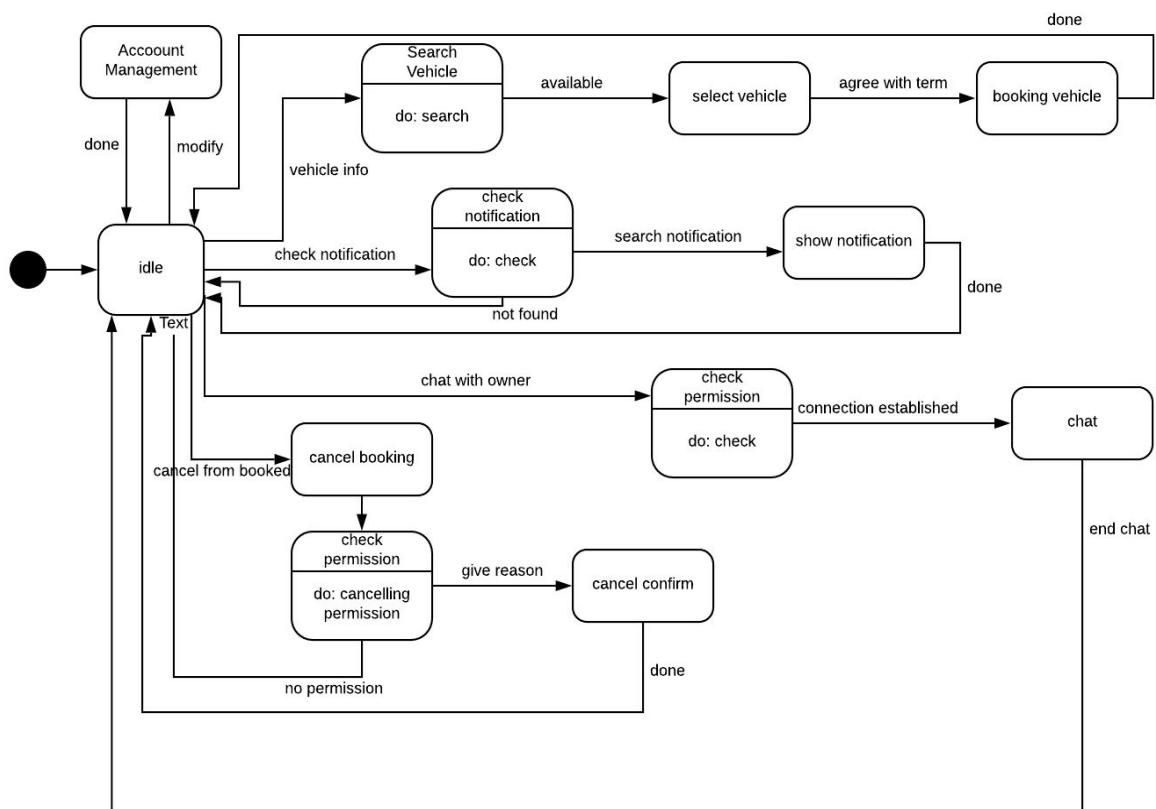


Figure 68: State Diagram(Borrower)

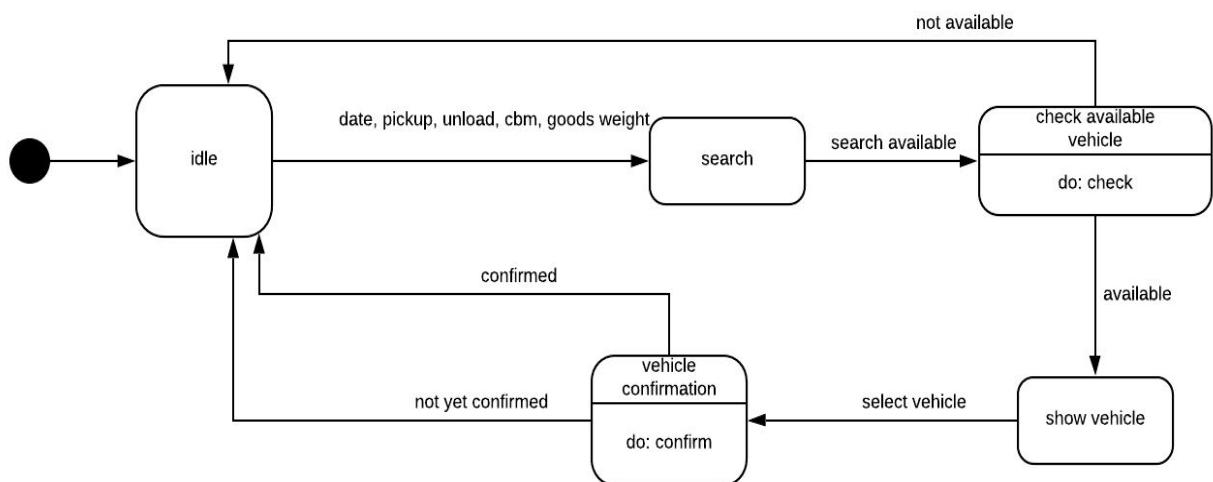


Figure 69: State Diagram(Search)

