Car rental franchise app

Requirements Specification

Version 1.0

April 19, 2021

**Table of Contents**

[**CAR RENTAL FRANCHISE APP REQUIREMENTS SPECIFICATION 1**](#_heading=h.19c6y18)

[**VERSION 1.0 1**](#_heading=h.3tbugp1)

[**APRIL 19, 2021 1**](#_heading=h.28h4qwu)

[**1.**](#_heading=h.nmf14n) **EXECUTIVE SUMMARY 3**

[1.1](#_heading=h.37m2jsg) Project Overview 3

[1.2](#_heading=h.1mrcu09) Purpose and Scope of this Specification 3

[**2.**](#_heading=h.46r0co2) **PRODUCT/SERVICE DESCRIPTION 3**

[2.1](#_heading=h.2lwamvv) Product Context 3

[2.2](#_heading=h.111kx3o) User Characteristics 3

[2.3](#_heading=h.3l18frh) Assumptions 3

[2.4](#_heading=h.206ipza) Constraints 3

[2.5](#_heading=h.4k668n3) Dependencies 4

[**3.**](#_heading=h.2zbgiuw) **REQUIREMENTS 4**

[3.1](#_heading=h.1egqt2p) Functional Requirements 5

[3.2](#_heading=h.3ygebqi) Non-Functional Requirements 5

[*3.2.1*](#_heading=h.2dlolyb) *Product Requirements 5*

[**3.2.1.1**](#_heading=h.44sinio) **User Interface Requirements** 6

[**3.2.1.2**](#_heading=h.2jxsxqh) **Usability** 6

[**3.2.1.3**](#_heading=h.z337ya) **Efficiency** 6

[3.2.1.3.1](#_heading=h.3j2qqm3) Performance Requirements 6

[3.2.1.3.2](#_heading=h.1y810tw) Space Requirements 6

[**3.2.1.4**](#_heading=h.4i7ojhp) **Dependability** 6

[**3.2.1.5**](#_heading=h.2xcytpi) **Security** 7

[*3.2.2*](#_heading=h.sqyw64) *Organizational Requirements 7*

[**3.2.2.1**](#_heading=h.3whwml4) **Environmental Requirements** 7

[**3.2.2.2**](#_heading=h.2bn6wsx) **Operational Requirements** 7

[**3.2.2.3**](#_heading=h.qsh70q) **Development Requirements** 7

[*3.2.3*](#_heading=h.3cqmetx) *External Requirements 7*

[**3.2.3.1**](#_heading=h.1pxezwc) **Regulatory Requirements** 7

[**3.2.3.2**](#_heading=h.49x2ik5) **Ethical Requirements** 7

[**3.2.3.3**](#_heading=h.2p2csry) **Legislative Requirements** 7

[3.2.3.3.1](#_heading=h.147n2zr) Accounting Requirements 7

[3.2.3.3.2](#_heading=h.3o7alnk) Security Requirements 7

[3.3](#_heading=h.1rvwp1q) Domain Requirements 7

[**4.**](#_heading=h.4bvk7pj) **USER SCENARIOS/USE CASES 7**

# Executive Summary

## Project Overview

The objective of this project is to create a web-based software program that can control a franchise for car rentals. With the help of the software, the franchise will be able to keep track of every vehicle in their fleet, including information on the make, model, year of manufacture, color, type of transmission, cost per day, fuel type, number of doors and seats, and more. The franchise will also be able to handle reservations, including dates, hours, and places, as well as create client invoices and contracts.

The app will help workers of a car rental franchise coordinate their work and reduce the usage of paper-based documents. An agent can store all of his transactions in dedicated spaces within the application and can share them with other agents and with administrators. Moreover, the app is directly connected to the invoice system, so every transaction is kept track of, and a bill can be issued per customer’s choice. Another crucial feature for the agents is that they can now update all the vehicles that are being rented so that there is no contradiction between customers.

In addition, our software will include other features that will facilitate the usage of the customers. One of them is the payment system which will be easy to use and completely safe for any user. The system will also support a catalog from which customers can read details for each car such as price and other specifications and make their decision. It is of paramount importance that the application maintains a reasonable ratio between being user-friendly and being professional. That being said the menus for the customers should be as friendly and as simple as possible however the views of agents and especially administrators must contain their dosage of professionalism since it is likely that the data provided to them will be used to compile official financial and legal documentation.

Key importance when developing this app will be given to data security from the creation of a secure database to the encryption of sensitive information such as employees’ SSNs, etc. The users should at first be certain about their integrity and security when using the app and then enjoy the experience that the software offers.

Project timeline: The extent and complexity of the software's features will determine the project's timeline. For development and testing, a general estimate for a project like this may be between 6 and 12 months.

Development team: A project manager, software architects, UI/UX designers, frontend developers, backend developers, database developers, quality assurance engineers, and a technical writer will likely make up the development team for this project. For GPS tracking, marketing/advertising integration, and payment processing, the team could also have to collaborate with outside providers.

## Purpose and Scope of this Specification

The purpose of this specification is to provide an evaluative insight into the current state of product design. It will also serve as a document to keep track of all the activities and processes related to the planning and drafting of each element of the project. The target audience of this specification is the development team and stakeholders such as investors or managing parties.

Within the scope of this documentation lie all aspects of the planning process discussed in detail such as:

* Features and functionalities included in the product.
* An outline of the application’s processes and interface.
* User characteristics, constraints, dependencies and assumptions. (All discussed in part 2 of the specification)
* List of requirements alongside evaluation and ranking. (Discussed in part 3)
* Full requirements’ description based on the type and importance of each requirement. (Discussed in part 3)
* Detailed information for the flow of action and flow of data within the application.
* User scenarios and descriptions of how the user will be able to operate the system. (Use case diagrams and use case explanations discussed in the last part of the specification)

There are other aspects that fall outside of the scope including:

* Financial considerations including but not limited to: financial outcome calculations, auditing, cost of each feature, etc.
* Issues related to product liability such as terms and agreements. (Note that we do refer to the inclusion of terms and agreements as a legal requirement, but the content of such legislative documentation will not be part of the project specification)
* Anything related to product marketing

Note: All the items that are not within the boundaries of the scope are either already dealt with by other parties included in the project or are not a direct responsibility of the project development team.

# Product/Service Description

## Product Context

This software application intends to facilitate and improve the process of car renting and the interaction of the car rental franchise with the customers. There are many other apps with the same purpose in the current market, however, we as a team have designed this software with the goal of gathering all the available features that have proven to be useful and desirable by the user while also adding new functionalities that will distinguish our app from others and revolutionize the involvement of technology in the car rental domain.

The software has a user-friendly GUI which is easy to use and compatible with all devices in the market. The fairly simple design of the interface is certain to help workers adapt to using it and to attract more clients since it does not require much concentration to be used.

For the customers, it will provide interactive menus from which they can make their reservations quickly and safely. While for other users such as administrators and agents, the focus will be on the easiness of communication within the organization. The system will provide a grid of messages and notifications to keep them up to date and will provide commands that allow them to inform each other in real-time.

## User Characteristics

**Administrator:**

* Has the authority to manage everything that agents and drivers do.
* Has the authority to manage employees’ information.
* Can view data about total incomes and total costs.
* Needs access to data about all the cars, reservations, and customer information.
* Changes the prices of the cars and other specifications.
* Adds and removes cars from the catalog.

**Car Rental Agent:**

* Needs access to data about available cars and reservations.
* Can issue invoices and rental contracts.
* Needs access to GPS device data for tracking cars.
* Receives notifications about the day’s reservations and rented cars.
* Needs to check for fines and ensure customers pay them.
* Notifies other agents when a car becomes unavailable.

**Customer:**

* Needs to log in or sign up to make a reservation.
* Can see available cars and make reservations through the website.
* Needs to provide personal information, such as name, surname, birthday, ID, email, and phone number.
* Can pay through credit card.
* Is obliged to return the car in good condition and pay for any fines incurred.
* Notifies the agency for any problems he might have with the car  
  .
* Can inquire information about prices and cars specifications.
* May have different levels of subscription (regular, premium, or platinum) with different rental options and extra features

**Driver:**

* Needs access to reservation information.
* Can check off whether the car has been delivered or not.
* Is responsible for checking the car before and after delivering it to the customer and ensuring it is in good condition.
* Reports any damage or issue to the administrator.
* Reports about any investment that is needed for any of the cars.

## Assumptions

1. It is assumed that the staff and customers are familiar with the English or Albanian language.
2. The car rental software application assumes the availability of a stable and reliable internet connection for both the staff and customers to access the system.
3. It is assumed that the application will be available on popular mobile devices with the latest iOS and Android operating systems.
4. The application assumes that the staff and customers have basic knowledge of operating the application and troubleshooting any issues that may arise.
5. It is assumed that the car rental company will have a valid and up-to-date insurance policy for all their vehicles and customers, and that customers will be made aware of the insurance coverage and limitations.
6. It is assumed that the car rental company will have enough vehicles available to meet the demand of their customers during peak times.
7. It is assumed that the car rental company will have a customer support system in place to address any concerns or issues that customers may have while using the application or renting a vehicle.
8. The application assumes that the customers will return the rented car on time and in the same condition as when it was rented, and that any damages or additional charges will be promptly communicated and resolved.

## Constraints

The system will have the following constraints:

* Every manager, agent, driver, or customer must log in and/or register with a username and password.
* The software should be able to integrate with other systems such as financial ones so that the customers can pay online for the reservations.
* To make a reservation, the customer needs to create an account or log into an existing one.
* The software should be able to properly handle a huge number of customers, reservations, and other operations, especially during busy season.
* The software should be secure and protect the data of the customers, reservations, and payment history.
* It should provide backup in case of crashes and errors.

## Dependencies

1. In order to print the bill contract, the car rental agent must enter the information of the customer and the available car.
2. If a car is not available, the car rental agent cannot make a reservation for it. In order to make a rental reservation, the customer needs to log in, pick an available car and pay for the provided service.
3. If a chosen car is not currently available, the customer may choose another car for reservation.
4. In order to finalize the return of the vehicle, the customer must pay every fine placed on the vehicle.
5. Customers are dependent on their subscription so that they can pick a certain car model.
6. The customer cannot pick any car they want if they do not have a subscription on the service.
7. In order to receive a discount on the next rental, the customer must have made reservations at least 5 times.
8. A car cannot be available for reservations, if the driver has confirmed it is under use by another customer.
9. The car’s availability is dependent on the condition of the car post-rental, checked by the driver, in order to be available for another time.

# ***Requirements***

* Describe all system requirements in enough detail for designers to design a system satisfying the requirements and testers to verify that the system satisfies requirements.
* Organize these requirements in a way that works best for your project. See Appendix DAppendix D, Organizing the Requirements for different ways to organize these requirements.
* Describe every input into the system, every output from the system, and every function performed by the system in response to an input or in support of an output. (Specify what functions are to be performed on what data to produce what results at what location for whom.)
* Each requirement should be numbered (or uniquely identifiable) and prioritized.

See the sample requirements in Functional Requirements, and System Interface/Integration, as well as these example priority definitions:

**Priority Definitions**

The following definitions are intended as a guideline to prioritize requirements.

* Priority 1 – The requirement is a “must have” as outlined by policy/law
* Priority 2 – The requirement is needed for improved processing, and the fulfillment of the requirement will create immediate benefits
* Priority 3 – The requirement is a “nice to have” which may include new functionality

It may be helpful to phrase the requirement in terms of its priority, e.g., "The value of the employee status sent to DIS **must be** either A or I" or "It **would be nice** if the application warned the user that the expiration date was 3 business days away". Another approach would be to group requirements by priority category.

* A good requirement is:
* Correct
* Unambiguous (all statements have exactly one interpretation)
* Complete (where TBDs are absolutely necessary, document why the information is unknown, who is responsible for resolution, and the deadline)
* Consistent
* Ranked for importance and/or stability
* Verifiable (avoid soft descriptions like “works well”, “is user friendly”; use concrete terms and specify measurable quantities)
* Modifiable (evolve the Requirements Specification only via a formal change process, preserving a complete audit trail of changes)
* Does not specify any particular design
* Traceable (cross-reference with source documents and spawned documents).

## Functional Requirements

In the example below, the requirement numbering has a scheme - BR\_LR\_0## (BR for Business Requirement, LR for Labor Relations). For small projects simply BR-## would suffice. Keep in mind that if no prefix is used, the traceability matrix may be difficult to create (e.g., no differentiation between '02' as a business requirement vs. a test case)

The following table is an example format for requirements. Choose whatever format works best for your project.

For Example:

| **Req#** | **Requirement** | **Comments** | **Priority** | **Date Rvwd** | **SME Reviewed / Approved** |
| --- | --- | --- | --- | --- | --- |
| BR\_LR\_05 | The system should associate a supervisor indicator with each job class. | Business Process = “Maintenance | 3 | 7/13/04 | Bob Dylan, Mick Jagger |
| BR\_LR\_08 | The system should handle any number of fees (existing and new) associated with unions. | Business Process = “Changing Dues in the System”  An example of a new fee is an initiation fee. | 2 | 7/13/04 | Bob Dylan, Mick Jagger |
| BR\_LR\_10 | The system should capture and maintain job class status (i.e., active or inactive) | Business Process = “Maintenance”  Some job classes are old and are no longer used. However, they still need to be maintained for legal, contract and historical purposes. | 2 | 7/13/04 | Bob Dylan, Mick Jagger |
| BR\_LR\_16 | The system should assign the Supervisor Code based on the value in the Job Class table and additional criteria as specified by the clients. | April 2005 – New requirement. It is one of three new requirements from BR\_LR\_03. | 2 |  |  |
| BR\_LR\_18 | The system should provide the Labor Relations office with the ability to override the system-derived Bargaining Unit code and the Union Code for to-be-determined employee types, including hourly appointments. | April 2005 – New requirement. It is one of three new requirements from BR\_LR\_04.  5/11/2005 – Priority changed from 2 to 3. | ~~2~~  3 |  |  |

## Non-Functional Requirements

**In here try to use the Structure given at slide 13 in Requirements Engineering Lecture Slides, with main categories of:**

### Product Requirements

* + Requirements which specify that the delivered product must behave in a particular way e.g. execution speed, reliability, etc.

#### **User Interface Requirements**

In addition to functions required, describe the characteristics of each interface between the product and its users (e.g., required screen formats/organization, report layouts, menu structures, error and other messages, or function keys).

