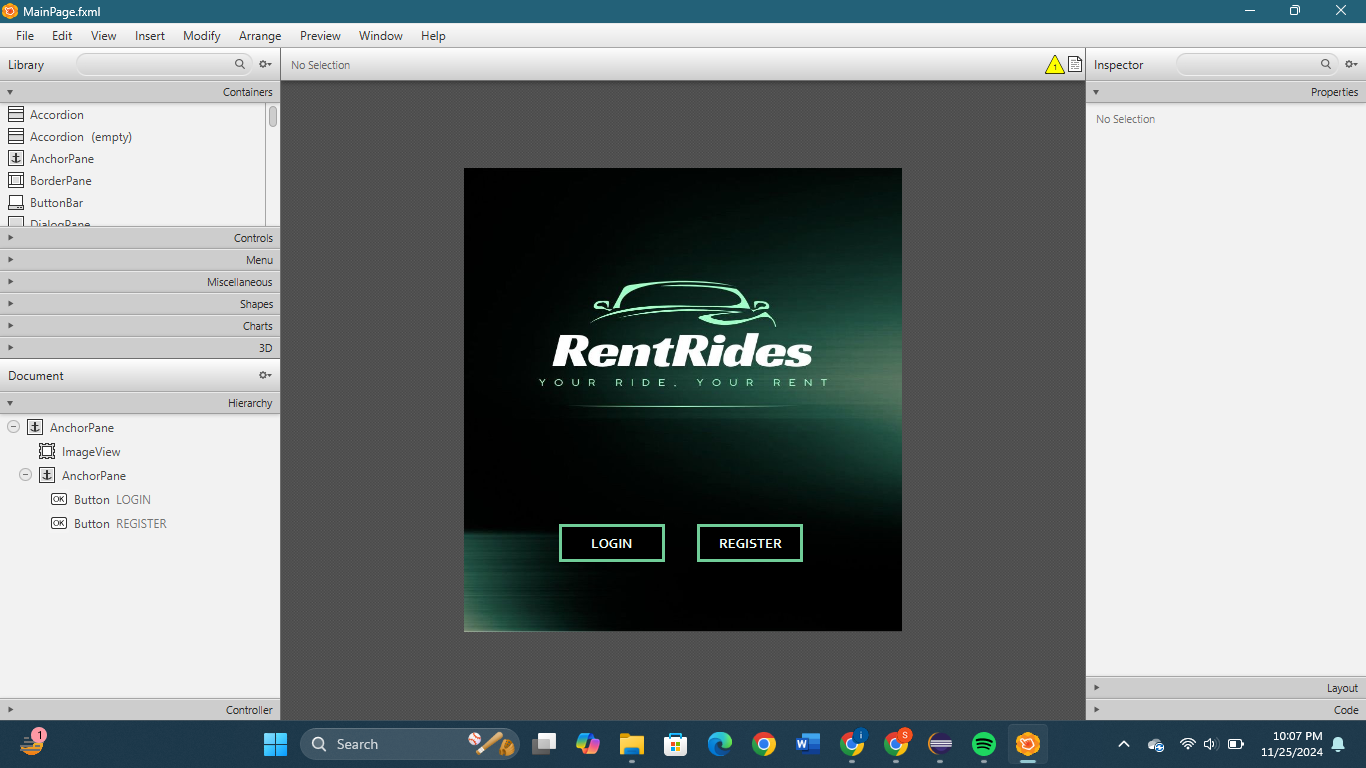
# Software Requirements and Design Document

## RentRides (Vehicle Rental App)

## 



**Prepared by:** Ayesha Kiani(22i-1283), Sana Fatima(22i-1160), M. Abdulrehman(22i-1182)

**FAST NUCES - ISLAMABAD**

**Date: November 2024**

# Table of Contents

1. Introduction  
 1.1 Purpose  
 1.2 Product Scope  
 1.3 Title  
 1.4 Objectives  
 1.5 Problem Statement

2. Overall Description  
 2.1 Product Perspective  
 2.2 Product Functions  
 2.3 List of Use Cases  
 2.4 Extended Use Cases  
 2.5 Use Case Diagram

3. Other Nonfunctional Requirements  
 3.1 Performance Requirements  
 3.2 Safety Requirements  
 3.3 Security Requirements  
 3.4 Software Quality Attributes  
 3.5 Business Rules  
 3.6 Operating Environment  
 3.7 User Interfaces

4. Domain Model  
  
5. System Sequence Diagram  
  
6. Sequence Diagram  
  
7. Class Diagram

# 1. Introduction

## 1.1 Purpose

This document outlines the software requirements and design for RentRides, a JavaFX-based vehicle rental application integrating cars and bikes for middle- and lower-income individuals in Islamabad and Rawalpindi. It addresses vehicle ownership challenges and unreliable public transportation through a user-friendly, secure platform.

## 1.2 Product Scope

RentRides consolidates services from various rental companies into one platform, offering vehicle rentals at affordable rates. It facilitates secure transactions, promotes environmental sustainability by reducing vehicle ownership, and ensures reliable customer support.

## 1.3 Title

**RentRides: Your Ride, Your Rent**

## 1.4 Objectives

The primary objectives of RentRides are:  
- To offer an accessible and affordable platform for vehicle rentals.  
- To reduce carbon emissions by promoting rentals over ownership.  
- To ensure secure transactions and a seamless rental experience.  
- To provide reliable customer support and emergency assistance.

## 1.5 Problem Statement

In Islamabad and Rawalpindi, high vehicle ownership costs and unreliable public transportation present significant mobility challenges. RentRides addresses these issues by providing a flexible, affordable, and secure vehicle rental solution.

# 2. Overall Description

## 2.1 Product Perspective

RentRides is a standalone application designed to bridge the gap in the vehicle rental market. It integrates car and bike rentals, targeting users who face mobility constraints due to financial or infrastructure limitations.

## 2.2 Product Functions

**Key functions of RentRides include:**  
- User registration and profile management.  
- Vehicle browsing and reservation.  
- Secure payment processing.  
- Vehicle return and feedback submission.  
- Rental company and driver management.

## 2.3 List of Use Cases

1. Register User  
2. Browse Vehicle  
3. Make Reservation  
4. Contact Customer Support  
5. Return Vehicle  
6. Give Feedback  
7. Manage Vehicle  
8. Manage Reservation

## 2.4 Extended Use Cases

**Fully Dressed Use Cases**

**Use Case #1:**

**Use Case Name:** Register User

**Scope:** Rent Rides

**Level:** User Goal

**Primary Actor:** User (Renter or Rental Company)

**Stakeholders and Interests:**

* **User (Renter or Rental Company):** The user wants to create an account quickly and securely to access rental services or manage their rental fleet
* **Rental Company**: Require accurate renter information for security, background checks, and potential future interactions with renters.

**Precondition:**

* The user is not already registered on the platform.
* The user has access to the "RentRides" app or website.

**Postcondition:**

* The user is registered successfully on the platform.
* The user can log in and access available services.
* Admin has a record of the registered user for monitoring and compliance purposes.

**Main Success Scenario:**

| **Actor Actions** | **System Responses** |
| --- | --- |
| 1. **User** clicks on the "Sign Up" button on the platform. |  |
|  | 2. **System** displays a registration form for user details. |
| **3. User** fills out the registration form, providing personal details (name, email, password, phone number, address). |  |
| **4. User** submits the form. |  |
|  | **5. System** processes the submission. |
|  | **6. System** stores the user’s details and activates their account. |
|  | **7. System** displays a success message and allows the user to log in. |
| **8. User** can now login using the details provided in the registration form |  |

**Extensions:**

***Invalid Email or Password Format***

* **Actor Action**:

The user provides an invalid email address or username that does not meet the platform's security requirements.

* **System Response**:

The system displays an error message: "Invalid email format" or "username can not contain spaces".

***Email Already Registered***

* **Actor Action**:

The user enters an email that is already associated with an existing account.

* **System Response**:

The system displays an error message: "This email is already registered. Please enter a different username."

***Incomplete Information Entered***

* **Actor Action**:

The user enters incomplete information, e.g. missing address or phone number.

* **System Response**:

The system displays an error message: "Please fill in all the details".

**Use Case #2:**

**Use Case Name:** Browse Vehicle

**Scope:** Rent Rides

**Level:** User Goal

**Primary Actor:** Renter

**Stakeholders and Interests:**

* **Renter:** Wants to easily browse available vehicles, filter options, compare prices, and view vehicle details.
* **Rental Company**: Wants to display its available vehicles and ensure the listing includes sufficient details (e.g., price, availability) to attract renters.
* **Rentrides Platform:** Interested in providing a seamless, user-friendly interface for searching and browsing vehicles, ensuring all rental companies' listings are accessible.

**Preconditions:**

* The renter is registered and logged into the platform.
* There are available vehicles listed on the platform by rental company's site.