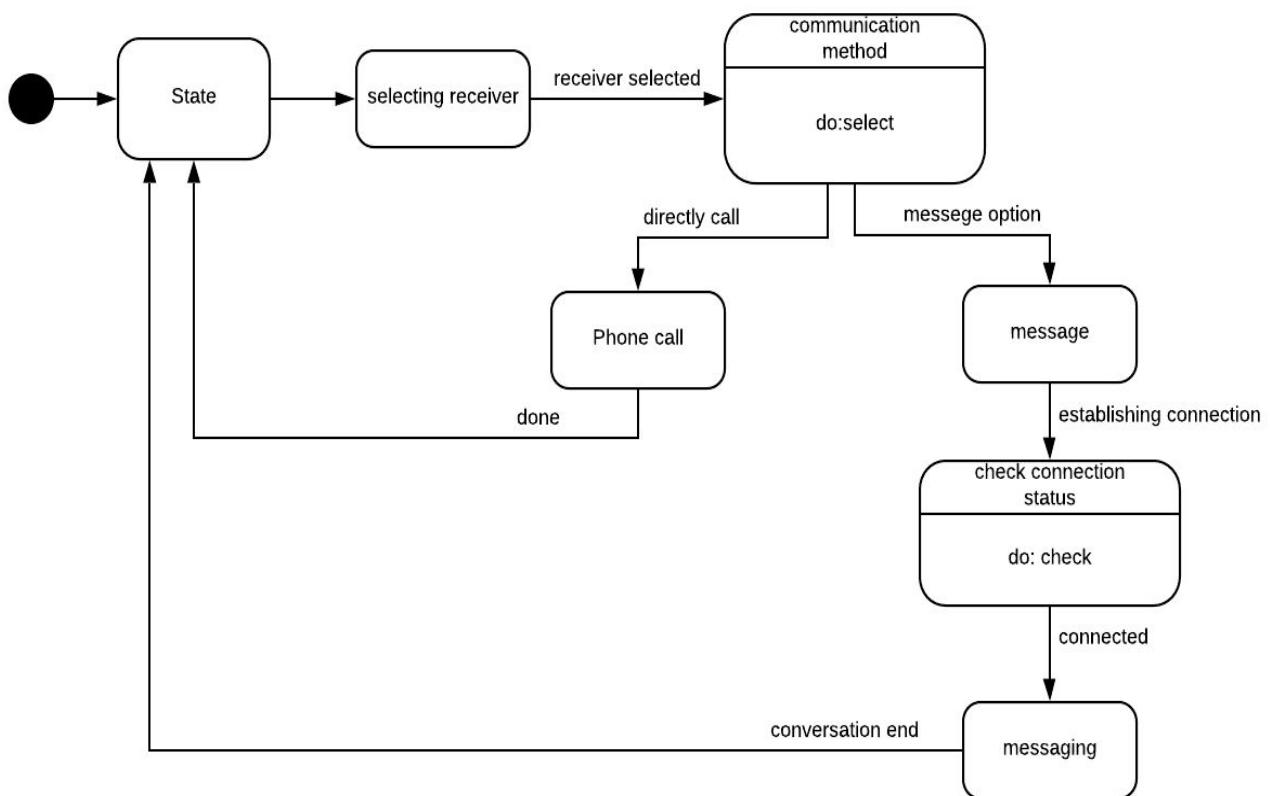


Figure 70: State diagram (Communication)

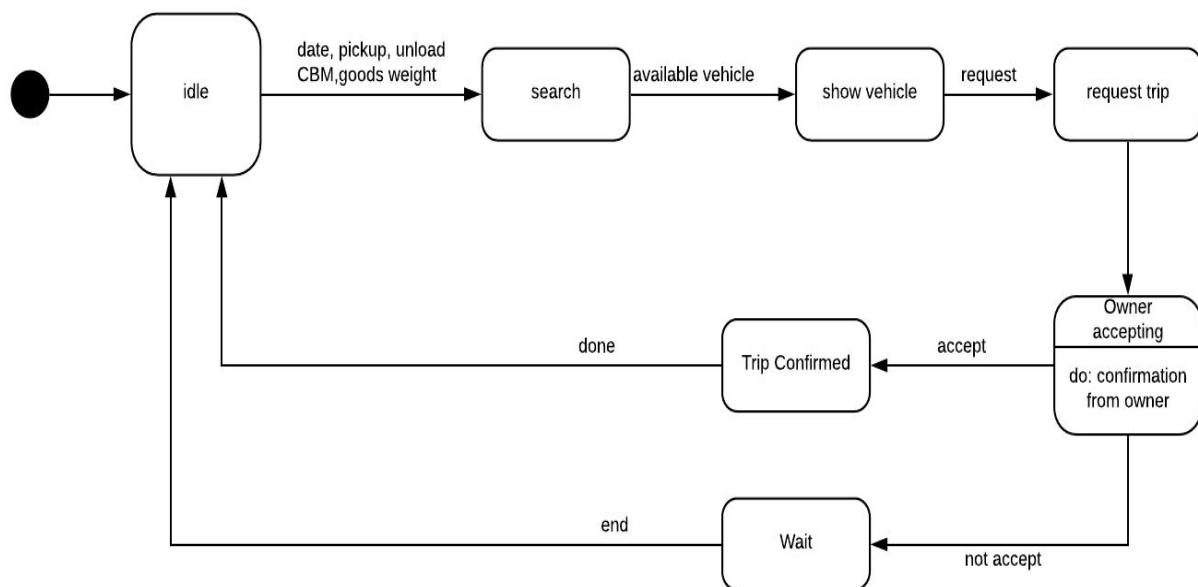


Figure 71: State diagram(Booking)

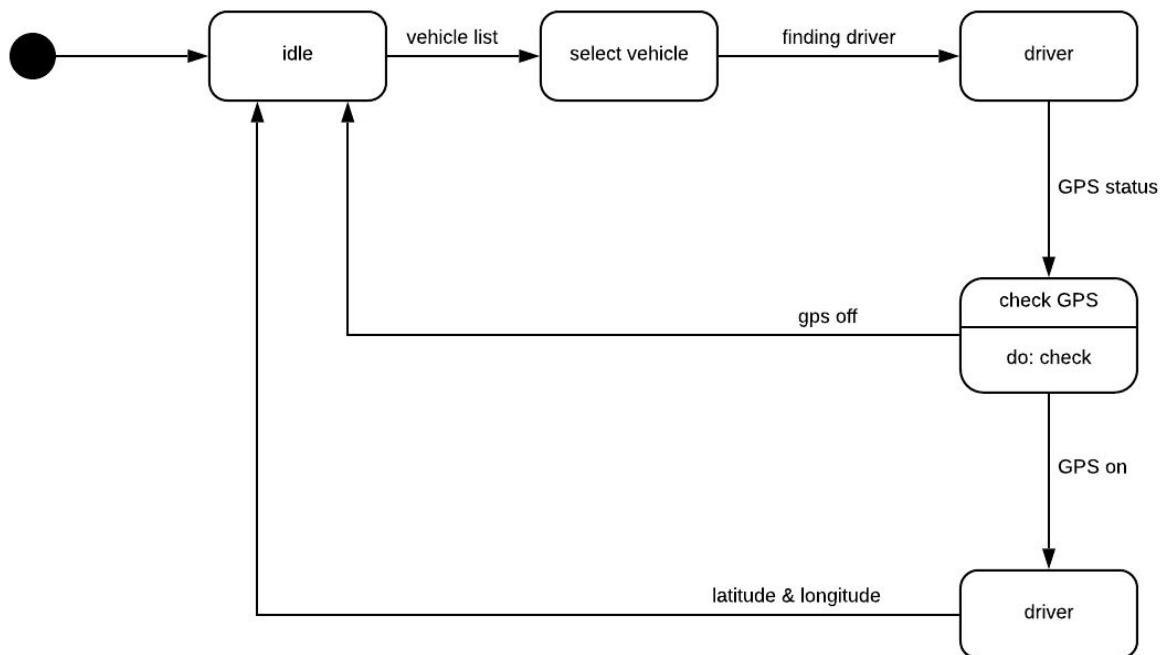


Figure 72: State diagram (Tracking)