#### **Usability**

Include any specific usability requirements, for example,

Learnability

* The user documentation and help should be complete
* The help should be context sensitive and explain how to achieve common tasks
* The system should be easy to learn

(See <http://www.usabilitynet.org/>)

#### **Efficiency**

##### Performance Requirements

Specify static and dynamic numerical requirements placed on the system or on human interaction with the system:

* Static numerical requirements may include the number of terminals to be supported, the number of simultaneous users to be supported, and the amount and type of information to be handled.
* Dynamic numerical requirements may include the number of transactions and tasks and the amount of data to be processed within certain time period for both normal and peak workload conditions.

All of these requirements should be stated in measurable form. For example, "95% of the transactions shall be processed in less than 1 second" rather than “an operator shall not have to wait for the transaction to complete”.

##### Space Requirements

#### **Dependability**

**Availability**

Include specific and measurable requirements for:

* Hours of operation
* Level of availability required
* Coverage for geographic areas
* Impact of downtime on users and business operations
* Impact of scheduled and unscheduled maintenance on uptime and maintenance communications procedures
* reliability (e.g., acceptable mean time between failures (MTBF), or the maximum permitted number of failures per hour).

**Reliability**

**Monitoring**

Include any requirements for product or service health monitoring, failure conditions, error detection, logging, and correction.

**Maintenance**

Specify attributes of the system that relate to ease of maintenance. These requirements may relate to modularity, complexity, or interface design. Requirements should not be placed here simply because they are thought to be good design practices.

**Integrity**

#### **Security**

Specify the factors that will protect the system from malicious or accidental access, modification, disclosure, destruction, or misuse. For example:

* encryption
* activity logging, historical data sets
* restrictions on intermodule communications
* data integrity checks

Specify the Authorization and Authentication factors. Consider using standard tools such as PubCookie.

### Organizational Requirements

Requirements which are a consequence of organisational policies and procedures e.g. process standards used, implementation requirements, etc

#### **Environmental Requirements**

#### **Operational Requirements**

#### **Development Requirements**

### External Requirements

* + Requirements which arise from factors which are external to the system and its development process e.g. interoperability requirements, legislative requirements, etc.

#### **Regulatory Requirements**

#### **Ethical Requirements**

#### **Legislative Requirements**

Specify the requirements derived from existing standards, policies, regulations, or laws (e.g., report format, data naming, accounting procedures, audit tracing). For example, this could specify the requirement for software to trace processing activity. Such traces are needed for some applications to meet minimum regulatory or financial standards. An audit trace requirement may, for example, state that all changes to a payroll database must be recorded in a trace file with before and after values

##### Accounting Requirements

##### Security Requirements

## Domain Requirements

Everything related to the domain that might be needed in the project shall be mentioned here. Sometimes the domain Requirements might be thought of as part of either functional or non-functional requirements.

Please provide all necessary non-functional requirements, similar to the requirements explained in the lesson slides or in the textbook.

# User Scenarios/Use Cases

* 1. **Use case diagrams**

Diagram

Description automatically generated

|  |  |
| --- | --- |
| Use Case (UC\_1.1): | *Log In* |
| Summary | The agents need to log in the system to later use it. |
| Actors | Primary Actor: agent |
| Preconditions | The agent should open the website |
| Description of the Main Sequence | * *Step 1: Enter the username.* * *Step 2: Enter the password.* * *Step 3: Press the log in button.* * *Step 4: The agent is logged in successfully.* |
| Description of the Alternative Sequence | * *Step 1: Enter the username.* * *Step 2: Enter the password.* * *Step 3: Press the log in button.* * *Step 4: The agent cannot log in.* |
| Nonfunctional requirements | *The device needs to be connected to the internet.*  *The system must be secure from unauthorized access.* |
| Postconditions | The agent should be logged in |

|  |  |
| --- | --- |
| Use Case (UC\_1.2): | *View available cars* |
| Summary | *The agent can view the available cars in the system.* |
| Actors | *Primary actor: Agent* |
| Preconditions | *Agent is logged in the system.* |
| Description of the Main Sequence | * *Step 1: After logging in, the agent clicks the button to view the cars.* * *Step 2: They put in the time period they want to look for.* * *Step 3: The system shows the cars available for that time period.* |
| Description of the Alternative Sequence | * *Step 1: After logging in, the agent clicks the button to view the cars.* * *Step 2: They put in the time period they want to look for.* * *Step 3: There are no available cars for that time period.* |
| Nonfunctional requirements | *The device needs to be connected to the internet.*  *The page must load within 2 seconds.* |
| Postconditions | *The agent has access to the available cars.* |

|  |  |
| --- | --- |
| Use Case (UC\_1.3): | *Makes reservations* |
| Summary | *The agent should be able to make reservations for the customers.* |
| Actors | *Primary Actor: Agent* |
| Preconditions | *The agent should have a customer asking for a reservation.* |
| Description of the Main Sequence | * *Step 1: The agent gets the information from the customer.* * *Step 2: They open the reservation section on the website.* * *Step 3: They choose the car and the time period.* * *Step 4: They fill in the required information.* * *Step 5: They save the reservation.* |
| Description of the Alternative Sequence | * *Step 1: The agent gets the information from the customer.* * *Step 2: They open the reservation section on the website.* * *Step 3: They choose the car and the time period.* * *Step 4: The car is not available, and the agent gets a notification and does not continue the reservation.* |
| Nonfunctional requirements | *The system should be able to maintain a huge number of reservations.*  *The reservation should be saved within 3 seconds.* |
| Postconditions | *The reservation is saved on the system.* |

|  |  |
| --- | --- |
| Use Case (UC\_1.4): | *Change the status of reservation to paid/unpaid* |
| Summary | *After making the reservation, the agent should change the status to paid when it gets paid or leave it unpaid otherwise.* |
| Actors | *Primary actor: Agent* |
| Preconditions | *There should be a reservation, so that the agent can change its status.* |
| Description of the Main Sequence | * *Step 1: The agent gets the payment from the customer.* * *Step 2: The agent changes the status to paid.* |
| Description of the Alternative Sequence | * *Step 1: If the customer decides to extend the rental period, the reservation status should become unpaid, until the extension is paid for.* |
| Nonfunctional requirements | *The system should save the changes within 3 seconds.* |
| Postconditions | *The reservation is showed in the system as paid.* |

|  |  |
| --- | --- |
| Use Case (UC\_1.5): | *Create and print bills* |
| Summary | *The agent should create the bill for a certain reservation and print it for the customer.* |
| Dependency | *This use case depends on making reservation use case and changing the status of reservation to paid or unpaid. In order to create a bill, you should first have a reservation and the reservation should be paid or should be to be paid. This is why the use case is included in those use cases.* |
| Actors | *Primary actor: Agent* |
| Preconditions | *There should be a reservation and the customer should be paying for the reservation.* |
| Description of the Main Sequence | * *Step 1: The agent clicks the button that creates a bill after making a reservation.* * *Step 2: The website itself calculates the amount to be paid and creates the bill in the programmed format.* * *Step 3: The agent prints the bill and hand it to the customer.* |
| Description of the Alternative Sequence | * *The agent clicks the button that creates a bill after making a reservation.* * *Step 2: The website cannot create the bill because of a certain problem.* |
| Nonfunctional requirements | *The system should be accurate in order to create a correct bill with the correct amount.*  *The system should show the amount in euros and leke.* |
| Postconditions | *The bill is created and printed out. The customer has a detailed bill for the services.* |

|  |  |
| --- | --- |
| Use Case (UC\_1.6): | *Print and sign the contract* |
| Summary | *After making the reservation, the system provides a contract with every legal issue regarding renting the car. The agent should print it, sign it and give the customer to sign it. There should be two copies of the contract, one for the agency and the other for the customer.* |
| Dependency | *This use case depends on making reservation use case. In order to print the contract, you should first have a reservation. This is why it is included in making reservation use case.* |
| Actors | *Primary Actor: Agent, Customer* |
| Preconditions | *There should be a reservation.* |
| Description of the Main Sequence | * *Step 1: The agent clicks the button to print the contract.* * *Step 2: The website creates the contract based on the programmed format.* * *Step 3: The contract is printed and signed by both sides.* |
| Description of the Alternative Sequence | * *Step 1: The agent clicks the button to print the contract.* * *Step 2: The website cannot create the contract.* |
| Non functional requirements | *Narrative description of nonfunctional requirements, such as performance and security requirements.* |
| Postconditions | *The contract is created and printed out. The customer has a detailed legal information about the services provided.* |

|  |  |
| --- | --- |
| Use Case (UC\_1.7): | *Check the location of the cars through GPS* |
| Summary | *In order to always have knowledge about the cars and make sure that the customers are not misusing them, the agents should be able to check the location of the cars through GPS installed in every car.* |
| Dependency | *To make this use case work, the website should have embedded the software provided by the GPS company to check the car location.* |
| Actors | *Primary Actor: Agent* |
| Preconditions | *The cars should have GPS installed in them.* |
| Description of the Main Sequence | * *Step 1: The agent has access to the embedded system by clicking the appropriate button.* * *Step 2: The agent searches for the desired car and checks the location.* |
| Description of the Alternative Sequence | * *Step 1: The agent has access to the embedded system by clicking the appropriate button.* * *Step 2: The GPS is off, or the GPS system is down.* |
| Nonfunctional requirements | *It should be available 24 hours.* |
| Postconditions | *The agent is provided with the information about the car location.* |

|  |  |
| --- | --- |
| Use Case (UC\_1.8): | *Filter the cars according to different characteristics* |
| Summary | *When searching for a car, the agents should be able to filter the cars using car model, year of production, color, price etc. according to their needs.* |
| Actors | *Primary Actor: Agent* |
| Preconditions | *The system should keep information about the car details to be able to filter cars according to them.* |
| Description of the Main Sequence | * *Step 1: The agent clicks the filter button and chooses the characteristic they want to find.* * *Step 2: Navigate through the generated responses.* |
| Description of the Alternative Sequence | * *Step 1: The agent clicks the filter button and chooses the characteristic they want to find.* * *Step 2: There is no car with those characteristics.* |
| Nonfunctional requirements | *Easily navigable.* |
| Postconditions | *The agent is provided with the cars list that have the required characteristic.* |

|  |  |
| --- | --- |
| Use Case (UC\_1.9): | *Access customer’s information* |
| Summary | *The agent may need to access customer’s information such as phone number that are obtained during the reservation.* |
| Actors | *Primary Actor: Agent* |
| Preconditions | *The customer must have done a reservation in order to access their information.* |
| Description of the Main Sequence | * *Step 1: The agent clicks the details of the reservation.* * *Step 2: It accesses the information given to them during the filling of the reservation form.* |
| Description of the Alternative Sequence | * *The reservation is deleted so there is no other way to access the information.* |
| Nonfunctional requirements | *If the reservation is cancelled or deleted, the system should delete the customer’s information.* |
| Postconditions | *The agent is provided with the needed information.* |

|  |  |
| --- | --- |
| Use Case (UC\_1.10): | *Delete a reservation* |
| Summary | *If the customer cancels the reservation or for some reasons, it is impossible to provide the service required, the agent should be able to delete the reservation.* |
| Actors | *Primary Actor: Agent* |
| Preconditions | *In order to delete the reservation, it should firstly exist and there should be a reason to delete it.* |
| Description of the Main Sequence | * *Step 1: The agent clicks the details of the reservation.* * *Step 2: By clicking the delete button at the end, the reservation is deleted.* |
| Description of the Alternative Sequence | * *Step 1: The agent clicks the details of the reservation.* * *Step 2: The reservation cannot be deleted.* |
| Nonfunctional requirements | *The deletion of the reservation should not affect the other reservations.* |
| Postconditions | *The reservation is deleted from the system.* |

|  |  |
| --- | --- |
| Use Case (UC\_1.11): | *Check for fines* |
| Summary | *During the rental periods, it Is possible that the customers can get a fine for breaking the law. This is why the agent should check for fines and get the money from the customer when he delivers the car at the end of the rental period.* |
| Dependency | *To make this use case work, the website should have embedded software to check the fines provided by the state.* |
| Actors | *Primary Actor: Agent* |
| Preconditions | *There should be cars rented for that period.* |
| Description of the Main Sequence | * *Step 1: The agent clicks the button that gives them access to check the fines.* * *Step 2: The agent should search by the license plate.* * *Step 3: They should keep a note of the cars with fines.* |
| Description of the Alternative Sequence | * *Step 1: The agent clicks the button that gives them access to check the fines.* * *Step 2: The agent should search by the license plate.* * *Step 3: No car has fines, or the embedded system is down.* |
| Nonfunctional requirements | *The software should be up to date and available 24 hours.* |
| Postconditions | *The agent has information about the fines and the customers pay for the fines.* |

|  |  |
| --- | --- |
| Use Case (UC\_1.12): | *Log Out* |
| Summary | *The agent at the end of the shift logs out of the system.* |
| Actors | *Primary Actor: Agent* |
| Preconditions | *The agent should be logged in the system.* |
| Description of the Main Sequence | * *Step 1: The agent goes to their profile and logs out successfully by clicking the log out button.* |
| Description of the Alternative Sequence | * *Step 1: The agent fails to log out.* |
| Nonfunctional requirements | *The device needs to be connected to the internet.* |
| Postconditions | *The agent is logged out of the website.* |

Diagram

Description automatically generated

|  |  |
| --- | --- |
| Use Case (UC\_2.1): | *Log In /Sign in* |
| Summary | The customers need to log in or sign in the system to later use it. |
| Actors | *Primary Actor: Customer* |
| Preconditions | The customer should open the website. |
| Description of the Main Sequence | * *Step 1: Enter the username.* * *Step 2: Enter the password/ other required fields.* * *Step 3: Press the log in/sign in button.* * *Step 4: The agent is logged in successfully.* |
| Description of the Alternative Sequence | * *Step 1: Enter the username.* * *Step 2: Enter the password.* * *Step 3: Press the log in button.* * *Step 4: The customer could not log in.* |
| Nonfunctional requirements | *The device needs to be connected to the internet.*  *The system must be secure from unauthorized access.* |
| Postconditions | *The customer should be logged in/ signed in.* |

|  |  |
| --- | --- |
| Use Case (UC\_2.2): | *Make reservation* |
| Summary | *The customer should be able to make reservations through the website.* |
| Dependency | *This optional section describes whether the UC depends on other UCs.* |
| Actors | *Primary Actor: Customer* |
| Preconditions | *The customer should have chosen the car.* |
| Description of the Main Sequence | * *Step 1: They open the reservation section on the website.* * *Step 2: They choose the car and the time period.* * *Step 3: They fill in the required information.* * *Step 4: They save the reservation.* |
| Description of the Alternative Sequence | * *Step 1: They open the reservation section on the website.* * *Step 2: They choose the car and the time period.* * *Step 3: The car is not available and the customer does not continue the reservation.* |
| Nonfunctional requirements | *The system should be able to maintain a huge number of reservations.*  *The reservation should be saved within 3 seconds.* |
| Postconditions | *The reservation is saved on the system.* |