**Postconditions:**

* The renter has browsed available vehicles and may proceed to either:
* Select a vehicle for rent.
* Refine their search or continue browsing.

**Main Success Scenario:**

| **Actor Actions** | **System Responses** |
| --- | --- |
| 1. **Renter** navigates to the "Browse Vehicles" section in the app. |  |
|  | 2. **System** displays a list of available vehicles, including cars and bikes. |
| **3. Renter** selects filters or search criteria (e.g., type of vehicle, price range, location, availability dates). |  |
|  | **4. System** applies the filters and updates the list of available vehicles. |
| **5. Renter** clicks on a specific vehicle to view details (e.g., model, price per day, features, rental company, reviews). |  |
|  | **6. System** displays the vehicle details, including photos, price breakdown, and renter reviews. |
| **7. Renter** decides to:  - Save the vehicle for later.  - Proceed to rent the vehicle (leading to the "Make Reservation" use case).  - Refine the search or continue browsing. |  |
|  | **8.** The system performs the chosen action:  - Saves the vehicle to the renter’s list.  - Redirects the renter to the rental process.  - Updates the list with refined search criteria. |

**Extensions:**

***No Available Vehicles Matching Search Criteria***

* **Actor Action**:

The renter applies search filters (e.g., a specific price range or vehicle type), but no vehicles match the criteria.

* **System Response**:

The system displays a message: "No vehicles found matching your search criteria" and suggests alternative options (e.g., adjusting filters or expanding the search range).

***Renter Cancels Browsing***

* **Actor Action**:

The renter decides to cancel browsing before choosing a vehicle.

* **System Response**:

The system exits the browsing process and returns the renter to the main page.

**Use Case #3:**

**Use Case Name:** Make Reservation

**Scope:** Rent Rides

**Level:** User Goal

**Primary Actor:** Renter

**Stakeholders and Interests:**

* **Renter:** Wants to quickly and securely reserve a vehicle for the desired dates and times.
* **Rental Company**: Wants to confirm vehicle availability and ensure secure payment for the reservation.
* **Rentrides Platform:** Ensures a smooth reservation process, integrating vehicle availability, secure payment, and rental terms.

**Preconditions:**

* The renter is registered and logged into the Rentrides platform.
* The renter has already browsed and selected a vehicle to reserve (leading from "Browse Vehicles" use case).
* The selected vehicle is available for the desired rental period.

**Postconditions:**

* The renter successfully reserves the vehicle, receives confirmation, and is notified of the pickup details.
* The vehicle is marked as reserved in the system, and renter proceeds to initiate payment.
* If a driver was requested, the designated rental company is also notified.

**Main Success Scenario:**

| **Actor Actions** | **System Responses** |
| --- | --- |
| 1. **Renter** selects the vehicle they wish to reserve from the available list. |  |
|  | 2. **System** displays the Vehicle Details |
| **3. Renter** clicks on "Rent Vehicle" and selects the desired rental dates, and time period. They may select to request a driver for the duration as well. |  |
|  | **4. System** checks vehicle availability for the specified dates and confirms that it is available. |
|  | **5. System** checks available drivers for the selected vehicle and dates. |
| **6. Renter** confirms the vehicle and driver selection (if chosen) and submits the reservation request |  |
|  | **7.** **System** confirms the vehicle and driver availability and creates a reservation record. |
|  | **8. System** sends notifications to both the renter and rental company. The reservation status is updated in the renter’s account. |
| **9. Renter** receives a confirmation message with reservation details. |  |

**Extensions:**

***Vehicle Unavailable for Selected Dates***

* **Actor Action**:

The renter selects a vehicle and attempts to reserve it, but the vehicle is unavailable for the specified dates.

* **System Response**:

The system displays an error message: "This vehicle is unavailable for the selected dates. Please choose different dates or select another vehicle."

***Renter Chooses Not to Add a Driver***

* **Actor Action**:

The renter does not need a driver and skips the driver selection step.

* **System Response**:

The system skips the driver selection step and proceeds with the vehicle reservation only.

**Extensions:**

***Payment Authorization Fails***

* **Actor Action**:

The renter enters payment details, but the payment gateway cannot authorize the transaction due to issues like insufficient funds or an invalid card.

* **System Response**:

The system displays an error message: "Payment authorization failed. Please check your payment details or try another payment method."

***Renter Cancels Payment Process***

* **Actor Action**:

The renter decides to cancel the payment process before confirming the transaction.

* **System Response**:

The system cancels the payment and notifies the renter that the reservation has been cancelled.

**Use Case #4:**

**Use Case Name:** Contact Customer Support

**Scope:** Rent Rides

**Level:** User Goal

**Primary Actor:** Renter

**Stakeholders and Interests:**

* **Renter:** Wants to resolve any issues quickly and easily, whether it's related to reservations, payments, or other services.
* **Rental Company**: Interested in addressing any operational or service issues that renters raise and ensuring customer satisfaction.
* **Admin:** Interested in efficiently resolving user issues and providing satisfactory resolutions to improve customer experience.

**Preconditions:**

* The renter is registered and logged in to the Rentrides platform.
* The renter has access to the "Contact Support" option through the platform.

**Postconditions:**

* The renter successfully contacts customer support and either receives a resolution or is notified of next steps (follow-up).

**Main Success Scenario:**

| **Actor Actions** | **System Responses** |
| --- | --- |
| 1. **Renter** navigates to the “Contact Support” option from the app menu. |  |
|  | **2. System** opens a form for renter to enter relevant details about the issue encountered. |
| **2. Renter** provides the necessary details about the issue ( problem description). |  |
|  | **3.**  **System** connects the renter to the appropriate customer support agent (part of the Admin). |
|  | **6. System** logs the issue and provides the agent with the renter’s account details and issue history. |
| **7. Customer Support Agent** interacts with the renter, aiding or requesting additional information. |  |
|  | **8. System** logs the support interaction, updates the renter’s account if necessary, and sends a follow-up confirmation email or message to the renter. |

**Extensions:**

***Technical Issue with Form Submission***

* **Actor Action**:

The renter fills the information form, but the system encounters a technical error while submitting it to the customer support agent.

* **System Response**:

The system displays an error message: “Unable to connect to support currently. Please try again later.” It logs the issue for further investigation.

**Use Case #5:**

**Use Case Name:** Return Vehicle

**Scope:** Rent Rides

**Level:** User Goal

**Primary Actor:** Renter

**Stakeholders and Interests:**

* **Renter:** Wants to return the vehicle promptly, ensure it meets the rental conditions, and avoid any additional charges.
* **Rental Company**: Interested in receiving the vehicle back in good condition, verifying mileage, and ensuring proper documentation of the return process.
* **Driver (if applicable):** If a driver was hired, they need to finalize their part of the rental agreement and return to the rental company's location.

**Preconditions:**

* The renter is logged into their RentRides account and has the necessary details about the rental.
* The renter has completed their rental period and is ready to return the vehicle.

**Postconditions:**

* The vehicle is successfully returned to the rental company, inspected, and logged into the system, and the renter receives confirmation of the return.

**Main Success Scenario:**

| **Actor Actions** | **System Responses** |
| --- | --- |
| 1. **Renter** navigates to the “Return Vehicle” section on the app. |  |
|  | 2. **System** displays the “Return Vehicle” page with input fields for registration number and renter’s username. |
| **3. Renter** enters all the information and proceeds to return the vehicle. |  |
|  | **4. System** validates the input data and ensure the data provided is correct. |
| **5. Renter** submits the vehicle return request. |  |
|  | **6. System** confirms the vehicle return request and sends a follow up confirmation message. |

**Extensions:**

***Renter Returns Vehicle Late***

* **Actor Action**:

The renter returns the vehicle after the scheduled return time.

* **System Response**:

The system calculates a late fee based on the rental company’s policies and notifies the renter about the extra charge. The system updates the total amount due and prompts the renter to pay the additional fee before completing the return process.

**Use Case #6:**

**Use Case Name:** Give Feedback

**Scope:** Rent Rides

**Level:** User goal

**Primary Actor:** Renter

**Stakeholders and Interests:**

* **Renter:** Wants to provide feedback on the vehicle or service
* **Rental Company:** Interested in renter feedback to improve services

**Preconditions:**

* The renter has a valid account and is logged into the platform.
* History of previous reservation exists.
* The renter has returned the rented vehicle.

**Postconditions:**

* The feedback is recorded and stored in the system.
* The rental company can view and respond to the feedback.
* The feedback is logged for future renter interactions and company evaluations.

