REQUIREMENTS SPECIFICATIONS DOCUMENT

FACULTY OF ECONOMICS AND ADMINISTRATIVE SCIENCE

BUSINESS INFORMATICS

CEN 302- SOFTWARE ENGINEERING

Tiranë

April 2019

Software Documentation by:

Alisa Doçi (adoci16@epoka.edu.al)

Amelia Bullari (abullari16@epoka.edu.al)

Besnik Sulmataj (bsulmataj16@epoka.edu.al)

Klaus Çepani (kcepani16@epoka.edu.al)

Klementina Idrizi (kidrizi16@epoka.edu.al)

Received by:

Igli Hakrama

TABLE OF CONTENTS

1. Executive Summary

1.1 Project Overview

This project consists of giving an idea and providing a solution to minimize cost and time for big construction companies, such as “Salillari Co”, which give earth moving heavy equipment for construction.

By creating a software in which we are going to analyze every economic and programing detail, we tend to achieve a successful software application to be used by the company as the most efficient way to handle their business affair for renting earth-moving heavy equipment for construction.

The project is going to provide every economic information needed, that said the project is going to include the number of workers in the company, the cost of every equipment that the company has, the daily price for the equipment and also the packet price for renting them. In the project we are also thinking to add a register, in which every data about the small companies that rent equipment from our company is provided. This solution is going to minimize time, because a company that has rented before does not have to repeat itself giving information when they want to rent again from us.

1.2 Purpose and Scope of this Specification

The purpose of this specification is the easy access of the products that are for rent and/or sale for the customers and also provide a better way for the managers of the company to control the products, employers and customers.

This specification encompasses several aspects of the process being discussed in as broad scope as possible. The following lines are addressed in the scope:

- In depth documentation of the features of the product.
- Details provided about the cost and the information needed for the produc.t
- Components & Functional/non-functional requirements.

- Definition of users' means of using and accessing the product.
- Dependencies and Constraints.

Some aspects that are not included in the scope are a:

- The legislative requirements for the product.
- Auditing and financial considerations of the product provided.

2. Product/Service Description

2.1 Product Context

The main service of the company is renting the earth-moving heavy equipment for construction purposes. Some of the products are: excavators, backhoe loaders, bulldozers, skid-steer loaders, motor graders, crawler loaders, trenchers and scrapers. As a big company, Salillari has enough equipment to rent to small companies. This items are going to be rented daily, weekly and monthly. Every equipment has its own rent price and when combined with other equipment may have a discounted price.

The software aims to help every step of the process of renting and/or buying earth-moving heave equipment for construction companies or individuals.

2.1 User Characteristics

Our customers include the entire staff of the company and also the guests (manager, economist, employee and guests(clients)):

- Administrator / Manager / Financier
- Employees
- Clients / Guests

2.2 Assumptions

The software is going to be implemented in a website. That means that every one of the users that are going to use the software will have to sign up in the software to be able to use the software.

It is assumed that the guests have knowledge on using a website and signing up.

2.3 Constraints

This system will be potentially constrained by:

- Security concerns on data accessibility.
- The need for a good internet connection.

2.4 Dependencies

Dependencies that affect the requirements:

This product is dependent on a successful communication between the guests and the managers or administrators of the company.

The software is dependent on the users' Internet connection.

3. Requirements

3.1 Functional Requirements

Functional Requirements for this application are stated as below:

Req_Code	Requirement	Comments	Priority	Date	SME Reviewed /Approved
FR_01	The software shall have different views for different user levels.	The views for guests, employees and administrator should all be different and specific to their functions.	1	25/03/2019	Alisa Doci/ Klaus Cepani
FR_02	All accounts, for each level of user, shall be secured with a password.	The password will be stored in the database in a hashed form, in order to protect the ethics.	1	26/03/2019	Alisa Doci/ Klaus Cepani
FR_03	Each user will have an unique username.	No to users will have the same username in order to avoid ambiguity.	1	25/03/2019	Alisa Doci
FR_04	To protect the system from spam and abuse,	To prove that you are a human being you must fill the reCaptcha.	3	25/03/2019	Alisa Doci/ Klementina Idrizi

	a reCaptcha shall be used.				
FR_05	The application shall feature a Register page.	This will be the page for users to register.	1	25/03/2019	Alisa Doci
FR_06	The application shall feature an Information Page.	This page will show all the information about the company, products and the application. The Information Page will be visible for everyone.	3	25/03/2019	Alisa Doci/ Klaus Cepani
FR_07	The application shall feature a Contact Us page.	Through this page the user can give feedback.	2	25/03/2019	Alisa Doci
FR_08	After the registration, users can change their passwords, but not their usernames.	The username you firstly choose, will remain unchanged.	3	25/03/2019	Alisa Doci/ Amelia Bullari
FR_09	Each level of user's view shall have Home Page.	The home page shall contain all the functions of the user.	3	25/03/2019	Alisa Doci

FR_10	Admin's View shall have a Finance option, where he will be able to see all the profits.	Admin will be the only user who will have access to the Finance Page.	2	25/03/2019	Alisa Doci/ Besnik Sulmataj
FR_11	Admin's View shall have a Reports option, where he will be able to see detailed statistics of the system.	Admin will be the only user who will have access to the Reports Page.	2	26/03/2019	Alisa Doci
FR_12	The system shall be able to generate weekly reports on sales.	This reports will be expressed in in tables and/or charts.	2	26/03/2019	Alisa Doci/ Klementina Idrizi
FR_13	The system shall be able to generate user and employees rating lists.	Users will be rated according to their activity/ purchase of products.	3	26/03/2019	Alisa Doci
FR_14	Admin shall be able to add new Employees.	Admin will be the only user to possess this right.	1	25/03/2019	Alisa Doci/ Klaus Cepani
FR_15	Admin shall be able to edit Employees Information.	Admin will be the only user to possess this right.	1	26/03/2019	Alisa Doci/Amelia Bullari

FR_16	Admin shall be able to delete Employees.	Admin will be the only user to possess this right.	2	27/03/2019	Alisa Doci/ Klementina Idrizi
FR_17	Admin shall have his/her own profile page.	On this page they will be able to edit their own personal information.	2	25/03/2019	Alisa Doci/ Besnik Sulmataj
FR_18	The Customer's View will offer a catalog of machineries and equipment they may choose to rent.	This catalog with will be as user friendly as possible.	1	26/03/2019	Alisa Doci
FR_19	After choosing the Vehicle they need, they shall be able to reserve it for rent.	Customers can reserve Vehicles to rent later.	2	26/03/2019	Alisa Doci
FR_20	The User's View shall have a Payment option.	This page will show the total amount they will need to pay.	2	26/03/2019	Alisa Doci/ Klementina Idrizi
FR_21	The Payment page shall feature a card information form.	The user must fill in their credit card information if they wish to pay online.	2	26/03/2019	Alisa Doci

BR_22	The User's View shall have an History option.	This page will show the user's previous activities.	3	25/03/2019	Alisa Doci/ Amelia Bullari Klaus Cepani
BR_23	Every Customer shall have their own profile page.	On this page they will be able to edit their own personal information..	2	25/03/2019	Alisa Doci/ Besnik Sulmataj
FR_24	Customers shall have a Feedback Page.	They can live comments/reviews/ suggestions.	3	25/03/2019	Alisa Doci
FR_25	The Employee's View shall feature a Rentals Option.	This page will show the rental deal that are on process.	3	25/03/2019	Alisa Doci/ Klaus Cepani
FR_26	The Employee's View shall feature a Payday Option.	This page will show the amount of money the employee earned that week and when he shall be able to receive it.	3	25/03/2019	Alisa Doci/ Amelia Bullari
FR_27	Employee shall be able to add new Vehicles.	When new equipment is available, Employee must register them.	1	25/03/2019	Alisa Doci
FR_28	Employee shall be able to edit Vehicles.	When changes happen to the equipment,	1	25/03/2019	Alisa Doci

		Employee must register them.			
FR_29	Employee shall be able to delete Vehicles.	When equipment is no longer available, Employee must delete them.	1	25/03/2019	Alisa Doci
FR_30	Employee shall be able to approve Reservations.	When Customers make reservations, Employee must approve or deny base on availability of the Vehicles.	1	25/03/2019	Alisa Doci
FR_31	Employee shall be able to reply to Customer Feedback.	Employee shall be able to view and reply to Customer's reviews.	3	26/03/2019	Alisa Doci
FR_32	Employee shall have his/her own Profile Page.	They can add or edit their personal information.	3	25/03/2019	Alisa Doci
BR_33	The Employee' View shall feature an History option.	This page will show the employee's previous commitments.	3	25/03/2019	Alisa Doci/ Klementina Idrizi
FR_34	The application interface must be stylish and functional in different devices.	The interface's design must change for different screen sizes.	2	25/03/2019	Alisa Doci

3.2 Non-Functional Requirements

3.2.1 User Interface Requirements

The user interface of the software application will be designed to be compatible with any browser, including Chrome, Mozilla, Safari and Android. We will focus on anticipating what users might need to do and will ensure that the interface has elements that are easy to access, understand, and use to facilitate those actions, while still keeping the application's design fresh and stylish. The software will offer four main interfaces: Admin, Employee, User and Guest.

The Guest Interface shall be offered without the need to log in. As a Guest you will be able to access Home Page, Information Page and Contact Us Page.

Admin Interface shall include:

- Finance button.
- Reports button
- Edit/Modify button.
- Projects button.

Employee Interface shall include:

- Schedule button
- Payday button
- Edit profile button

User Interface shall include:

- View catalogs button.
- Payment button.
- Card Information button.
- History button.
- Edit profile button.

3.2.2 Usability

In software engineering, usability is the degree to which a software can be used by consumers to achieve objectives with effectiveness, efficiency, and satisfaction.

For our system to achieve usability it must be:

- Easy to learn
- Efficient to use
- Easy to remember
- Few errors
- Subjectively pleasing

The main aim is to produce a software which must be easy and practical to use. Administrator, employees and users shall find it very comfortable using the software on any device, including their smartphones and tablets, and all it will take are some very basic computer experience.

3.2.3 Learnability

Learnability is one of the attributes of a software product that contribute to general usability.

- The software shall be easy to learn and even intuitive to some degree, because the user will be able immediately grasp how to interact with the system.
- There shall me messages and instructions that will help users to easily complete each task.
- Icons and Manu Bars will facilitate the use of the software application.
- The application's interfaces shall use images and colors to further facilitate the usage for every type of user.

3.2.4 Accessibility

Taking into consideration the user-friendly interface, everything shall be easy to be accessed by different user groups. Every user level shall have their own page with different panels. The software tends to simplify as much as possible accessibility for slightly disabled users.

3.2.5 Memorability

Within the context of usability and interface design, memorability refers to the user's ability to leave a program and then remember how to use it whenever he chooses to return to it.

We are mostly going to use symbols, icons, and other visual presentation types that allows the user to make a free association with the task at hand. For example, the "Home" icon on most browser is a little house. The reason for this is that people assume that this house is their home. Thus the home concept is communicated to them via a visual cue. The same idea will be used in the design of the product in order to make the usage of the software as simple and memorable as possible.

3.2.6 Performance

When we speak about a software's performance, we need to take into consideration three very important factors: response time, workload and platform.

The tasks for each user shall be processed within a few milliseconds, a few seconds in the worst case. So, for the performance to be optimal, the response time shall be minimal.

Workload refers to the capability of the software to handle the maximum number of users interacting with the system at the same time. A good server is needed to support a lot of traffic in the system.

Lastly, a platform is defined as a combination of both the hardware and the software that will host the system. The platform must be able to support the system's functionality.

3.2.7 Capacity

This application will be developed to cover and support all the steps of the process of renting and/or buying earth-moving heave equipment for construction companies or individuals. The basic idea is very similar to that of an online shop. The application shall be able work at the same time and on real time for all available users.

All users will use the same database, therefore if too many requests are done at the same time, a slight delay might occur. The starting time of the software will be in no time and for all the other requests delays might depend on the user's internet connection strength/speed. The application shall be stored on the Web Server.

3.2.8 Availability

The application shall operate 24 hours a day, seven days a week and it shall be available to anyone who owns or has access to a digital device, be it a personal computer, lap top, tablet or smartphone. The software shall be compatible with every browser, such as Chrome, Mozilla, Safari etc.

The software shall not encounter downtime since the data used shall be very sensitive and time-varying. Also, the regular maintenance shall not affect the functionalities of the web site.

3.2.9 Monitoring

For all errors that might occur, the software application shall be able to provide the users and the administrators with the appropriate feedback, this way the maintenance group shall be able to fix the problem as soon as possible. Also, the maintains group shall conduct periodic

reports and that information shall be used to improve the application system. The system will be developed to be secure and reliable.

3.2.10 Maintenance

- The main platforms that will be used to maintain the application are MySQL for the database and the APACHE server.
- The application will be developed in a way that will make it easily extendable. This means that new features can be effortlessly implemented in the existing software.
- The application shall use logs and text files in order to maintain as much information as possible in case that something unexpected happens.

3.2.11 Operations

The main operations for this software application shall be:

- It must be available 24/7.
- Each user shall be able to login in the system.
- Each user shall have access to their information after being logged in.
- The sensitive information entering the system shall be secured.
- The sensate information shall be accessed only by authorized users.
- Only authorized users shall possess the right to edit/delete/modify information.

3.2.12 System Interface/Integration

The database management system is one of the most important parts of this software application. The whole program will have an admin that will see every logged in person and what he requires. The use who will be logged in shall be able to see if the machineries, they are interested in, are free or not. In case the equipment is occupied, they shall be notified about the

future time when the machine will be available for rent or purchase. Also, the management of the staff, will be available through the use of the database created.

3.2.13 Network and Hardware Interfaces

The application being a web application needs to be stored in a web server, so that the browser user agent would be able to create a TCP connection with server. The server should function with CentOS web server. The Doctrine ORM, which deals with databases needs to get installed for proper functionality.

3.2.14 Security

- Rules are set to allow only authenticated/authorized users to full read and write access rights to the system's database.
- To protect the database from abuse until it has time to customize the software rules or set up authentication.
- Security provided to the access of user private information.
- Security provided to access of data.
- Restriction provided to access of data.

3.2.15 Protection

Matching function are checked when adding a new employee or a new client. The specified functions are checked:

- A function that will check for a valid name.
- A function that will check for a valid email address.
- A function that will check for valid password.
- A function that will check for a valid firm.

3.2.16 Authorization and Authentication

In this part of the software this kind of features are added:

- A function that checks for valid reports.
- A function for entering new employees.
- Provide list of employees completing a specific report.
- List reports based on some criteria.

3.2.17 Data Management

- The main platforms that will be used are MySQL for database and the APACHE server.
- Other platforms being considered are Cloud Firestore and Realtime Database, which we will primarily use are NoSQL databases. NoSQL stands for “non SQL”, but also “not only SQL”, and it is a non-relational database, which stores data in a large file usually. It offers more flexibility in database design, which does not restrict us in the relations between entities. It also scales better horizontally, meaning that it is better distributed in a more efficient network of nodes.
- Data shall be saved in a JSON tree. This offers easy encoding and decoding directly from Java objects in Android.
- In Cloud Firestore, differently from the Realtime Database, data is stored in Documents and Collections, thus offering better query support, and an offline first approach.

3.2.18 Standards Compliance

Web site agreement is going to be a legal agreement that will protect user information’s and will not share and use them. Also will notify the user in case of new material or machinery available. The software will be designed so that requests can be filed at any hour. This will enable the employees, users and companies to use the system more effectively and freely, without being restricted to the Business working hours.

3.2.19 Portability

The main idea of the web application is to be used by everyone. The companies working with us that have the ability to rent or by construction material have are also technological equipped to support high connection internet and to have a good browser like Google Chrome. Our web application requires only a minimum amount of graphic card and an internet connection as mentioned above this is not a problem. The software will be able to run on every browser that the user might want to use

4. UML Diagrams

4.1 Use Cases

Use-case name	Login	
Actor	Employee, Customer and Admin	
Description	This use case describes how the employee, customer and admin to login in the system	
Precondition	The staff, customer and admin must be registered on the system	
Post-condition	The authenticated person gets the appropriate page	
Basic course of action	**User Action**	**System Response**
	1. Select the login link 3. Fill username and password, click on login button	2. The system displays the login form 4.The system verifies that all the filled have been filled out and valid. 5. The system display the page 6. Use case Exit
Alternate course of action	If in step 4, all fields are not filled properly and/or do not match, the system notifies the actor with a message. Go back or return to step 4 of basic course of action to enter again.	

Use-case name	Customer Creates Account	
Actor	Customer	
Description	This use case describes customer can create an account for his/her self.	
Precondition	The customer must have accessed the webpage and clicked “Create Account”.	
Post-condition	The customer can use the services the system offers.	
Basic course of action	**User Action**	**System Response**
	1. Select the Create Account link 3. Fill the personal information input boxes and click Register button or link.	2. The system displays the register form 4. The system verifies that all the fields have been filled out and are valid. 5. The system display the welcome page 6. Use case Exit
Alternate course of action	If in step 4, all fields are not filled properly and/or do not match, the system notifies the actor with a message. Go back or return to step 4 of basic course of action to enter again.	

Use-case name	Reserve Vehicle	
Actor	Customer	
Description	This use case permits customers to reserve vehicles for renting, based on the availability of the vehicle	
Precondition	Customer wants to reserve a vehicle and reservation of the customer have to be entered.	
Post-condition	Customer reserves successfully	
Basic course of action	**User Action**	**System Response**
	1.The customer wants to reserve a Vehicle. 2.The customer clicks Reservation Page. 4.The customer enters the necessary information. 6. The customer clicks reserve button. 8. The customer accepts the reservation and clicks Accept.	3. The system displays the Login form 5. The system verifies that all the filled have been filled out and valid. 7. The system display the page. 9. Use case Exit
Alternate course of action	If all fields are not filled out the system goes back or returns to step 4 of basic course of action.	

Use-case name	Feedback	
Actor	Customer	
Description	This use case permits customer to give feedback	
Precondition	None	
Post-condition	Send feedback	
Basic course of Action	**User Action**	**System Response**
	1. The customers click on feedback link. 3. The customers give feedback with their details.	2. The system displays form. 4. Then the system checks the field are filled. 5.Feedback is successfully sent. 6. Use case Exit..
Alternate course of action	If fields are not filled correctly the customer goes back or returns to step 3 of basic course of action	

Use-case name	Customer Returns Vehicle	
Actor	Customer, Employee	
Description	This use case permits Employees to mark Vehicles as Available after Return.	
Precondition	Customer returns the rented Vehicle.	
Post-condition	Vehicle is Available for other customers.	
Basic course of action	**User Action**	**System Response**
	1.The customer wants to return a Vehicle. 3.The employee clicks Vehicle Page. 4.The employee marks Vehicle as available.	2. The system displays the Vehicle information. 5. The system verifies and makes the changes into the database. 6. Use case Exit
Alternate course of action	No alternate course of action.	

Use-case name	Make Payment	
Actor	Customer	
Description	This use case permits customers to pay online for the service.	
Precondition	Customer wants to make the payment for the rental.	
Post-condition	Payment received Successfully.	
Basic course of action	**User Action**	**System Response**
	1.The customer wants to make the payment. 2.The customer clicks Payment Page. 4.The customer enters the necessary information. 5. The customer clicks Make Payment button.	2. The system displays the Payment detail form. 4. The system verifies that all the filled have been filled out and are valid. 5. The system display the page. 6. Use case Exit
Alternate course of action	If all fields are not filled out the system goes back or returns to step 4 of basic course of action.	