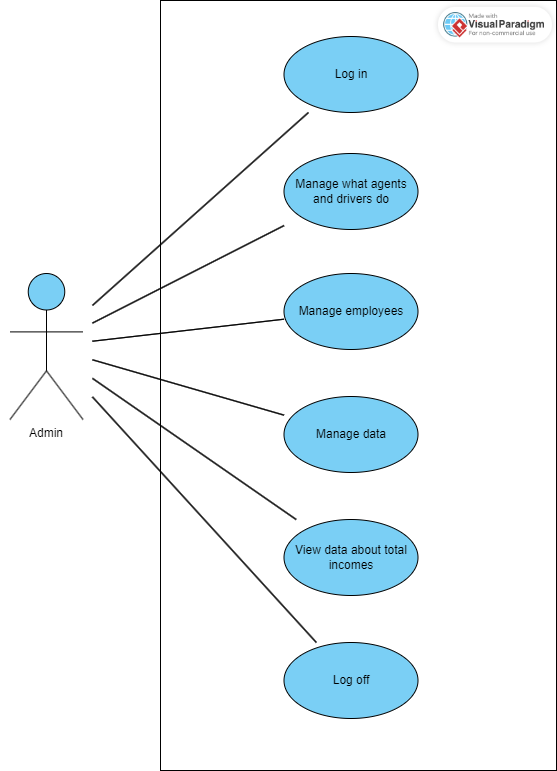
|  |  |
| --- | --- |
| Use Case (UC\_2.3): | *View available cars and services* |
| Summary | *The customer can view the available cars and services in the system.* |
| Actors | *Primary actor: Customer* |
| Preconditions | *Cutomer is logged in the system.* |
| Description of the Main Sequence | * *Step 1: After logging in, the customer clicks the button to view the cars.* * *Step 2: They put in the time period they want to look for.* * *Step 3: The system shows the cars available for that time period.* |
| Description of the Alternative Sequence | * *Step 1: After logging in, the customer clicks the button to view the cars.* * *Step 2: They put in the time period they want to look for.* * *Step 3: There are no available cars for that time period.* |
| Nonfunctional requirements | *The device needs to be connected to the internet.*  *The page must load within 2 seconds.* |
| Postconditions | *The customer has access to the available cars and services.* |

|  |  |
| --- | --- |
| Use Case (UC\_2.4): | *Pay for the reservation* |
| Summary | *After making the reservation, the customer should pay for the reservation online using credit card.* |
| Dependency | *To make this use case work, the website should support different online payment ways.* |
| Actors | *Primary Actor: Customer* |
| Preconditions | *The customer should have made a reservation.* |
| Description of the Main Sequence | * *Step 1: The customer makes the reservation.* * *Step 2: The customer clicks the payment way they want to choose.* * *Step 3: The customer follows the step to process the payment.* |
| Description of the Alternative Sequence | * *Step 1: The customer makes the reservation.* * *Step 2: The customer clicks the payment way they want to choose.* * *Step 3: The customer follows the step to process the payment, but it gets cancelled or it is not supported.* |
| Nonfunctional requirements | *The payment ways should be secure for both the agency and the customer.* |
| Postconditions | *The reservation is paid.* |

|  |  |
| --- | --- |
| Use Case (UC\_2.5): | *Cancel reservations* |
| Summary | *The customer may need to cancel the reservation for various reasons.* |
| Actors | *Primary Actor: Customer* |
| Preconditions | *In order to delete the reservation, it should firstly exist and there should be a reason to delete it.* |
| Description of the Main Sequence | * *Step 1: The customer clicks the details of the reservation.* * *Step 2: By clicking the delete button at the end, the reservation is deleted.* |
| Description of the Alternative Sequence | * *Step 1: The customer clicks the details of the reservation.* * *Step 2: The reservation cannot be deleted.* |
| Nonfunctional requirements | *The deletion of the reservation should not affect the other reservations.* |
| Postconditions | *The reservation is deleted from the system.* |

|  |  |
| --- | --- |
| Use Case (UC\_2.6): | *Filter the cars according to different characteristics* |
| Summary | *When searching for a car, the customer should be able to filter the cars using car model, year of production, color, price etc. according to their needs.* |
| Actors | *Primary Actor: Customer* |
| Preconditions | *The system should keep information about the car details to be able to filter cars according to them.* |
| Description of the Main Sequence | * *Step 1: The customer clicks the filter button and chooses the characteristic they want to find.* * *Step 2: Navigate through the generated responses.* |
| Description of the Alternative Sequence | * *Step 1: The customer clicks the filter button and chooses the characteristic they want to find.* * *Step 2: There is no car with those characteristics.* |
| Nonfunctional requirements | *Easily navigable.* |
| Postconditions | *The customer is provided with the cars list that have the required characteristic.* |

|  |  |
| --- | --- |
| Use Case (UC\_1.7): | *Log Out* |
| Summary | *The customer at the end of the reservation logs out of the system.* |
| Actors | *Primary Actor: Customer* |
| Preconditions | *The customer should be logged in the system.* |
| Description of the Main Sequence | * *Step 1: The customer goes to their profile and logs out successfully by clicking the log out button.* |
| Description of the Alternative Sequence | * *Step 1: The customer fails to log out.* |
| Nonfunctional requirements | *The device needs to be connected to the internet.* |
| Postconditions | *The customer is logged out of the website.* |

****

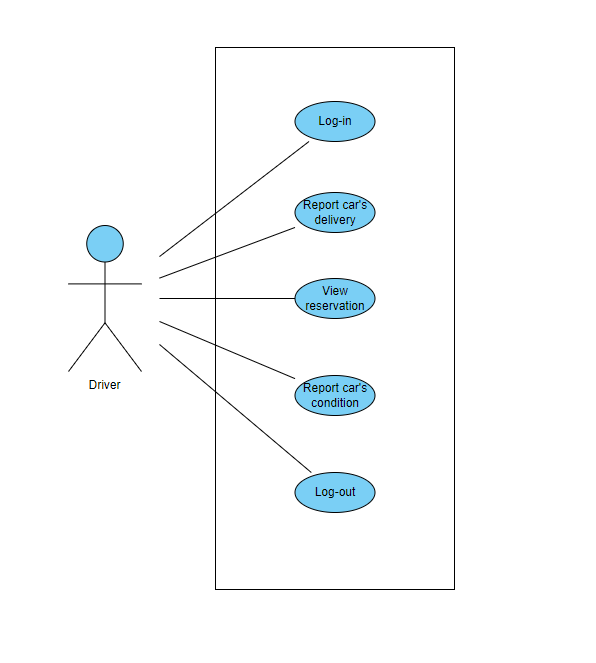
|  |  |
| --- | --- |
| Use Case (UC\_3.1): | *Log In* |
| Summary | The admin needs to log in the system to later use it. |
| Actors | Primary Actor: Admin |
| Preconditions | The admin should open the website |
| Description of the Main Sequence | * *Step 1: Enter the username.* * *Step 2: Enter the password.* * *Step 3: Press the log in button.* * *Step 4: Authentication* * *Step 5: The admin is logged in successfully.* |
| Description of the Alternative Sequence | * *Step 1: Enter the username.* * *Step 2: Enter the password.* * *Step 3: Press the log in button.* * *Step 4: Authentication* * *Step 5: The admin cannot log in.* |
| Nonfunctional requirements | *The device needs to be connected to the internet.*  *The system must be secure from unauthorized access.* |
| Postconditions | The admin should be logged in |

|  |  |
| --- | --- |
| Use Case (UC\_3.2): | *Manage what drivers and agents do* |
| Summary | *The admin manages all actions of drivers and agents.* |
| Actors | *Primary Actor: admin*  *Secondary Actor: driver, agent* |
| Preconditions | *Driver and agent should do their job.* |
| Description of the Main Sequence | * *Step 1: Admin checks employee list.* * *Step 2: Picks employee.* * *Step 3: Views his/her statistics.* * *Step 4: (Optional) Adds goals.* |
| Description of the Alternative Sequence | * *No alternative sequence* |
| Non-functional requirements | *All actors needs to be connected to the internet* |
| Postconditions | *The admin manages agents and drivers.* |

|  |  |
| --- | --- |
| Use Case (UC\_3.3): | *Manage employees* |
| Summary | The admin can register, modify, delete employees, and he can keep data about them |
| Actors | Primary Actor: admin  Secondary Actor: employees |
| Preconditions | The admin should open the website. |
| Description of the Main Sequence | *● Step 1: Admin checks employee list.*  *● Step 2: Picks employee.*  *● Step 3: Modifies his/her information.* |
| Description of the Alternative Sequence | *● Step 1: Admin checks employee list.*  *● Step 2: Picks employee.*  *● Step 3: Deletes employee.* |
| Description of the Alternative Sequence no 2 | *● Step 1: Admin checks employee list.*  *● Step 2: Makes register request.*  *● Step 3: Registers employee* |
| Nonfunctional requirements | *The device needs to be connected to the internet.*  *The system must be secure from unauthorized access.* |
| Postconditions | The admin can modify the data of employees |

|  |  |
| --- | --- |
| Use Case (UC\_3.4): | *View data about total incomes* |
| Summary | The admin can view analytical data about his/her business. |
| Actors | Primary Actor: admin |
| Preconditions | The admin should open the website. |
| Description of the Main Sequence | *● Step 1: Admin opens the financial statistics menu.*  *● Step 2: Chooses specific filters.*  *● Step 3: Views data, keeps notes and sets new goals.* |
| Description of the Alternative Sequence | *● No alternative sequence* |
| Nonfunctional requirements | *The device needs to be connected to the internet.*  *The system must be secure from unauthorized access.* |
| Postconditions | Administrator has analyzed the financial data |

|  |  |
| --- | --- |
| Use Case (UC\_3.5): | *Log Out* |
| Summary | *The admin at the end of the shift logs out of the system.* |
| Actors | *Primary Actor: Admin* |
| Preconditions | *The admin should be logged in the system.* |
| Description of the Main Sequence | * *Step 1: The admin goes to their profile and logs out successfully by clicking the log out button.* |
| Description of the Alternative Sequence | * *Step 1: The admin saves and terminates any tasks that might be still going on.* * *Step 2: The admin goes to their profile and logs out successfully by clicking the log out button.* |
| Nonfunctional requirements | *The device needs to be connected to the internet.* |
| Postconditions | *The admin is logged out of the website.* |

****

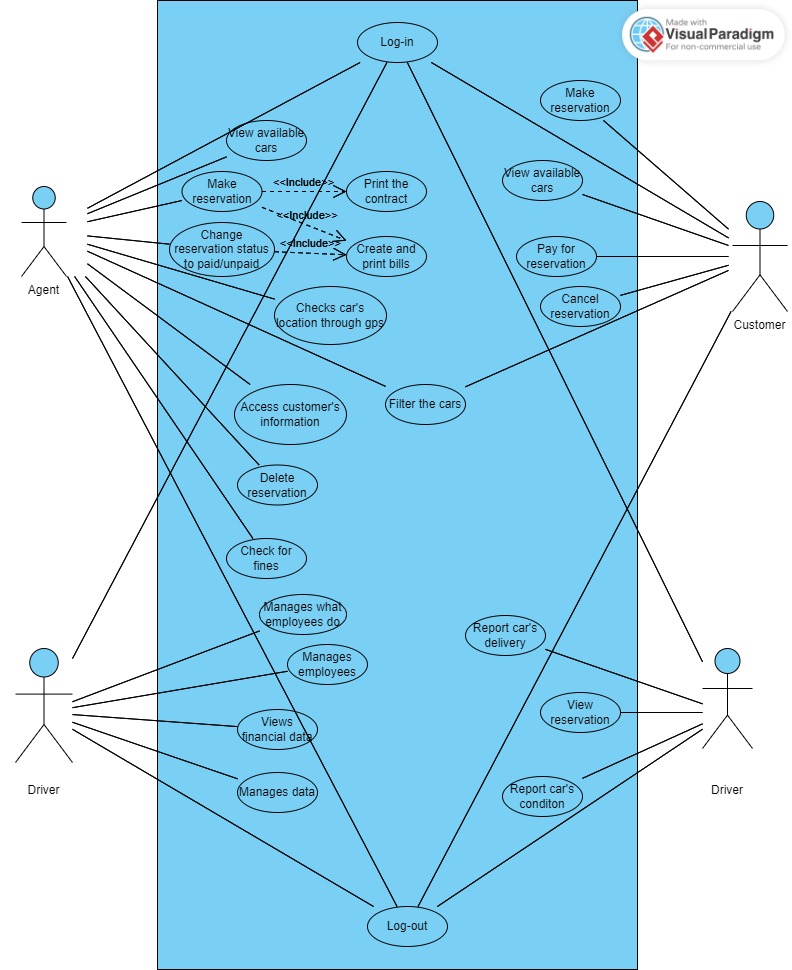
|  |  |
| --- | --- |
| Use Case (UC\_4.1): | *Log In* |
| Summary | The driver needs to log in the system to later use it. |
| Actors | Primary Actor: Driver |
| Preconditions | The Driver should open the website |
| Description of the Main Sequence | * *Step 1: Enter the username.* * *Step 2: Enter the password.* * *Step 3: Press the log in button.* * *Step 4: Authentication* * *Step 5: The driver is logged in successfully.* |
| Description of the Alternative Sequence | * *Step 1: Enter the username.* * *Step 2: Enter the password.* * *Step 3: Press the log in button.* * *Step 4: Authentication* * *Step 5: The driver cannot log in.* |
| Nonfunctional requirements | *The device needs to be connected to the internet.*  *The system must be secure from unauthorized access.* |
| Postconditions | The driver should be logged in |

|  |  |
| --- | --- |
| Use Case (UC\_4.2): | *View reservation* |
| Summary | The driver can view reservations in order to make sure he delivers on time and in the correct place |
| Actors | Primary Actor: driver |
| Preconditions | The driver should be logged in the system. |
| Description of the Main Sequence | *● Step 1: Driver opens the reservations menu*  *● Step 2: Chooses the reservation using reservation details.*  *● Step 3: Gets reservation information that he/she needs* |
| Description of the Alternative Sequence | *● No alternative sequence* |
| Nonfunctional requirements | *The device needs to be connected to the internet.*  *The system must be secure from unauthorized access.* |
| Postconditions | Driver has retrieved the information about time and place of delivery. |

|  |  |
| --- | --- |
| Use Case (UC\_4.3): | *Report cars delivery* |
| Summary | The driver should report about any car he has delivered to the customers |
| Actors | Primary Actor: driver |
| Preconditions | The driver should be logged in |
| Description of the Main Sequence | *● Step 1: Driver opens the reservations menu.*  *● Step 2: Finds the reservation using reservation’s details.*  *● Step 3: Marks the reservation as delivered.* |
| Description of the Alternative Sequence | *● No alternative sequence* |
| Nonfunctional requirements | *The device needs to be connected to the internet.*  *The system must be secure from unauthorized access.* |
| Postconditions | Driver has reported a successful delivery. |