**Main Success Scenario:**

| **Actor Actions** | **System Responses** |
| --- | --- |
| 1. **Renter** initiates the process of logging into the platform. |  |
|  | 2. The **system** verifies the renter's login credentials. |
| 3. **Renter** navigates to the section for vehicle return. |  |
|  | 4. The **system** confirms that the vehicle return process has started. |
| 5. **Renter** completes the vehicle return. |  |
|  | 6. The **system** updates the reservation status to "returned." |
| 7. **Renter** waits for a prompt to provide feedback. |  |
|  | 8. The **system** prompts the renter to give feedback on the rental experience. |
| 9. **Renter** fills out and submits the feedback form. |  |
|  | 10. The **system** securely stores the feedback. |
|  | 11. The **system** makes the feedback available to the rental company for review. |

**Extensions:**

***Vehicle Not Returned***

* **Actor Action:**  
  The renter attempts to provide feedback without returning the vehicle.
* **System Response:**  
  The system informs the renter that feedback cannot be submitted until the vehicle is returned: "Please return the vehicle before submitting feedback."

***Feedback Submission Failure***

* **Actor Action:**  
  The renter submits the feedback form.
* **System Response:**  
  The system encounters an error and retries submission, notifying the renter of the failure: "We encountered an issue submitting your feedback. Please try again."

***No Response to Feedback Prompt***

* **Actor Action:**  
  The renter does not respond to the feedback prompt.
* **System Response:**  
  The system sends a reminder notification to the renter after a specified time: "We'd love your feedback! Please take a moment to share your experience."

**Use Case #7:**

**Use Case Name:** Manage Vehicle

**Scope:** Rent Rides

**Level:** User goal

**Primary Actor:** Rental Company

**Stakeholders and Interests:**

* **Rental Company:** Needs to manage the vehicle inventory.
* **Admin:** Oversees the quality of listings.
* **Renters:** Need access to updated vehicle information.

**Preconditions:**

* The company is registered and approved on the platform.
* The company has all the valid documents.
* The company has a valid login and is authenticated.
* The platform supports vehicle management features.
* The company has at least one vehicle to add, edit, or remove.

**Postconditions:**

* The vehicle list is updated on the platform.
* Renters can see updated information for the vehicles.
* The system records changes made for future reference.
* The company is notified if any updates fail.

**Main Success Scenario:**

| **Actor’s Action** | **System Responses** |
| --- | --- |
| 1. The **company** logs into the platform. |  |
|  | 2. The **system** authenticates the company. |
| 3. The **company** selects the option to manage vehicles. |  |
|  | 4. The **system** displays the company’s current vehicle list. |
| 5. The **company** chooses to add a new vehicle by entering details (make, model, rental price, availability). |  |
|  | 6. The **system** verifies the accuracy and completeness of the new vehicle details. |
| 7. The **company** confirms the addition of the new vehicle. |  |
|  | 8. The **system** updates the vehicle list and makes the new vehicle visible to renters. |
| 9. The **company** selects an existing vehicle to edit and updates details like rental price or availability. |  |
|  | 10. The **system** verifies the updated details and saves the changes, updating the vehicle information visible to renters. |
| 11. The **company** chooses to remove a vehicle from the listing. |  |
|  | 12. The **system** checks if the vehicle has any active reservations. If not, the system removes the vehicle from the platform and notifies the company. |

**Extensions:**

***Incorrect Or Missing Vehicle Details***

* **Actor Action:**  
  The company attempts to add or update vehicle details that are invalid.
* **System Response:**  
  The system displays an error message: "Please ensure all vehicle details are correct."

***Update Failure***

* **Actor Action:**  
  The company submits updates, but the system encounters an error during processing.
* **System Response:**  
  The system retries the update and, if unsuccessful, informs the company: "Vehicle update failed. Please try again."

***Vehicle Removal Blocked***

* **Actor Action:**  
  The company attempts to remove a vehicle that is currently reserved.
* **System Response:**  
  The system prevents the removal and displays a message: "Vehicle cannot be removed due to active reservations."

**Use Case #8:**

**Use Case Name:** Manage Reservation

**Scope:** Rent Rides

**Level:** User goal

**Primary Actor:** Rental Company

**Stakeholders and Interests:**

* **Rental Company:** Needs to manage bookings and availability.
* **Renters:** Need confirmation and updates on their reservations.
* **Admin:** Monitors booking activities.

**Preconditions:**

* The company has listed vehicles and drivers on the platform.
* The company is logged into the platform with valid credentials.
* The system has access to current reservation data.
* The company has received requests from renters who made reservations.
* The company has a process in place for handling vehicle reservations, cancellations and updates.

**Postconditions:**

* The reservation status is updated in the system (confirmed, pending or cancelled).
* Renters receive notifications regarding the status of their reservations.
* The system logs the changes.
* The company can view the updated reservation list.
* Any conflicts regarding vehicle or driver availability are resolved and communicated to the renter.

**Main Success Scenario:**

| **Actor Action** | **System Responses** |
| --- | --- |
| 1. The **company** logs into the platform. |  |
|  | 2. The **system** authenticates the company’s login credentials. |
| 3. The **company** navigates to the reservation management section. |  |
|  | 4. The **system** displays the list of all current reservations. |
| 5. The **company** adds a new reservation or selects an existing one to edit or remove. |  |
|  | 6. The **system** verifies the details of the reservation or edits. |
| 7. The **company** confirms the action (add, edit, or remove). |  |
|  | 8. The **system** updates the reservation list accordingly:  - If ***added***, the new reservation is saved, and the renter is notified.  - If ***edited***, the updated details are reflected, and the renter is notified.  - If ***removed****,* the system cancels the reservation and informs the renter of the cancellation. |
| 9. The **company** checks vehicle after any reservation changes. |  |
|  | 10. The **system** displays the latest availability and adjusts vehicle status as needed. |

**Extensions:**

***Reservation Cancellation by Company***

* **Actor Action:**  
  The company decides to cancel a reservation made by a renter.
* **System Response:**  
  The system informs the renter of the cancellation and suggests alternative vehicles or dates: "Your reservation has been cancelled. Here are some alternative options."

***Vehicle Unavailability***

* **Actor Action:**  
  The company attempts to confirm a reservation, but the vehicle is no longer available.
* **System Response:**  
  The system prompts the company to provide a replacement vehicle or alternative options: "The selected vehicle is unavailable. Please choose a different vehicle."

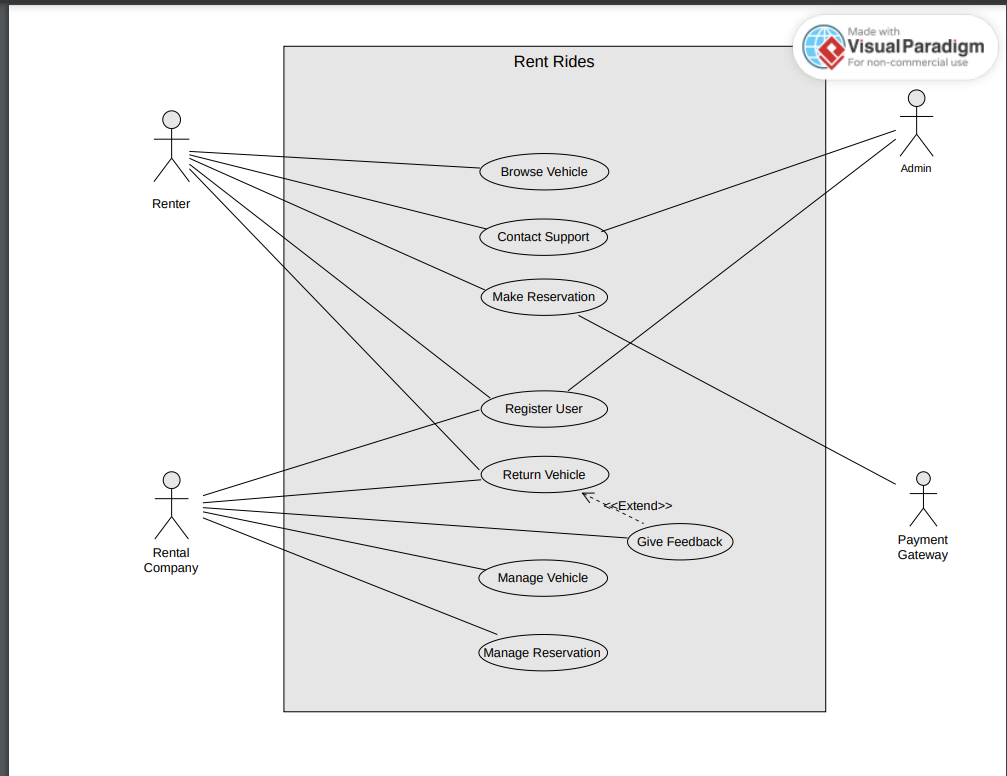
***System Error During Update***

* **Actor Action:**  
  The company tries to update the status of a reservation, but the system encounters an error.
* **System Response:**  
  The system notifies the company of the issue and logs the error: "Unable to update reservation status at this time. Please try again later."

***Renter Notification Failure***

* **Actor Action:**  
  The company updates a reservation status, but the notification to the renter fails.
* **System Response:**  
  The system logs the error and attempts to resend the notification while informing the company: "Renter notification failed. We are retrying."