Use-case name	Cancel a reservation	
Actor	Customer	
Description	This use case permits a customer to cancel a reservation	
Precondition	Customer already has reserved and wants to cancel the reservation	
Post-condition	Customer successfully cancel a vehicle	
Basic course of Action	**User Action**	**System Response**
	1. The customer wants to cancel reservation. 2.The customer opens reservation page and clicks Cancel reservation link. 4. The customer enters reservation confirmation number and clicks Cancel Reservation button. 6. Clicks Yes when asked "Are you sure you want to cancel Reservation?"	3. The system displays a form. 5. The system verifies the field has been filled out correctly and checks validity of the confirmation number. 7. The system cancels the reservation and display a message, showing that the reservation is canceled. 8. Use case exit.
Alternate course of action	If the customer enters an invalid number, the system goes back or returns to step 4 of the basic course of action. If the customer clicks "NO" reservation cancelation will be terminated.	

Use-case name	Customer Deletes Account	
Actor	Customer	
Description	This use case permits Customer to delete their account.	
Precondition	The customer must have entered the webpage and clicked "Delete Account".	
Post-condition	The customer can no longer use the services the system offers.	
Basic course of action	**User Action**	**System Response**
	1. Log in and select the Delete Account link. 3. Clicks Yes when asked whether he/she would like to delete the account.	2. The system shows a warning message. 4. The system 5. The system deletes Customer from Database. 6. Use case Exit.
Alternate course of action	If in step 3, user clicks No, the case terminates.	

Use-case name	Process Rent	
Actor	Employee	
Description	This use case permits to register rental information of the customers and the vehicle that the customer rents.	
Precondition	Login	
Post-condition	Customer rent information	
Basic course of action	**User Action**	**System Response**
	1. The customer wants to take the reserved vehicle. 2. The employee open rent page. 4. The employee enters Full name, nationality, Country, City, Identification number, phone, and plate No. Daily price, rent date, return date, total rent day, total payment. 5. The employee clicks on rent button.	3. The system displays a form to be filled out for renting the vehicle. 4. The system prompts to enter the following information. 6. The system verifies that basic fields have been filled out. 7. The system displays successful rent summary. 8. Use case exit
Alternate course of action	If Full name, Nationality, country, city, ID, car number, daily price, rent date, return date and total payment are not filled correctly the system goes back to step 4.	

Use-case name	Vehicle registration	
Actor	Employee	
Description	This use case permits staff to register new vehicles to the system with detail descriptions about the vehicle such as model, brand and price per day.	
Precondition	New vehicle purchased	
Post-condition	New vehicle information stored successfully.	
Basic course of action	**User Action**	**System Response**
	1. The staff wants to add a new vehicle. 2. Goes to Add Vehicle form page. 4. The employee enters the proper following information in the form. 5. The staff clicks or presses on the save or insert button.	3. The system displays a form to be filled out for vehicle registration. 6. The system verifies that the fields have been filled out correctly. 7. The system displays a Successfully Stored message to the employee. 8. Use case exit
Alternate course of action	If all fields are not filled out correctly the system goes back or returns to step 4 of basic course of action. The customer needs to refill the invalid or the empty field.	

Use-case name	Edit Vehicle Information	
Actor	Employee	
Description	This use case permits staff to alter available vehicle information.	
Precondition	Vehicle has underwent some changes.	
Post-condition	Vehicle information stored successfully.	
Basic course of action	**User Action**	**System Response**
	1. The staff wants to edit vehicle information. 2.Chooses vehicle and goes to Edit Vehicle. 4. The employee enters the proper following information in the form and clicks Save.	3. The system displays a form to be filled out with vehicle information. 6. The system verifies that the fields have been filled out correctly. 7. The system displays a Successfully Stored message to the employee. 8. Use case exit
Alternate course of action	If all fields are not filled out correctly the system goes back or returns to step 4 of basic course of action. The employee needs to refill the invalid or the empty field.	

Use-case name	Delete Vehicle	
Actor	Employee	
Description	This use case permits staff to delete vehicle from database.	
Precondition	Vehicle has been sold or it is no longer available.	
Post-condition	Vehicle deleted successfully.	
Basic course of action	**User Action**	**System Response**
	1. The staff wants to delete vehicle. 2.Chooses vehicle and goes to Edit Vehicle. 4. The employee enters the proper following information in the form and clicks Save.	3. The system displays a form to be filled out with vehicle information. 6. The system verifies that the fields have been filled out correctly. 7. The system displays a Successfully Stored message to the employee. 8. Use case exit
Alternate course of action	If all fields are not filled out correctly the system goes back or returns to step 4 of basic course of action. The employee needs to refill the invalid or the empty field.	

Use-case name	Update rent	
Actor	Employee	
Description	This use case permits staff to update or modify rent information in case there is a need for editing.	
Precondition	Need to change information	
Post-condition	Successful update message	
Basic course of Action	**User Action**	**System Response**
	1. Employee wants to update rent 2. Goes to the rent page 3.Searches by unique attribute which is given to customer during rent. 5. The staff updates the information. 6. Click on update button.	4. The system displays the rent Information. 7. The system validates updated Information in to the database. 8. Exit use case.
Alternate course of action	If match is not found, go back to basic course of action 3. If the entered information is invalid the system goes back to basic course of action 5.	

Use-case name	Respond to Customer Feedback	
Actor	Employee	
Description	This use case permits Employee to view and reply to Customer Feedback.	
Precondition	None	
Post-condition	Send feedback	
Basic course of Action	**User Action**	**System Response**
	1. The Employee clicks on feedback link. 3. The Employee views customer feedback and clicks reply. 5. Employee fills in the Reply.	2. The system displays Customer Feedback. 4. Then the system displays a comment input box. 6. Message is successfully sent. 7. Use case Exit..
Alternate course of action	If the field is not filled correctly Employee goes back or returns to step 5 of basic course of action.	

Use-case name	Generate Transaction Report	
Actor	Admin	
Description	This use case allows admin to generate a Transaction report about the renting information.	
Precondition	Login	
Post-condition	Generate monthly transaction report information.	
Basic course of Action	**User Action**	**System Response**
	1. The admin wants to generate report. 2.The admin clicks Report Page and chooses Transaction Report. 5. Then on the rent page the admin specifies the month and then clicks on the generate button.	3. The system displays the Transaction Report. 7. Use case exit.
Alternate course of action	No alternate course of action.	

Use-case name	Generate Employee Report	
Actor	Admin	
Description	This use case allows admin to generate an Employee Report about the staff performance.	
Precondition	Login	
Post-condition	Generate monthly Employee report information.	
Basic course of Action	**User Action**	**System Response**
	1. The admin wants to generate a report. 2.The admin clicks Report Page and chooses Employee Report. 5. Then on the rent page the admin specifies the month and then clicks on the generate button.	3. The system displays the Employee Report. 7. Use case exit.
Alternate course of action	No alternate course of action.	

Use-case name	Generate Customer Report	
Actor	Admin	
Description	This use case allows admin to generate a Customer report about customer's activity.	
Precondition	Login	
Post-condition	Generate monthly transaction report information.	
Basic course of Action	**User Action**	**System Response**
	1. The admin wants to generate report. 2.The admin clicks Report Page and chooses Customer Report. 5. Then on the rent page the admin specifies the month and then clicks on the generate button.	3. The system displays the Customer Report. 7. Use case exit.
Alternate course of action	No alternate course of action.	

Use-case name	Employee registration	
Actor	Admin	
Description	This use case permits Admin to register new employees to the system.	
Precondition	New employee hired.	
Post-condition	New Employee information stored successfully.	
Basic course of action	**User Action**	**System Response**
	1. The Admin wants to add a new staff member. 2. Goes to Add Employee form page. 4. The Admin enters the proper following information in the form. 5. The Admin clicks or presses on the Save button.	3. The system displays a form to be filled out for Employee registration. 6. The system verifies that the fields have been filled out correctly. 7. The system displays a Successfully Stored message to the Admin. 8. Use case exit.
Alternate course of action	If all fields are not filled out correctly the system goes back or returns to step 4 of basic course of action. The admin needs to refill the invalid or the empty field.	

Use-case name	Edit Employee Information	
Actor	Admin	
Description	This use case permits Admin to alter Employee information.	
Precondition	Employee Information has changed.	
Post-condition	Employee information stored successfully.	
Basic course of action	**User Action**	**System Response**
	1. The Admin wants to edit employee information. 2.Chooses Employee and goes to Edit Employee Information. 4. The Admin enters the proper following information in the form and clicks Save.	3. The system displays a form to be filled out with Employee information. 6. The system verifies that the fields have been filled out correctly. 7. The system displays a Successfully Stored message to the admin. 8. Use case exit.
Alternate course of action	If all fields are not filled out correctly the system goes back or returns to step 4 of basic course of action. The admin needs to refill the invalid or the empty field.	

Use-case name	Delete Employee	
Actor	Admin	
Description	This use case permits Admin to delete Employee from database.	
Precondition	Employee no longer works for the company.	
Post-condition	Employee deleted successfully from the database.	
Basic course of action	**User Action**	**System Response**
	1. The Admin wants to delete Employee. 2.Chooses Employee and goes to Edit Employee. 4. The Admin enters the proper following information in the form and clicks Save.	3. The system displays a form to be filled out with Employee information. 6. The system verifies that the fields have been filled out correctly. 7. The system displays a Successfully Stored message to the Admin. 8. Use case exit.
Alternate course of action	If all fields are not filled out correctly the system goes back or returns to step 4 of basic course of action. The Admin needs to refill the invalid or the empty field.	

4.2 Use Case Diagrams

Figure 1 Main Use Case Diagram

In this use case diagram, we have shown the main use cases that the three actors of our system, consisting of:

- Customer,
- Admin,
- Employee,

are required to accomplish on daily basis, while interacting with the system.

Figure 2 Use Case Dependencies Diagram

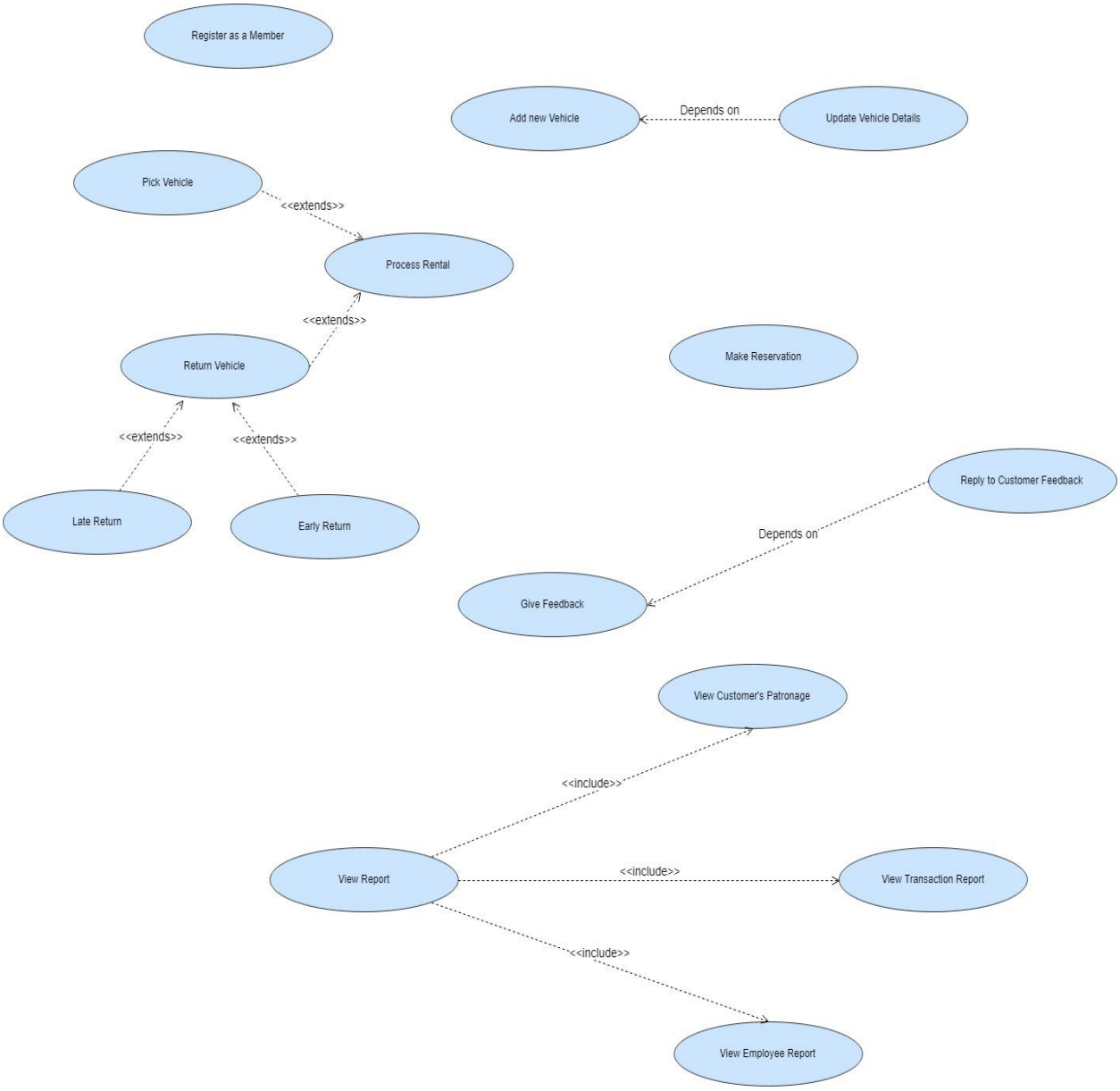


Figure 3 Rental Process Use Case Diagram

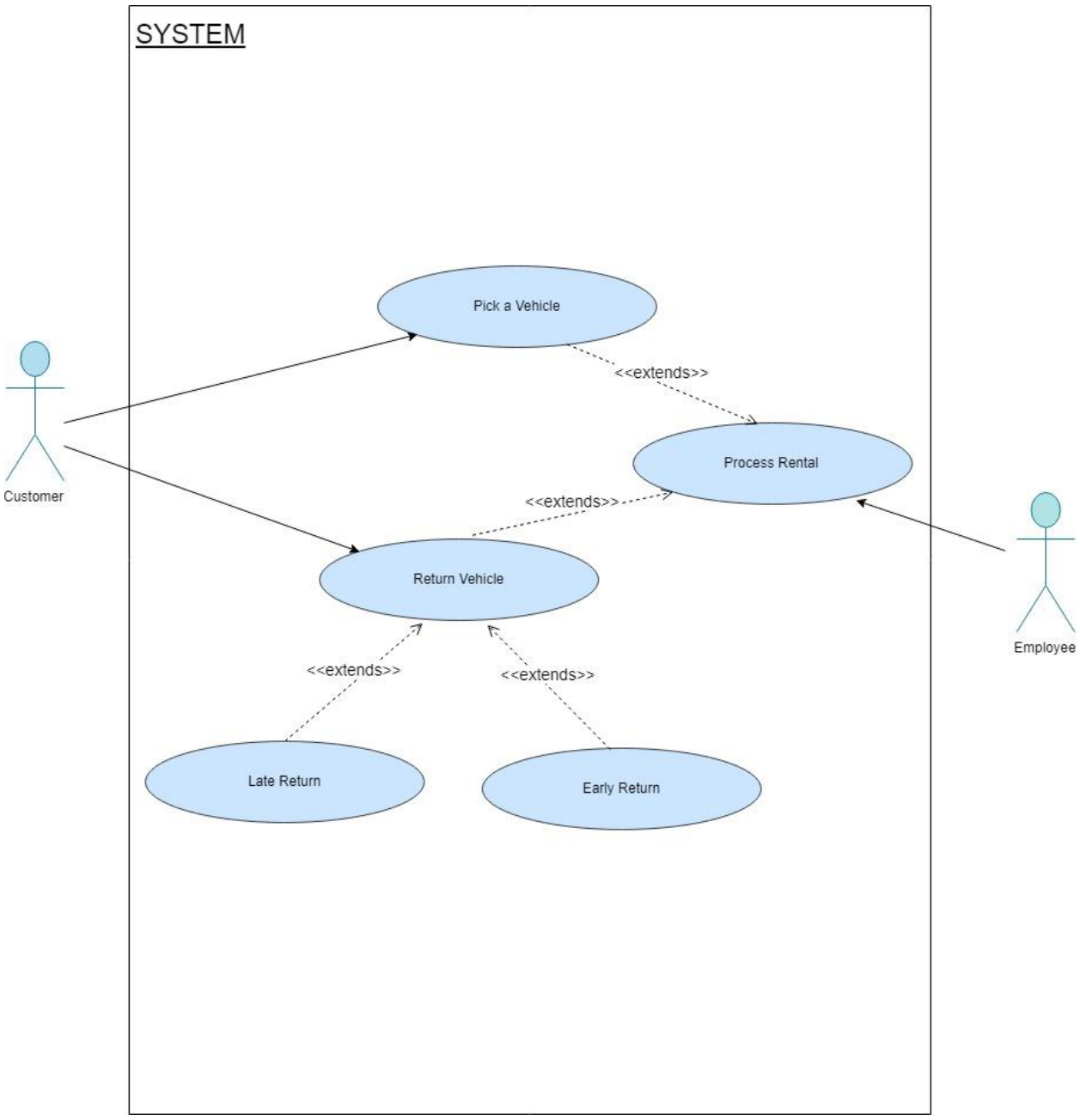


Figure 4 Report Use Case Diagram

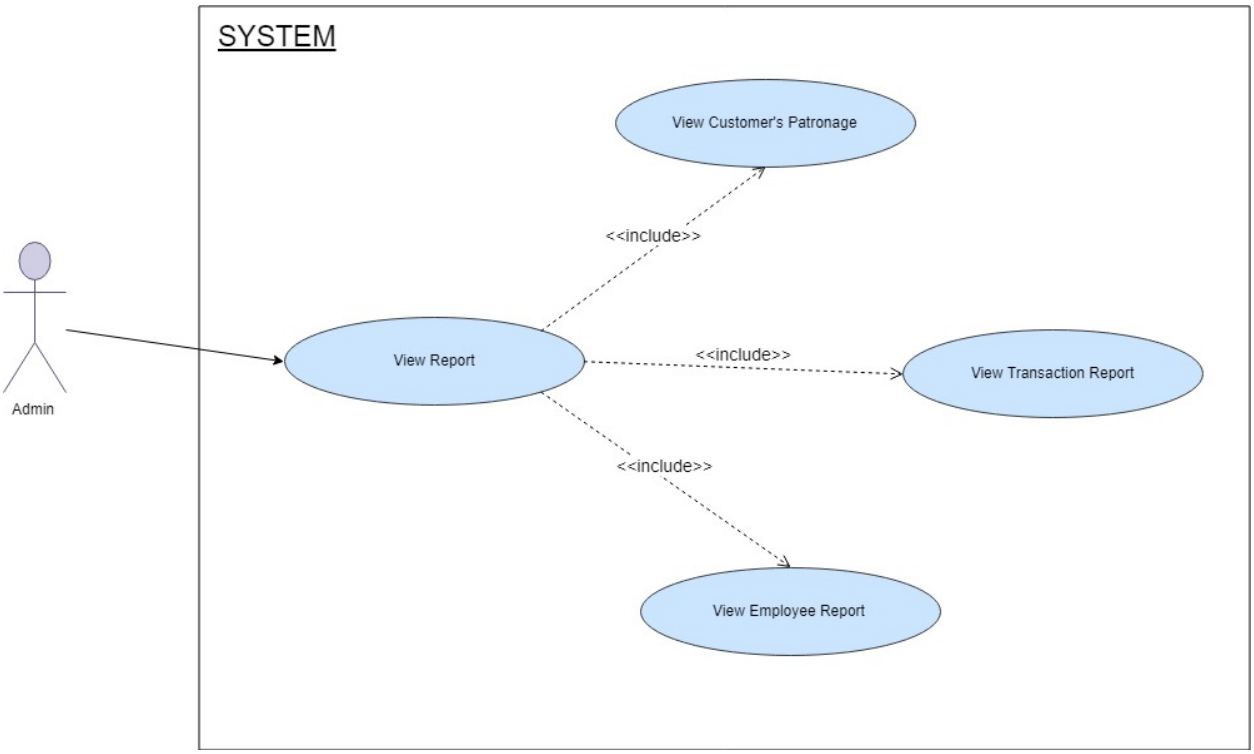
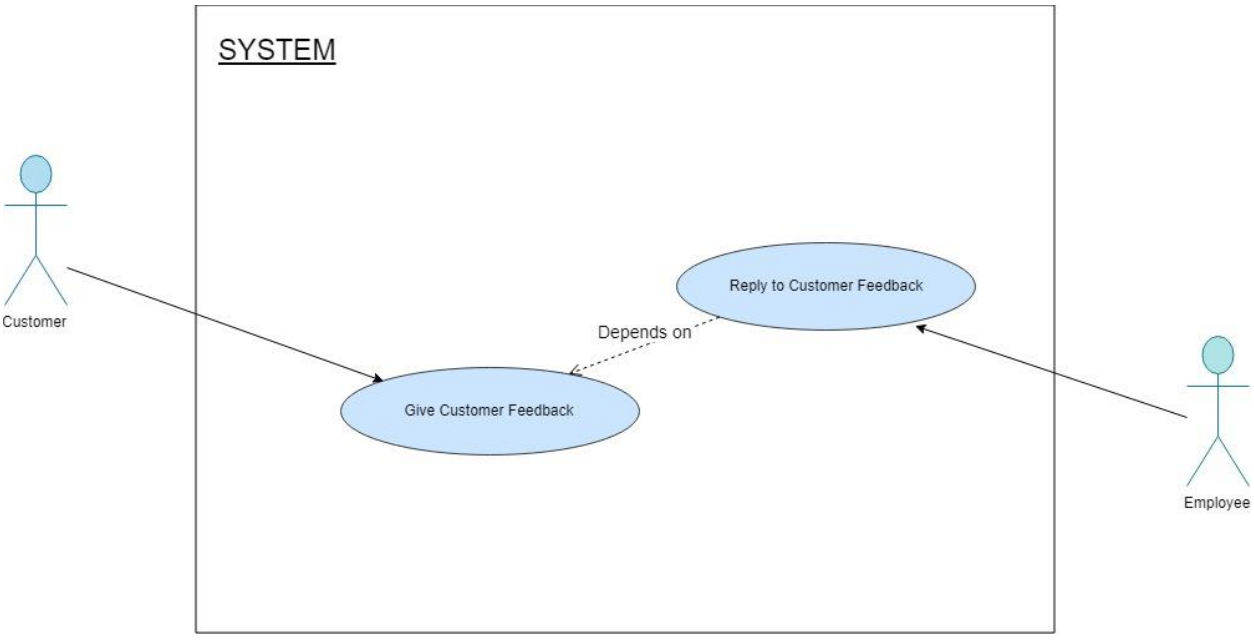