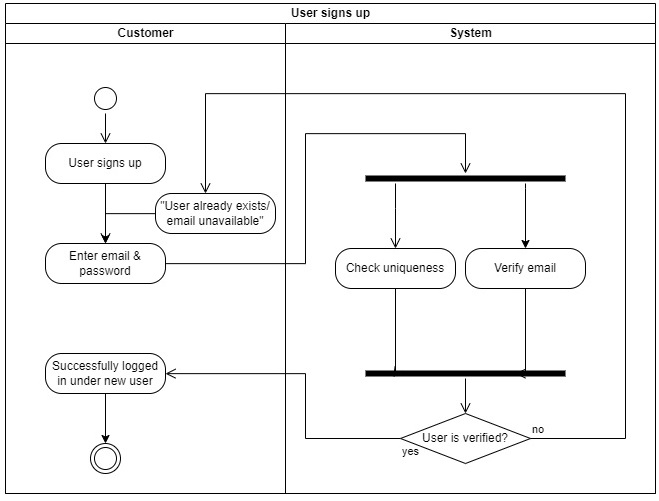
|  |  |
| --- | --- |
| Use Case (UC\_4.4): | *Report the car’s condition* |
| Summary | The driver should report any damage that has happened to the car after the client returns it |
| Actors | Primary Actor: driver |
| Preconditions | The driver should be logged in. |
| Description of the Main Sequence | *● Step 1: Driver opens the cars menu*  *● Step 2: Finds the specific car by its license number*  *● Step 3: Makes a condition report request.*  *● Step 4: Writes any information regarding the car’s condition.* |
| Description of the Alternative Sequence | *● No alternative sequence* |
| Nonfunctional requirements | *The device needs to be connected to the internet.*  *The system must be secure from unauthorized access.* |
| Postconditions | Driver has reported the conditions of the car |

|  |  |
| --- | --- |
| Use Case (UC\_4.5): | *Log Out* |
| Summary | *The driver at the end of the shift logs out of the system.* |
| Actors | *Primary Actor: Driver* |
| Preconditions | *The Driver should be logged in the system.* |
| Description of the Main Sequence | * *Step 1: The driver goes to their profile and logs out successfully by clicking the log out button.* |
| Description of the Alternative Sequence | * *Step 1: The driver saves and terminates any tasks that might be still going on.* * *Step 2: The driver goes to their profile and logs out successfully by clicking the log out button.* |
| Nonfunctional requirements | *The device needs to be connected to the internet.* |
| Postconditions | *The driver is logged out of the website.* |

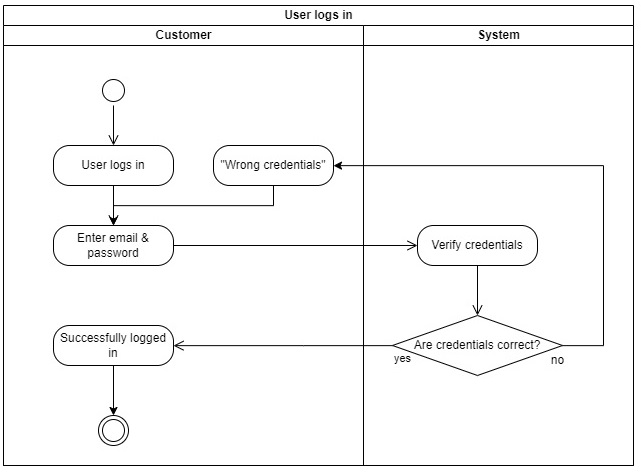
****

* 1. **Activity diagrams**

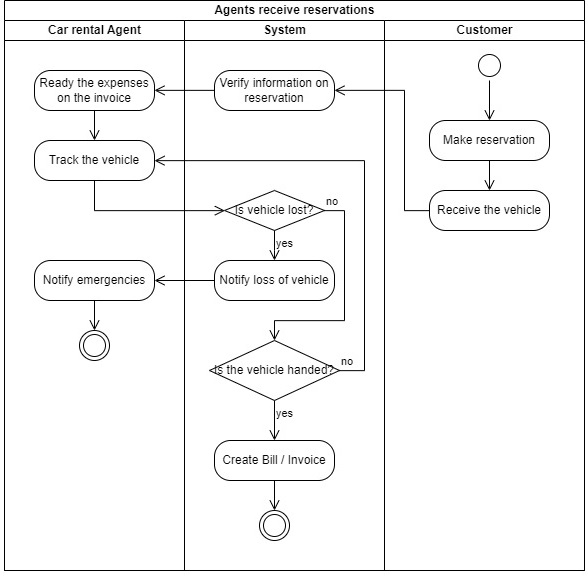
AC\_1: User signs up



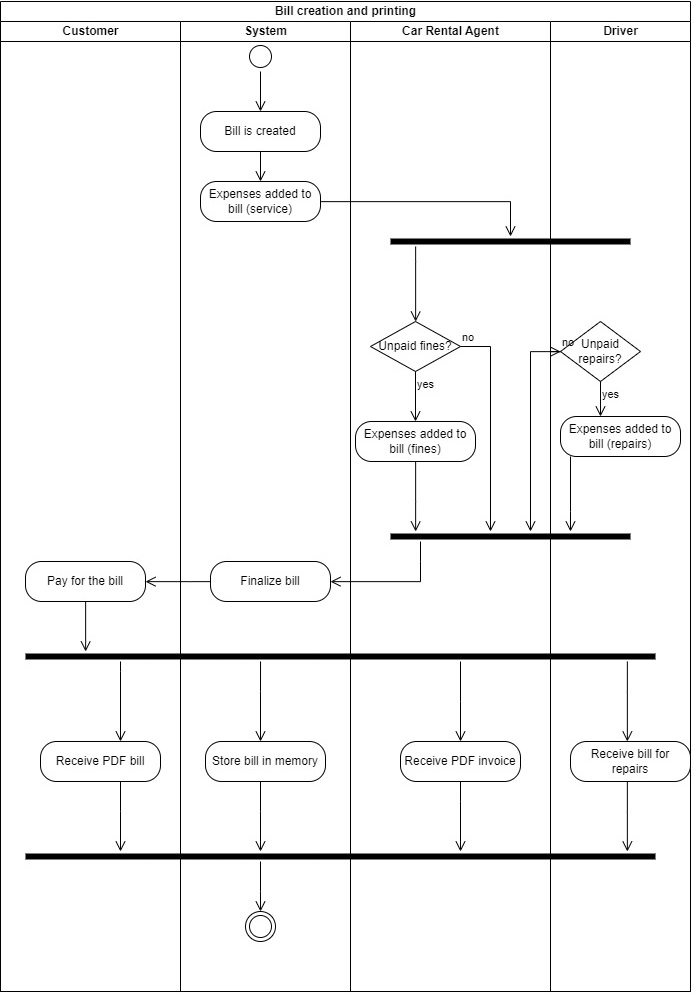
AC\_2: User logs in:



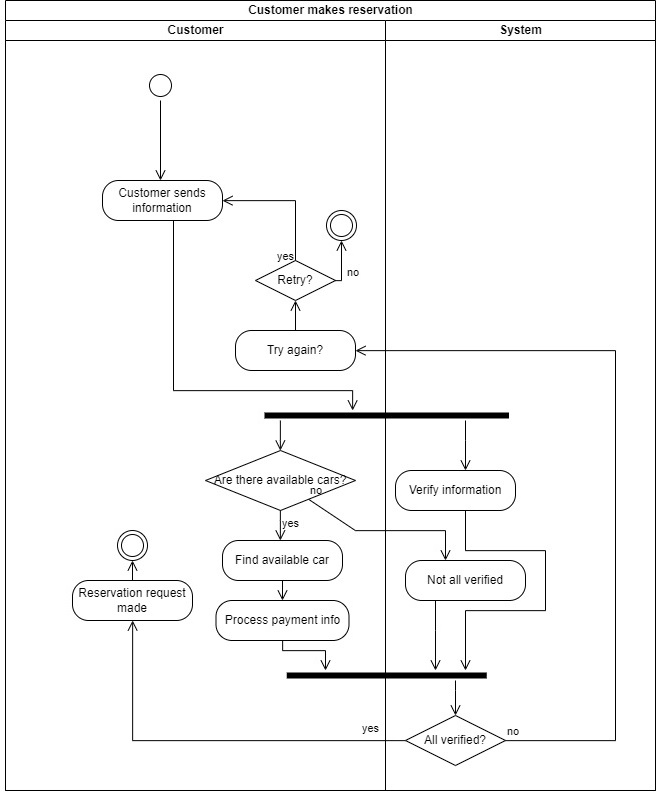
AC\_3: Agents receive reservations:



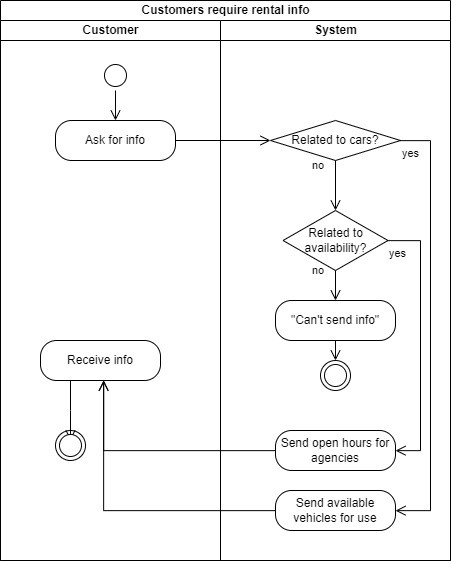
AC\_4: Bill is created:



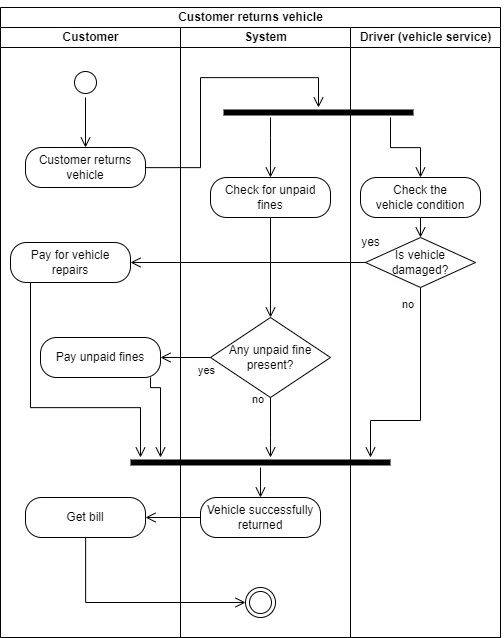
AC\_5: Customer makes reservation:



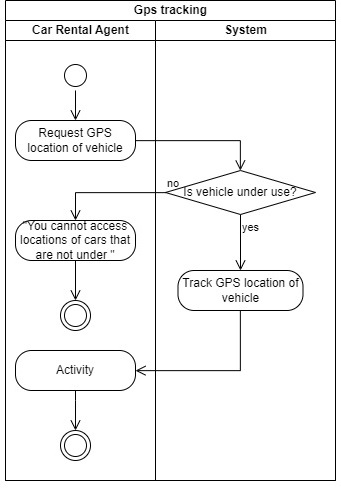
AC\_6 Customer requires rental information:



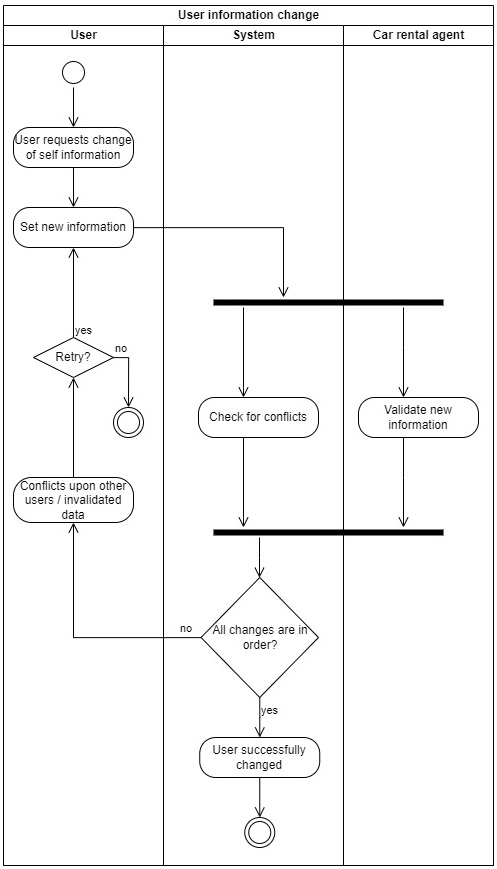
AC\_7: Customer returns vehicle



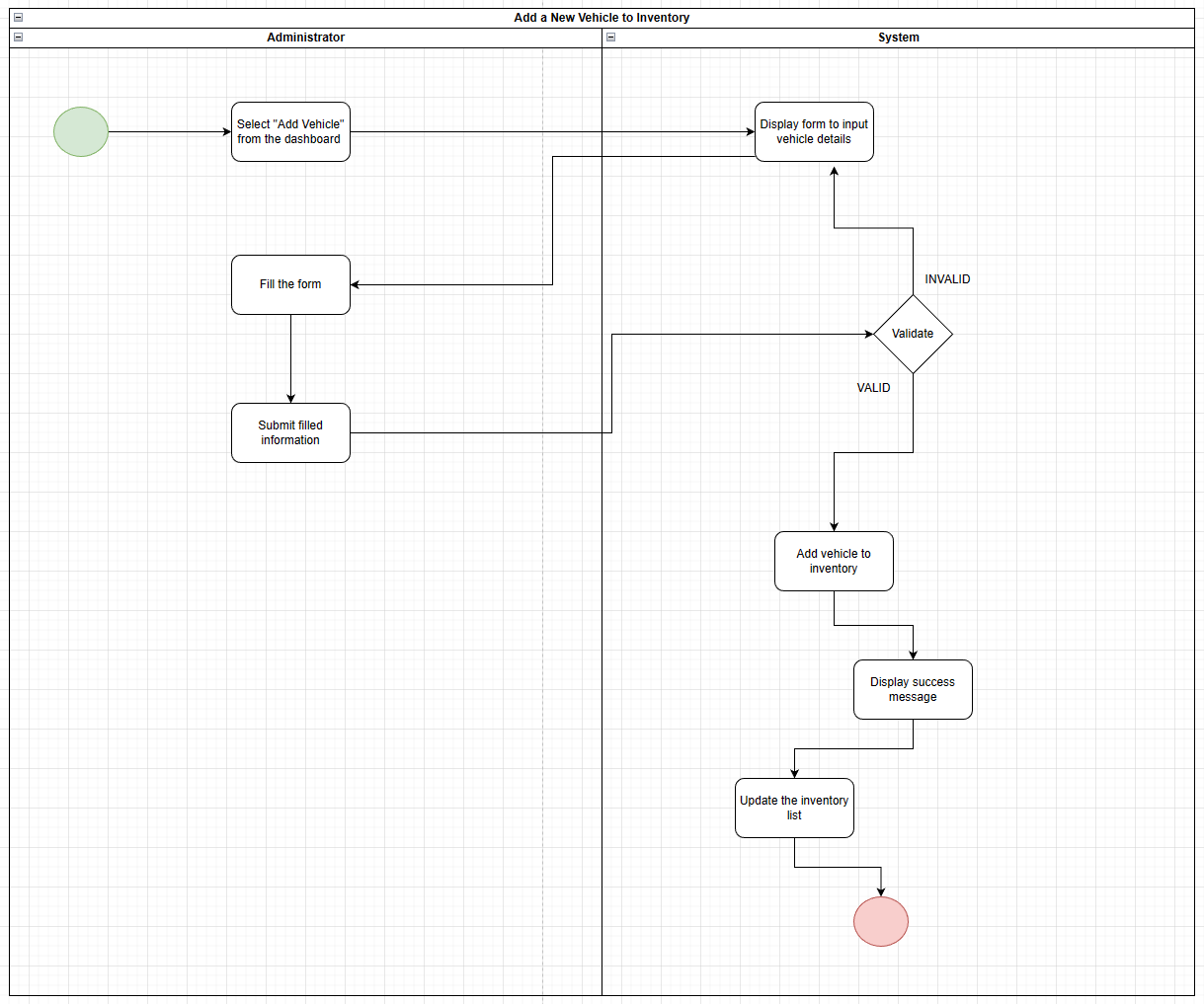
AC\_8: GPS-Tracking



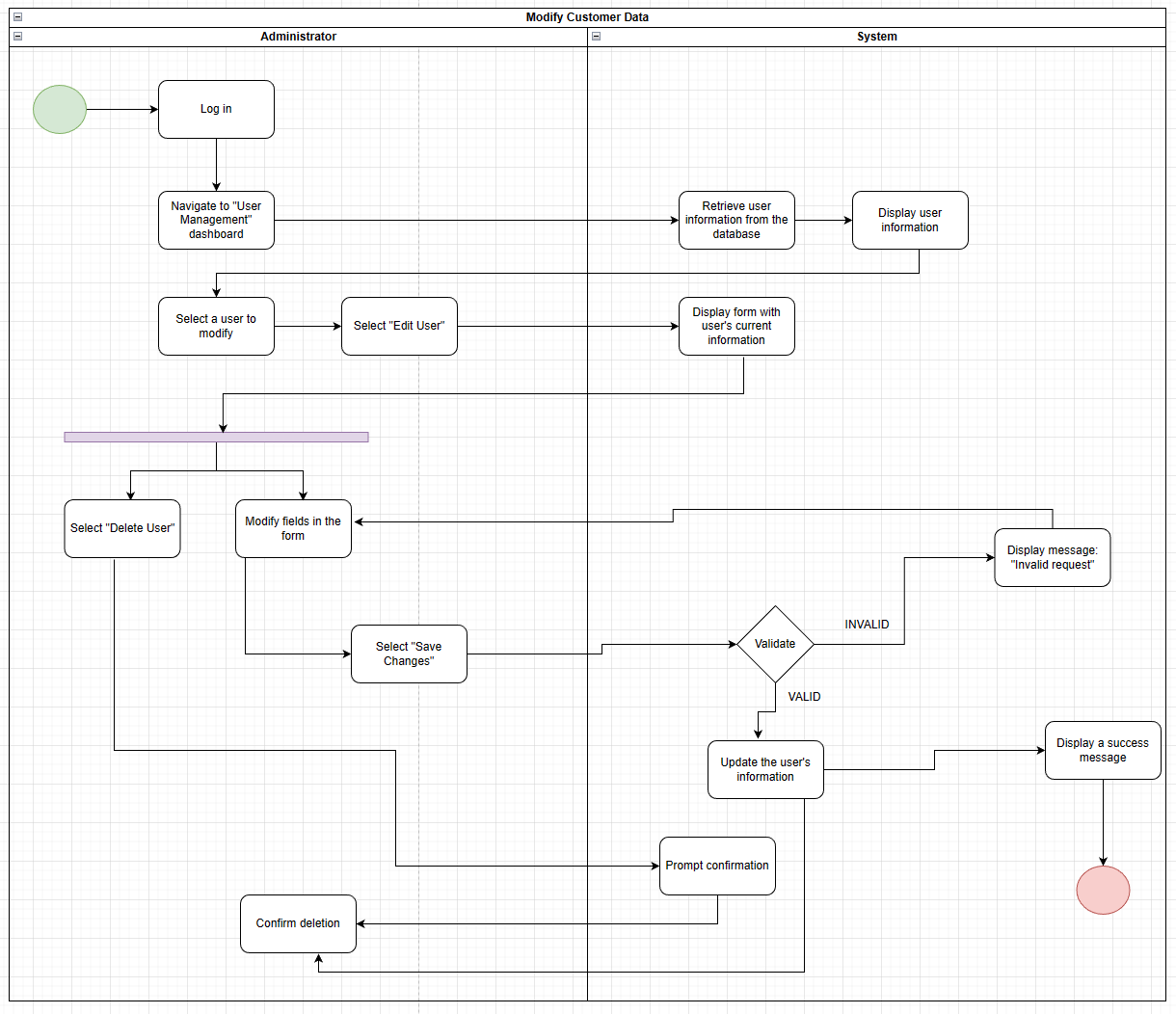
AC\_9: User changes profile information



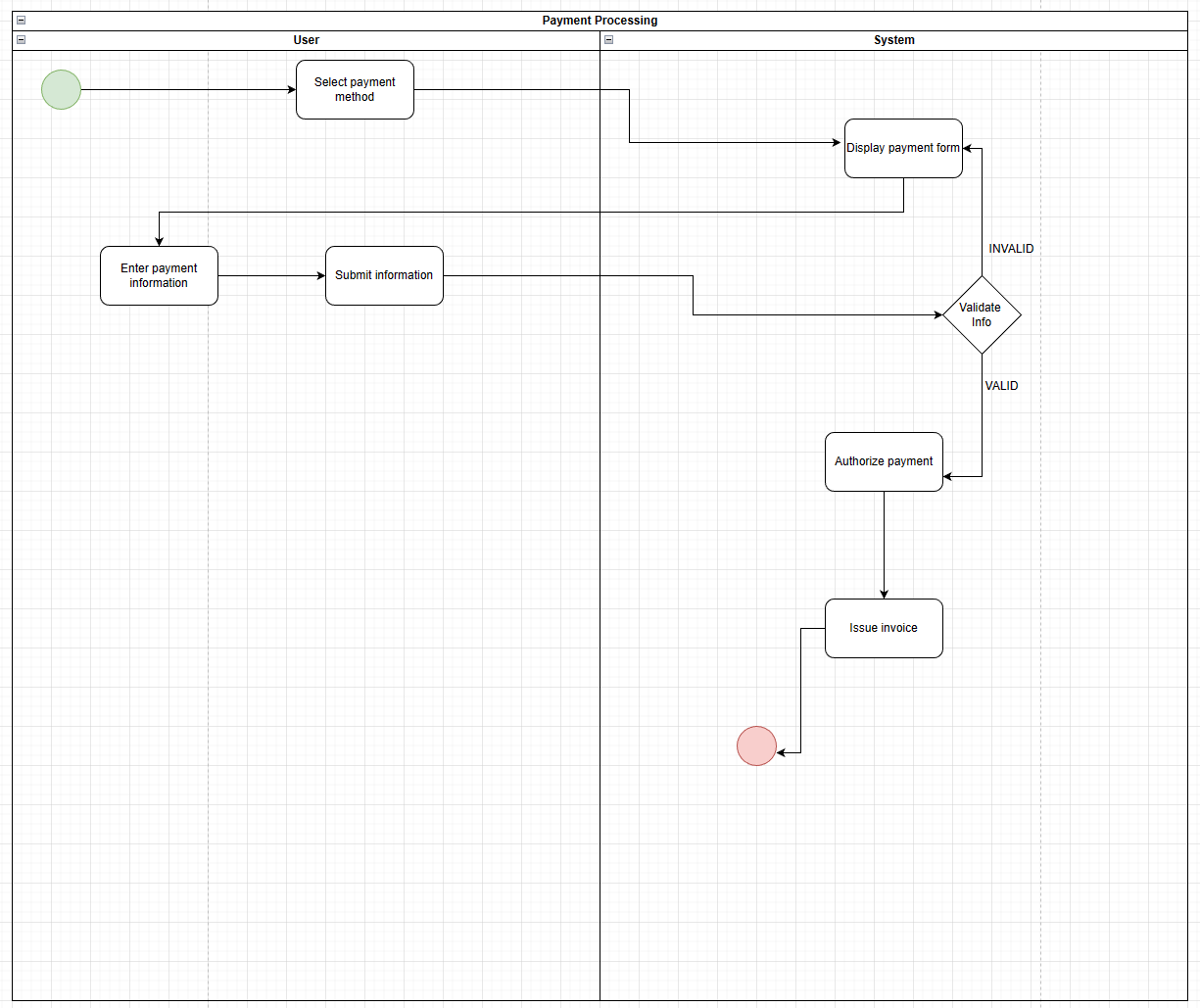
AC\_10: Admin adds new vehicle to inventory