## 2.5 Use Case Diagram



# 3. Other Nonfunctional Requirements

## 3.1 Performance Requirements

RentRides must ensure fast response times for user interactions, with a maximum page load time of 2 seconds. The system should handle up to 1,000 simultaneous users without performance degradation.

## 3.2 Safety Requirements

**The app ensures safety by:**  
- Validating user and vehicle information.  
- Encrypting payment transactions.  
- Providing verified contact details for emergency support.

## 3.3 Security Requirements

**Security measures include:**  
- Role-based access control.  
- Data encryption for sensitive information.

## 3.4 Software Quality Attributes

The app emphasizes reliability, usability, and maintainability:  
**Reliability**: Ensures minimal downtime and consistent performance.  
**Usability**: Intuitive design for ease of navigation.  
**Maintainability**: Modular code structure for scalability.

## 3.5 Business Rules

**Operating principles include:**  
- Renters must be verified before making reservations.  
- Rental companies are responsible for maintaining vehicle quality.  
- Secure communication protocols are mandatory for data exchange.

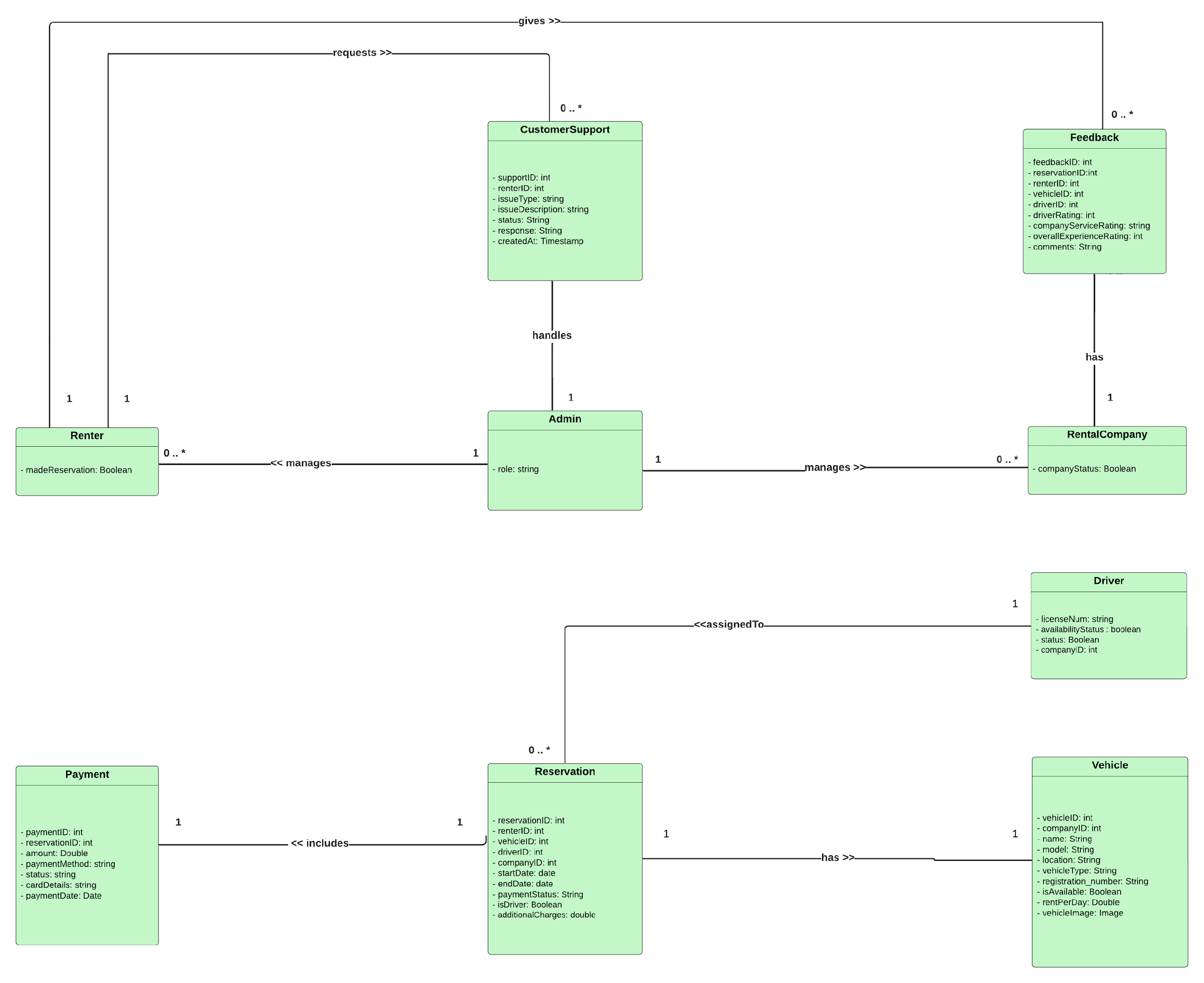
## 3.6 Operating Environment

The application is designed for desktop systems running Windows 10 or above with Java 8 or higher installed. It relies on a PostgreSQL backend hosted on secure cloud servers.

## 3.7 User Interfaces

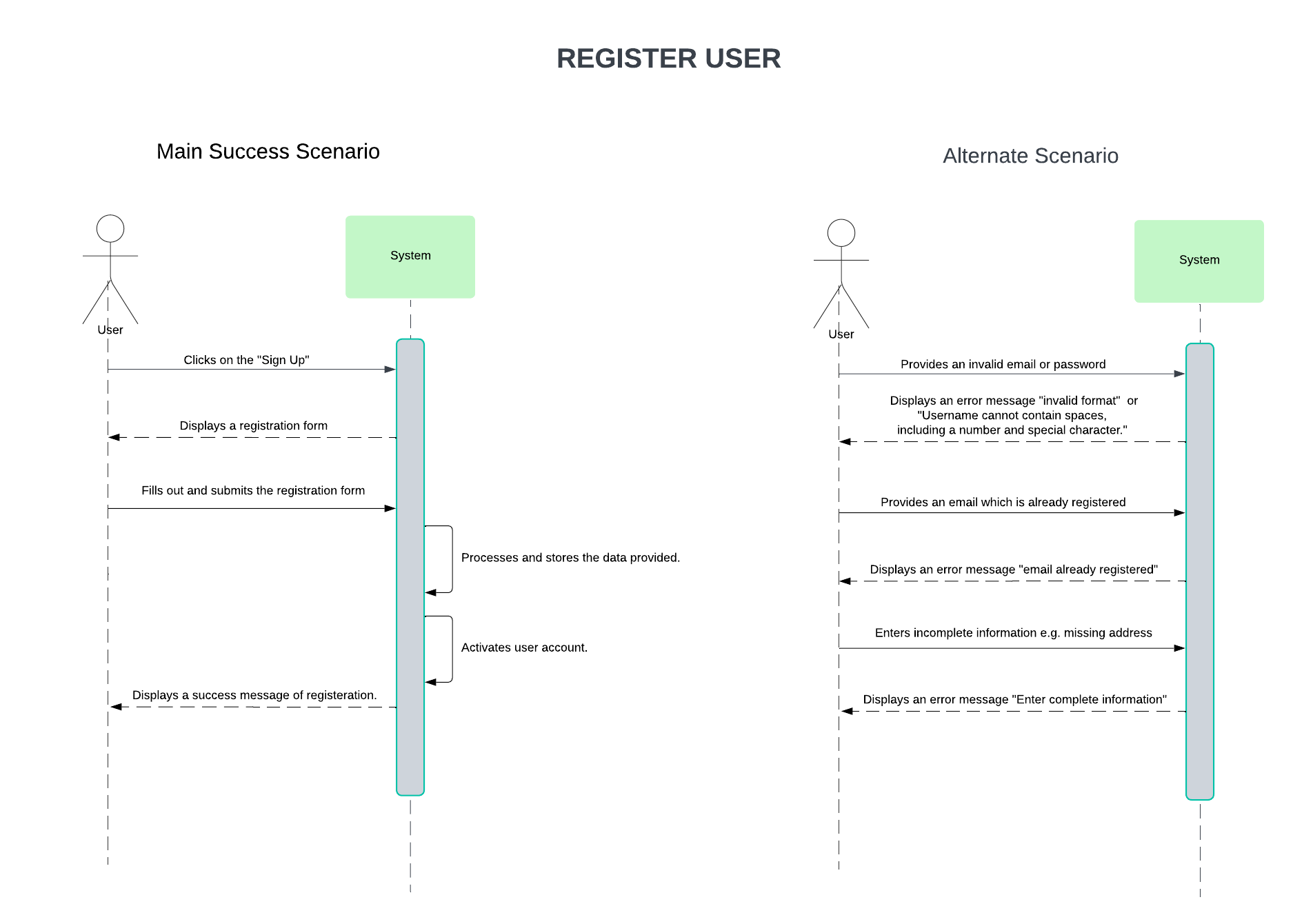
User interfaces include:  
- A clean, responsive GUI for renters, rental companies, and admins.  
- Features like dropdown menus, search filters, and real-time notifications.  
- Error messages and tooltips to guide user actions.