Figure 5 Feedback Use Case Diagram



4.3 Activity Diagrams

Figure 6 Customer Registration Activity Diagram

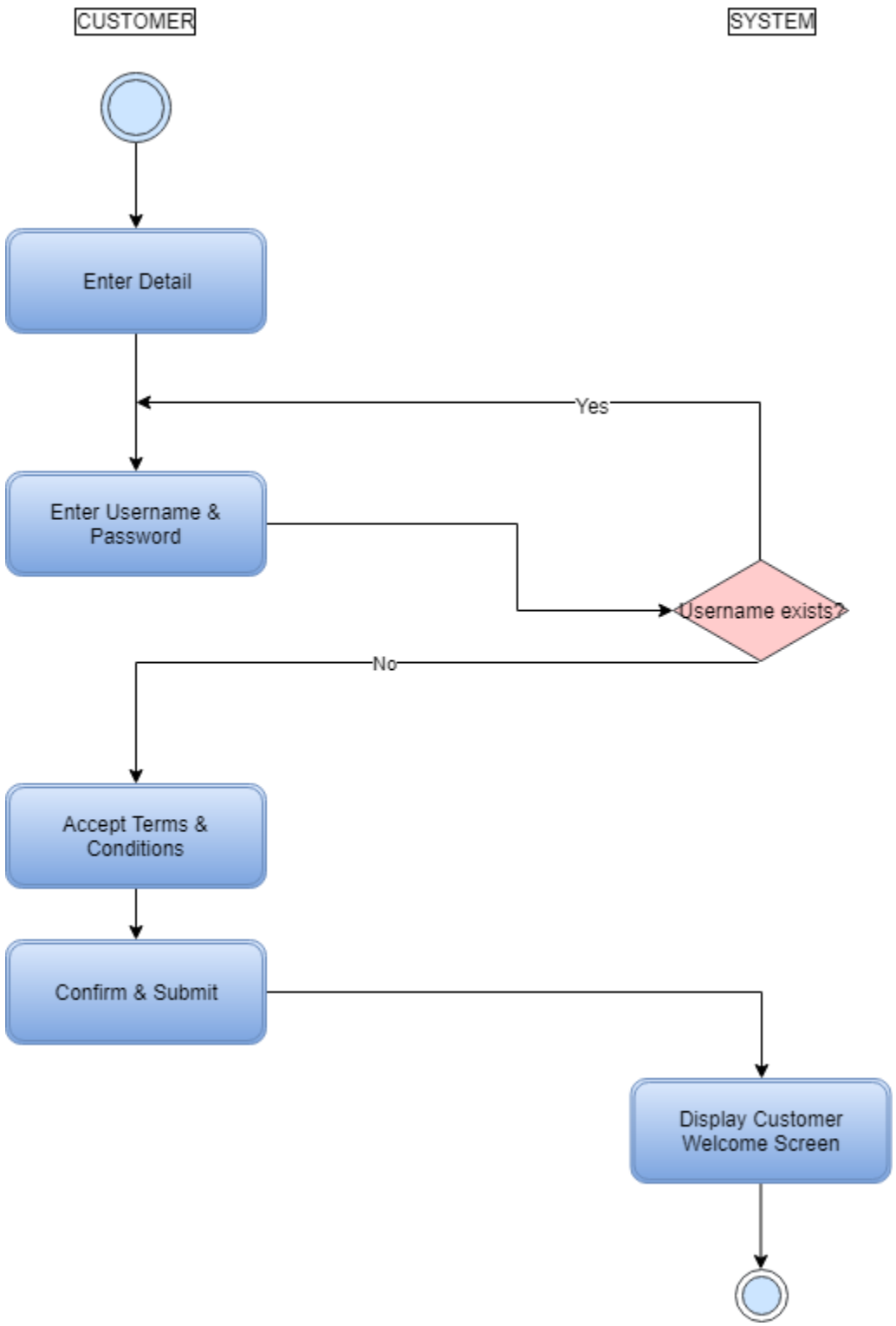


Figure 7 New Vehicle Activity Diagram

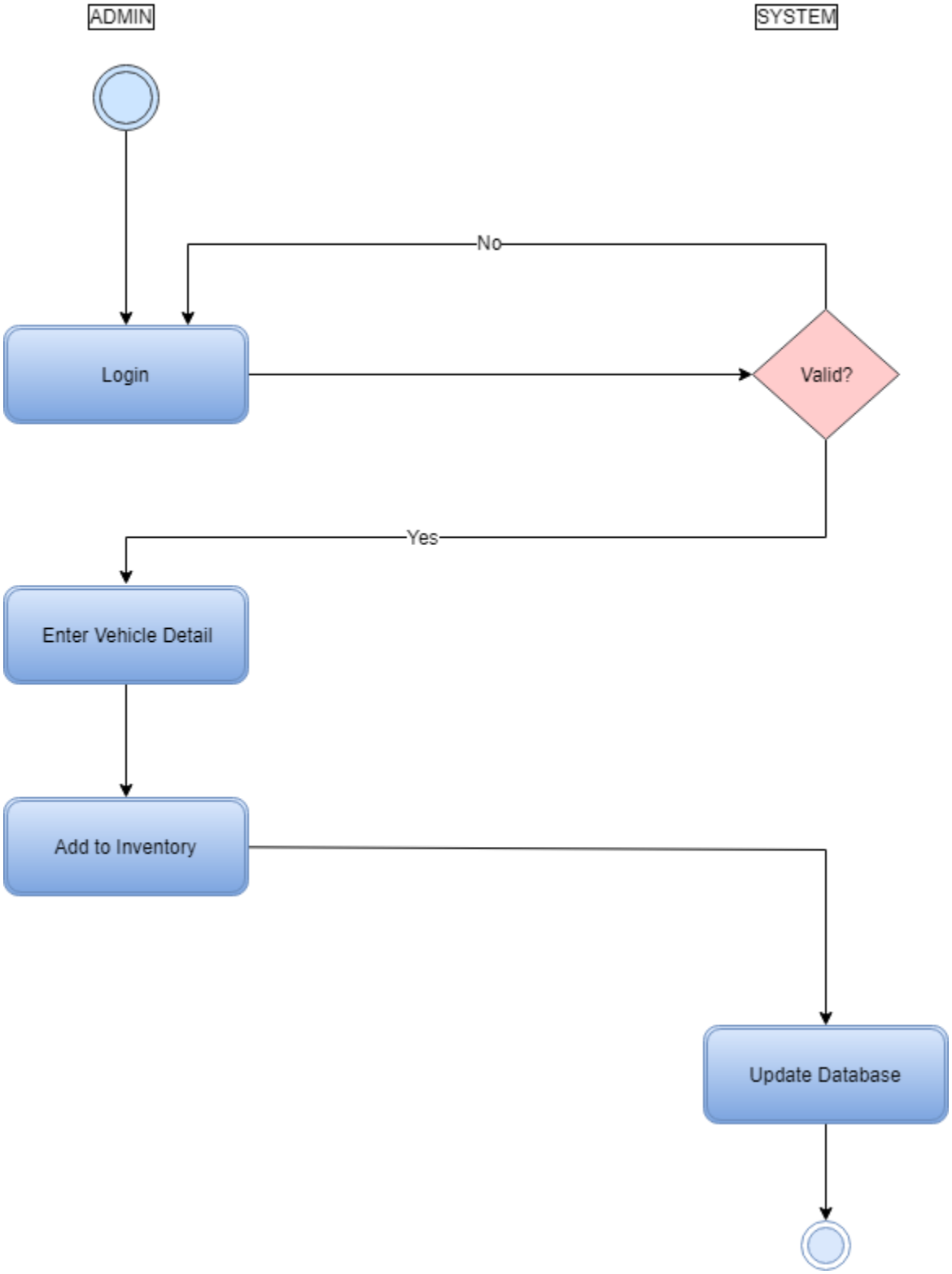


Figure 8 Vehicle Reservation Activity Diagram

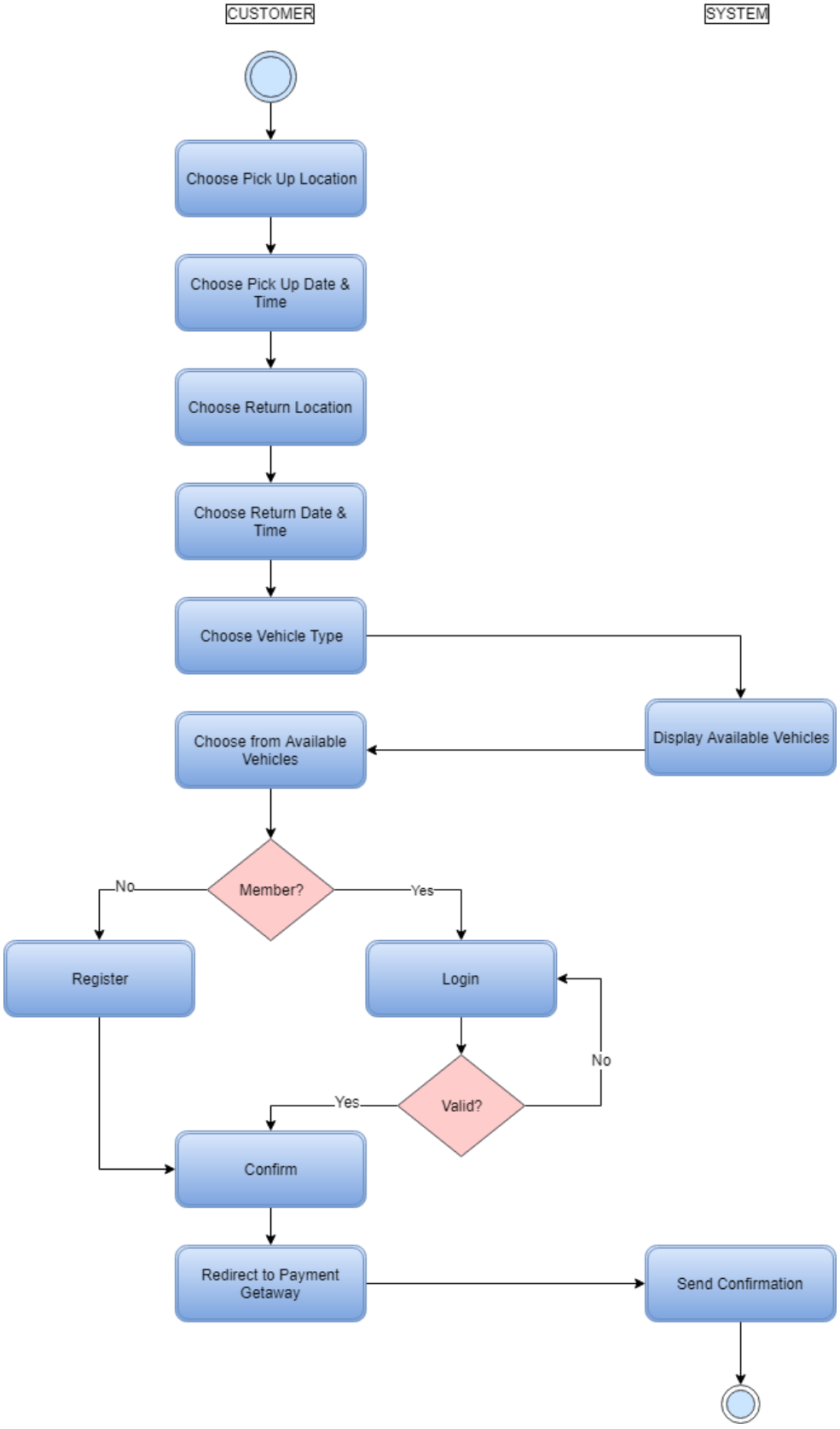


Figure 9 Payment Activity Diagram

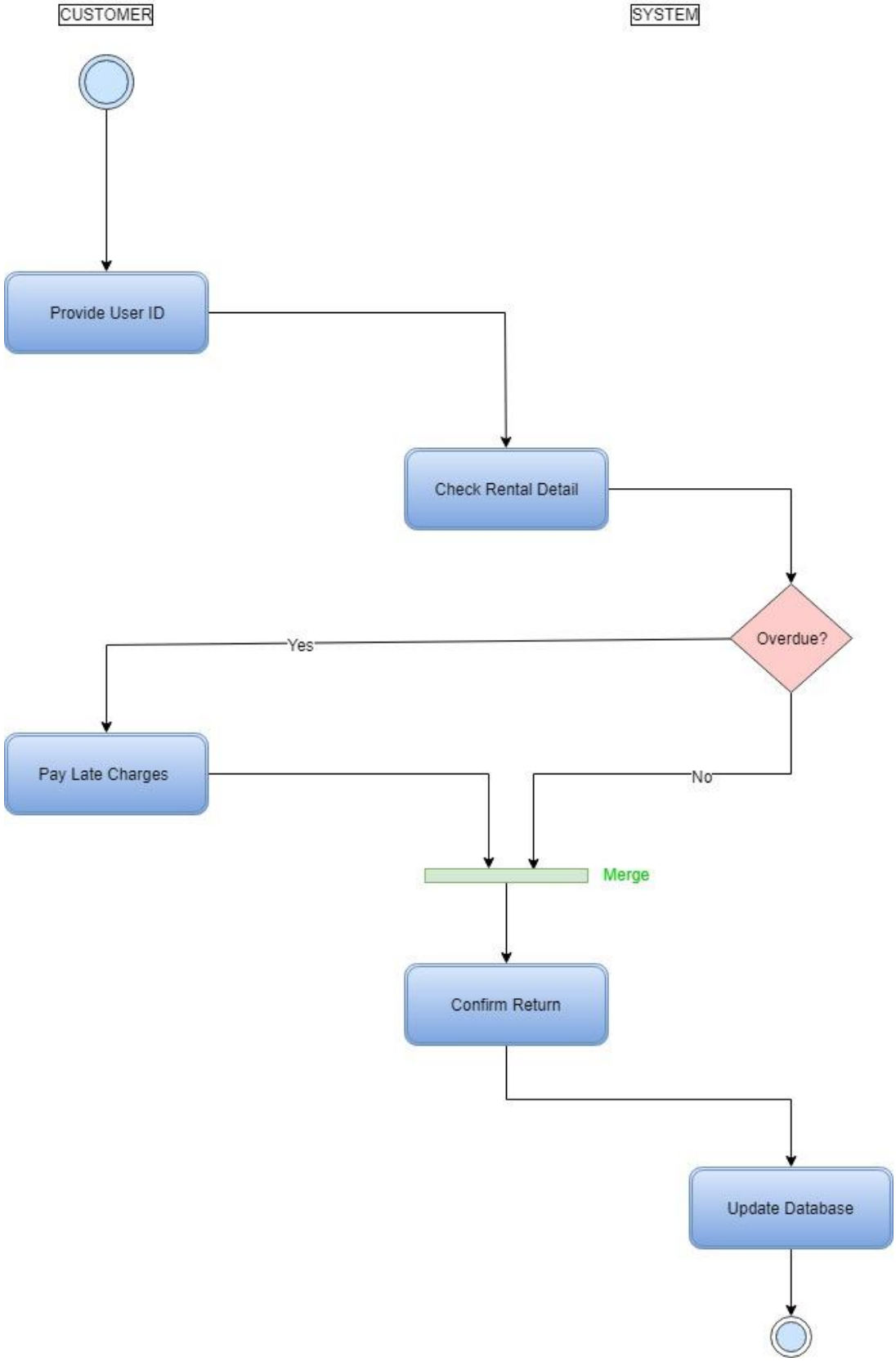


