Software Requirements Specification

For

“Vehicle Renting System”

Prepared By:

Ashi Bagedi (0801CS201022)

Date: 23/02/2023

Problem Statement :

The purpose of this vehicle renting system aims to offer an efficient and user-friendly solution for renting vehicles that meets the needs of both customers and rental companies.

Based on observations, customers who want to avail the rental services needs to go to nearby rental showrooms or have to search for it, due to lack of knowledge of their locality for the newcomer, someone who wants to rent a vehicle for some days or someone who wants a vehicle for traveling, finding such an offline store or searching for it would be a tedious task.

Small scale companies cannot able to build a massive garage for supplying their vehicles online and managing a website would make it more complex for running a small scale renting business and it makes it easier for them to supply vehicles in their locality only, we can linked all those small scale renting businesses and provide them a platform where they can able to rent their vehicles online.

With the help of the new system, rental companies will be able to expand their business by reaching a broader customer base, while customers can benefit from a convenient and hassle-free renting experience.

Table of Contents

1. Introduction

Purpose

- The purpose of this project is to provide customers with a convenient and flexible way to rent vehicles for their transportation needs. This can include short-term rentals for daily use or longer-term rentals for trips or events. The project aims to provide a user-friendly platform for customers to search and select the vehicles they need(on rent), and make reservations or bookings online.

- This document contains the software requirements for the Vehicle Rental System VRS.

- The main purpose of a Vehicle Rental System to be implemented is to manage information about Vehicles, Customers and the system users related to the company.

Document Conventions

- ➢ This document was prepared using the IEEE recommended practice for Software Requirements Specification.
- ➢ Here we have used font size 18 for title with bold format, font size for subtitle with bold format and font size 14 for the content .

Intended Audience and Reading Suggestions

The document is intended for all the stakeholder customers, administrater and companies. The Stakeholders are assumed to have basic knowledge of Mobile and Web Application.

Product Scope

- This system keeps detailed records of both the vehicle and the customers, the duration they rent the vehicle as well as the type of vehicle they rent.
- This system will be mainly designed for small companies that render vehicle rental services to customers.

- This system will have the ability to generate and print invoice for each successful transaction.
- Customers will be able to make good use of the system.

References

For the sample template - https://web.cs.dal.ca/~hawkey/3130/srs_template-ieee.doc

2. System Features

These are statements of services the system should provide, how the system should react to particular inputs, and how the system should behave in particular situations. It specifies the application functionality that the developers must build into the product to enable users to accomplish their tasks.

2.1 User Accounts

This system should allow users to register themselves using their email or mobile number.

2.1.1 Register

User: Customer, Vehicle owner, admin.

Input: In registration form enter your personal credentials (name, email id, and password).

Output: Registration successful and redirect to login page.

Alternative flow: If already registered then display the message already registered.

2.1.2 View account information

User: Customer, Vehicle owner, admin.

Input: View account option.

Output: Account information is displayed.

Alternative flow: None

2.1.3 Update account

User: Customer, Vehicle owner, admin.

Input: Enter account update details

Output: Account successfully updated.

Alternative flow: None

2.1.4 Login

User: Customer, Vehicle owner, admin.

Input: Email Id/Username, password

Output: Login Successfully

Alternative flow: Login failed and redirect to login page.

2.2 Rent Vehicle

2.2.1 Search

User: Customer

Input: Vehicle type, date and time.

Output: According to input, a list of available vehicles will be displayed.

Alternative flow: Vehicle not available of that type or on that date and time.

2.2.2 Booking

User: Customer

Input: After selecting the desired vehicle enter booking details, payment type and accept terms & conditions.

Output: Booking successful, booking details and a redirect button to the payment page will be displayed.

Alternative flow: Booking not available for this vehicle or Network error please refresh and try again.

2.2.3 Payment

User: Customer

Input: Enter desired payment details.

Output: Successful payment and vehicle is reserved.

Alternative flow: If payment failed then retry again.

2.2.4 Cancel Booking

User: Customer

Input: Enter Booking phone number.

Output: Booking cancel and refund has been processed.

Alternative flow: Booking cannot be canceled as rent date has been started.

2.3 Vehicle log

2.3.1 Add vehicle

User: Admin

Input: Enter vehicle details and condition.

Output: Vehicle successfully added and vehicle is listed.

Alternative flow: Vehicle is already added and can only be updated.

2.3.2 Update vehicle

User: Admin

Input: Enter updated details of vehicle.