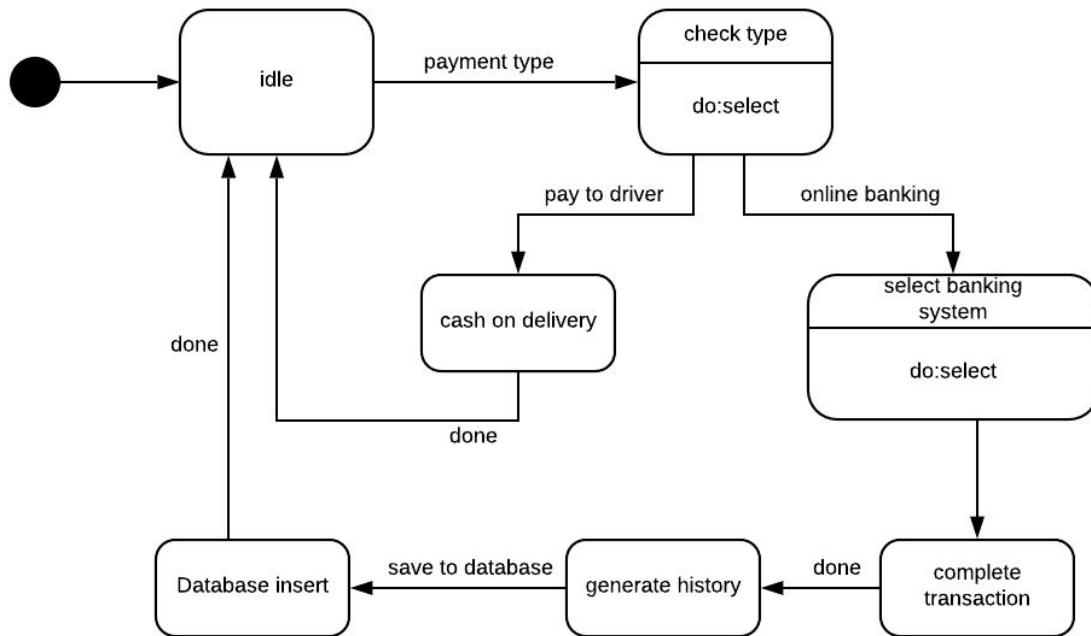


Figure 73: State diagram (Payment)

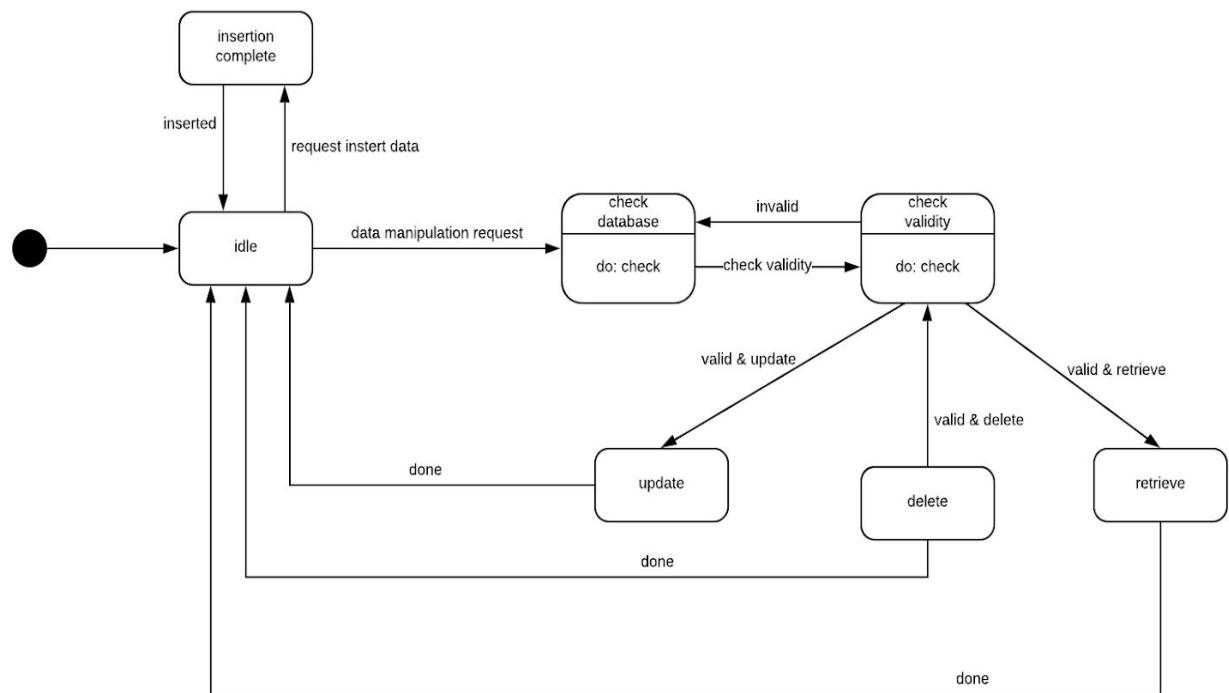


Figure 74: State diagram (Database)

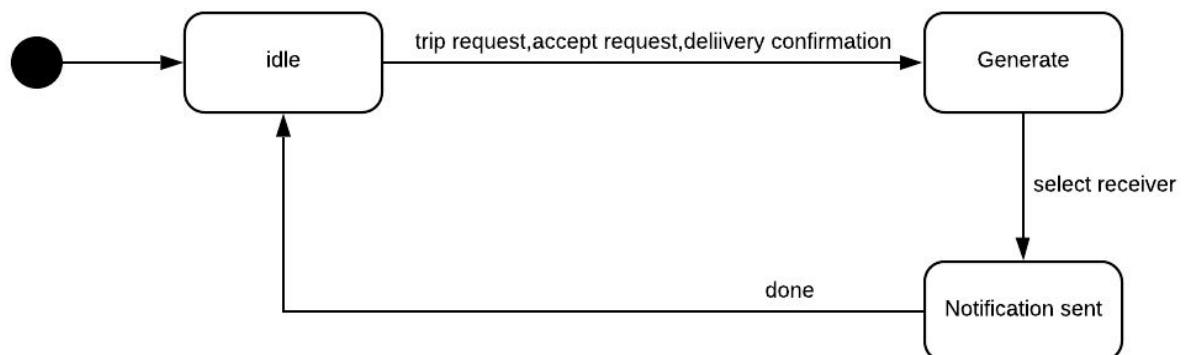


Figure 75: State diagram (Notification)

8.2 Sequence Diagram

Sequence diagram indicates how events cause transitions from object to object.