Figure 10 Customer Profile Modification Activity Diagram

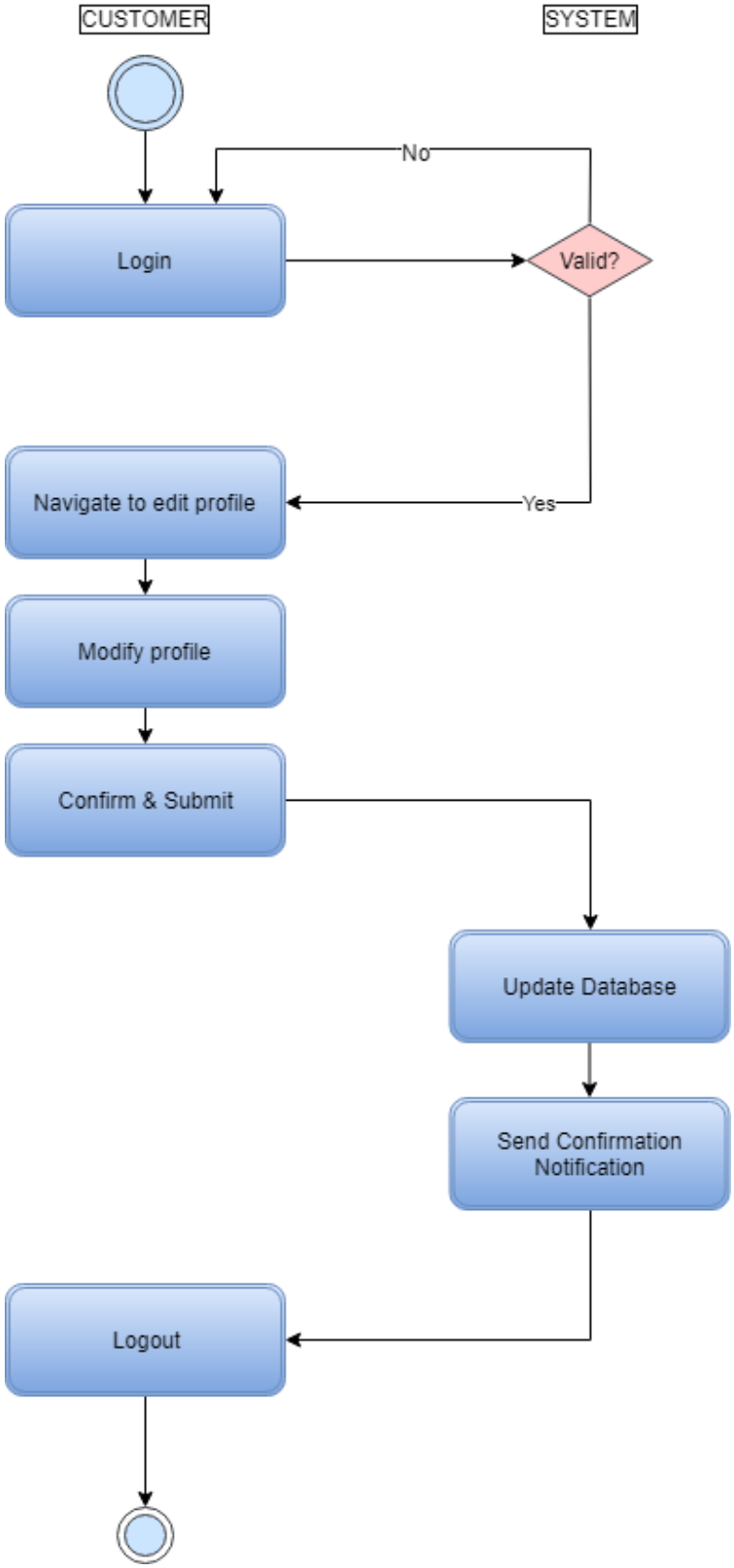


Figure 11 Placing Order Activity Diagram

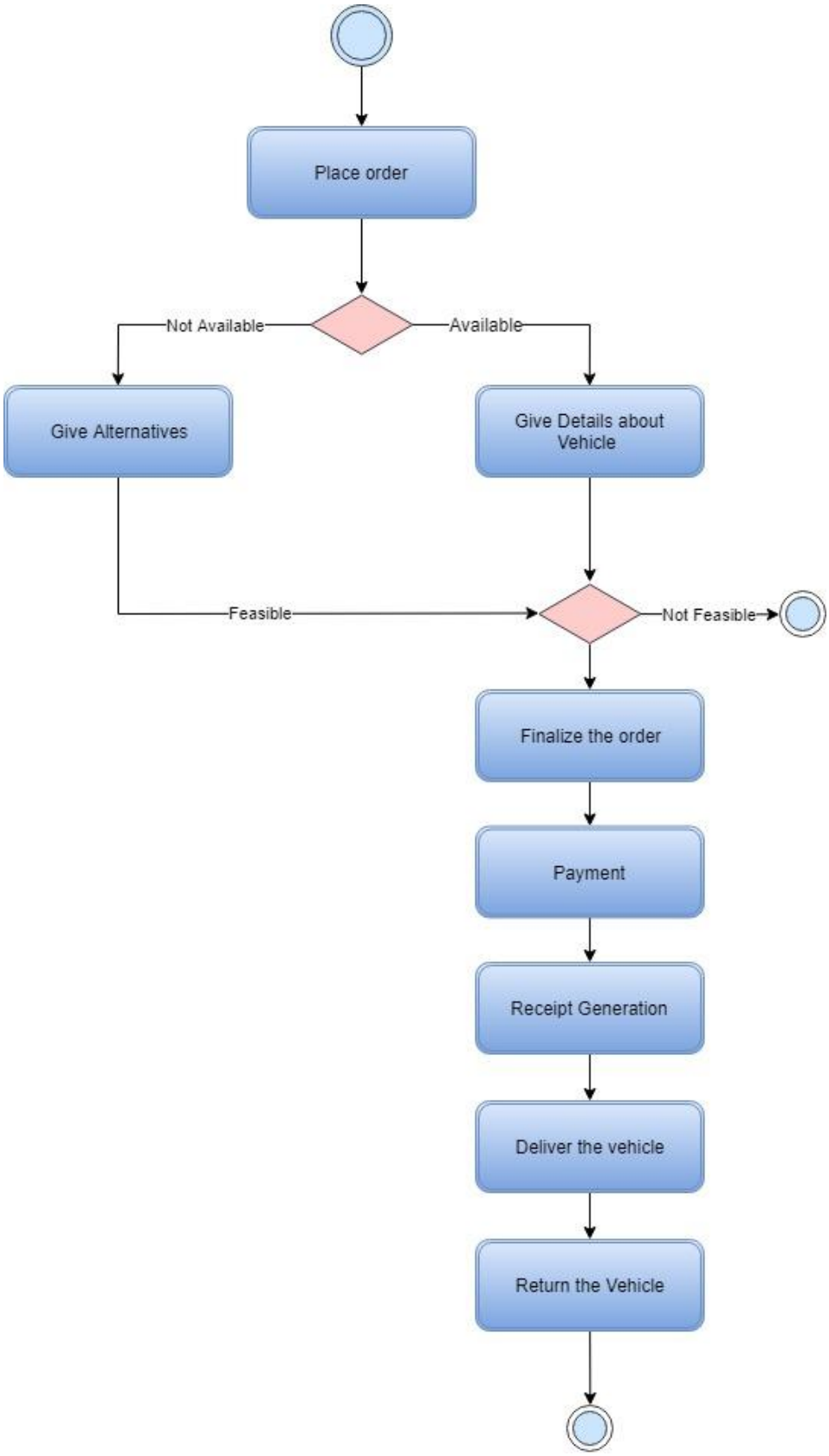


Figure 12 Generate Report Activity Diagram

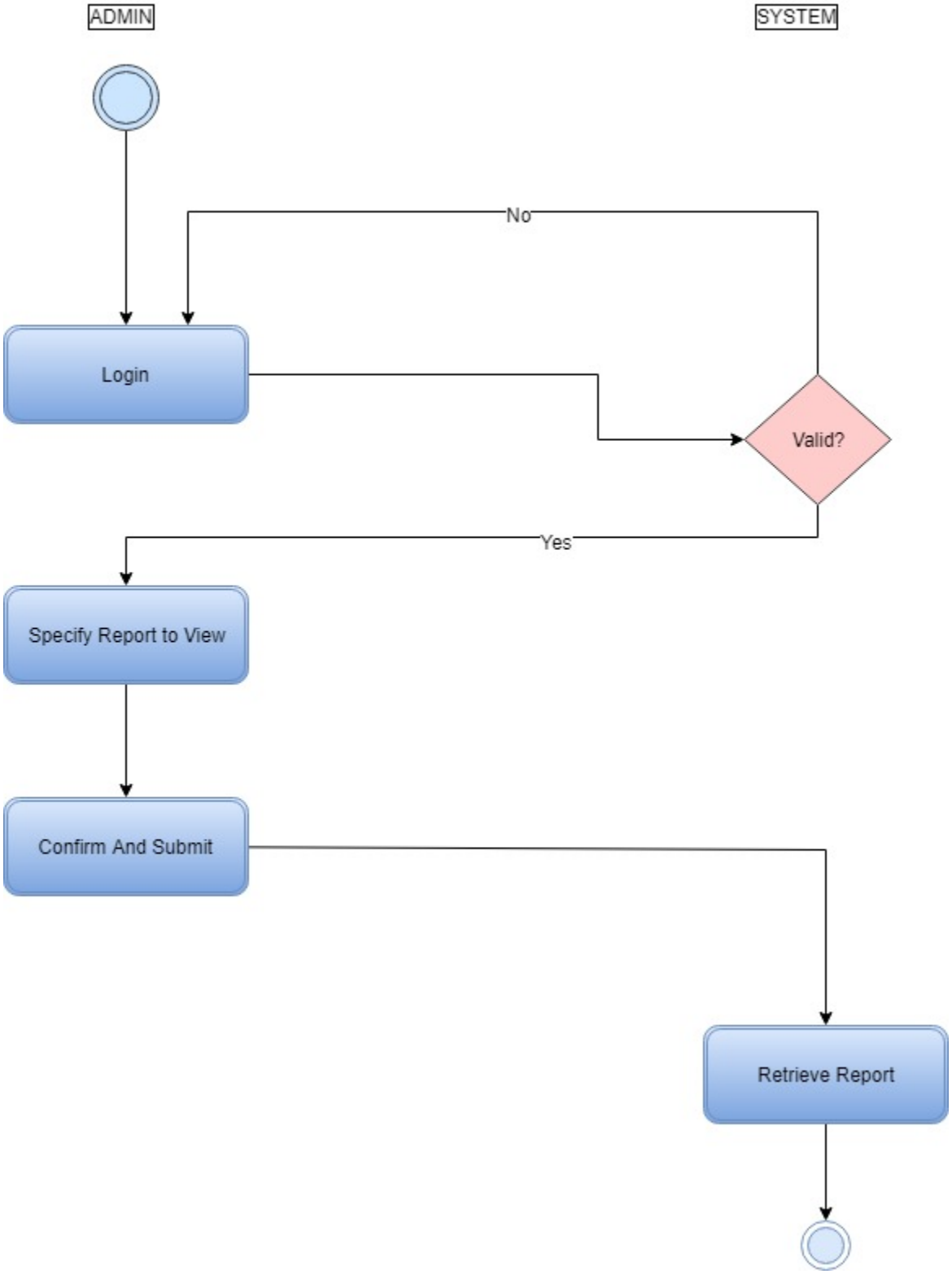


Figure 13 Customer Feedback Activity Diagram

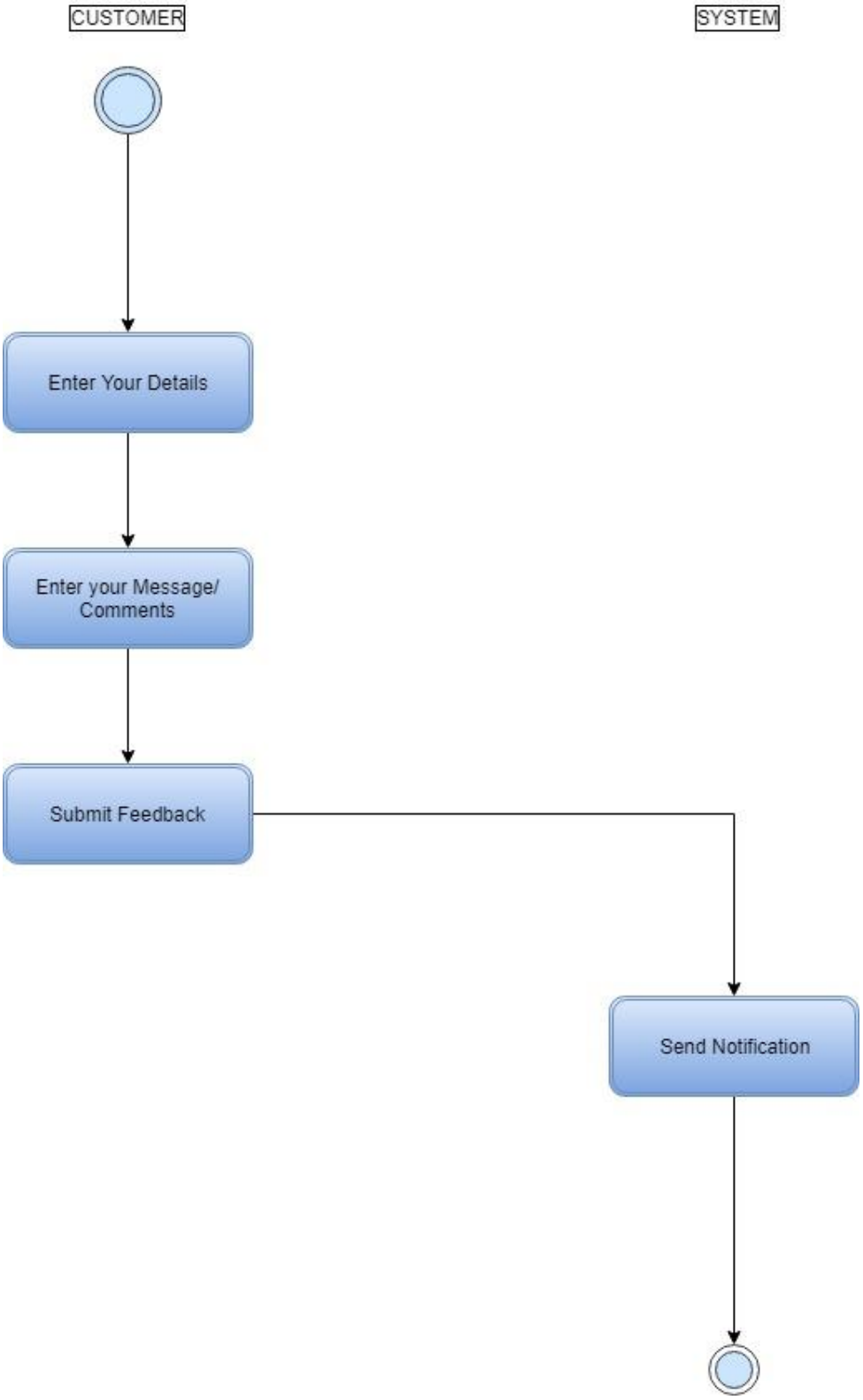
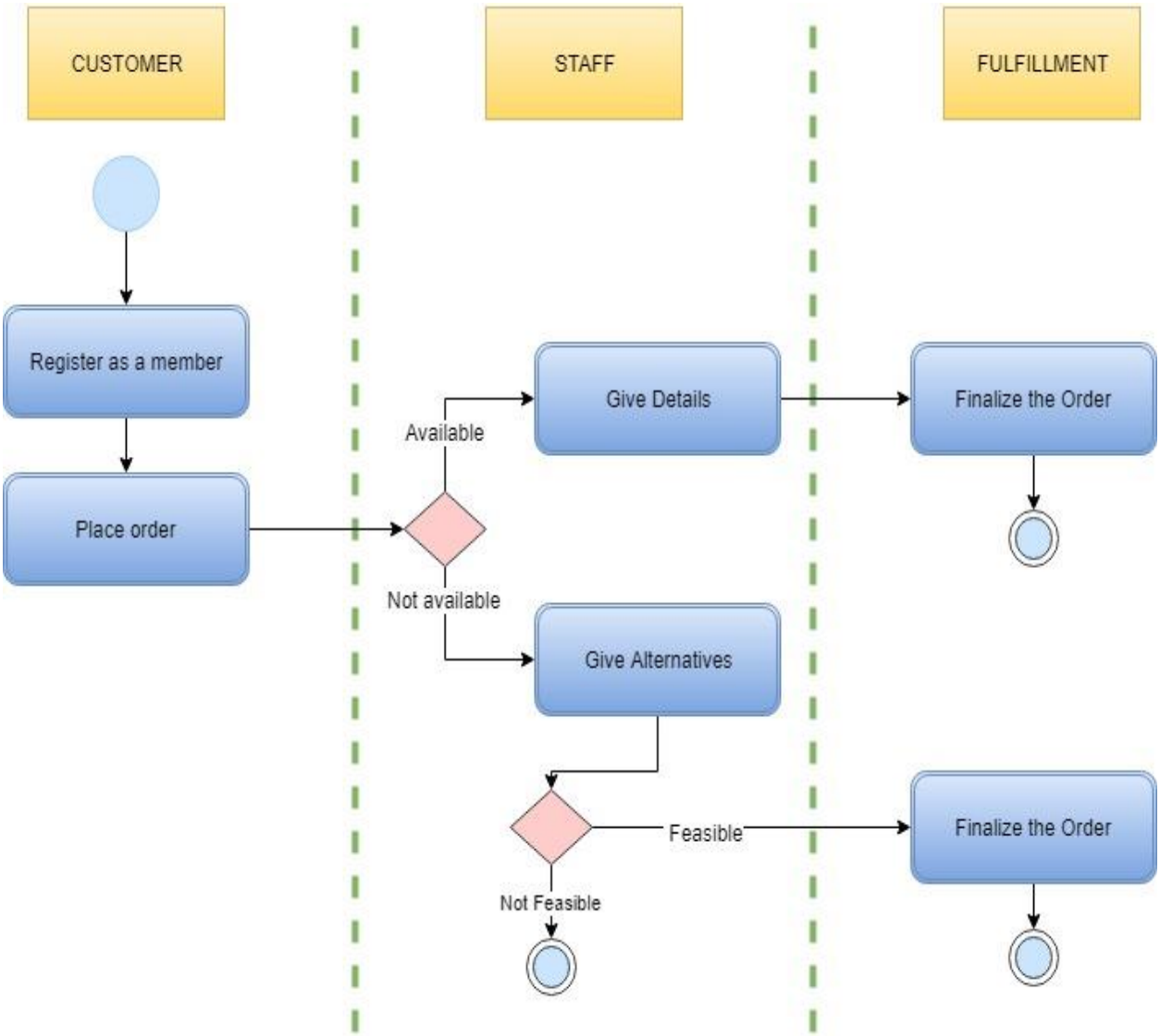