Output: Successfully updated

Alternative flow: Vehicle cannot be updated due to technical error.

2.3.3 Remove vehicle

User: Admin

Input: Enter reason for removal.

Output: Successfully removed and confirmation message is sent.

Alternative flow: Vehicle cannot be removed due to ongoing rental.

3. Other Nonfunctional Requirements

3.1 Performance

The performance of a website depends on the software's speed of response, throughput, execution time and storage capacity. The service levels comprising performance requirements are often based on supporting end-user tasks. As according to the future need the performance of our system can also be increased.

3.2 Availability

The system should always be available for access at 24 hours, 7 days a week. Also in the occurrence of any major system malfunctioning, the system should be available in 1 to 2 working days, so that the business process is not severely affected.

3.3 Security

Users will have double password authentication for individual security reasons.

3.4 Usability

The system provides a help and support menu in all interfaces for the user to interact with the system. The user can use the system by reading help and support.

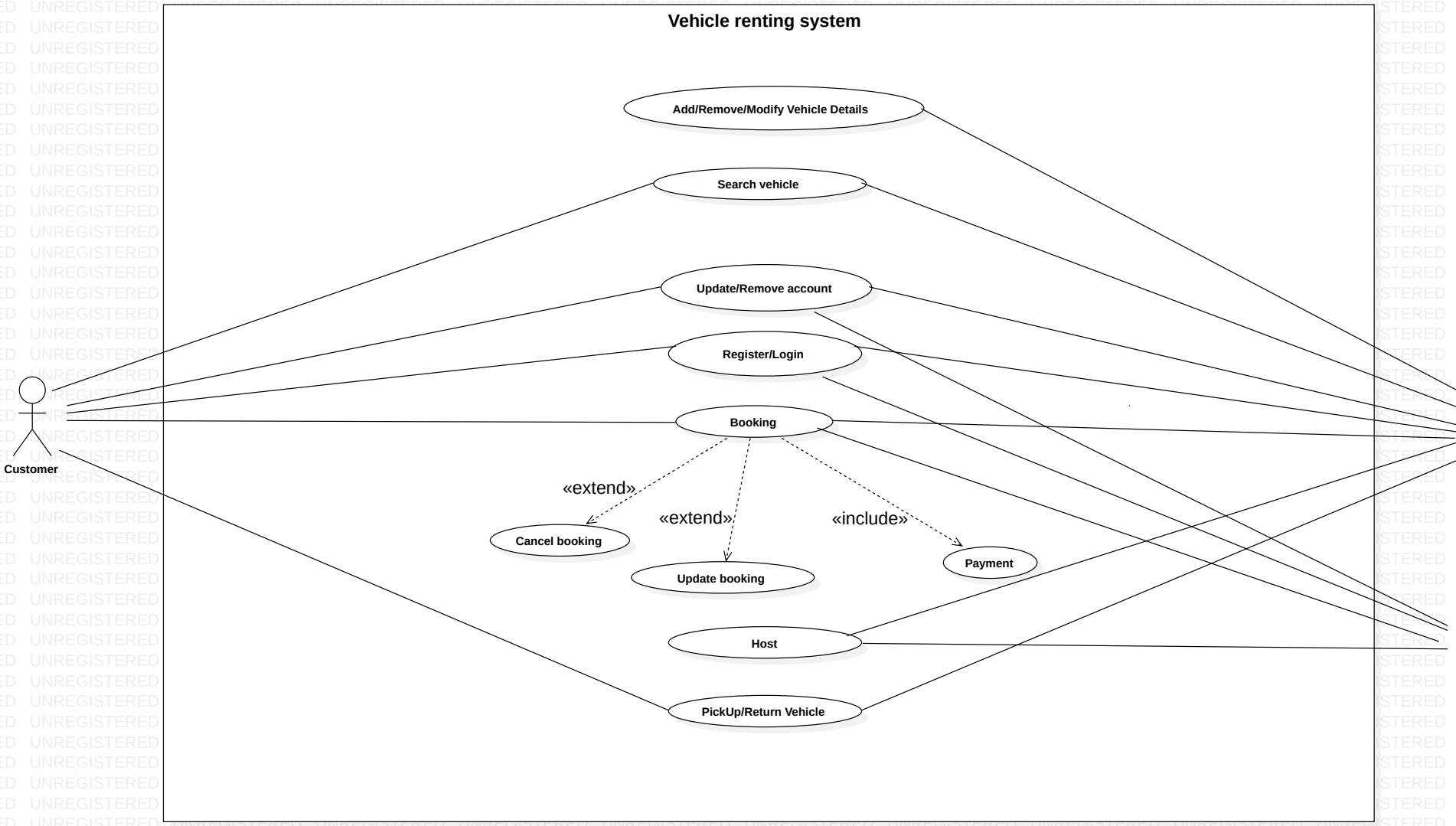
3.5 Modifiability

As according to the future requirement our website can be modified. We all know that with the passage of time the need for various features increases so modification is allowed in this website.