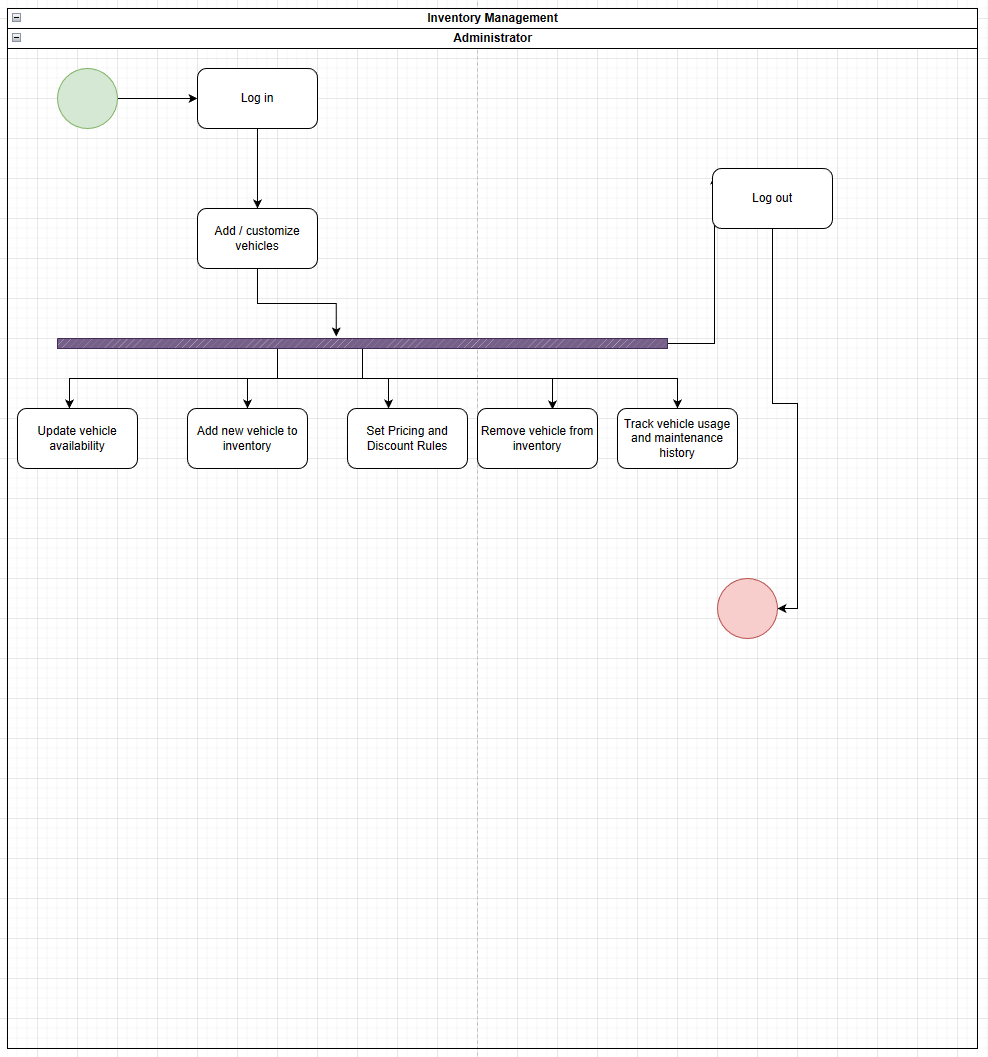
AC\_11: Admin modifies user’s data



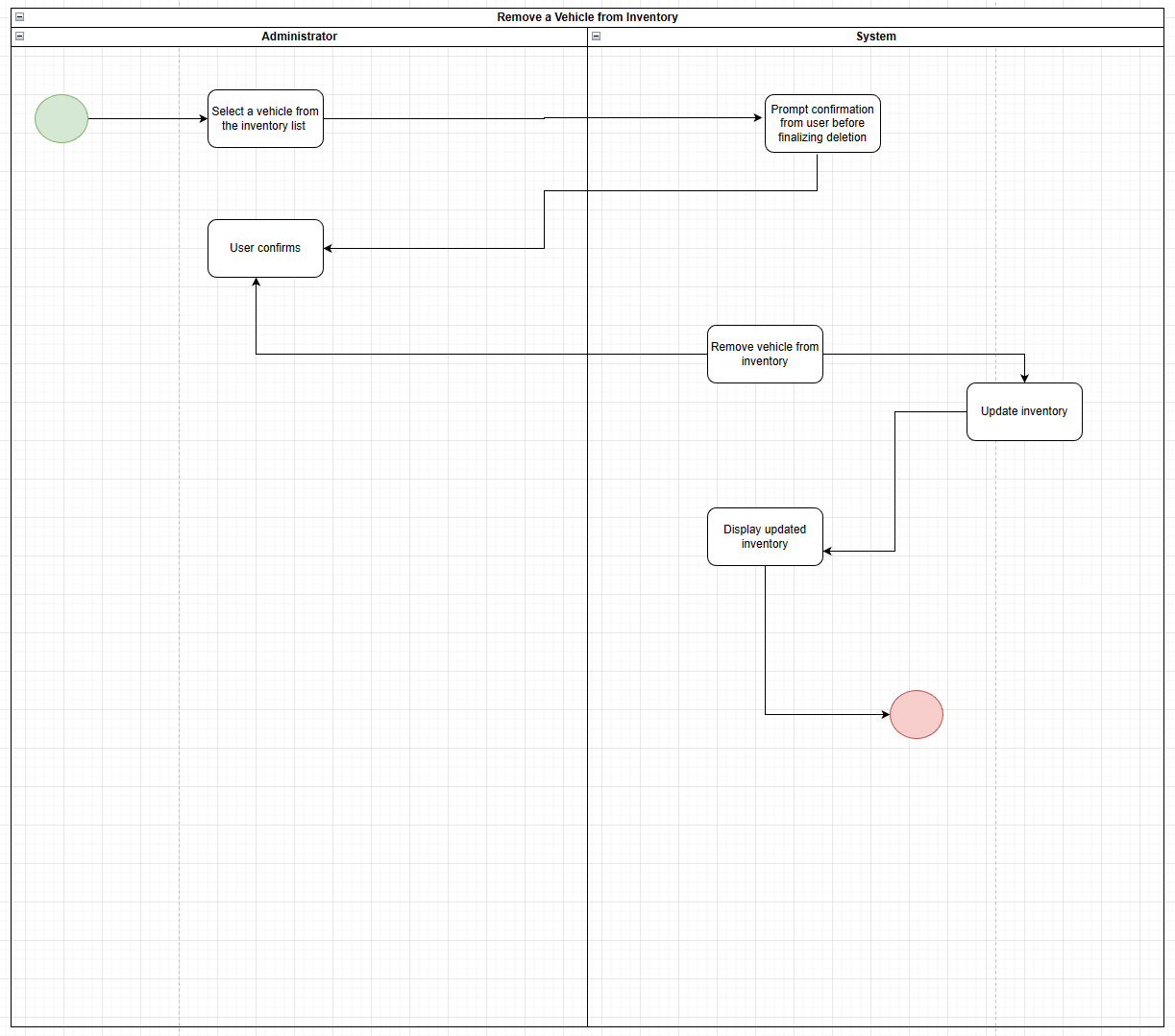
AC\_12: Payment processing



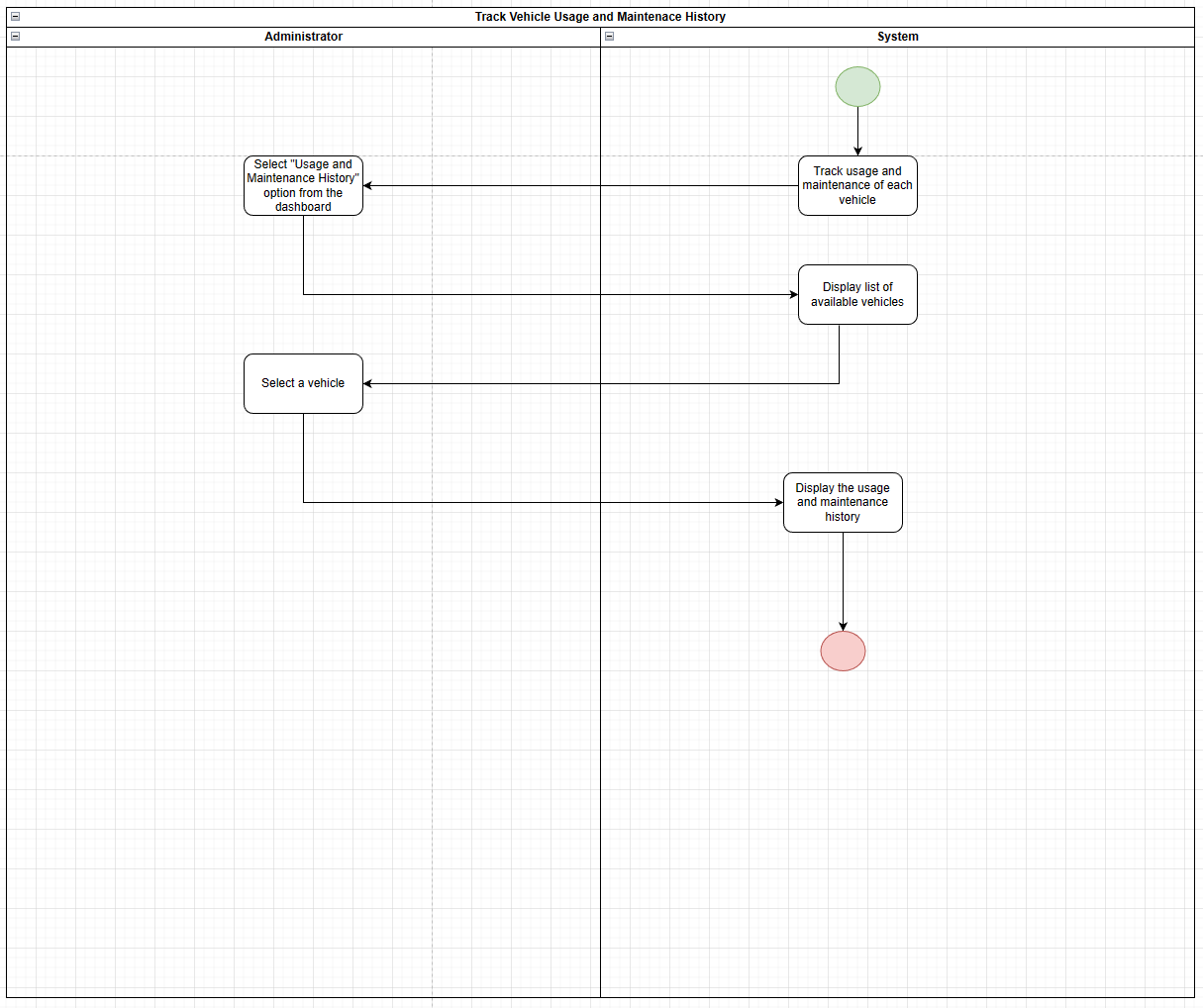
AC\_13: Administrator manages inventory



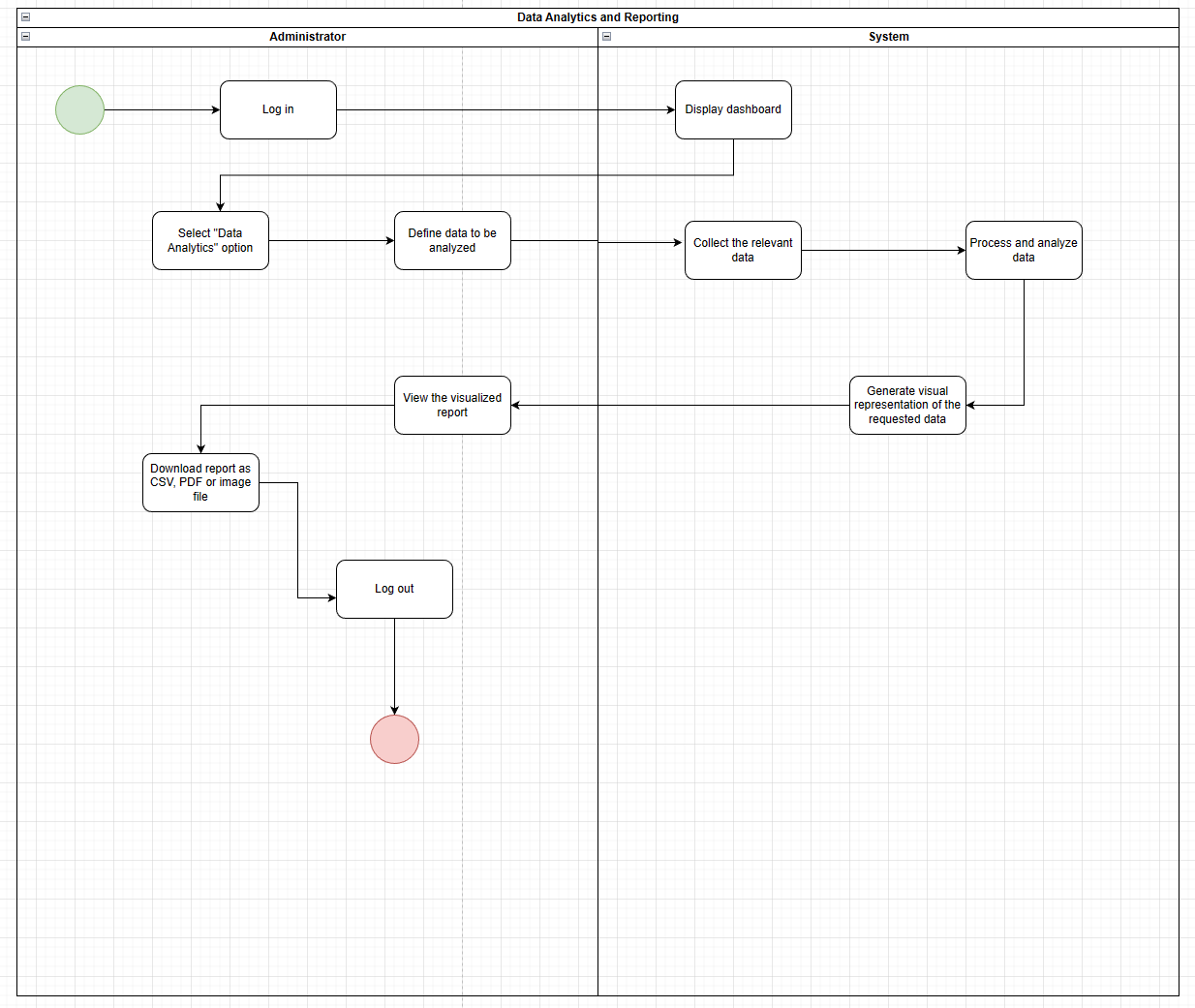
AC\_14: Administrator removes a vehicle from inventory



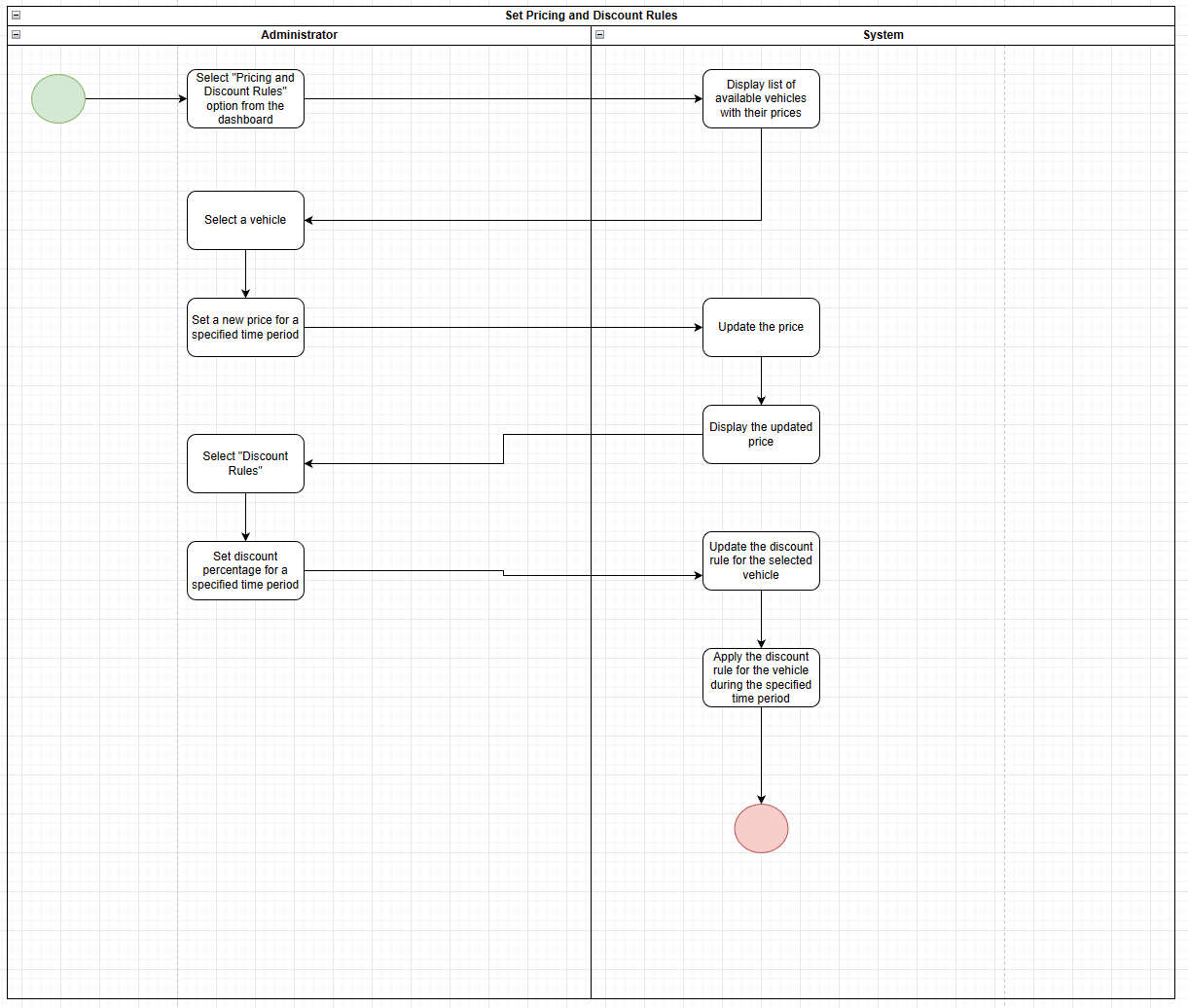
AC\_15: Tracking vehicle usage and maintenance history