Figure 14 Placing Order Swim-lane Activity Diagram



4.4 Sequence Diagrams

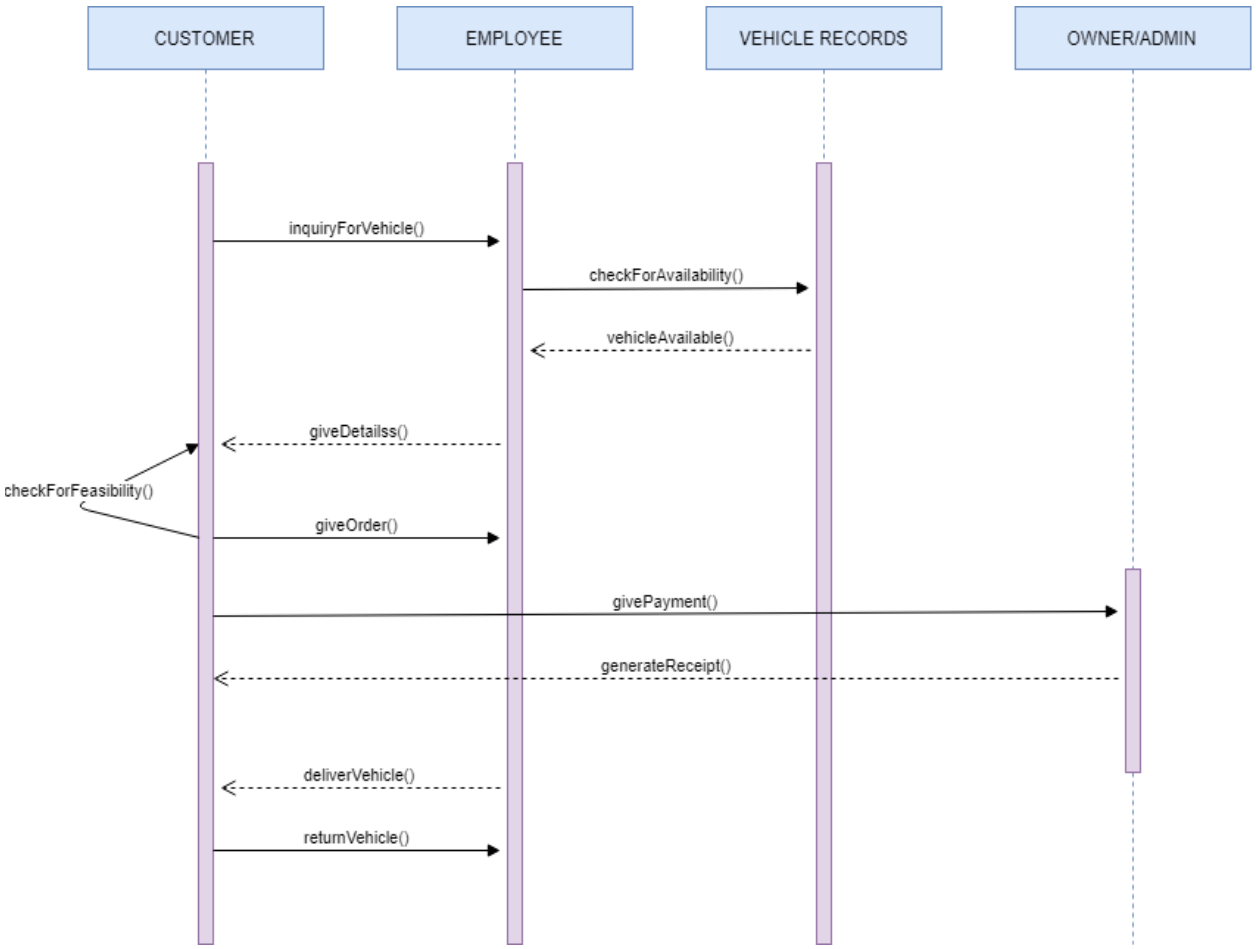


Figure 15 Overall Renting Process Sequence Diagram

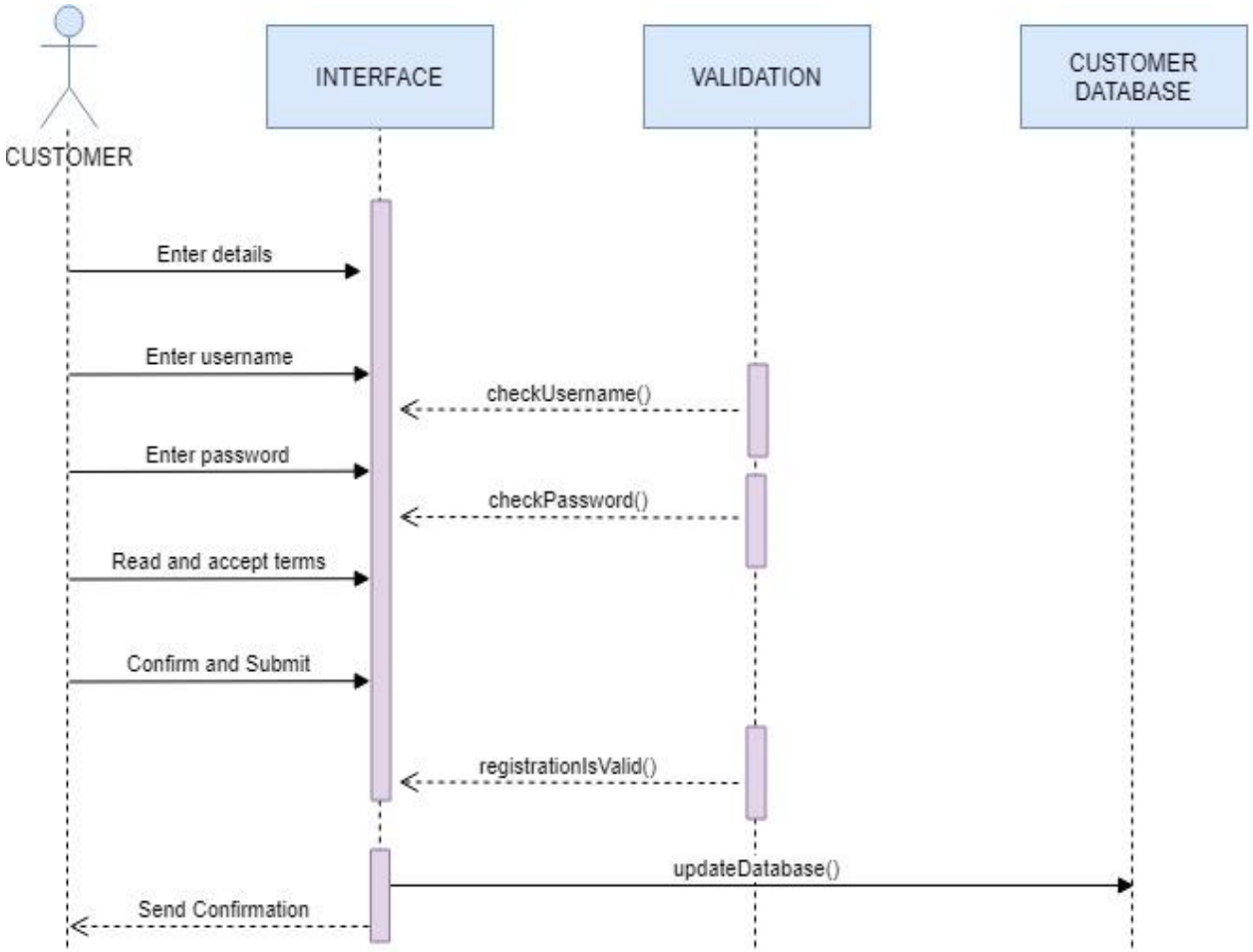


Figure 16 Log in Sequence Diagram

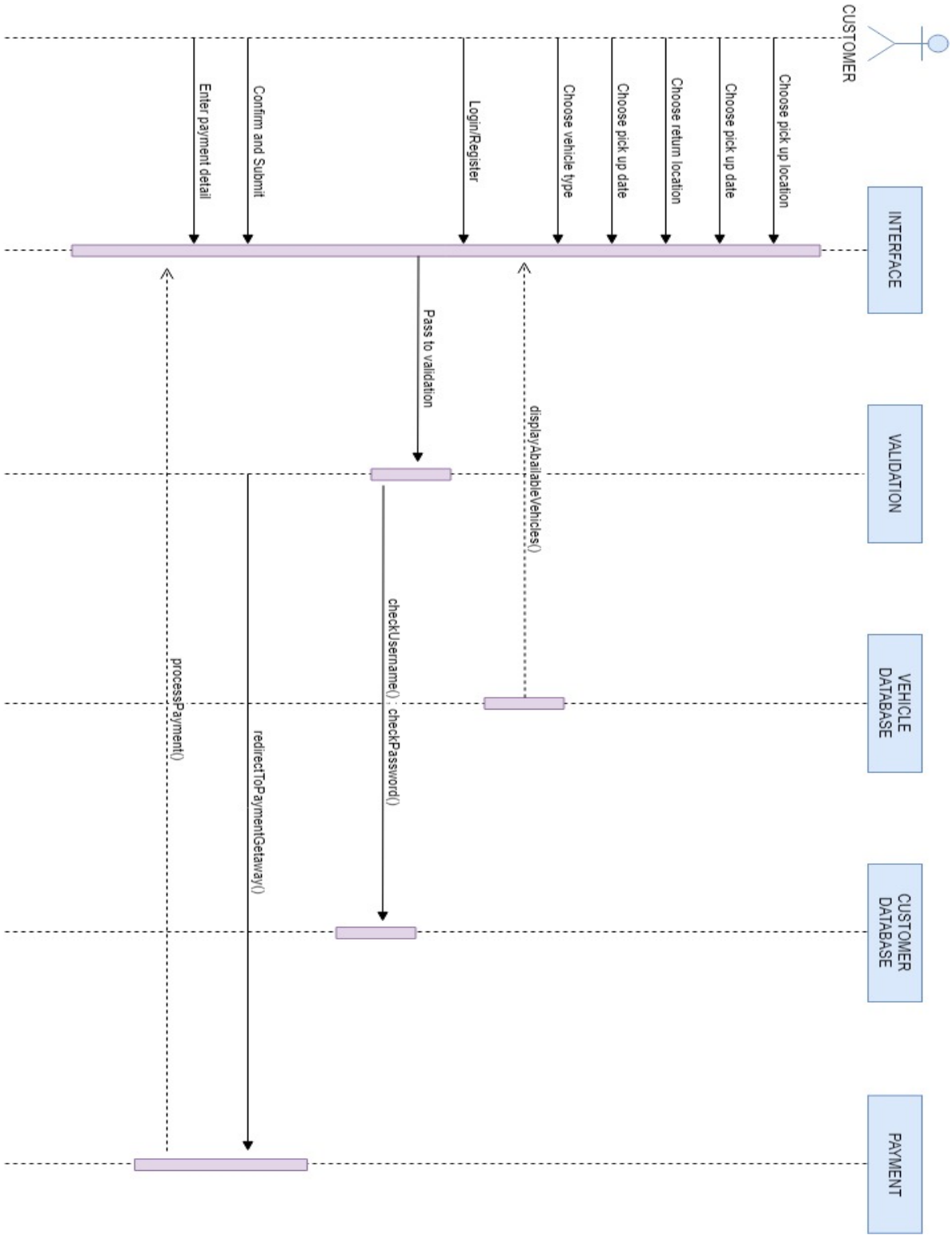


Figure 17 Vehicle Reservation Activity Diagram

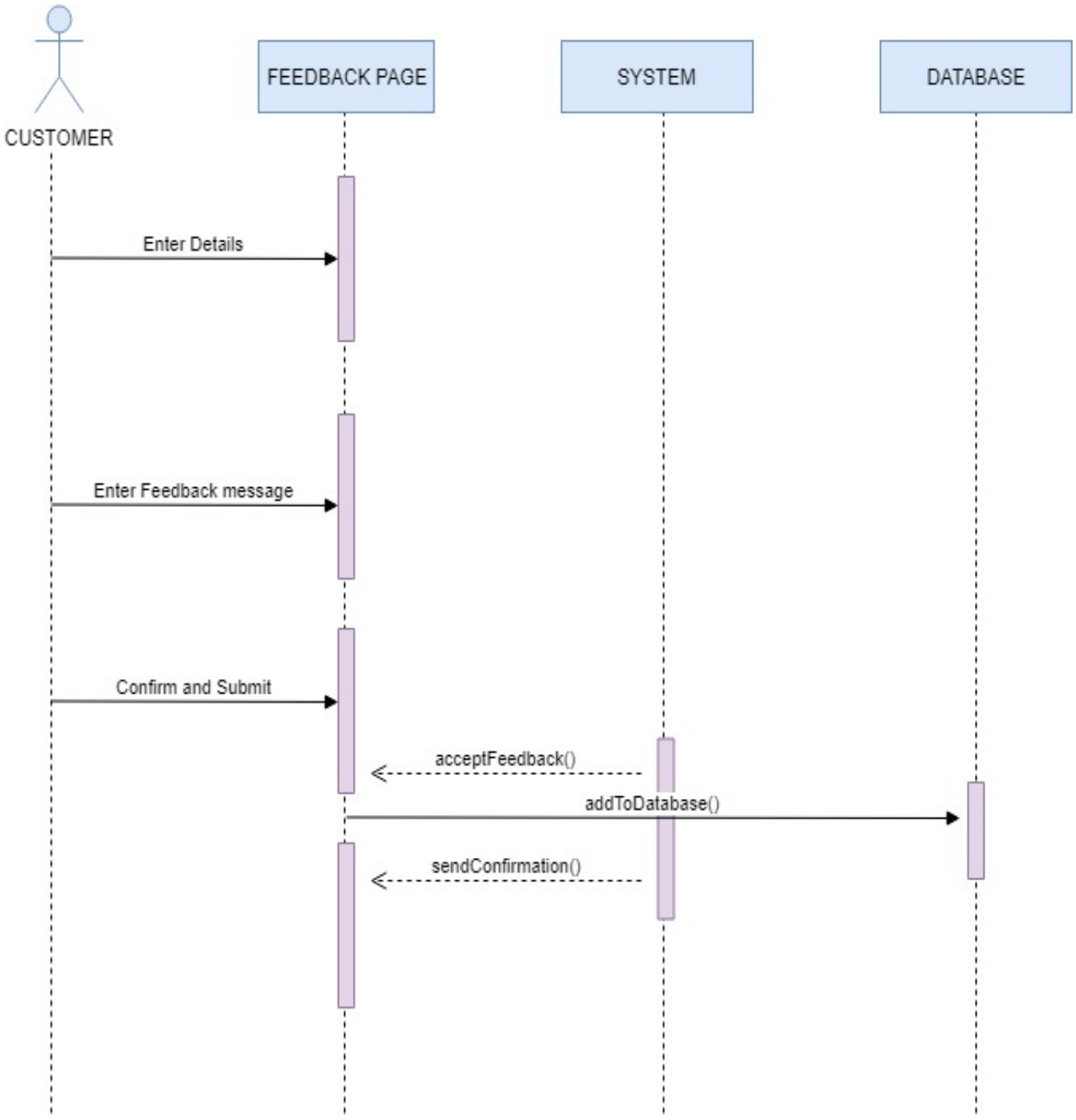


Figure 18 Feedback Sequence Diagram

Figure 19 Add New Vehicle Sequence Diagram

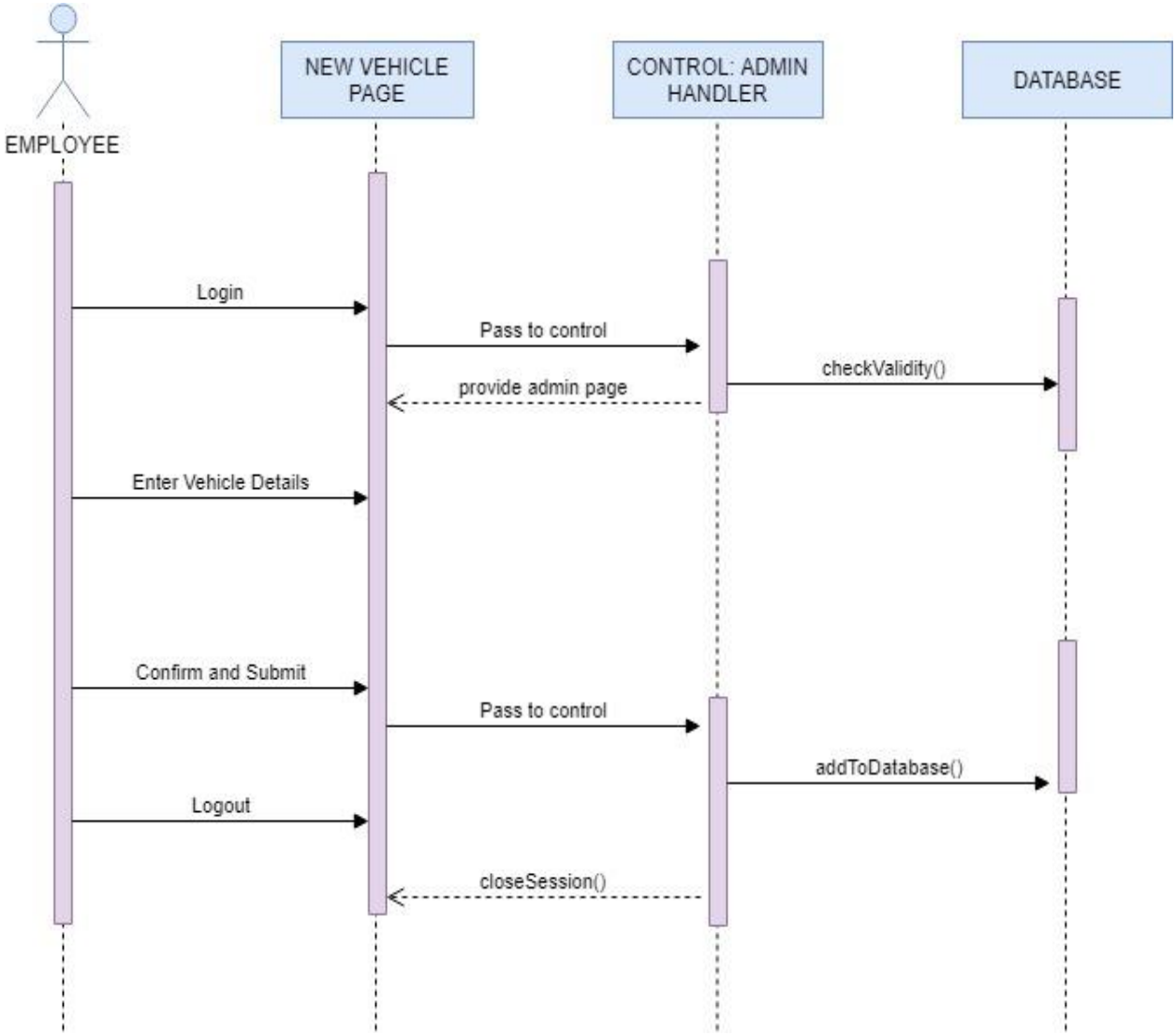


Figure 20 Return Vehicle Sequence Diagram

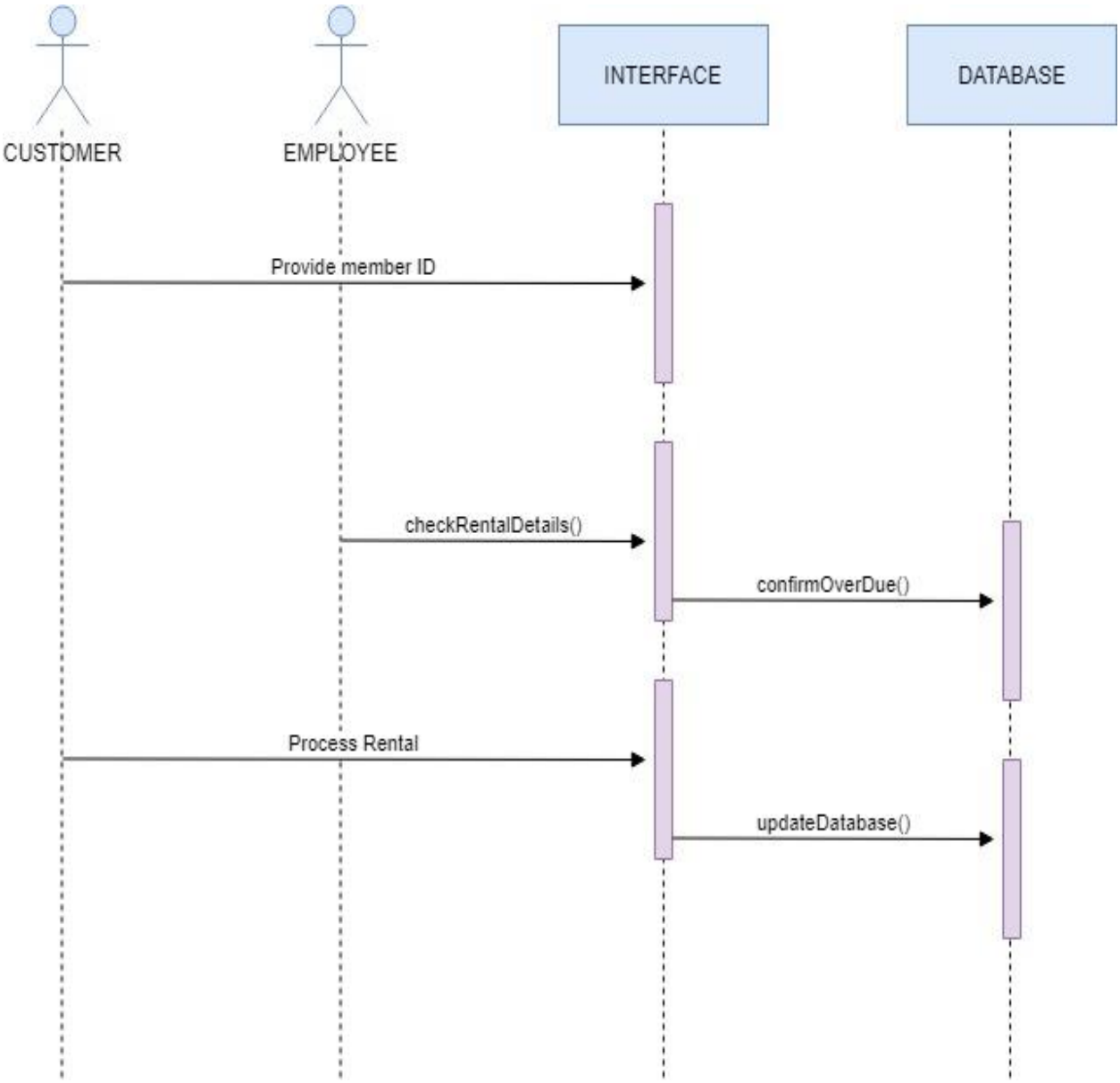
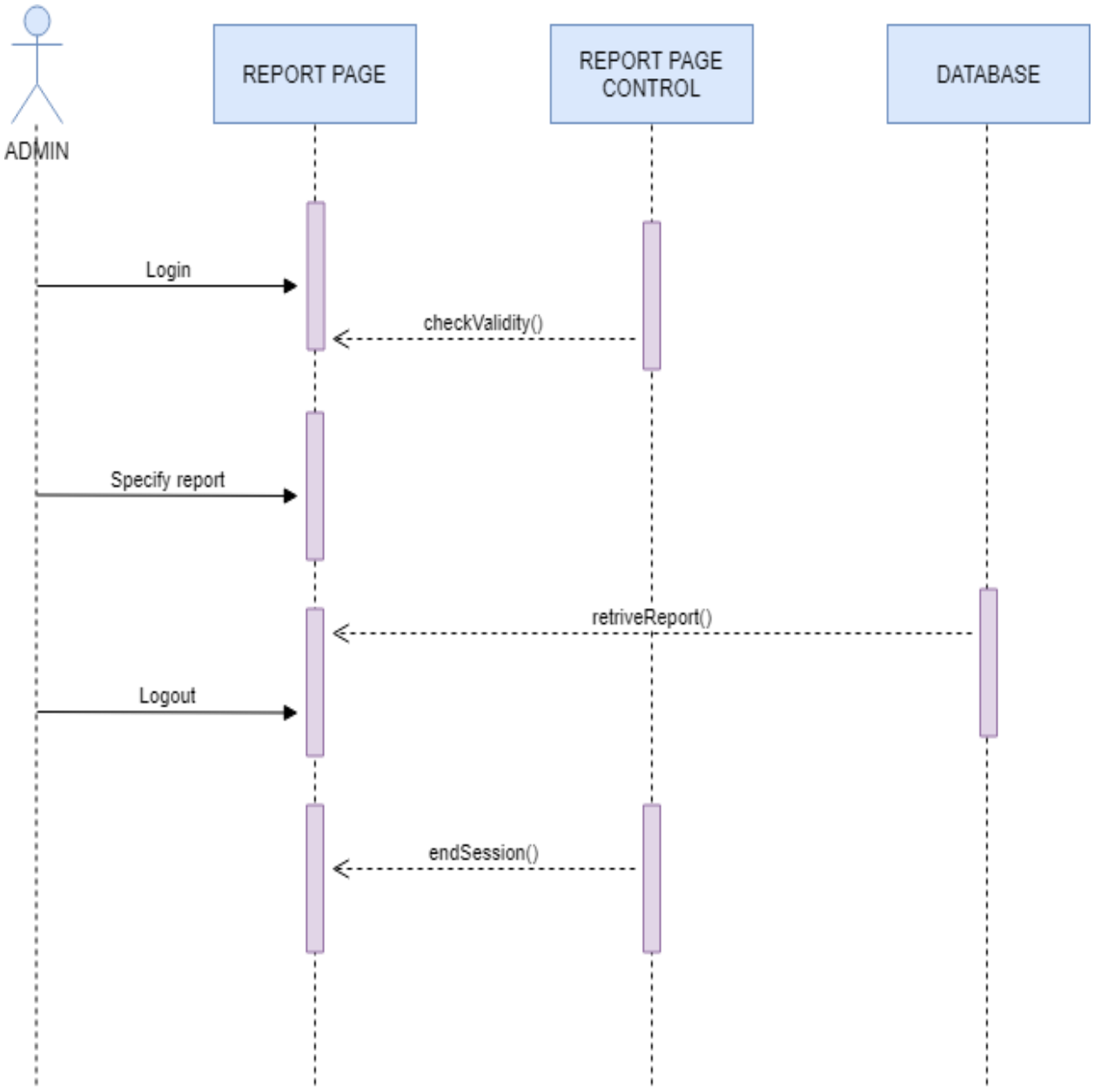