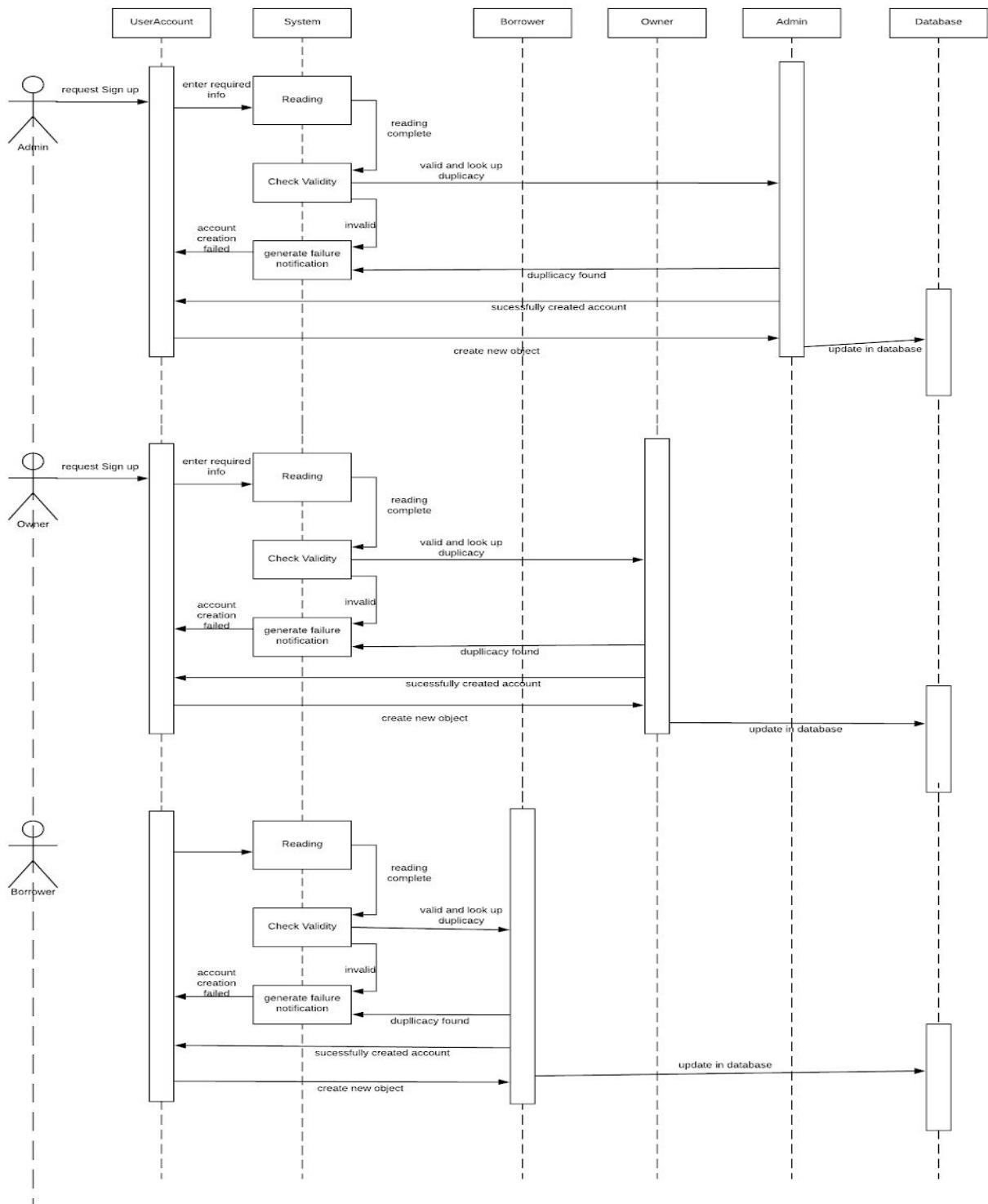


Figure 76: sequence diagram (sign up)

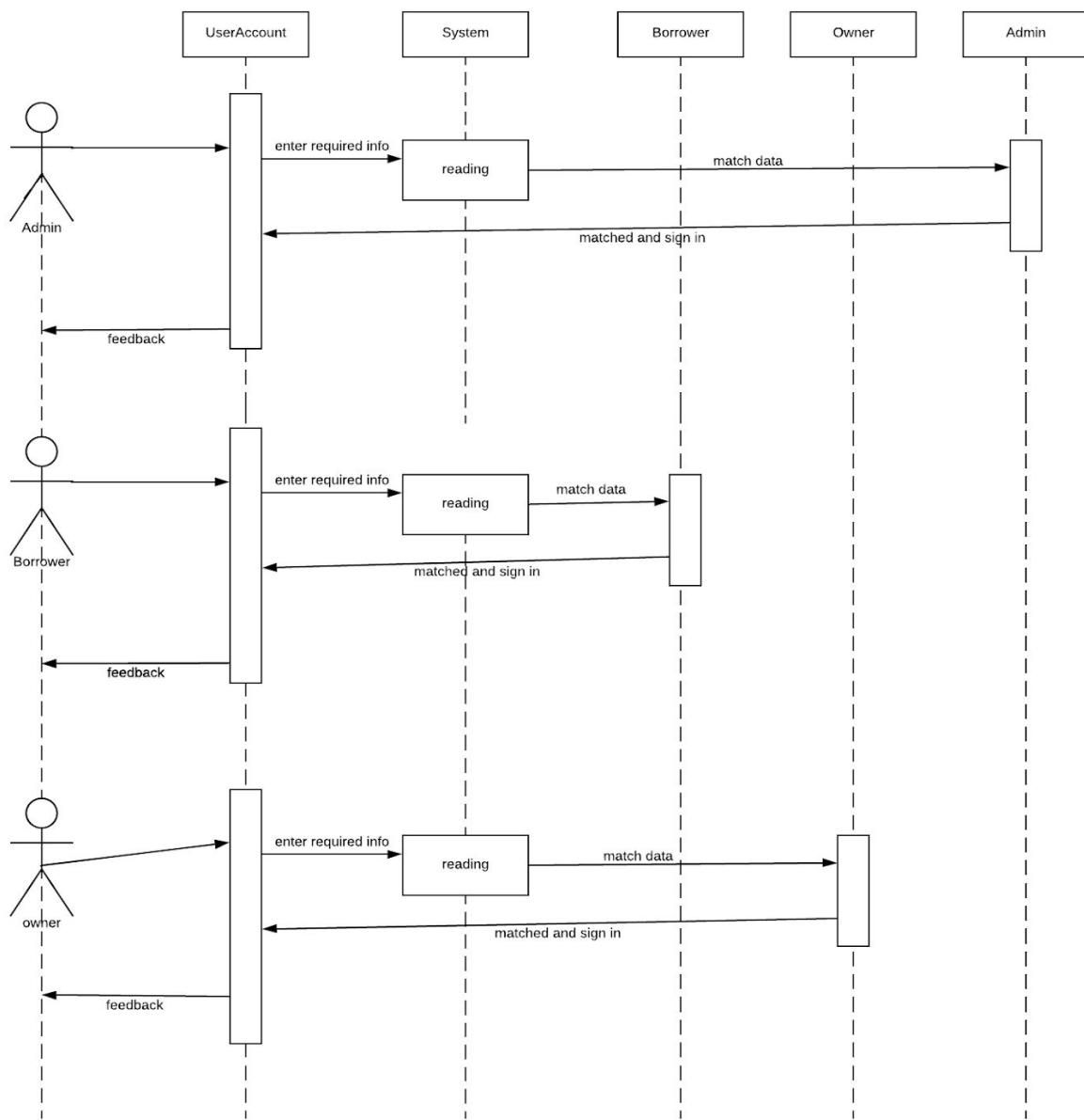


Figure 77 : sequence diagram (sign in)