# 4. Domain Model



# 5. System Sequence Diagram

1. Register User



1. Browse Vehicle

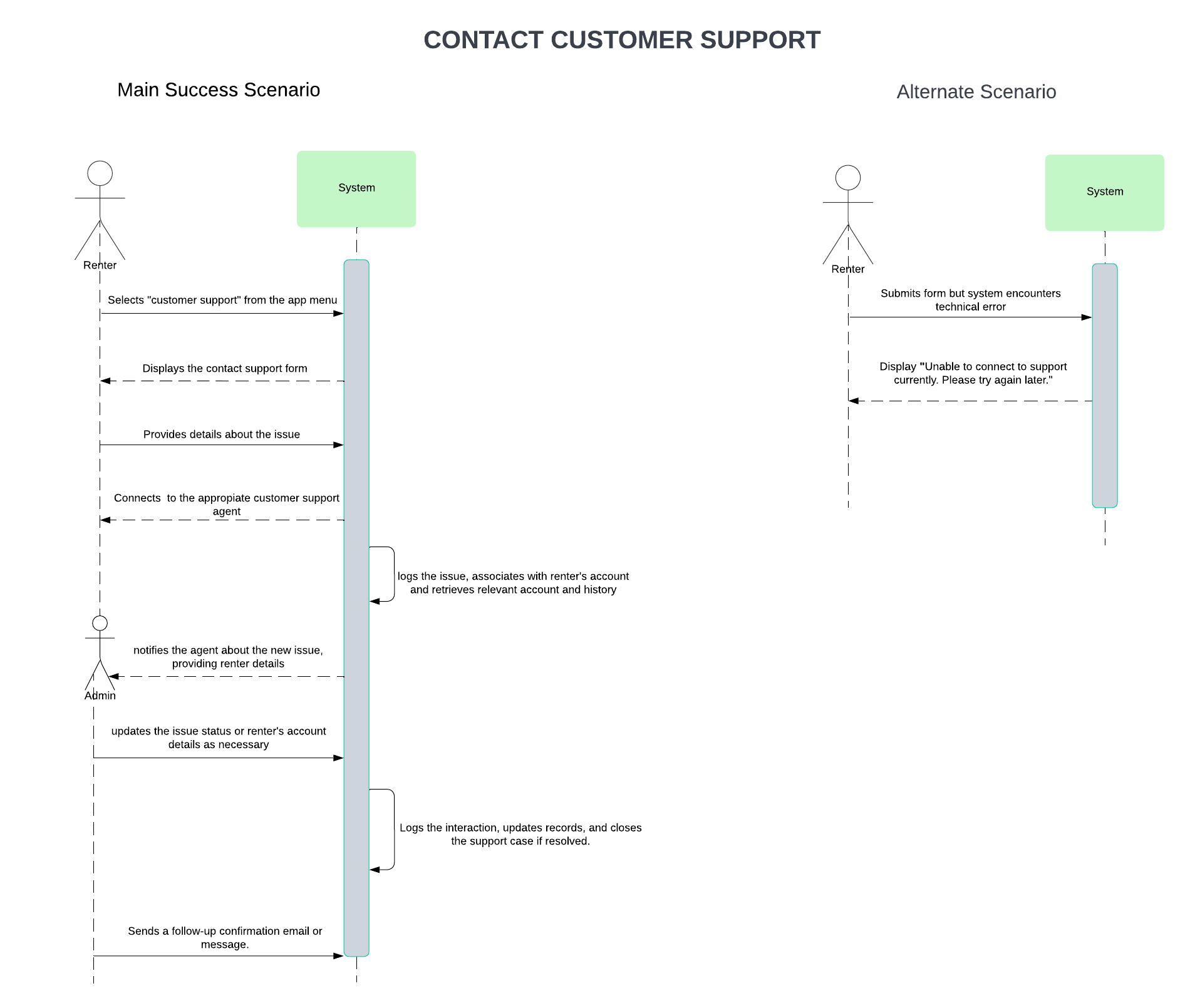
A diagram of a vehicle

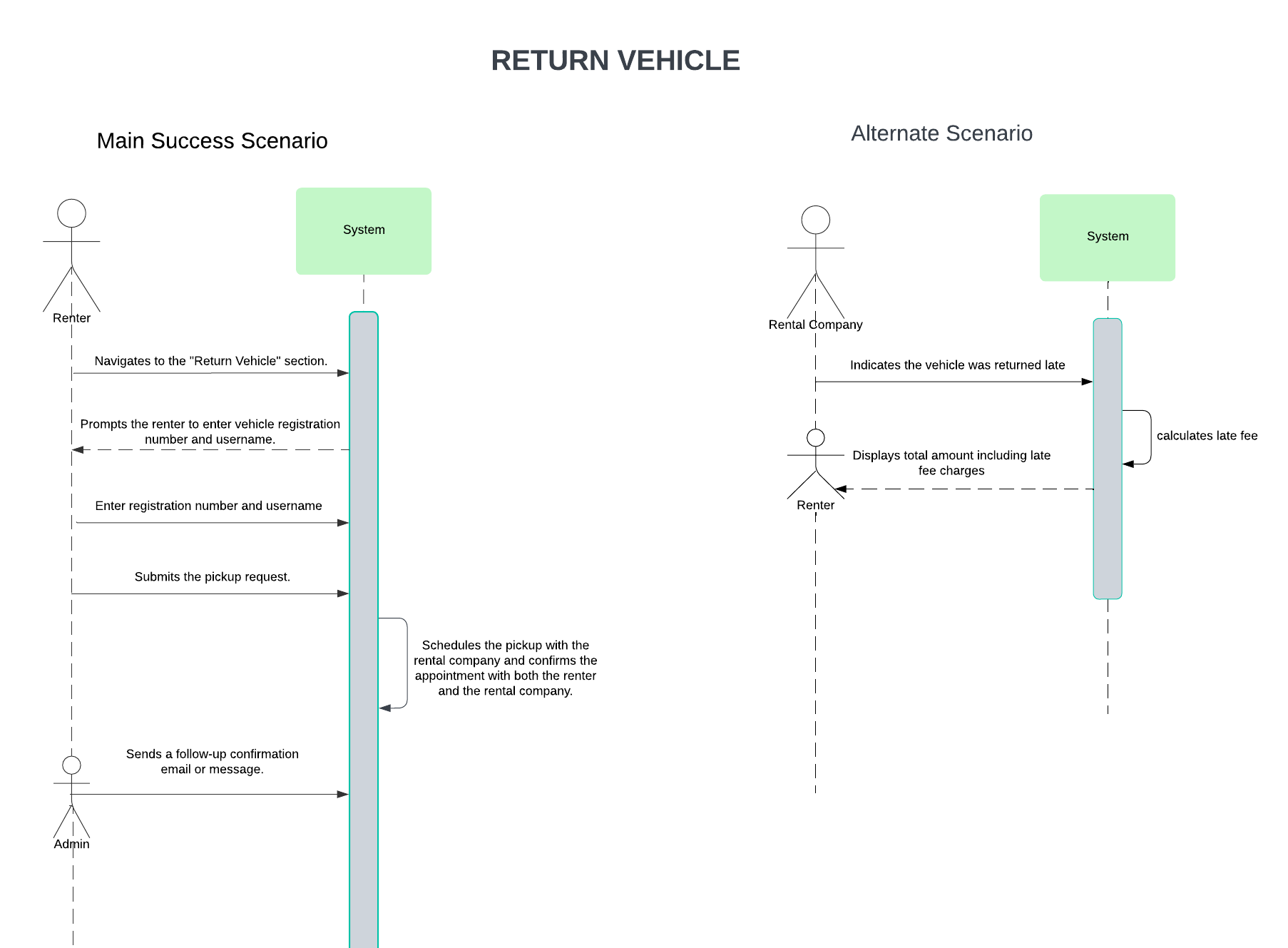
Description automatically generated

1. Make Reservation

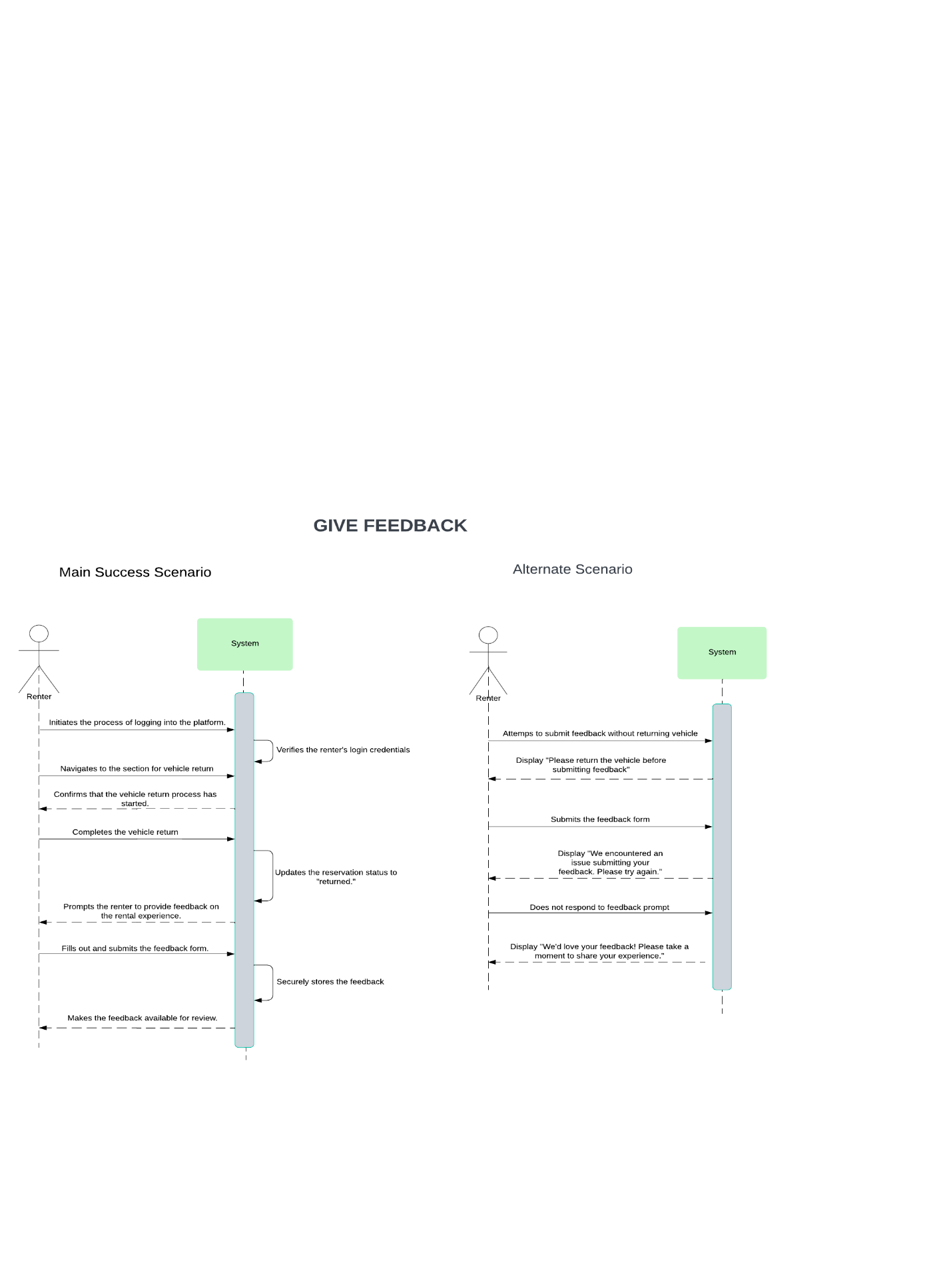