3.6 Ease of use

Considering the level of knowledge possessed by the users of this system, a simple but quality user interface should be developed to make it easy to understand and require less training.

Model1::UseCaseDiagram1



Use Case Description

Use case 1: Register/Login
Actors: Customer, vehicle owner, admin.
Main flow:

- Visit the registration page of the system.
- Provide their personal details such as name, email address, contact number, and password.
- Agree to the terms and conditions of the system.
- Submit the registration form.
- Visit the login page of the system.
- Enter login credentials.
- Clicks on the login button.

Pre condition:

- Access to the internet
- Valid/authenticate email address and phone number.
- Acceptance of terms and conditions

Post condition:

- Account creation
- Authentication
- Registration/login successfully

Alternative flow:

-

Use case 2: Update/Remove account
Actors: Customer, vehicle owner, admin.
Main flow:

- Update account by logging in.
- And navigate to settings.
- Remove account by deleting it.

Pre condition:

- Updated account information must be valid
- System must be operation and able to handle update/remove request
- Account should exist for removing account

Post condition:

- Should receive confirmation that the account was successfully updated/removed.

Alternative flow:

- Invalid information, the system could display an error message.

Use case 4: Booking
Actors: Customer, vehicle owner, admin.
Main flow:

- Searching for available vehicles
- Selecting a desired vehicle
- Booking it by providing personal and payment details
- Confirming the booking and picking up the vehicle
- Receiving a confirmation message.

Pre condition:

- User has internet access and a valid driver's licence.
- Vehicles listed should be available.

Post condition:

- User successfully books a vehicle.
- The rental company reserves the selected vehicle.

Alternative flow:

- System prompts users to try again in case of errors, and provides alternative vehicle options if the selected vehicle is unavailable.
- Allows users to make changes to their booking.

Use case 5: Search Vehicle
Actor: Customer
Main flow:

- Customer navigates to the "Search Vehicle" section in the system.
- Displays a search form where the customer can input search criteria such as location, type of vehicle, and rental period.
- Enters search criteria and submits the form.
- The system retrieves matching vehicles from the inventory database and displays them to the customer.
- The customer selects a vehicle.

- The system confirms the reservation and sends a confirmation message to the customer.

Pre condition:

- The customer is logged in to the vehicle renting system.

Post condition:

- The customer has successfully reserved a vehicle from the available inventory.

Alternative flow:

- If no vehicles match the customer's search criteria, the system displays a message informing the customer that no vehicles are available and prompts them to refine their search.

Use case 6: Add/Update/Remove vehicle log
Actor: vehicle owner
Main flow:

- The Vehicle Service owner navigates to the "Vehicle Log" section in the vehicle renting system.
- The system displays a list of vehicles and their current status.
- Owner selects the vehicle that needs to be updated/removed/added.
- Displays the details of the selected vehicle, including its current status.
- Selects the option to add/update/remove the vehicle's status.
- The system prompts the Vehicle Service owner to select a new status from a list of available options.
- The system validates the input and updates the vehicle's status in the database.
- The system displays a confirmation message to the Vehicle Service owner.

Pre condition:

- The Vehicle Service owner is logged in to the vehicle renting system.

Post condition:

- The vehicle's status has been successfully added/updated/removed from the system.

Alternative flow:

- If the Vehicle Service owner cancels the operation, the system returns to the list of vehicles without adding/updating/removing the status.

Use case 7: Cancel Booking
Actor: Customer
Main flow:
- User requests to cancel a booking
- System verifies the booking details
- System confirms cancellation and updates the booking status
- System processes any applicable refunds

Pre condition:
- User must have an active booking

Post condition:
- User's booking is cancelled and any applicable refunds are processed

Alternative flow:
- If the user cancels the booking within the cancellation window, the system does not process any refunds.

Use case 8: Update booking
Actor: Customer
Main flow:
- User requests to update their booking details
- System verifies the availability of the vehicle and any additional charges
- System confirms the updated booking details

Pre condition:
- User must have an active booking

Post condition:
- User's booking is updated with the new details

Alternative flow:
- If the requested update conflicts with another reservation, the system will suggest alternative times or vehicles.