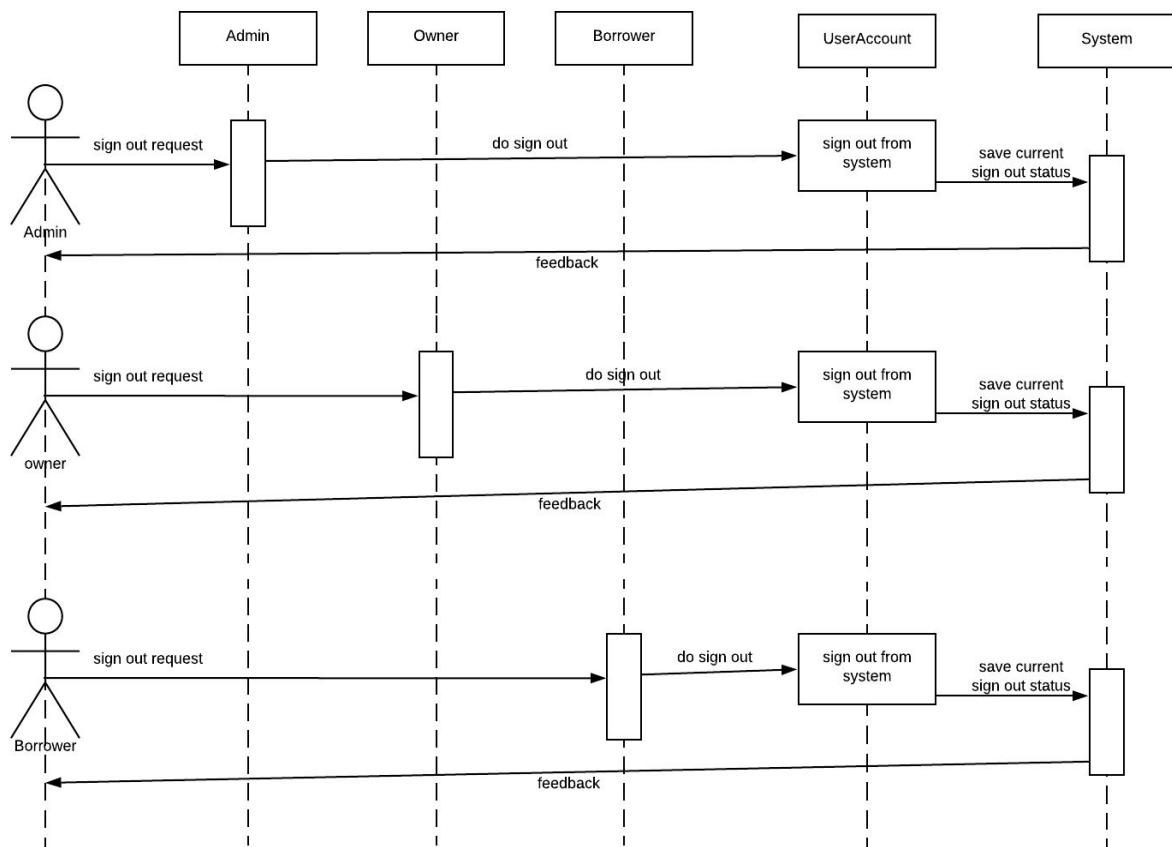


Figure 78: sequence diagram (sign out)

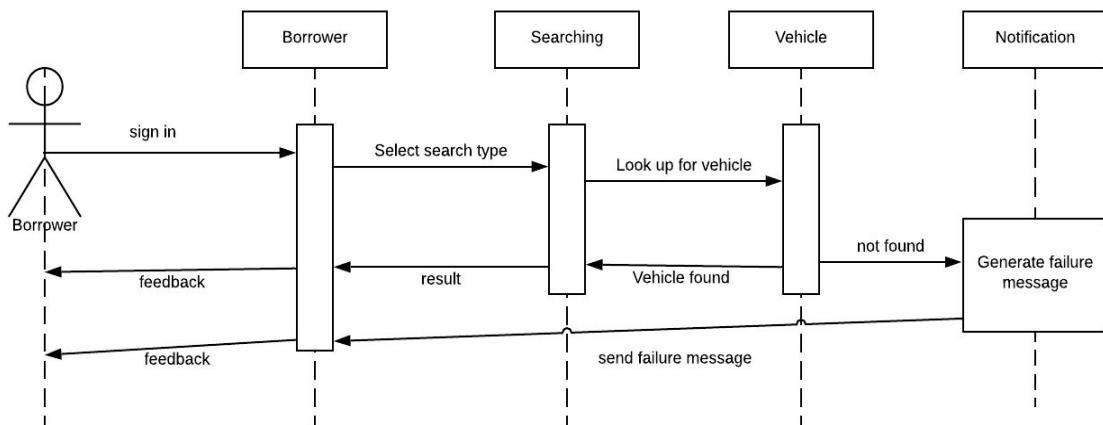


Figure 79: sequence diagram (searching)