A close-up of a paper

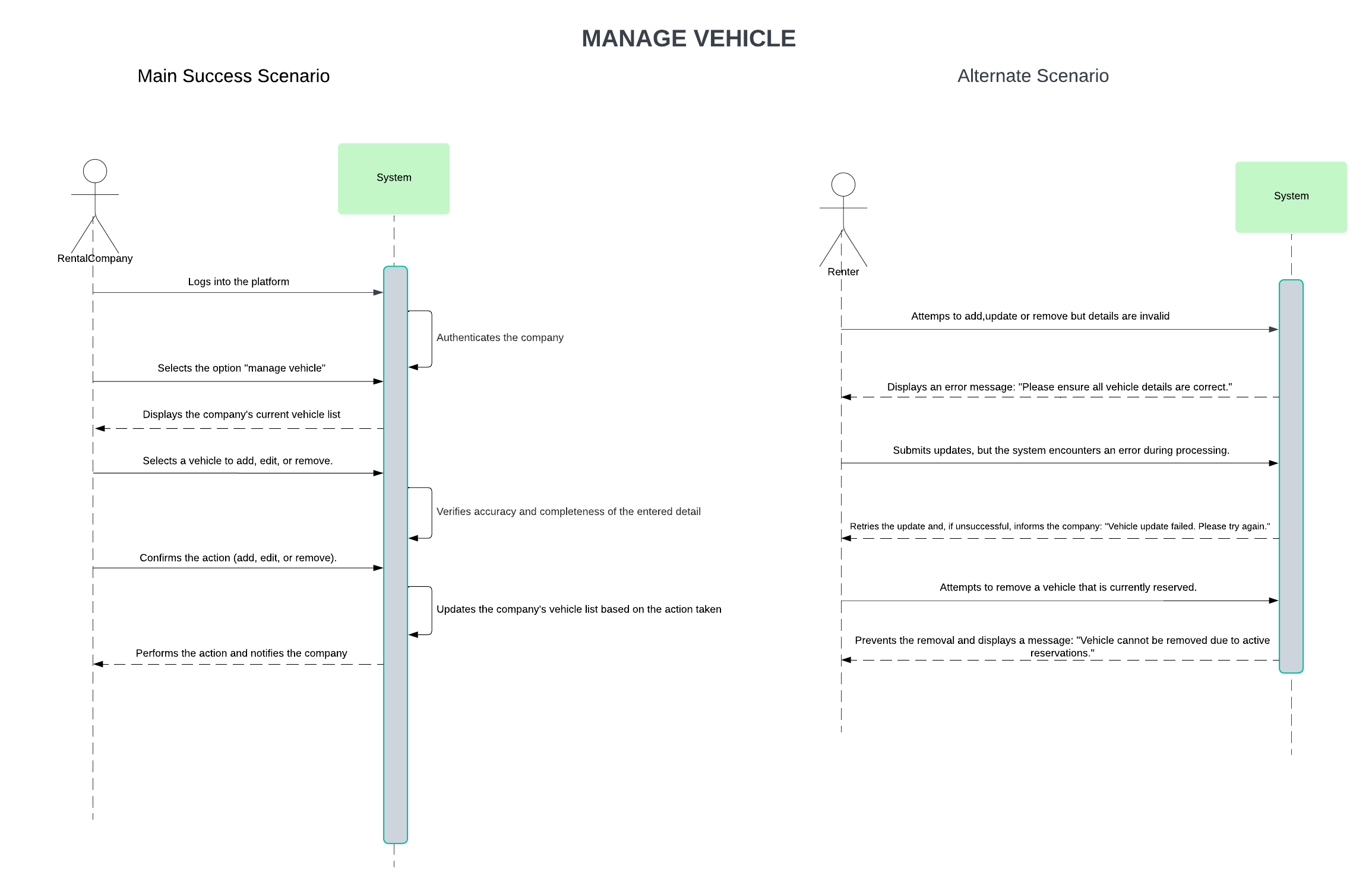
Description automatically generated

1. Contact Customer Support
2. Return Vehicle



1. Give Feedback

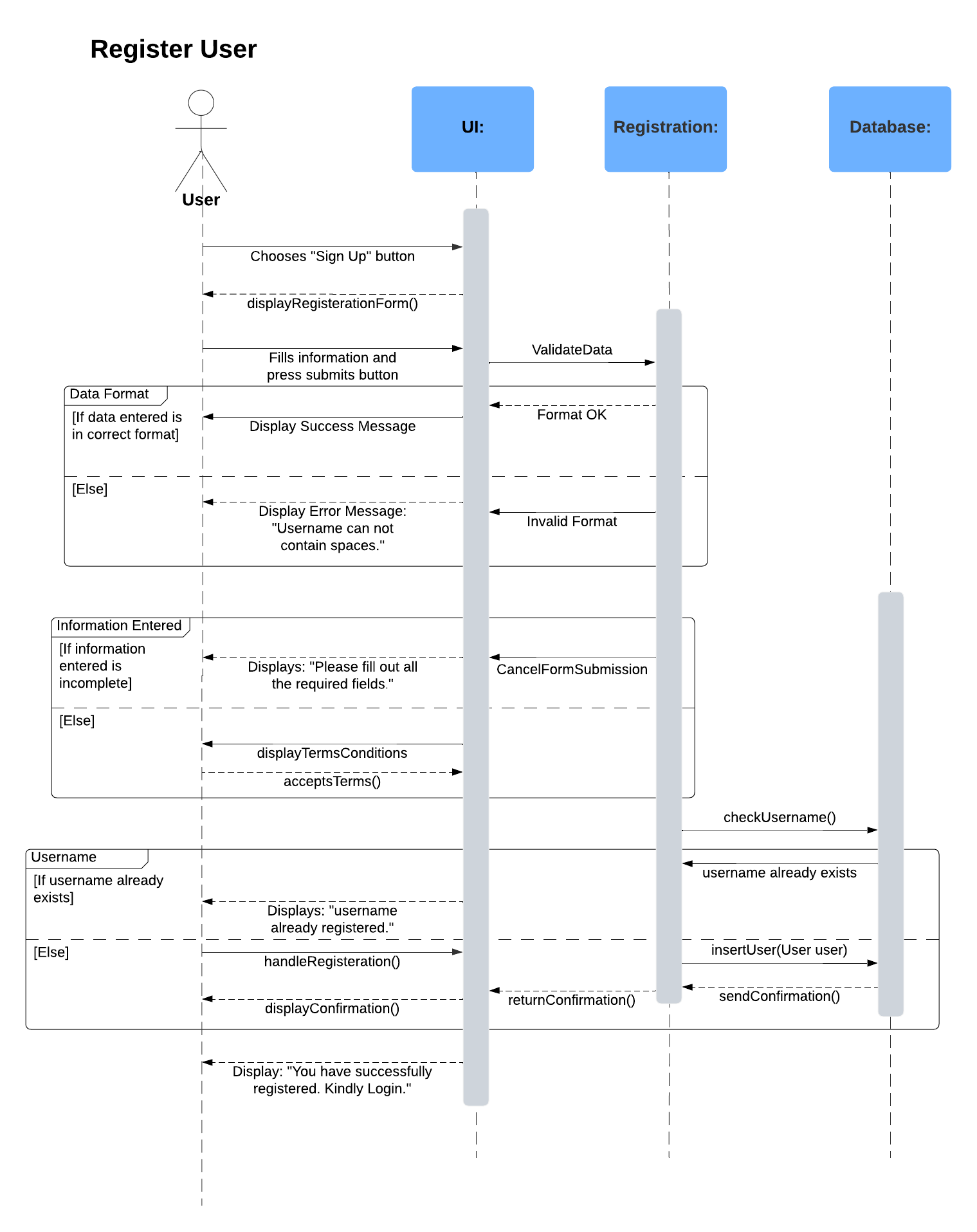


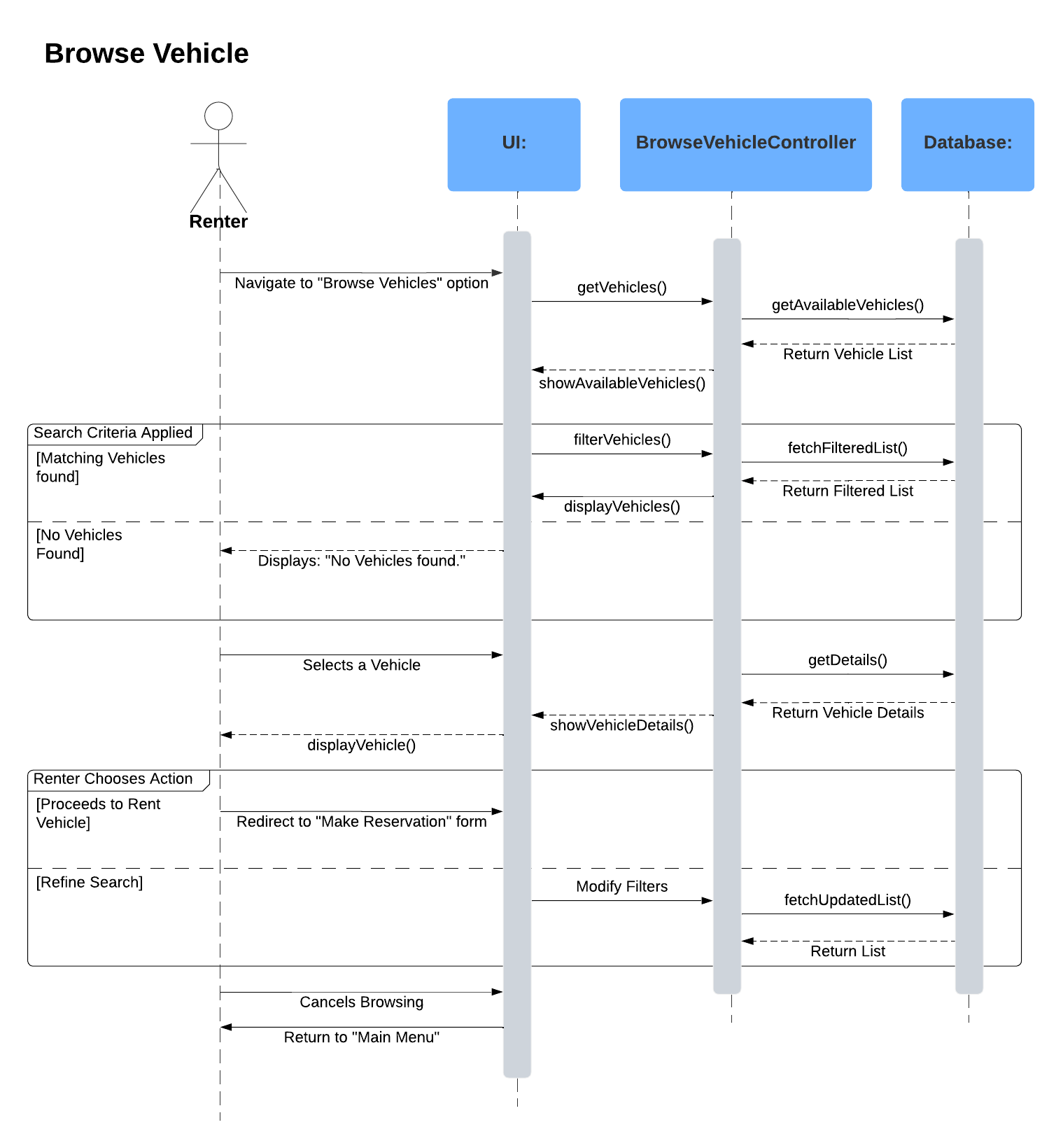
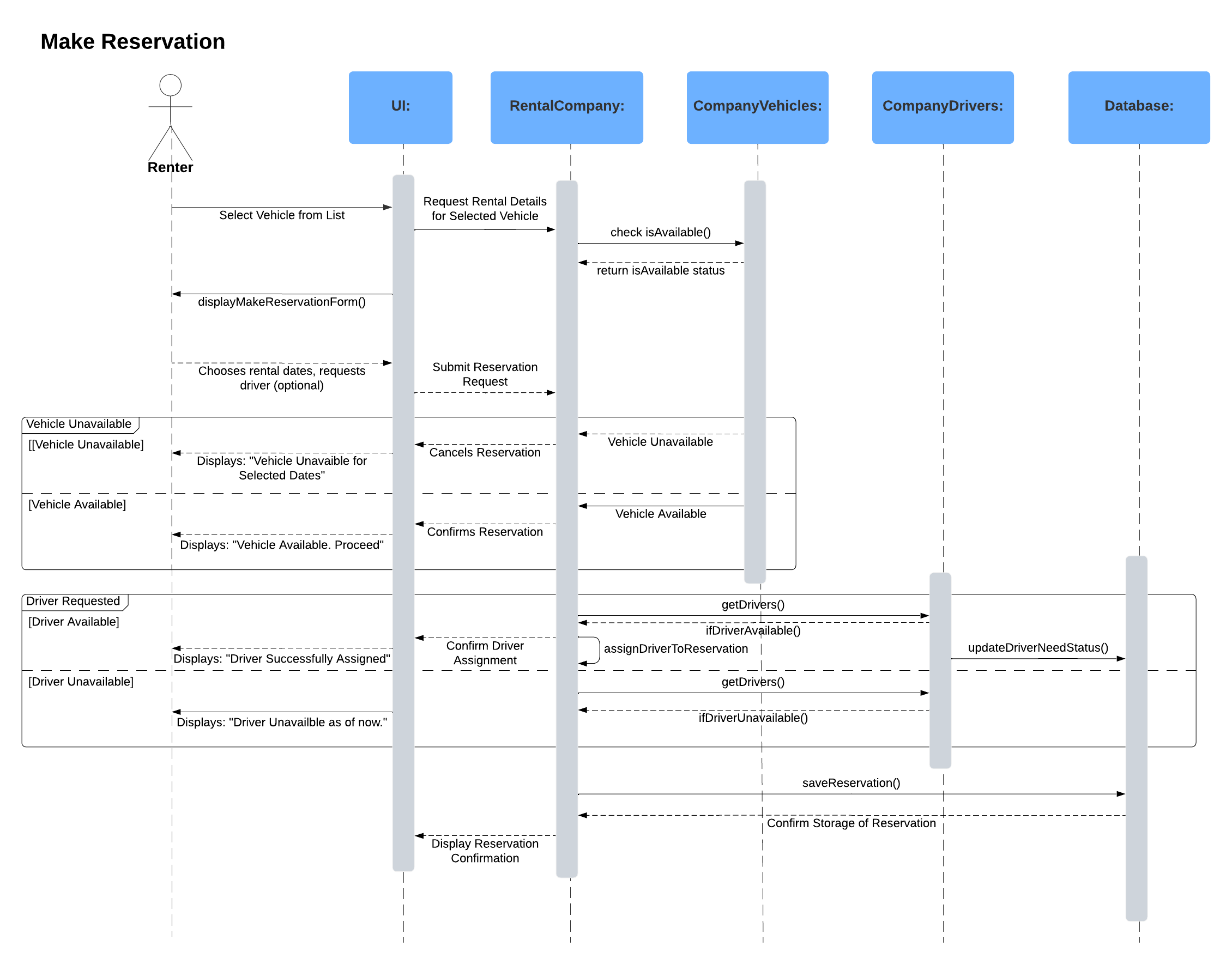