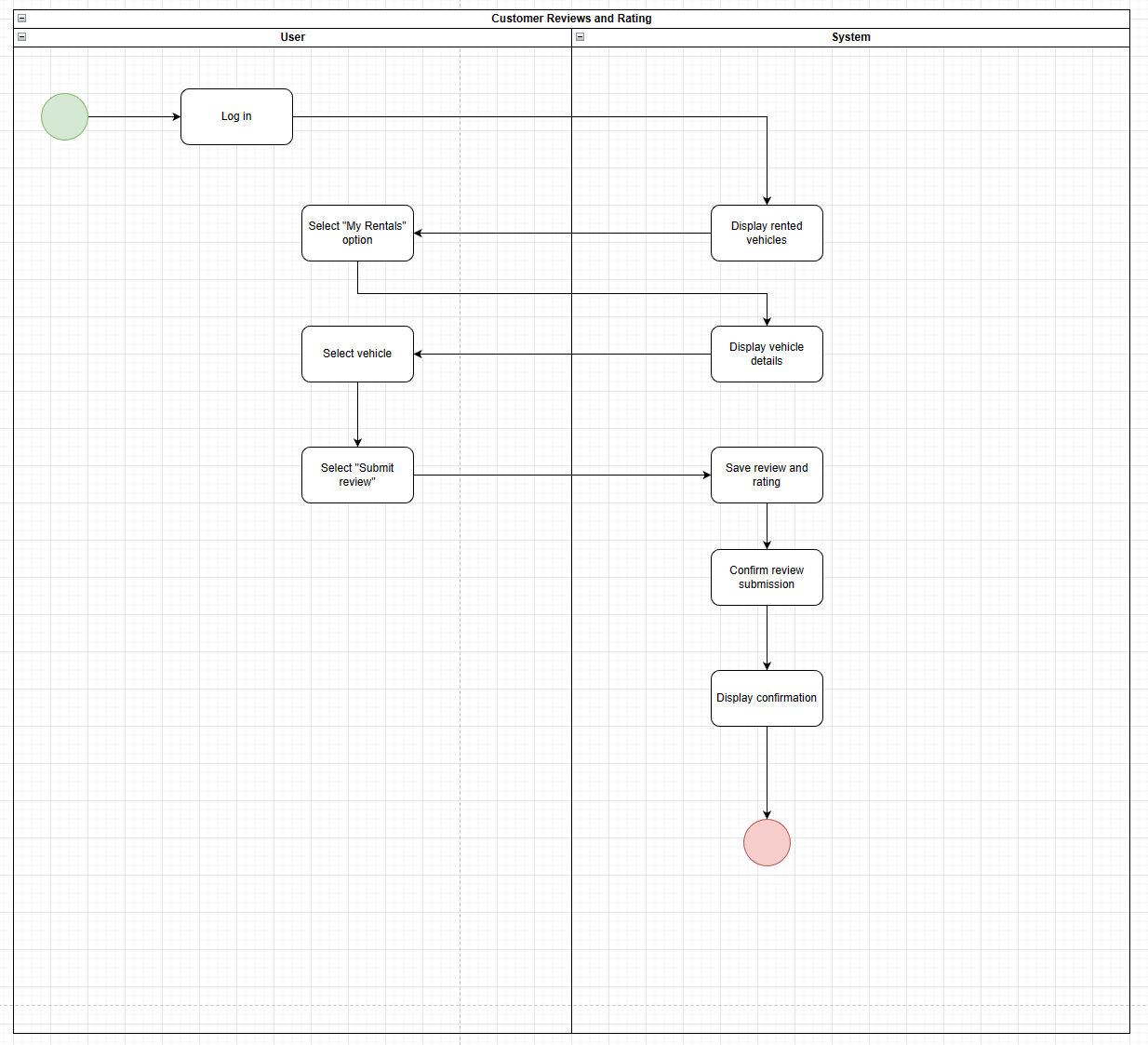
AC\_16: Administrator receives analytic data and reports



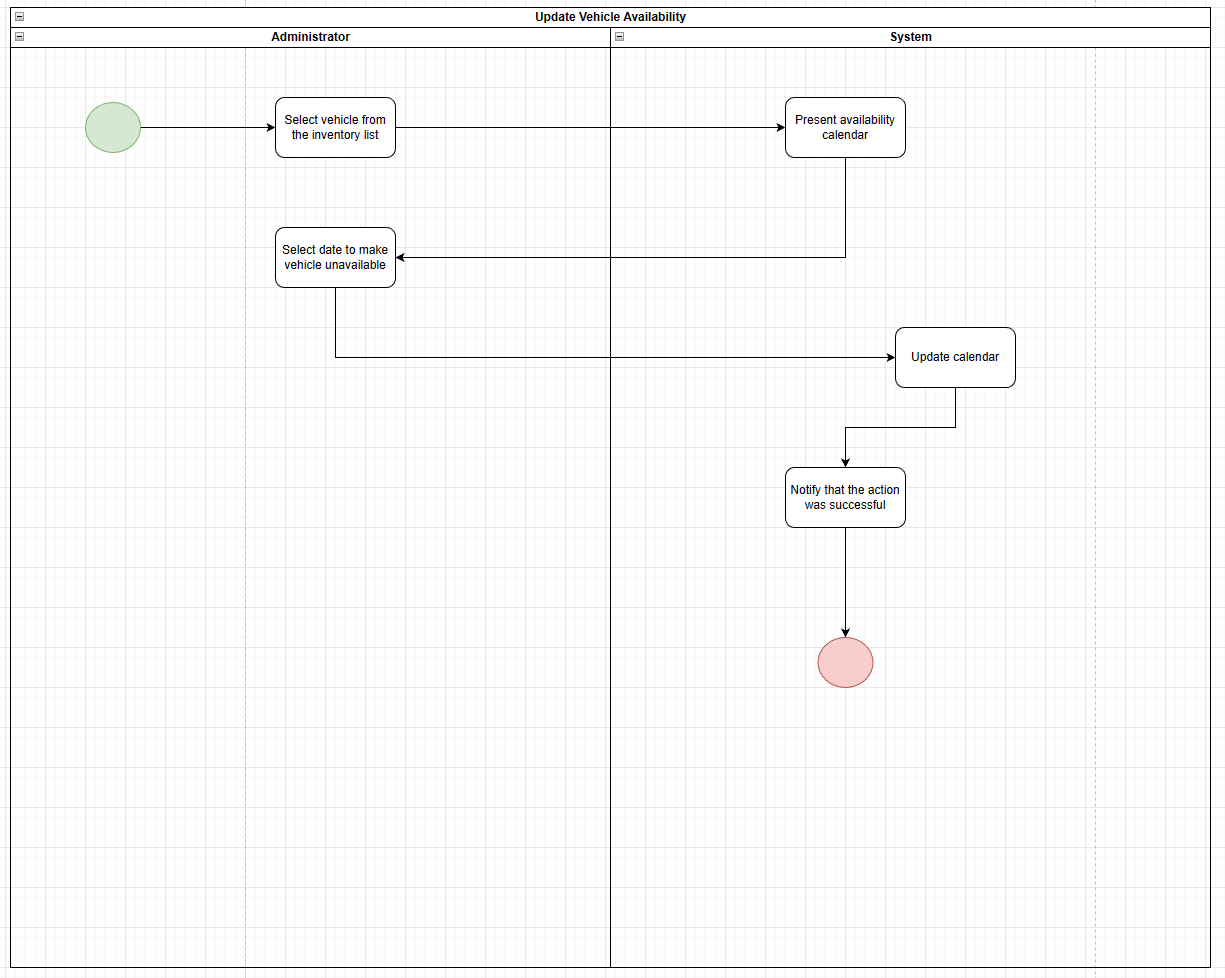
AC\_17: Setting prices and discount rules