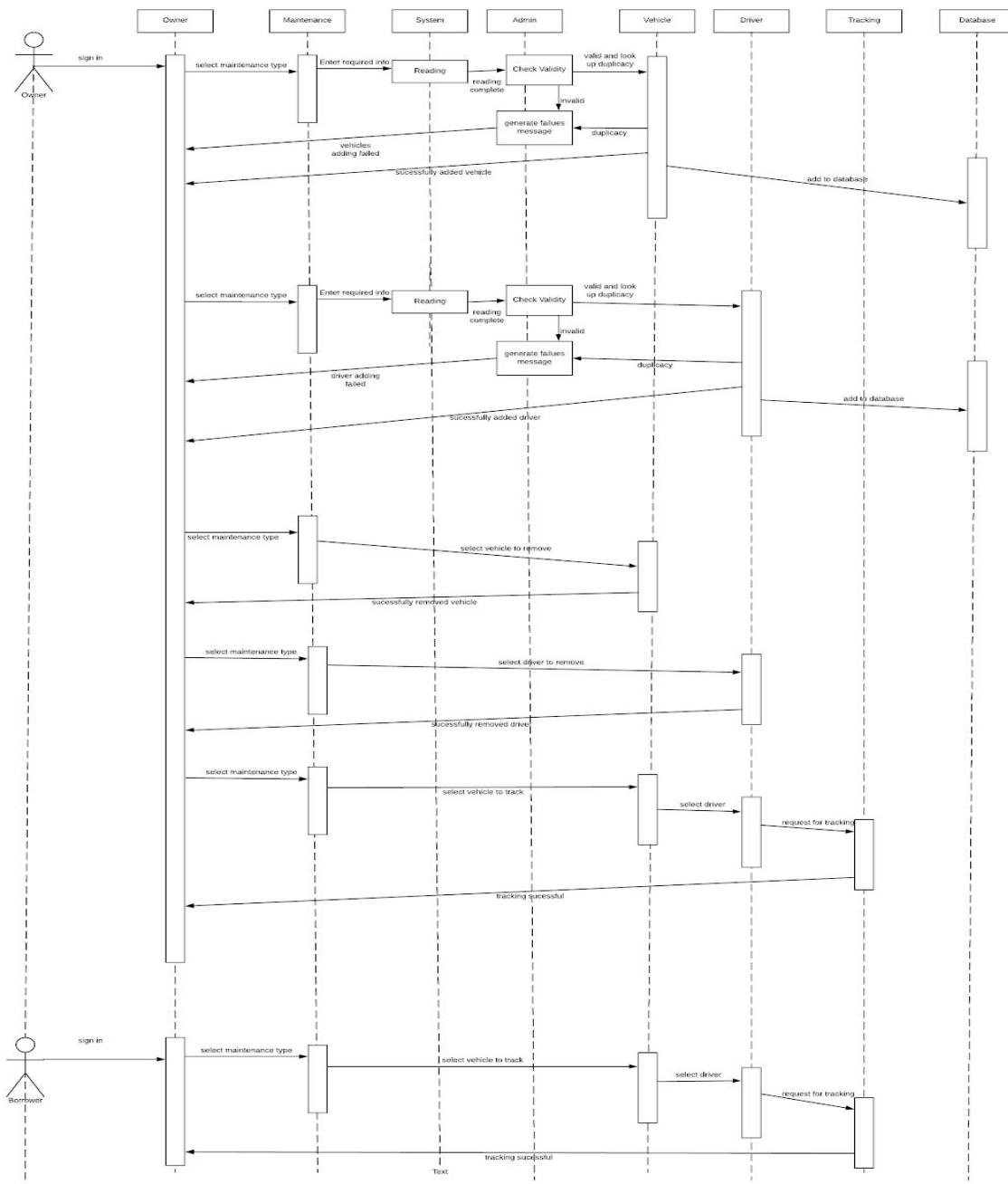


Figure 80: sequence diagram (maintenance)

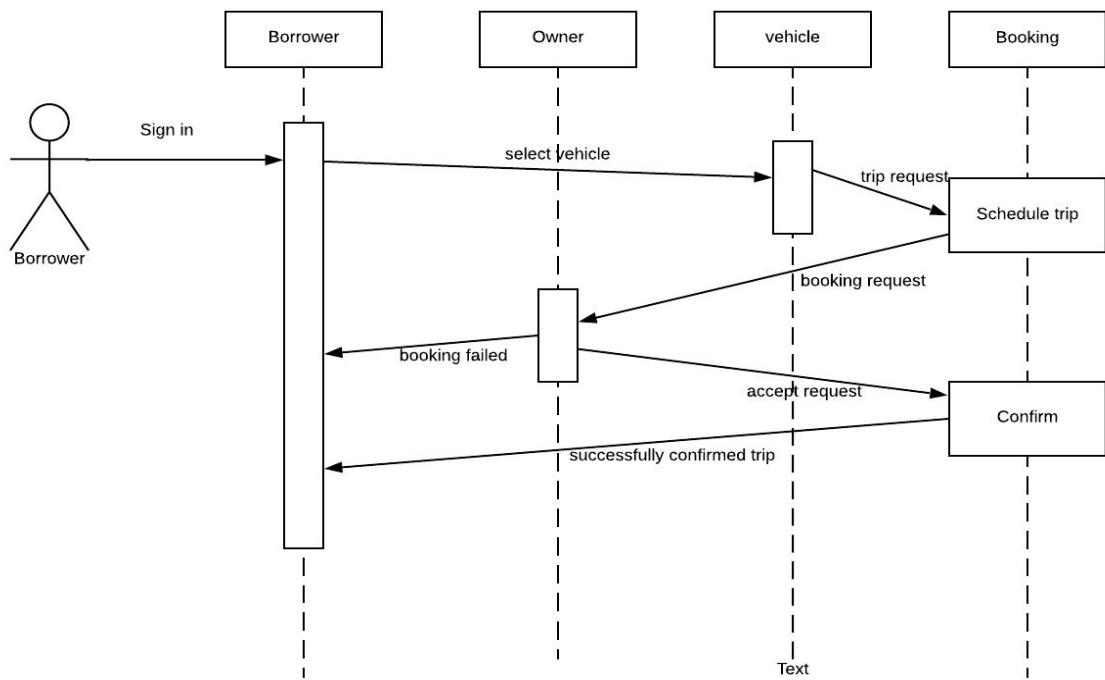


Figure 81: sequence diagram (booking)