Figure 21 Report Sequence Diagram



4.5 State Diagram

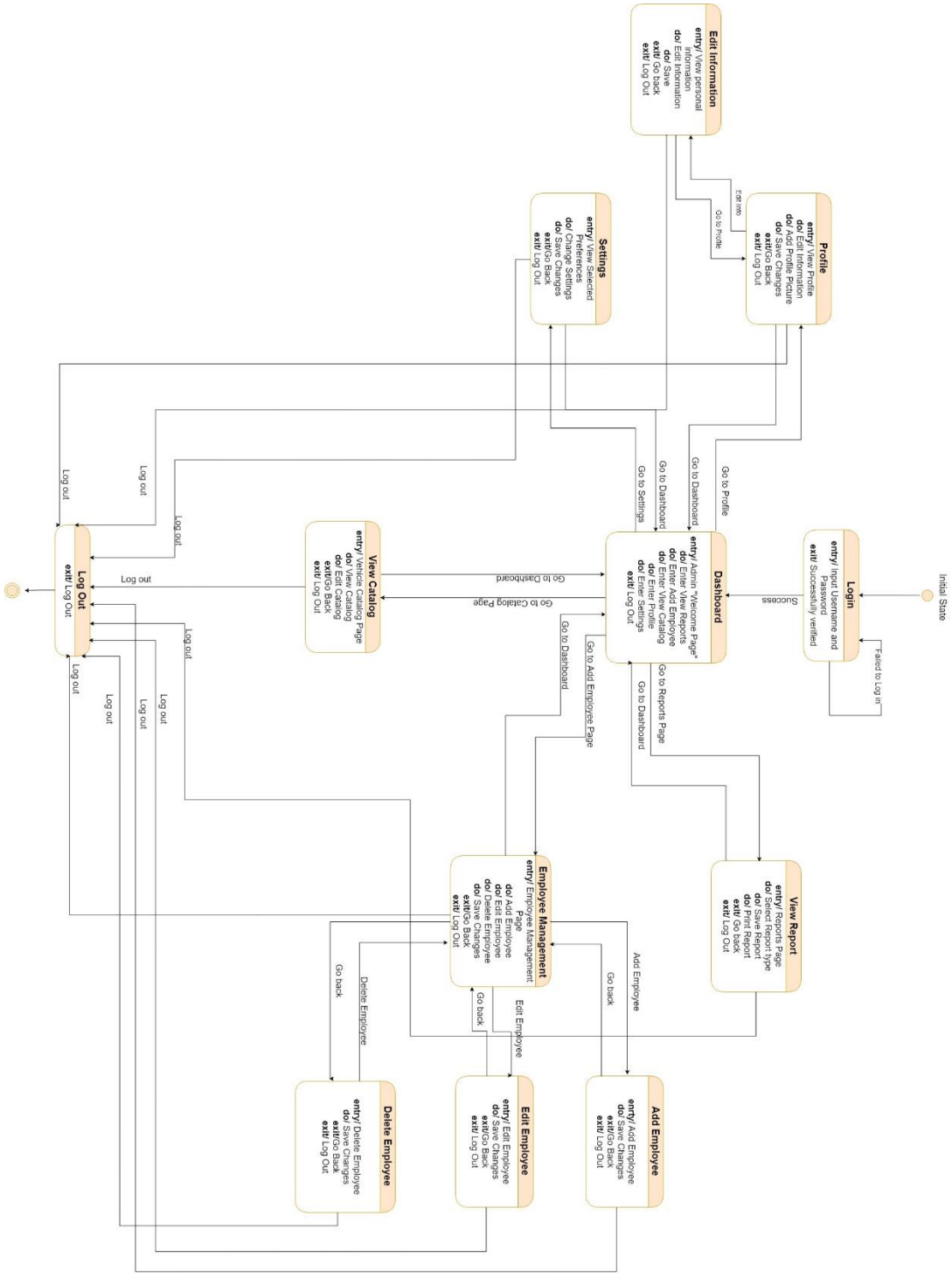


Figure 22 Admin State Diagram

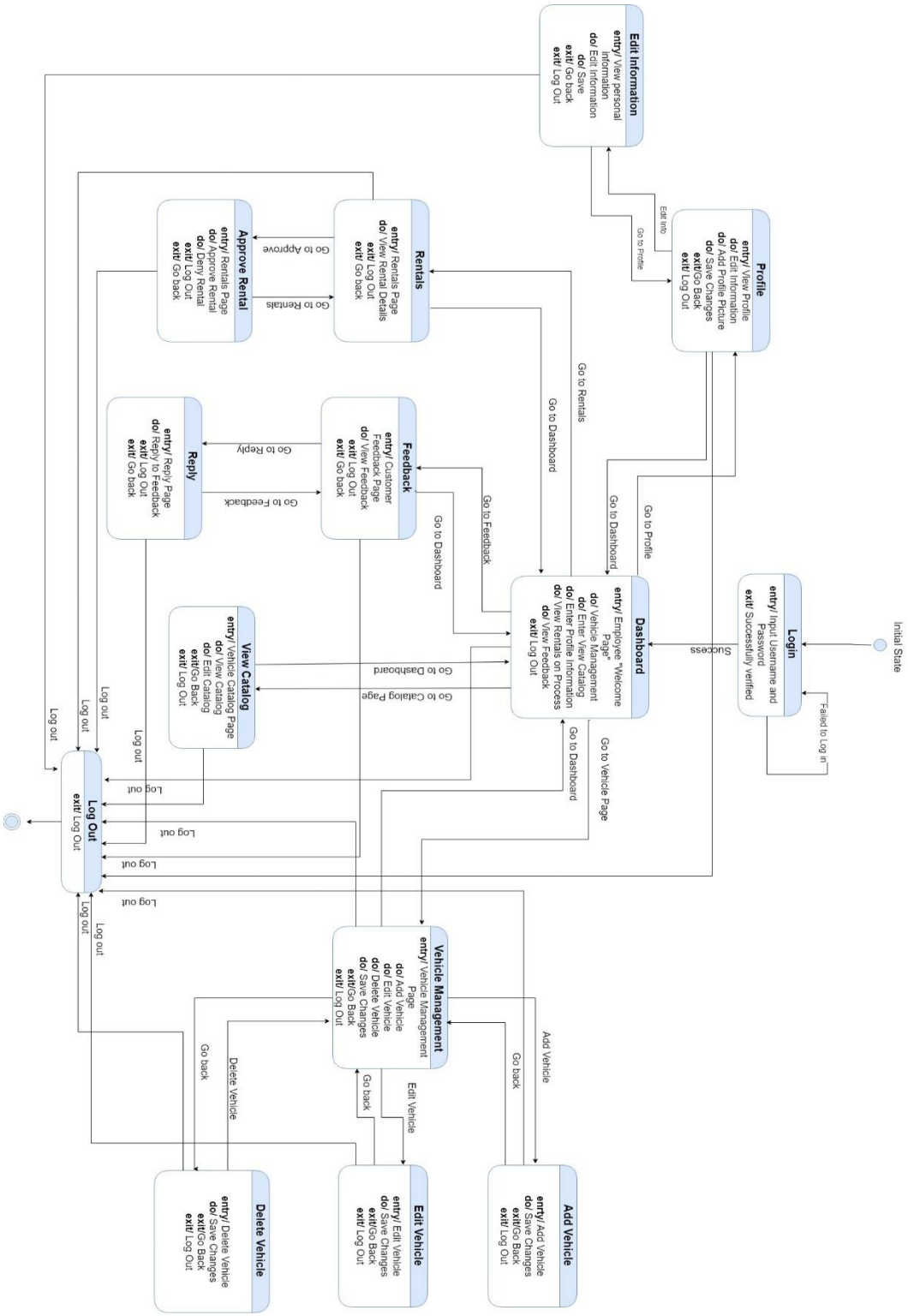


Figure 23 Employee State Diagram

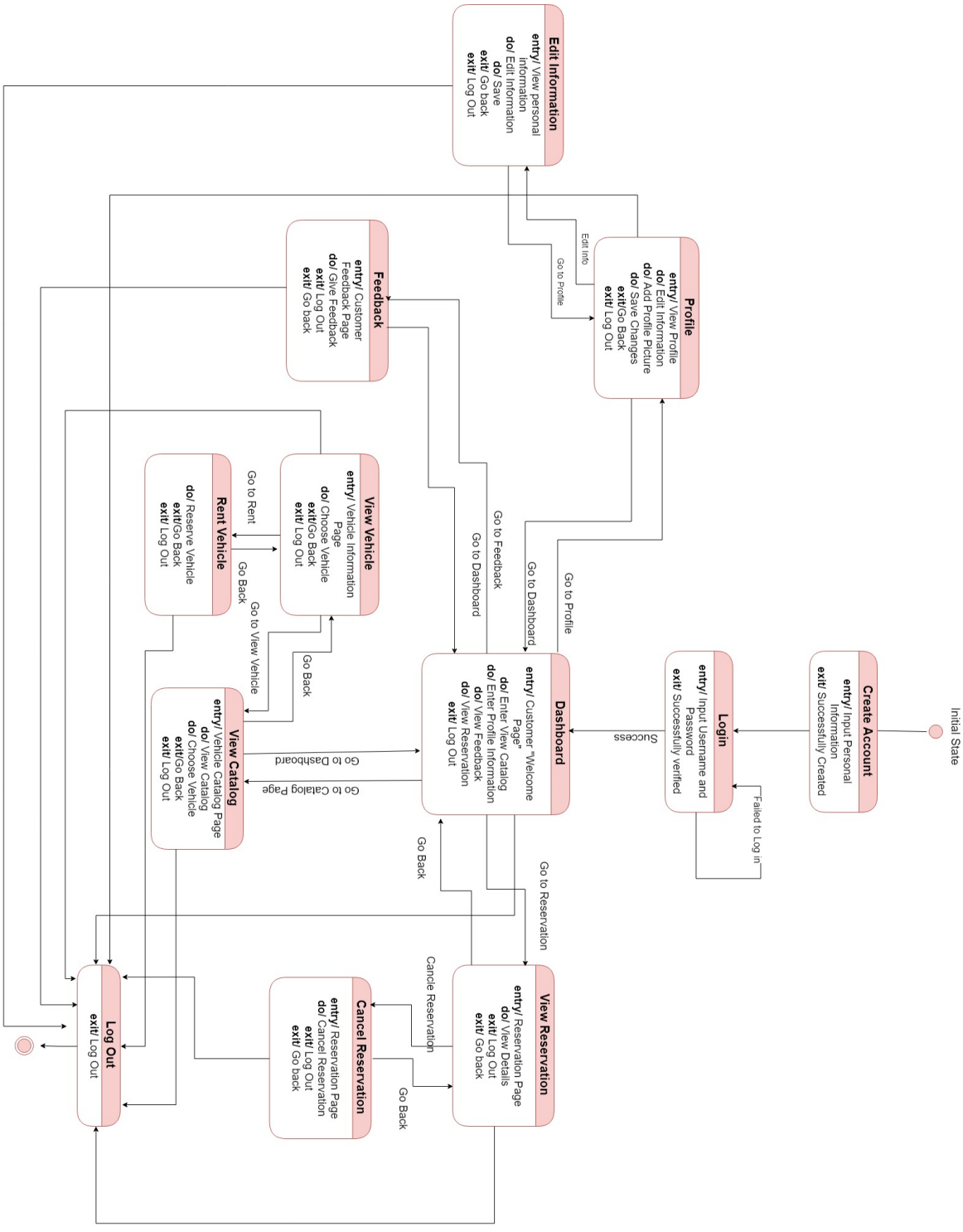


Figure 24 Customer State Diagram

4.6 Class Diagram

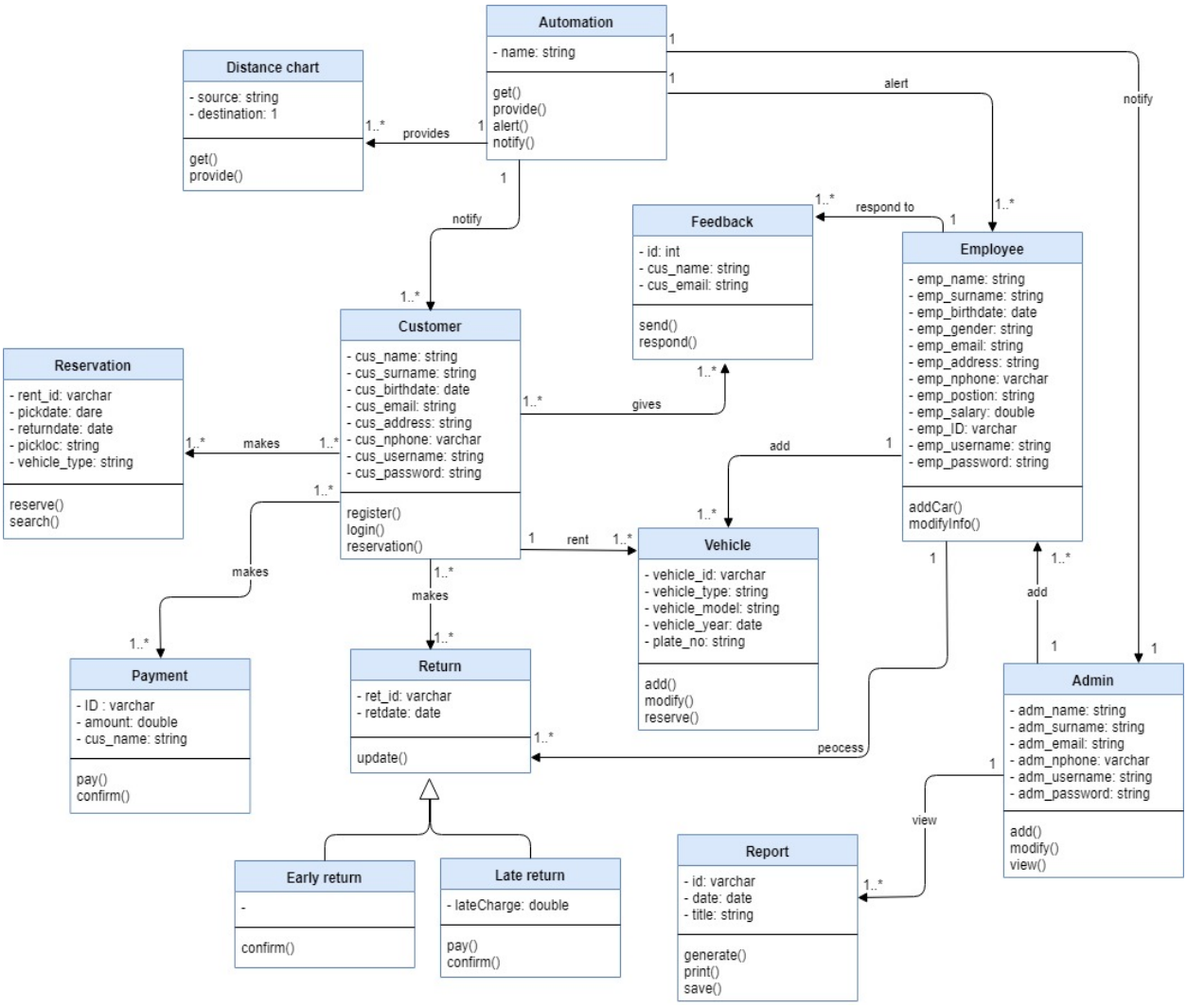


Figure 25 Class Diagram

4.7 Object Diagram