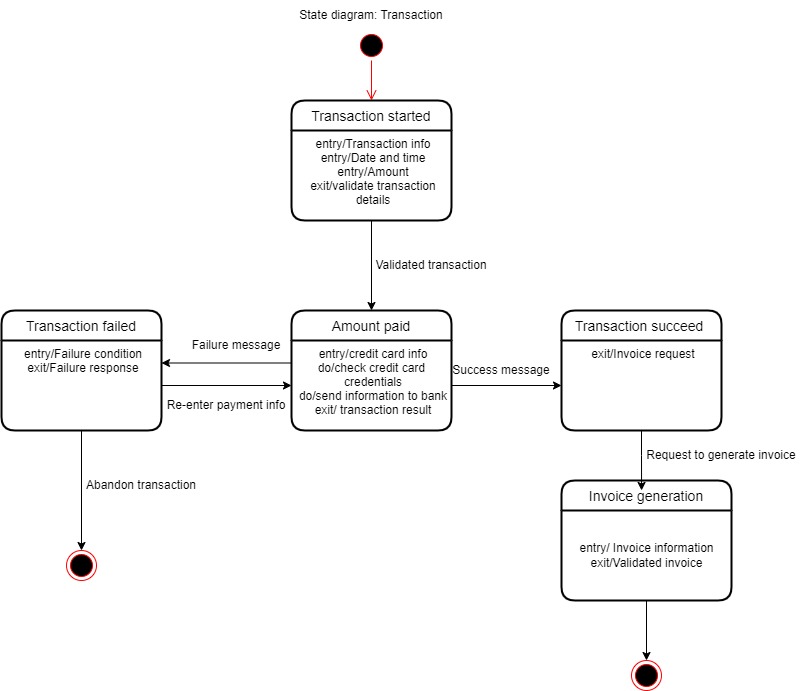


AC\_18: Customer reviews and rates the service

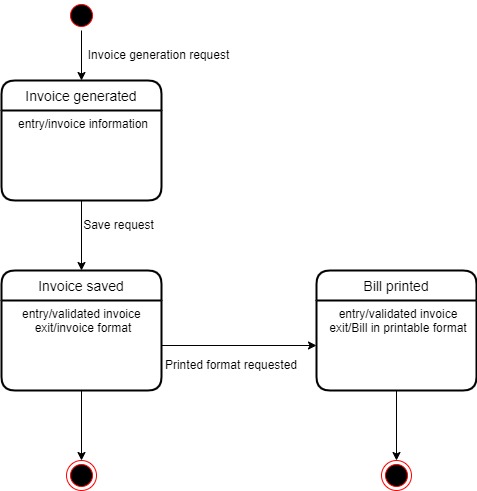


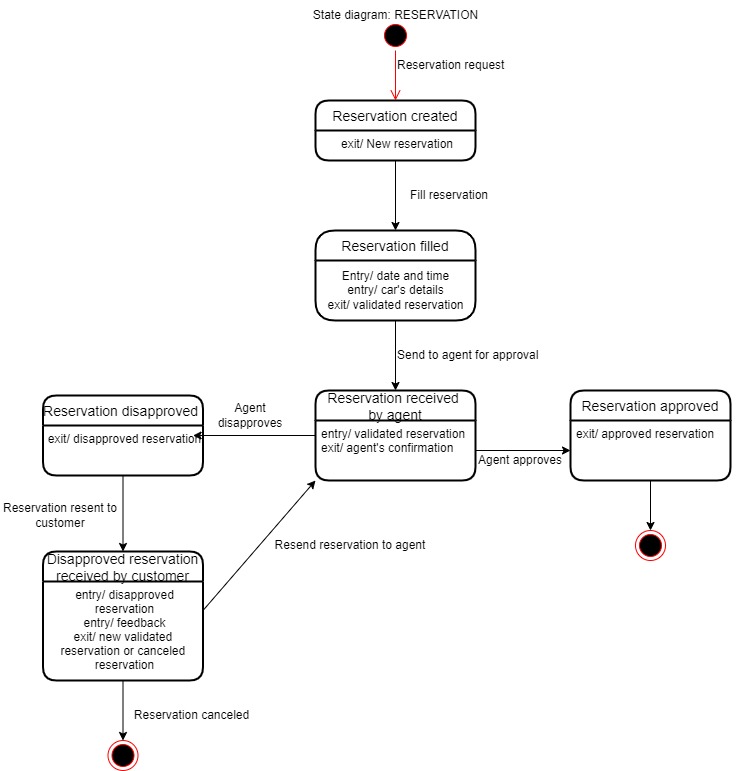
AC\_19: Update car’s availability

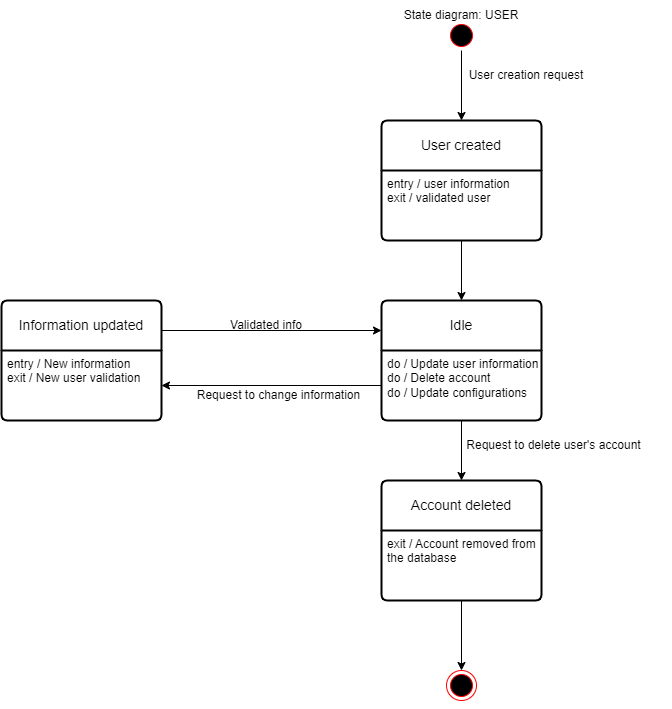


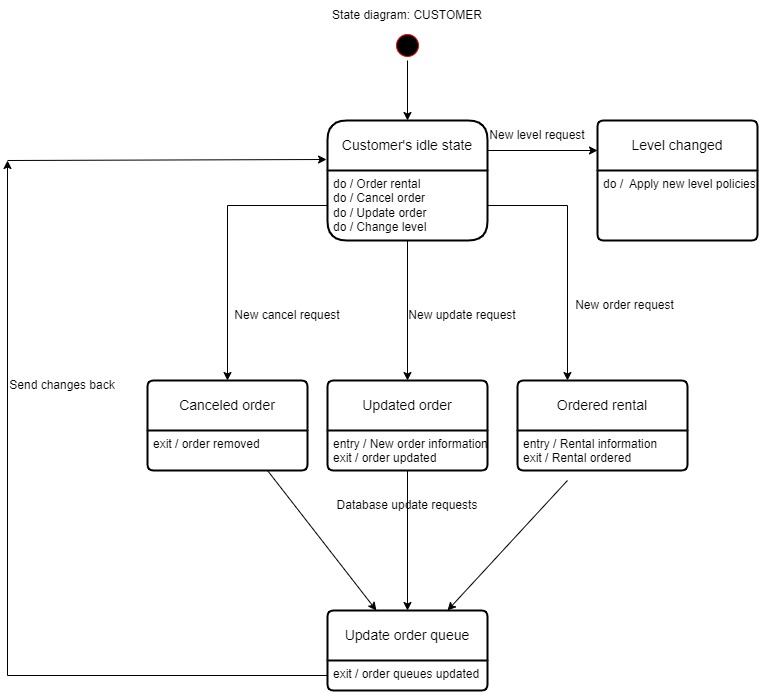
* 1. **State diagrams**

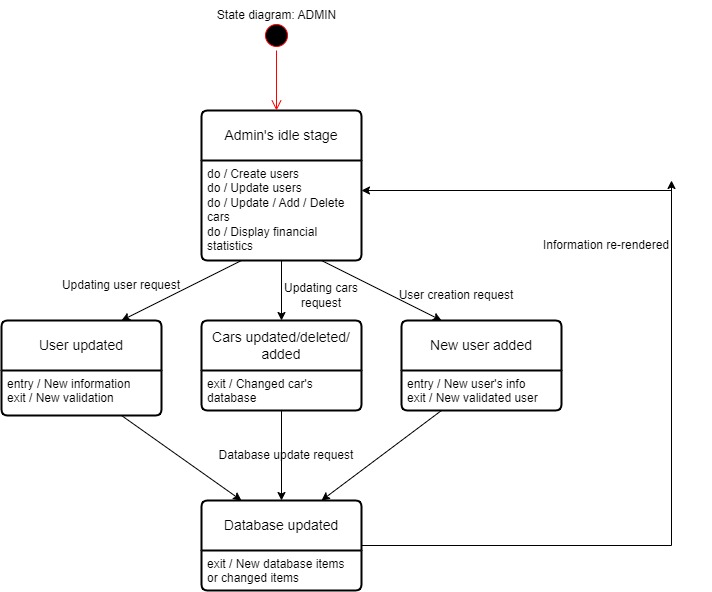
Invoice state diagram

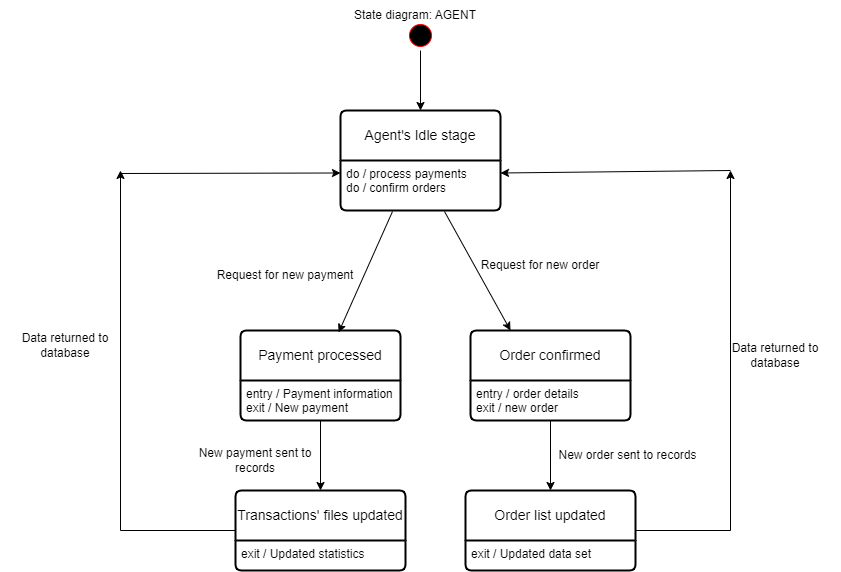
****

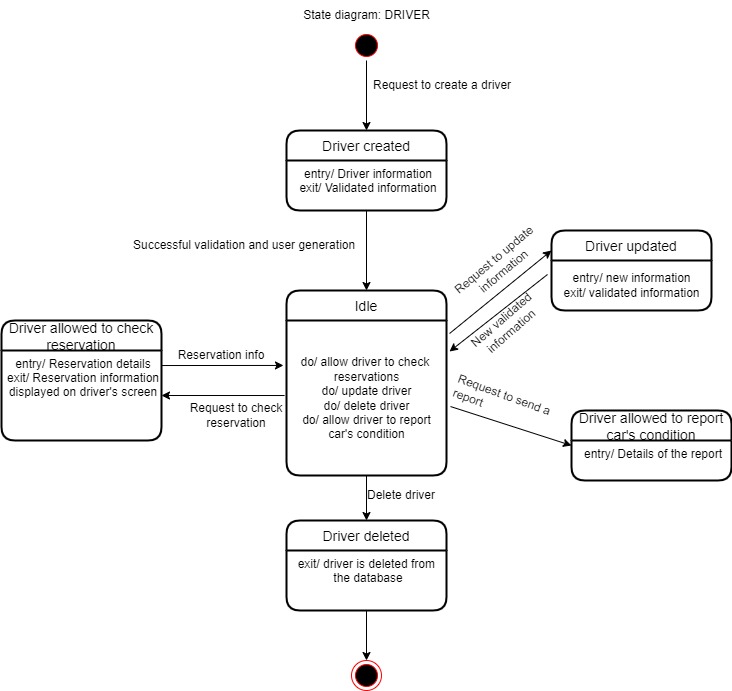


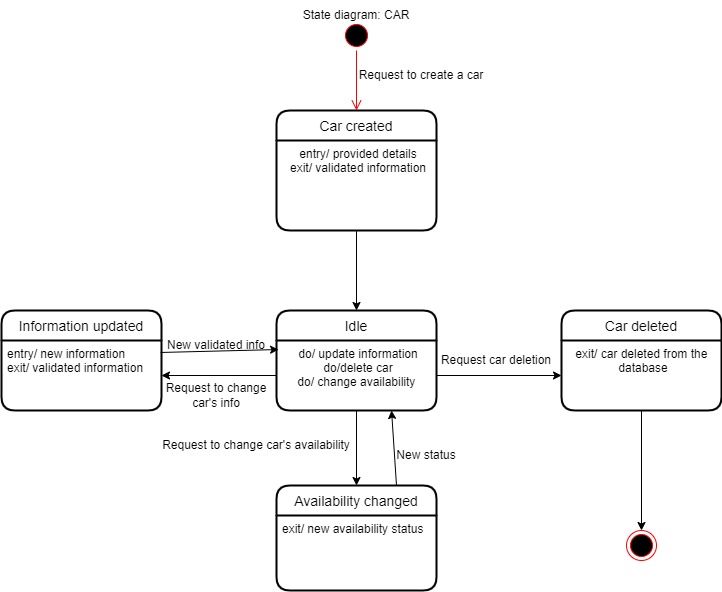












APPENDIX

The appendixes are not always considered part of the actual Requirements Specification and are not always necessary. They may include

* Sample input/output formats, descriptions of cost analysis studies, or results of user surveys;
* Supporting or background information that can help the readers of the Requirements Specification;
* A description of the problems to be solved by the system;
* Special packaging instructions for the code and the media to meet security, export, initial loading, or other requirements.

When appendixes are included, the Requirements Specification should explicitly state whether or not the appendixes are to be considered part of the requirements.

1. **Definitions, Acronyms, and Abbreviations**

Define all terms, acronyms, and abbreviations used in this document.

1. **References**

List all the documents and other materials referenced in this document.

1. **Requirements Traceability Matrix**

The following trace matrix examples show one possible use of naming standards for deliverables (FunctionalArea-DocType-NN). The number has no other meaning than to keep the documents unique. For example, the Bargaining Unit Assignment Process Flow would be BUA-PF-01.

For example (1):

| **Business Requirement** | **Area** | **Deliverables** | **Status** |
| --- | --- | --- | --- |
| BR\_LR\_01  The system should validate the relationship between Bargaining Unit/Location and Job Class.---Comments: Business Process = "Assigning a Bargaining Unit to an Appointment" (Priority 1) | BUA | BUA-CD-01  Assign BU Conceptual Design | Accepted |
| BUA-PF-01  Derive Bargaining Unit-Process Flow Diagram | Accepted |
| BUA-PF-01  Derive Bargaining Unit-Process Flow Diagram | Accepted |
| BR\_LR\_09  The system should provide the capability for the Labor Relations Office to maintain the job class/union relationship.---Comments: Business Process = "Maintenance" (Priority 1) | BUA | BUA-CD-01  Assign BU Conceptual Design | Accepted |
| BUA-PF-02  BU Assignment Rules Maint Process Flow Diagram | ReadyForReview |

For example (2):

| **BizReqID** | **Pri** | **Major Area** | **DevTstItems DelivID** | **Deliv Name** | **Status** |
| --- | --- | --- | --- | --- | --- |
| BR\_LR\_01 | 1 | BUA | BUA-CD-01 | Assign BU Conceptual Design | Accepted |
| BR\_LR\_01 | 1 | BUA | BUA-DS-02 | Bargaining Unit Assignment DB Modification Description | Accepted |
| BR\_LR\_01 | 1 | BUA | BUA-PF-01 | Derive Bargaining Unit-Process Flow Diagram | Accepted |
| BR\_LR\_01 | 1 | BUA | BUA-UCD-01 | BU Assign LR UseCase Diagram | ReadyForReview |
| BR\_LR\_01 | 1 | BUA | BUA-UCT-001 | BU Assignment by PC UseCase - Add Appointment and Derive UBU | Reviewed |
| BR\_LR\_01 | 1 | BUA | BUA-UCT-002 | BU Assignment by PC UseCase - Add Appointment (UBU Not Found) | Reviewed |
| BR\_LR\_01 | 1 | BUA | BUA-UCT-006 | BU Assignment by PC UseCase - Modify Appointment (Removed UBU) | Reviewed |
| BR\_LR\_09 | 1 | BUA | BUA-CD-01 | Assign BU Conceptual Design | Accepted |
| BR\_LR\_09 | 1 | BUA | BUA-DS-02 | Bargaining Unit Assignment DB Modification Description | Accepted |
| BR\_LR\_09 | 1 | BUA | BUA-PF-02 | BU Assignment Rules Maint Process Flow Diagram | Accepted |
| BR\_LR\_09 | 1 | BUA | BUA-UCD-03 | BU Assign Rules Maint UseCase Diagram | Reviewed |
| BR\_LR\_09 | 1 | BUA | BUA-UCT-045 | BU Assignment Rules Maint: Successfully Add New Assignment Rule | Reviewed |
| BR\_LR\_09 | 1 | BUA | BUA-UCT-051 | BU Assignment Rules MaintUseCase: Modify Rule | Reviewed |
| BR\_LR\_09 | 1 | BUA | BUA-UCT-053 | BU Assignment Rules MaintUseCase - Review Assignment Rules | Reviewed |
| BR\_LR\_09 | 1 | BUA | BUA-UCT-057 | BU Assignment Rules MaintUseCase: Inactivate Last Rule for a BU | Reviewed |
| BR\_LR\_09 | 1 | BUA | BUA-UI-02 | BU AssignRules Maint UI Mockups | ReadyForReview |
| BR\_LR\_09 | 1 | BUA | BUA-TC-021 | BU Assignment Rules Maint TestCase: Add New Rule (Associated Job Class Does Not Exist) - Success | ReadyForReview |
| BR\_LR\_09 | 1 | BUA | BUA-TC-027 | BU Assignment Rules Maint TestCase: Modify Rule - Success | ReadyForReview |
| BR\_LR\_09 | 1 | BUA | BUA-TC-035 | BU Assignment Rules Maint TestCase: Add New Rule (Associated Job Class Does Not Exist) - Error Condition | ReadyForReview |
| BR\_LR\_09 | 1 | BUA | BUA-TC-049 | BU Assignment Rules Maint TestCase: Modify Rule - Error Condition | ReadyForReview |

For example (3):

| **BizReqID** | **CD01** | **CD02** | **CD03** | **CD04** | **UI01** | **UI02** | **UCT01** | **UCT02** | **UCT03** | **TC01** | **TC02** | **TC03** | **TC04** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| BR\_LR\_01 |  |  | X |  | X |  | X |  |  | X |  | X |  |
| BR\_LR\_09 | X |  |  | X |  | X |  |  | X |  | X |  | X |
| BR\_LR\_10 | X |  |  | X |  |  |  |  | X |  | X |  |  |
| BR\_LR\_11 |  | X |  |  |  |  |  |  |  |  |  |  |  |

1. **Organizing the Requirements**

This section is for information only as an aid in preparing the requirements document.

Detailed requirements tend to be extensive. Give careful consideration to your organization scheme. Some examples of organization schemes are described below:

**By System Mode**

Some systems behave quite differently depending on the mode of operation. For example, a control system may have different sets of functions depending on its mode: training, normal, or emergency.

**By User Class**

Some systems provide different sets of functions to different classes of users. For example, an elevator control system presents different capabilities to passengers, maintenance workers, and fire fighters.

**By Objects**

Objects are real-world entities that have a counterpart within the system. For example, in a patient monitoring system, objects include patients, sensors, nurses, rooms, physicians, medicines, etc. Associated with each object is a set of attributes (of that object) and functions (performed by that object). These functions are also called services, methods, or processes. Note that sets of objects may share attributes and services. These are grouped together as classes.

**By Feature**

A feature is an externally desired service by the system that may require a sequence of inputs to affect the desired result. For example, in a telephone system, features include local call, call forwarding, and conference call. Each feature is generally described in a sequence of stimulus-response pairs, and may include validity checks on inputs, exact sequencing of operations, responses to abnormal situations, including error handling and recovery, effects of parameters, relationships of inputs to outputs, including input/output sequences and formulas for input to output.

**By Stimulus**

Some systems can be best organized by describing their functions in terms of stimuli. For example, the functions of an automatic aircraft landing system may be organized into sections for loss of power, wind shear, sudden change in roll, vertical velocity excessive, etc.

**By Response**

Some systems can be best organized by describing all the functions in support of the generation of a response. For example, the functions of a personnel system may be organized into sections corresponding to all functions associated with generating paychecks, all functions associated with generating a current list of employees, etc.

**By Functional Hierarchy**

When none of the above organizational schemes prove helpful, the overall functionality can be organized into a hierarchy of functions organized by common inputs, common outputs, or common internal data access. Data flow diagrams and data dictionaries can be used to show the relationships between and among the functions and data.

**Additional Comments**

Whenever a new Requirements Specification is contemplated, more than one of the organizational techniques given above may be appropriate. In such cases, organize the specific requirements for multiple hierarchies tailored to the specific needs of the system under specification.

There are many notations, methods, and automated support tools available to aid in the documentation of requirements. For the most part, their usefulness is a function of organization. For example, when organizing by mode, finite state machines or state charts may prove helpful; when organizing by object, object-oriented analysis may prove helpful; when organizing by feature, stimulus-response sequences may prove helpful; and when organizing by functional hierarchy, data flow diagrams and data dictionaries may prove helpful.