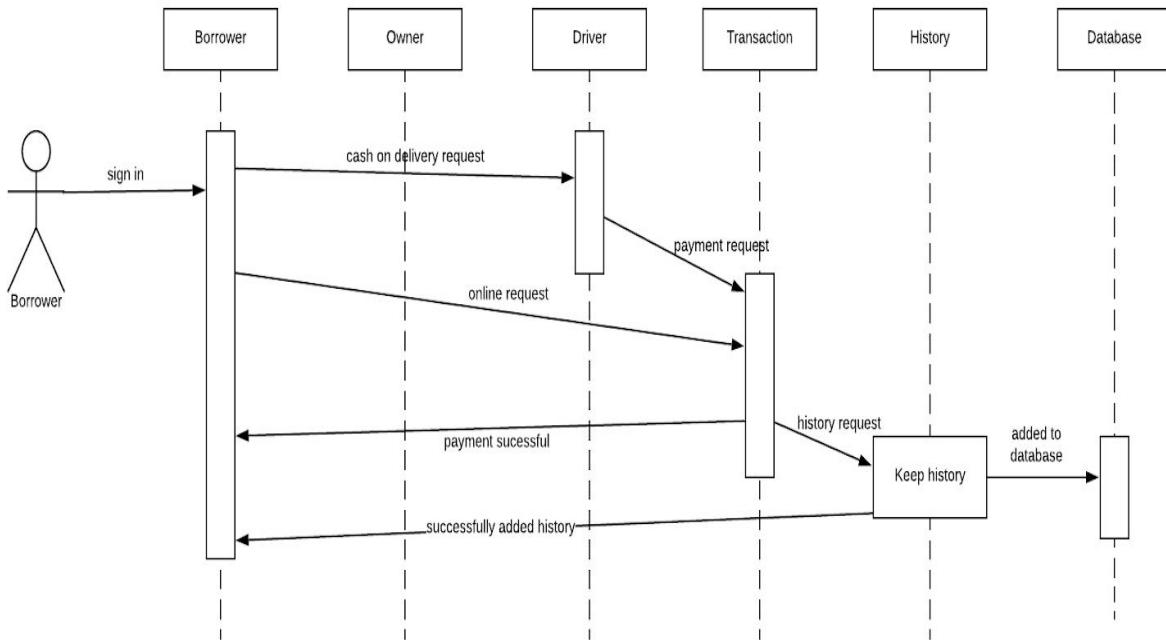


Figure 82: sequence diagram (payment)

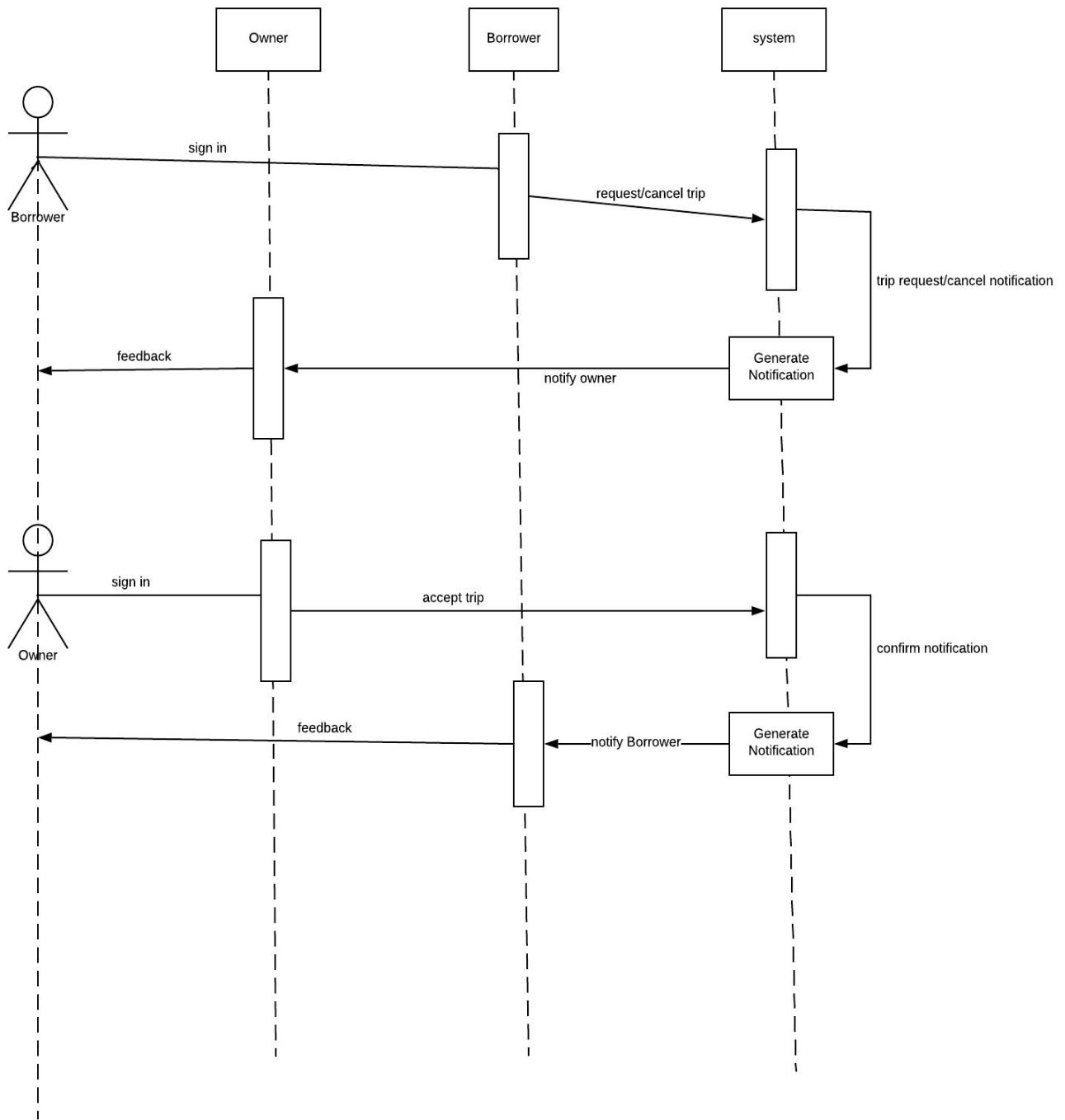


Figure 83: sequence diagram (notification)