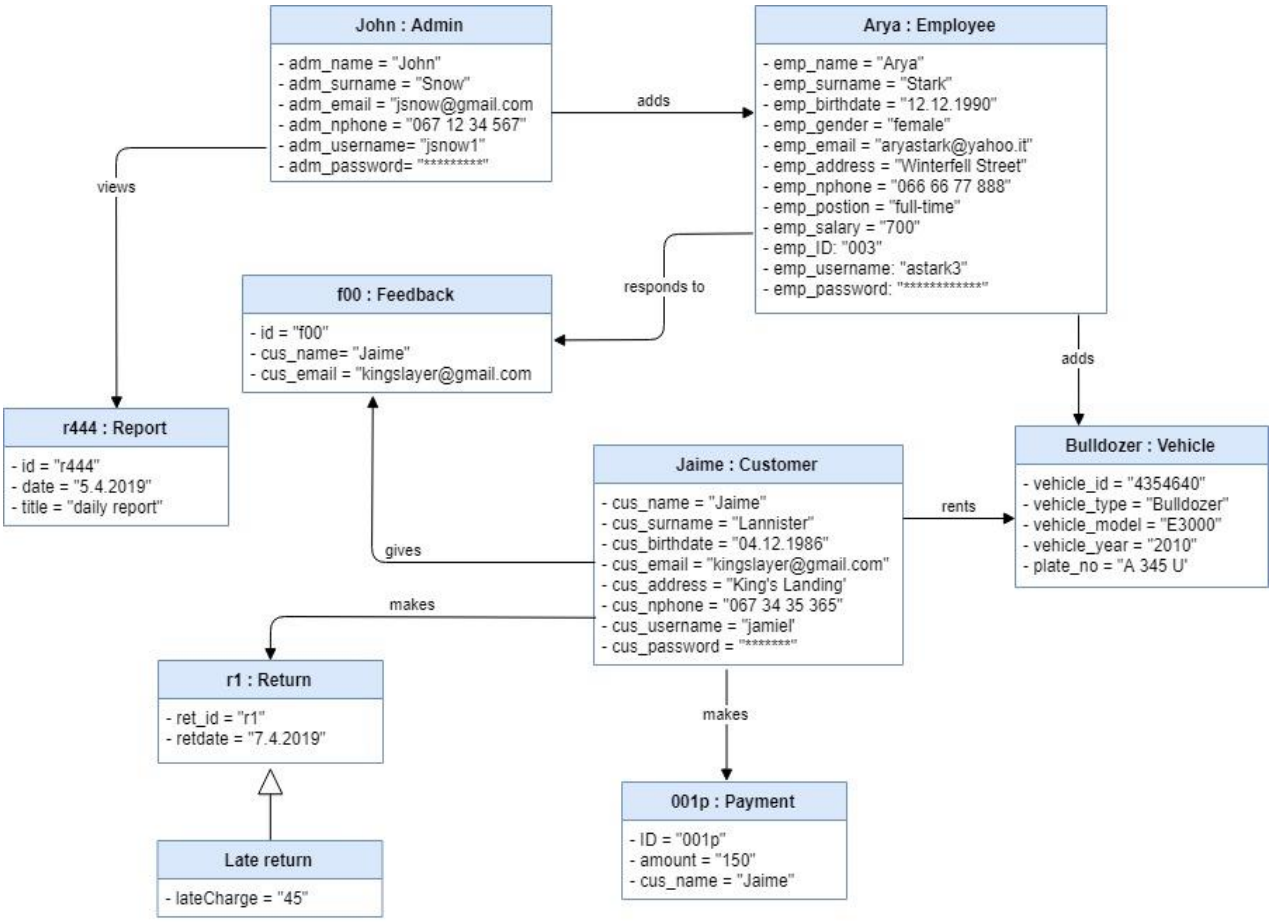


Figure 26 Object Diagram

4.8 Entity Relationship Diagram

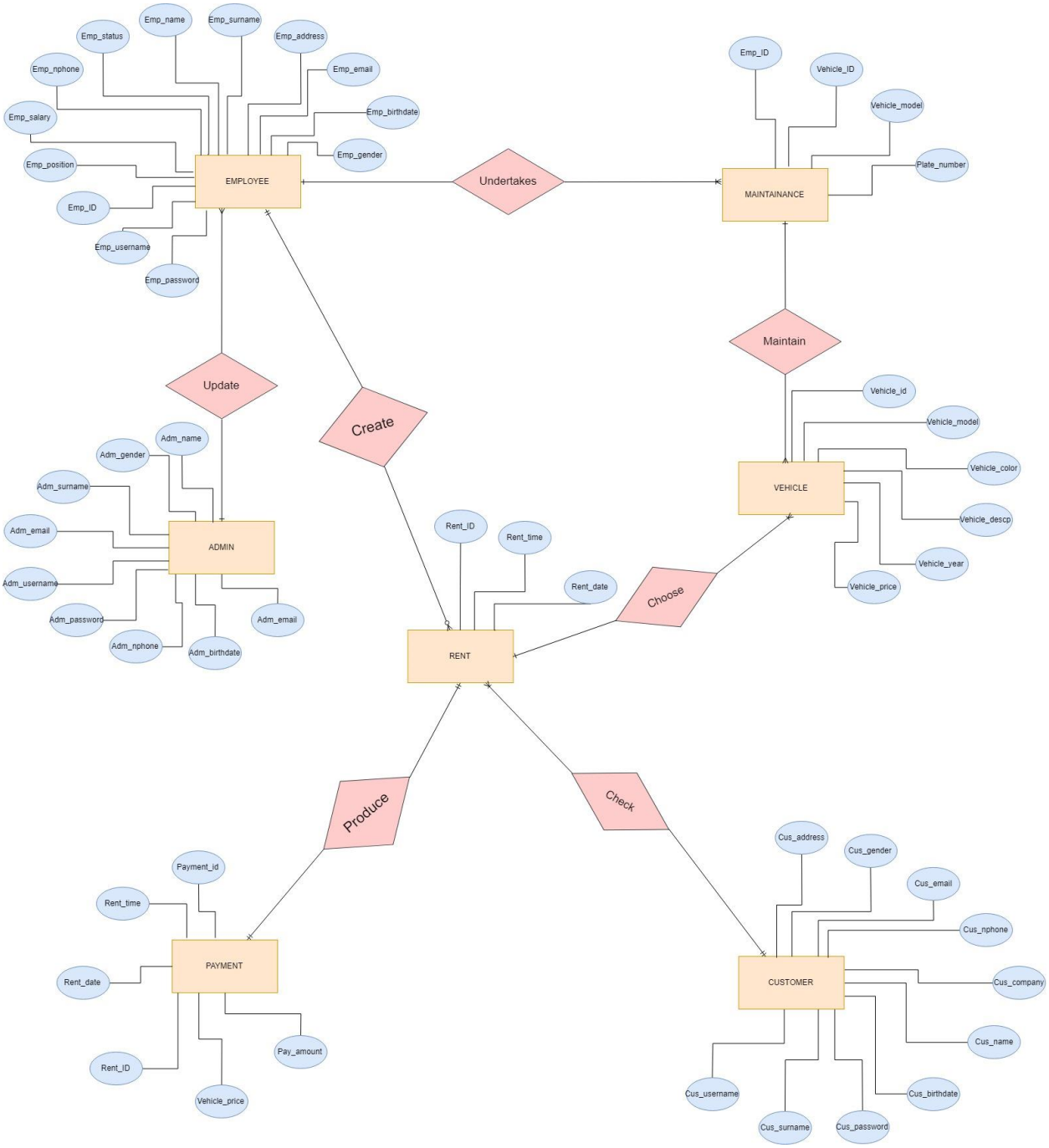


Figure 27 ERD

4.9 Data Flow Diagrams

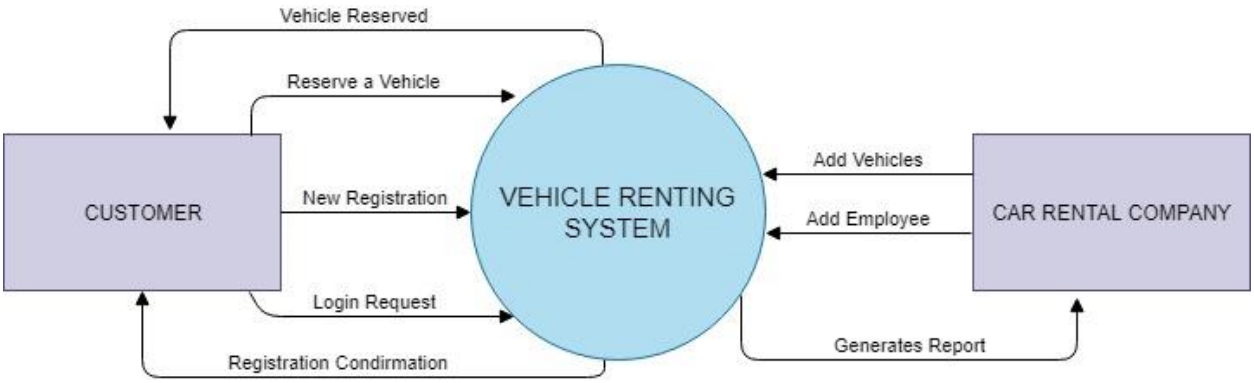


Figure 28 DFD Level 0

Figure 29 DFD Level 1

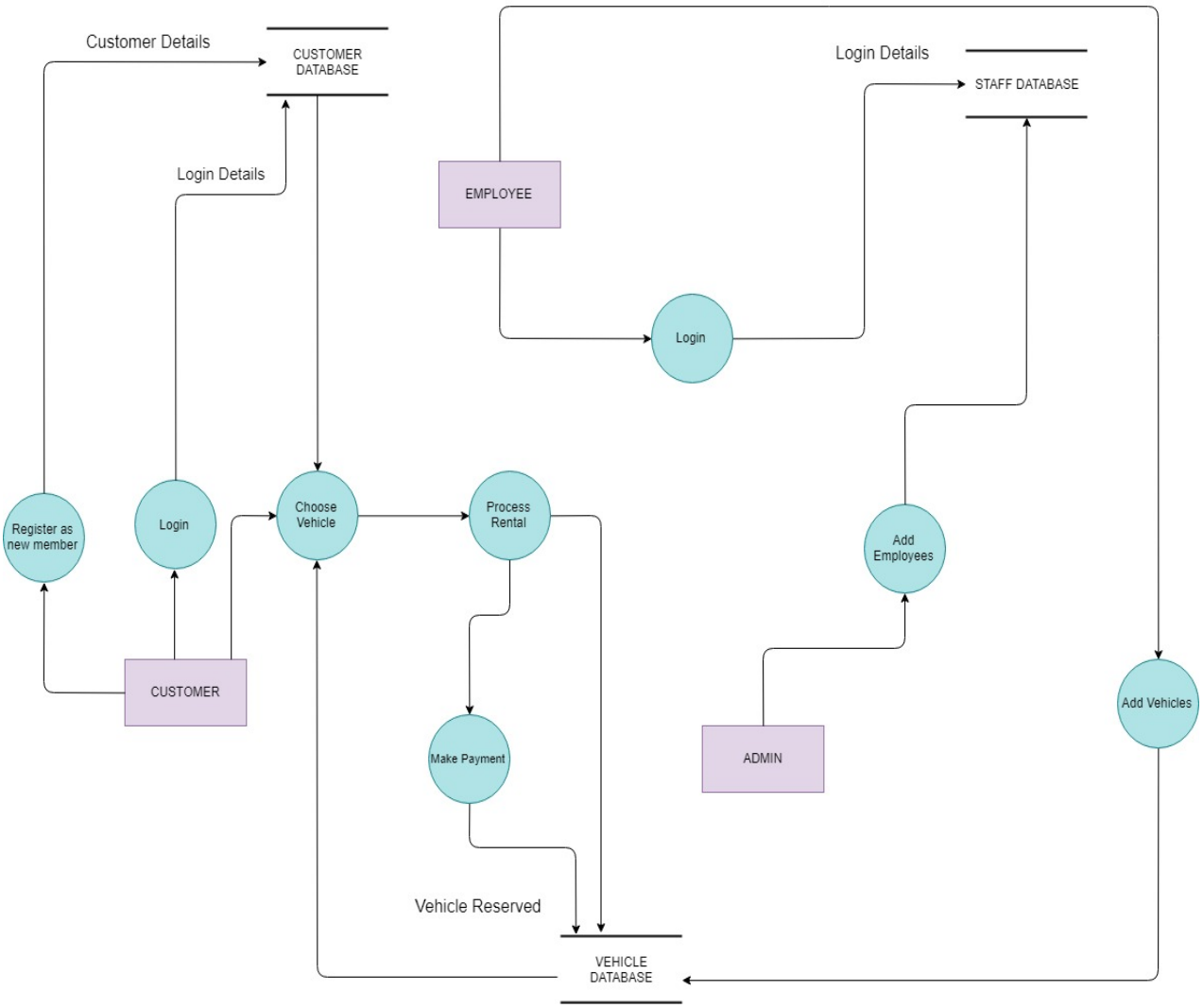


Figure 30 DFD Level 2

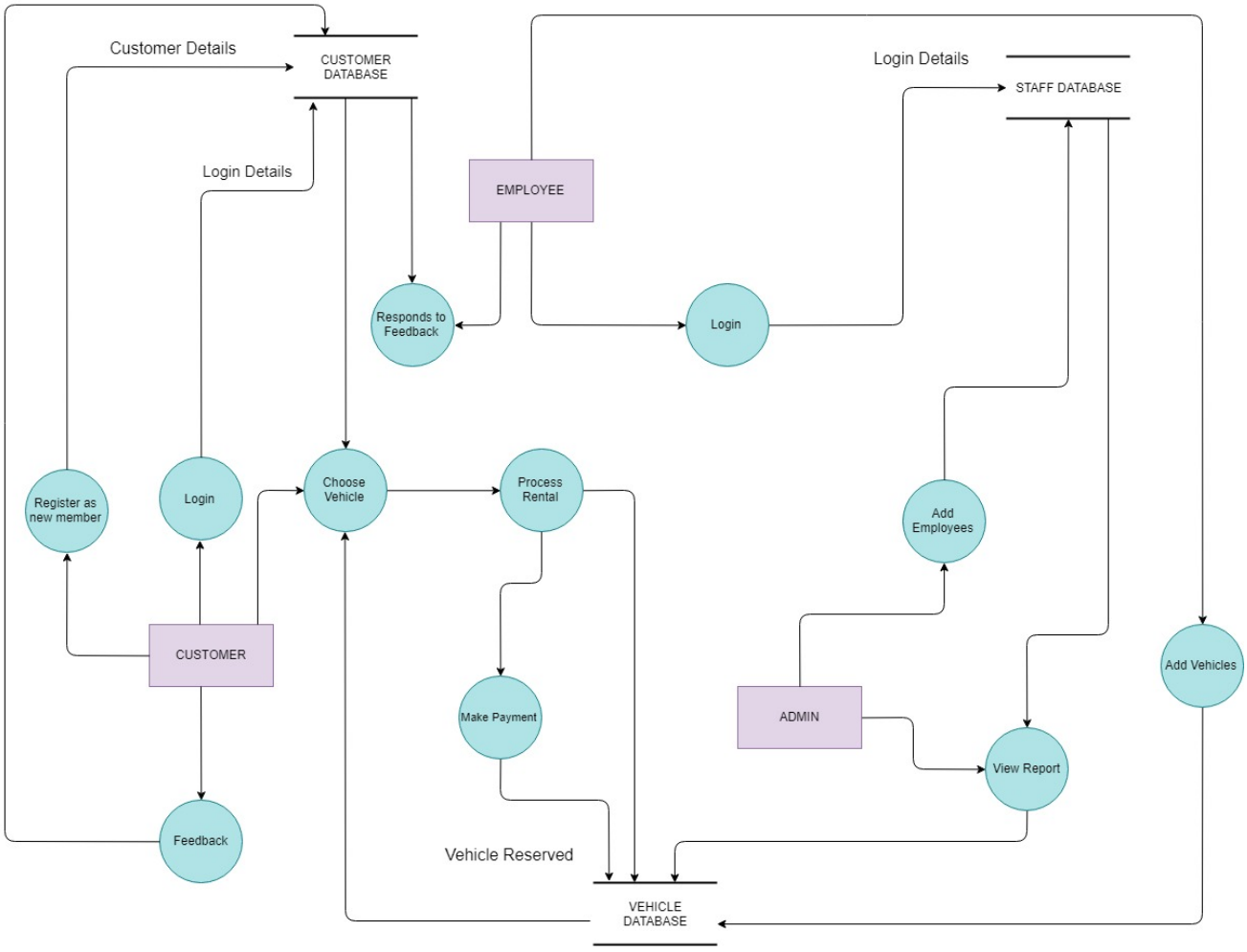
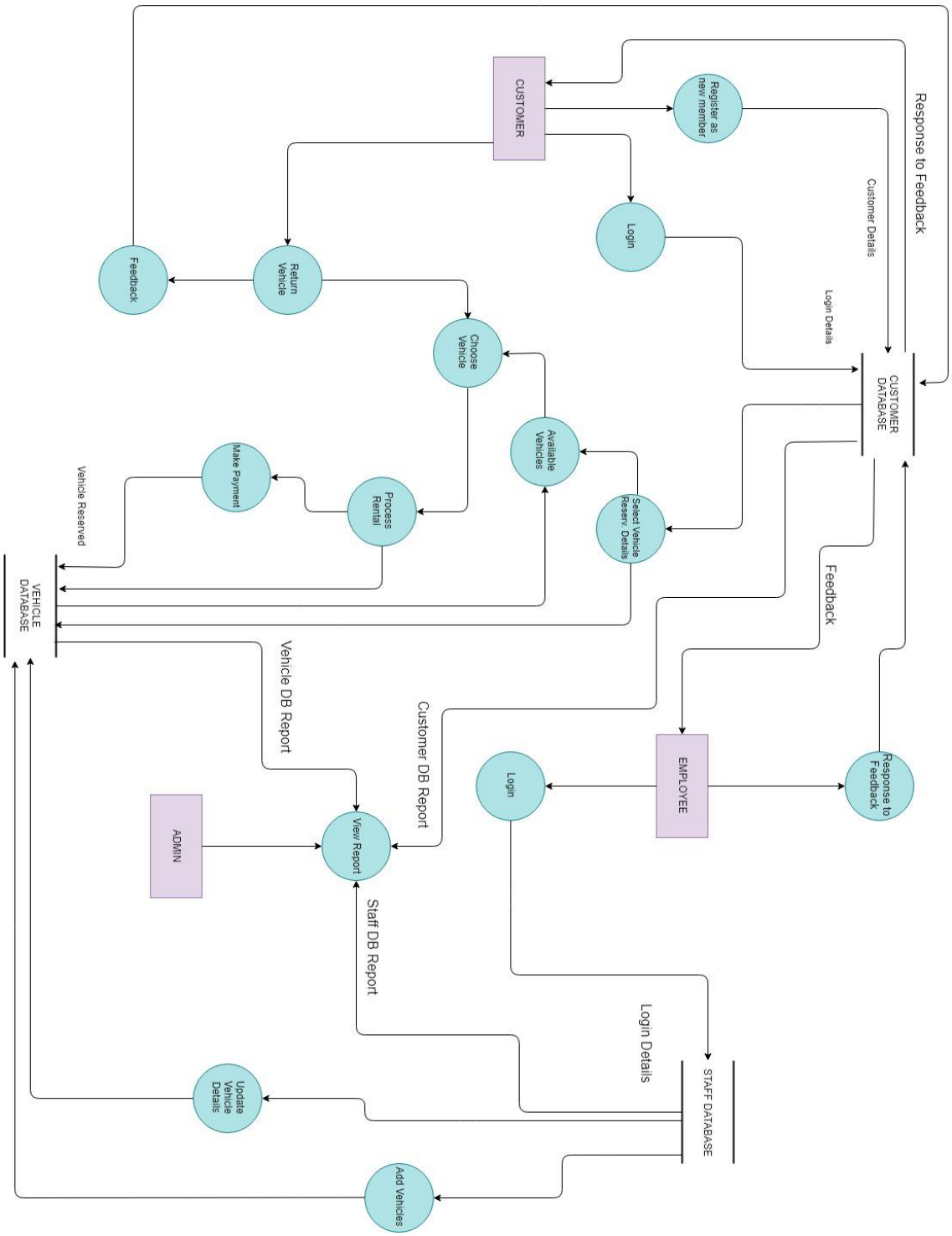