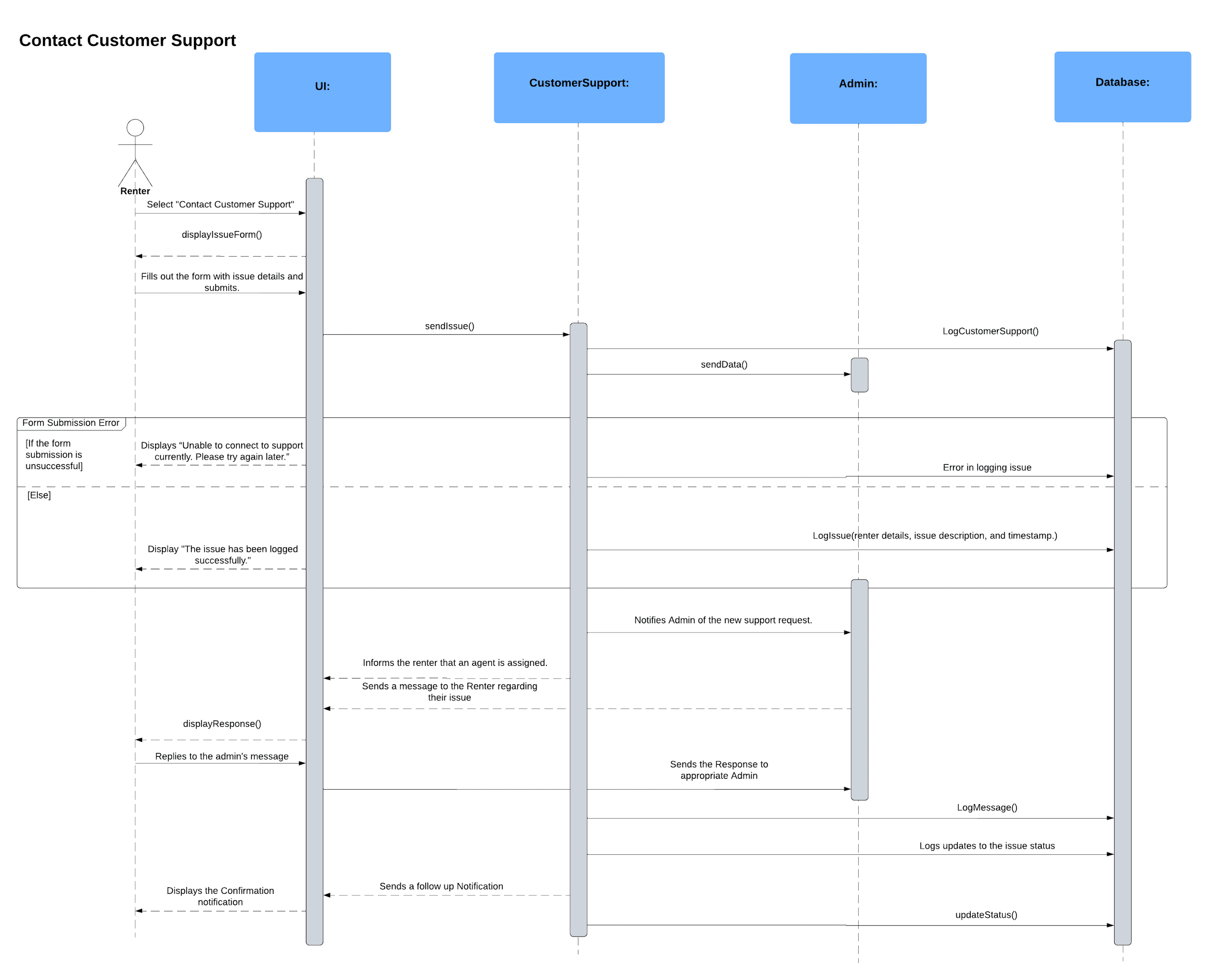
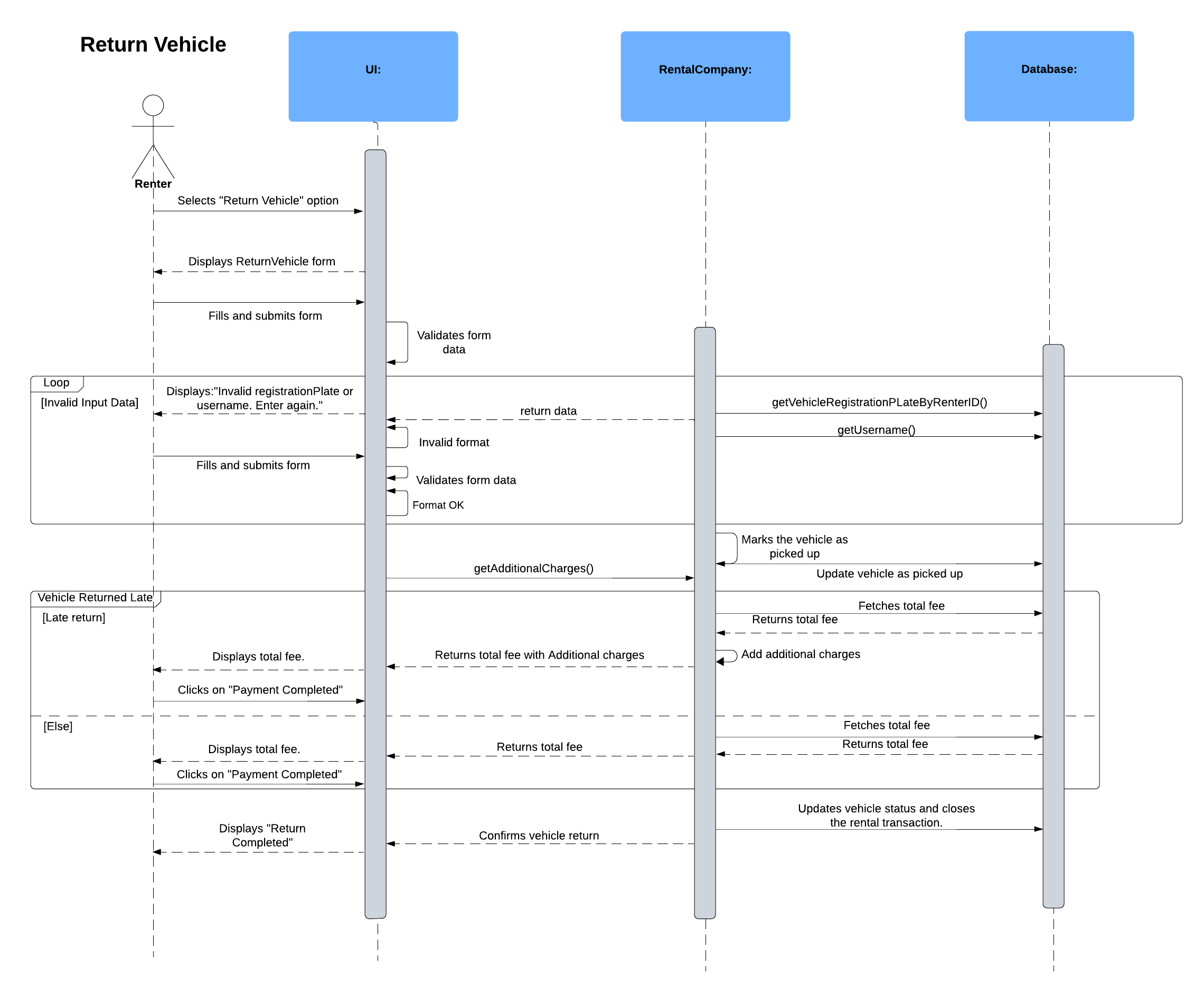
1. Manage Vehicle
2. Manage ReservationA close-up of a diagram

   Description automatically generated

# 6. Sequence Diagram

1. Register User



1. Browse Vehicle
2. Make Reservation
3. Contact Customer Support
4. Return Vehicle
5. Give FeedbackA screenshot of a computer

   Description automatically generated
6. Manage VehicleA close-up of a document

   Description automatically generated
7. Manage ReservationA close-up of a document

   Description automatically generated

# 7. Class Diagram