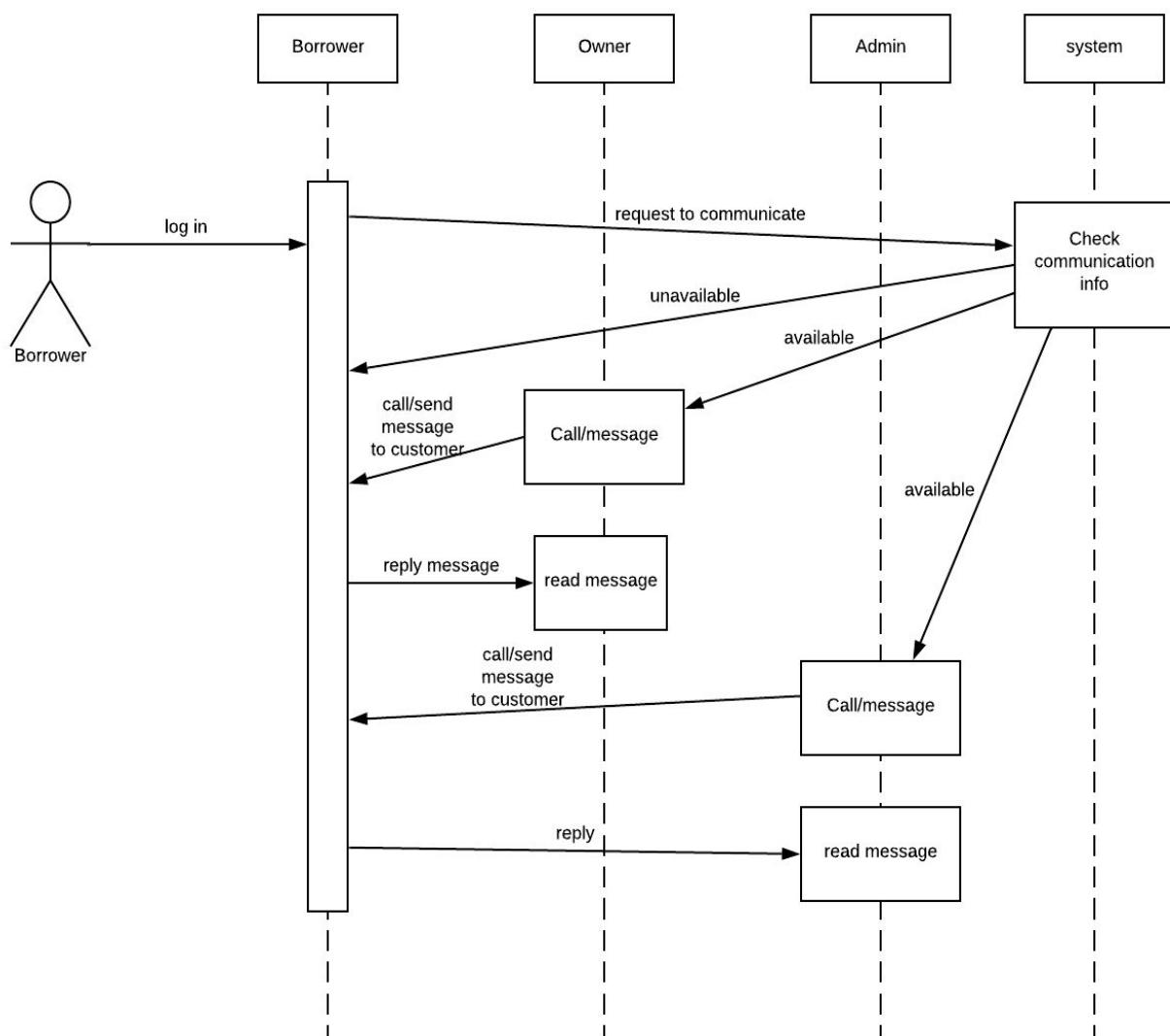


Figure 84: sequence diagram (communication)

Chapter 9: Conclusion

From this SRS report on Vehicle Renting and Management System, the readers will get a clear and easy view of the overall Management system. This SRS document can be used effectively to maintain the software development cycle. It will be very easy to conduct the whole project using SRS. We tried best to remove all dependencies and make an effective and fully designed SRS.

Meeting Schedule

Meeting 1

Date: 15-01-2019

Location: IIT, DU

Discussion:

- Introducing with each other in the group.
- Discussing about the basic of our project topic

Members:

Niraj Chaudhary BSSE 0942

Md. Musa Ali BSSE 0943

Meeting 2

Date: 22-01-2019

Location: IIT, DU

Discussion:

- Discussion with our coordinator regarding project information and stakeholder information
- Selecting stakeholders to visit for the project

Members:

Niraj Chaudhary BSSE 0942

Md. Musa Ali BSSE 0943

Meeting 3

Date: 30-01-2019

Location: DOHS, Mirpur-12

Discussion:

- Discussion with borrowers regarding project information
- Collecting requirement information from borrower side
- Discussing on how company manage day to day trip, transaction, trustworthy,

Members:

Niraj Chaudhary BSSE 0942

Md. Musa Ali BSSE 0943

Meeting 4

Date: 05-02-2019

Location: Banga Bazar

Discussion:

- Discussion with owner regarding project information
- Discussion related on payment, delay

Members:

Niraj Chaudhary BSSE 0942

Md. Musa Ali BSSE 0943

Meeting 5

Date: 10-02-2019

Location: Chairmanbari, Bonani

Discussion:

- Discussion with truck agencies regarding project information
- Discussion related on day to day trip, import/export charge, driver management, goods responsibility, truck capacity
- Offering extra additional features.

Members:

Niraj Chaudhary BSSE 0942

Md. Musa Ali BSSE 0943

Meeting 6

Date: 17-02-2019

Location: IIT, DU

Discussion:

- Finalizing the common requirements collected through survey

Members:

Niraj Chaudhary BSSE 0942

Md. Musa Ali BSSE 0943

Meeting 7

Date: 21-02-2019

Location: IIT, DU

Discussion:

- Focusing on the needs of stakeholders
- Discussing about the easiness and features they may expect in an automated system to deal with customers.

Members:

Niraj Chaudhary BSSE 0942

Md. Musa Ali BSSE 0943

Meeting 8

Date: 24-02-2019

Location: IIT, DU

Discussion:

- Identifying the sub system
- Specifying the use case scenarios
- Translating use case diagram

Members:

Niraj Chaudhary BSSE 0942

Md. Musa Ali BSSE 0943

Meeting 9

Date: 27-02-2019

Location: IIT, DU

Discussion:

- Activity diagram and swimlane diagram

Members:

Niraj Chaudhary BSSE 0942

Md. Musa Ali BSSE 0943

Meeting 10

Date: 30-02-2019

Location: IIT, DU

Discussion:

- Identified potential entities
- Finalizing entities and their attributes
- Identifying relationship between them

Members:

Niraj Chaudhary BSSE 0942

Md. Musa Ali BSSE 0943

Meeting 11

Date: 04-03-2019

Location: Sir P.J Hartog International Hall, DU

Discussion:

- Noun Identification
- Performing selection criteria
- Identifying potential classes
- Identifying associated nouns and verbs of each potential classes
- Identifying attributes of each potential classes
- Identifying methods of each potential classes
- Finalizing classes for the entire system
- Constructing class cards

• **Members:**

Niraj Chaudhary BSSE 0942

Md. Musa Ali BSSE 0943

Meeting 12

Date: 09-03-2019

Location: Amar Ekushey Hall, DU

Discussion:

- Flow-oriented Modeling
- Data Flow Diagram

Members:

Niraj Chaudhary BSSE 0942

Md. Musa Ali BSSE 0943

Meeting 13

Date: 13-03-2019

Location: Amar Ekushey Hall, DU

Discussion:

- Behavioral Model
- State Diagram
- Sequence Diagram

Members:

Niraj Chaudhary BSSE 0942

Md. Musa Ali BSSE 0943

Meeting 14

Date: 19-03-2019

Location: Sir P.J Hartog International Hall, DU

Discussion:

- Finalising SRS report

Members:

Niraj Chaudhary BSSE 0942

Md. Musa Ali BSSE 0943