Figure 31 DFD Level 3



4.10 Collaboration Diagrams

Figure 32 Admin-Employee-Vehicle-Customer Collaboration Diagram

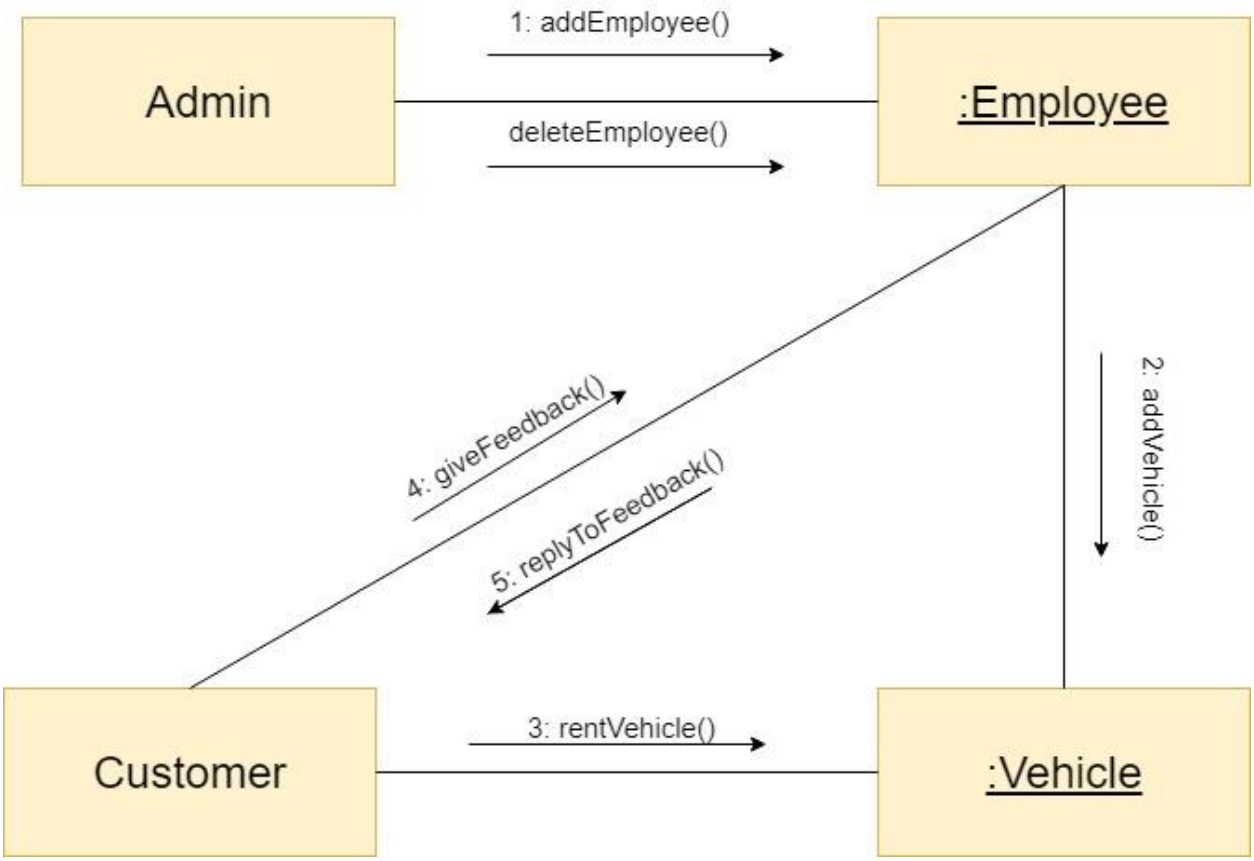


Figure 33Placing Order Collaboration Diagram

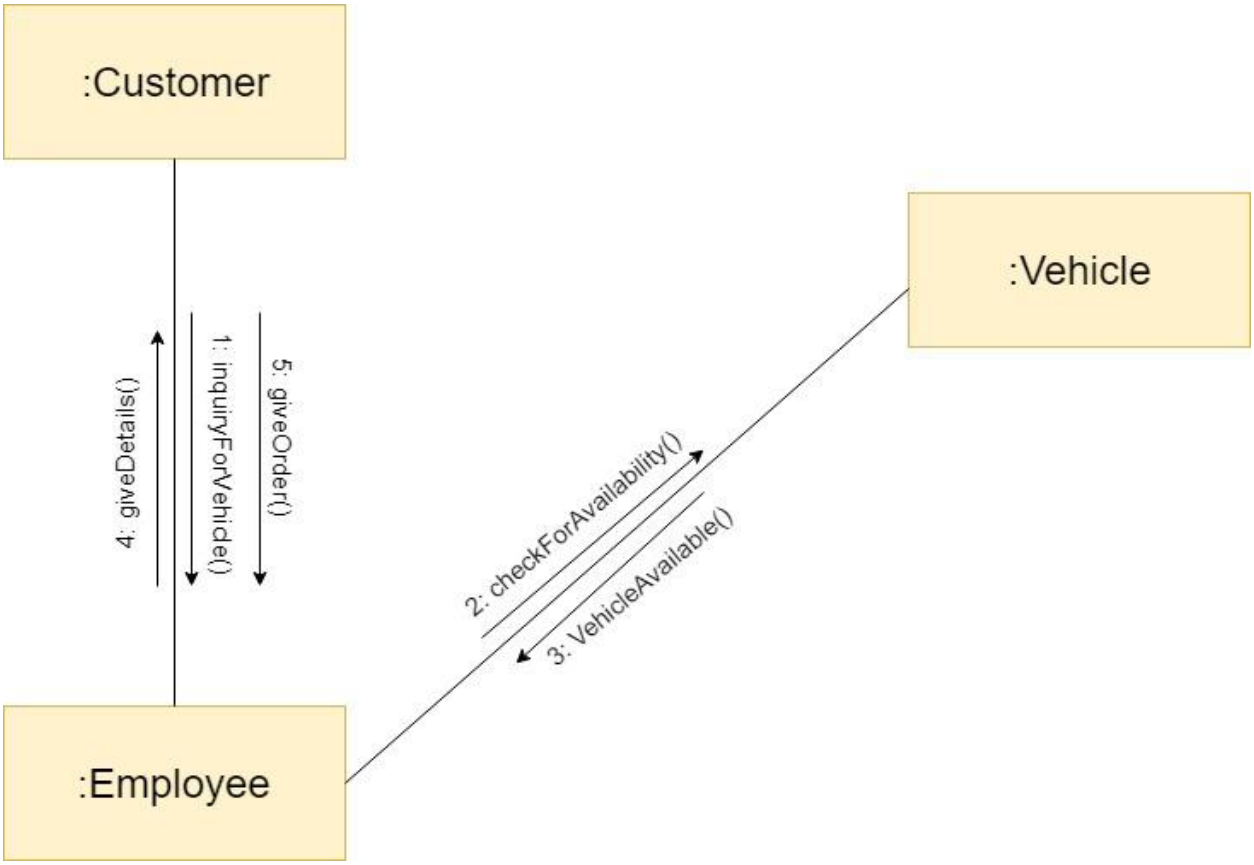
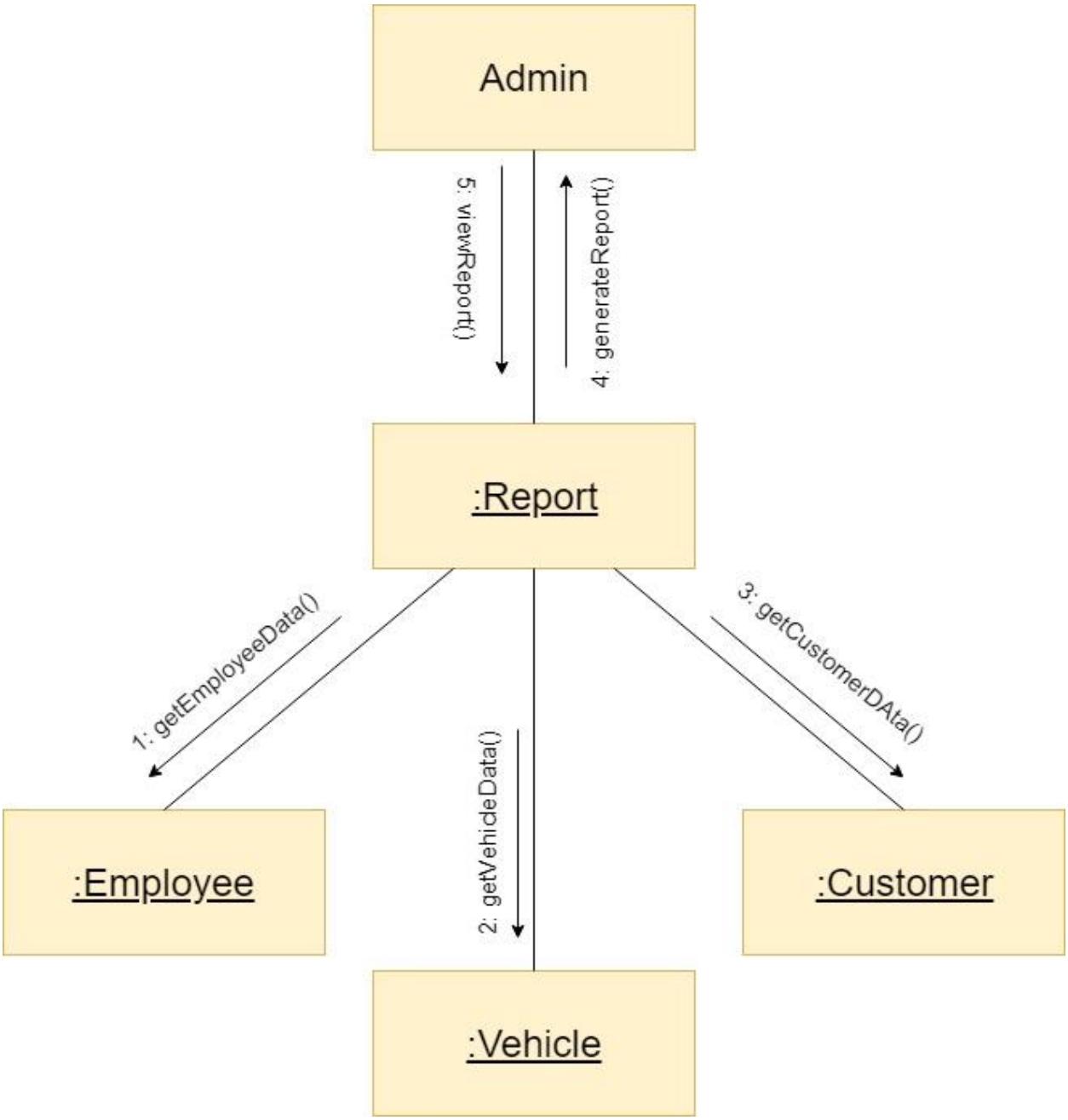


Figure 34 Report Collaboration Diagram



4.11 Component Diagram

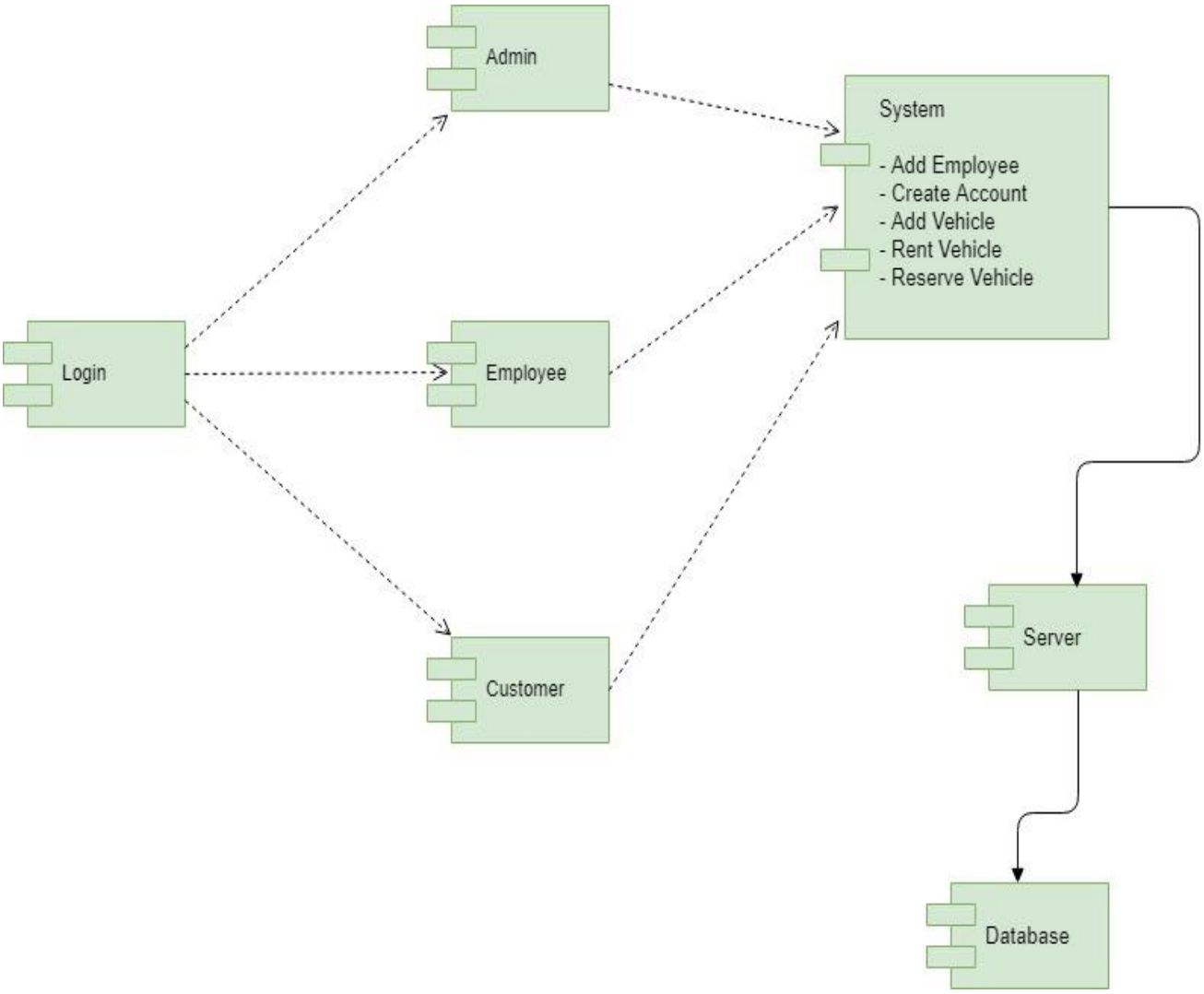


Figure 35 Component Diagram

4.12 Deployment Diagram

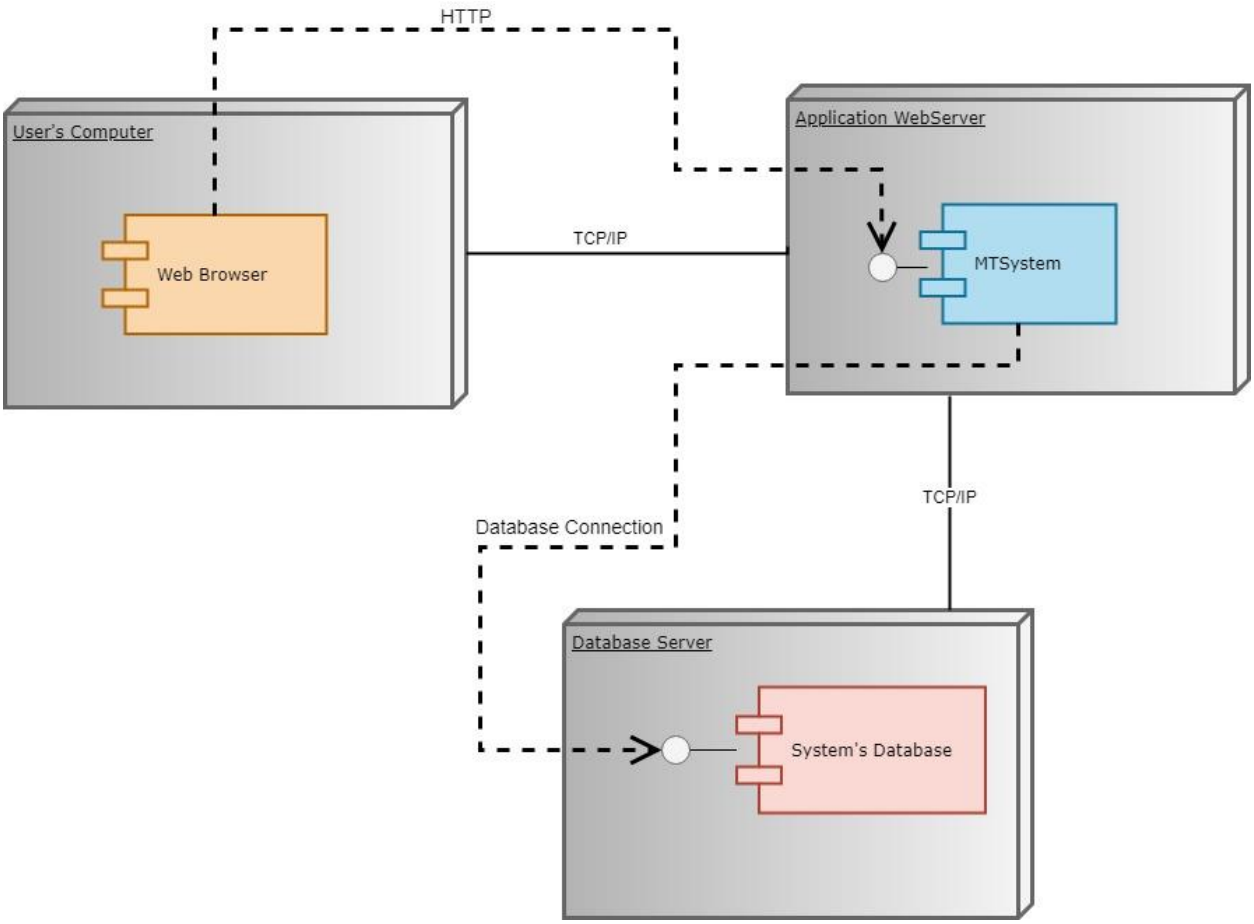


Figure 36 Deployment Diagram

TO BE CONTINUED…