Use case 9: Pickup/Return vehicle
Actor: Customer, vehicle owner.
Main flow:

- User arrives to pick up/return the vehicle
- System verifies the user's identity and payment information
- System hands over the keys to the vehicle owner/customer.

Pre condition:

- User must have an active booking
- User must have provided payment information

Post condition:

- User receives the keys to the vehicle and can begin using it

Alternative flow:

- If the user's payment information is invalid, the system will prompt the user to update their payment information before handing over the keys.

Use case 10: Payment
Actors:Customers, Vehicle owner
Main flow:

- Clients can upload a proof of payment (receipt, deposit slip, etc).
- Vehicle Owner can verify the payment done by the client.

Pre condition:

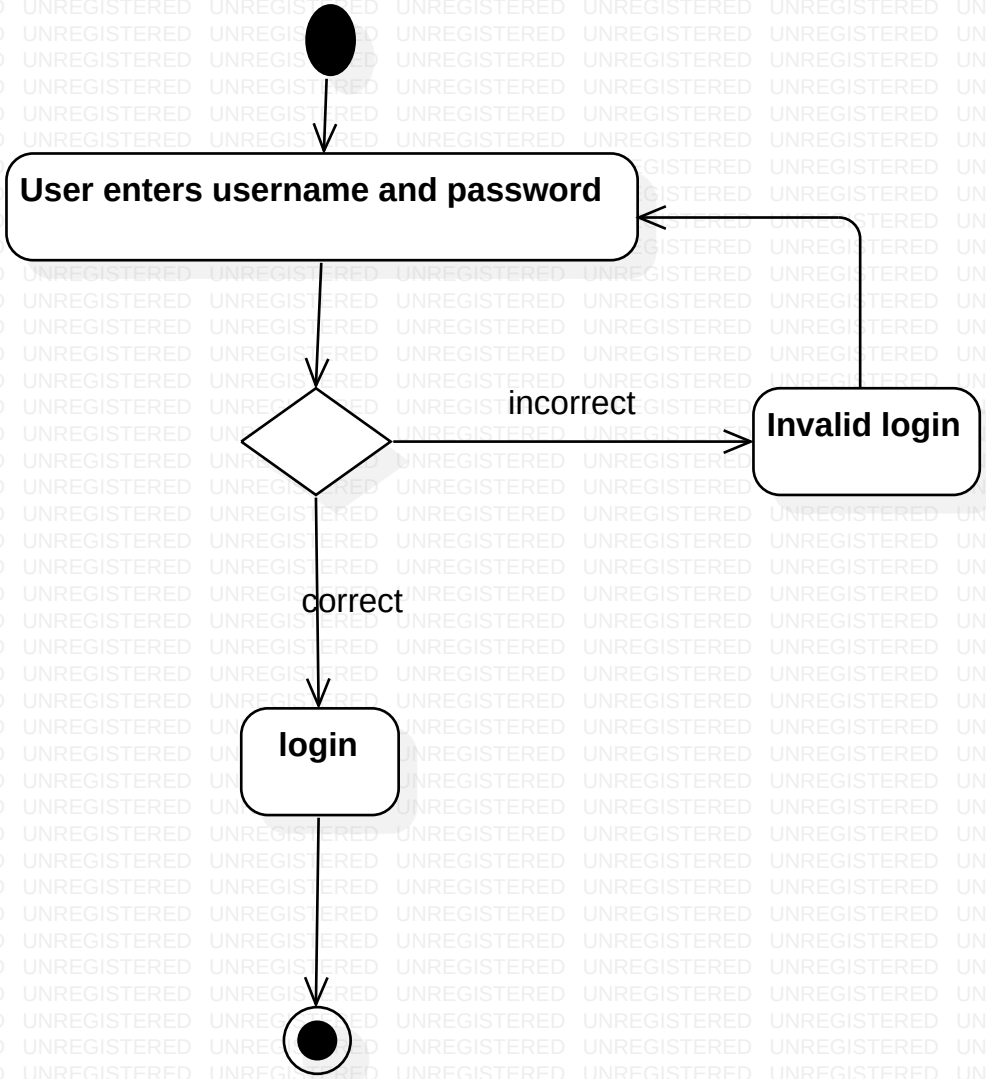
- Customers will need to login first in order to access the feature.
- The Vehicle Owner will need to verify also to manage the payment transactions.

Post condition:

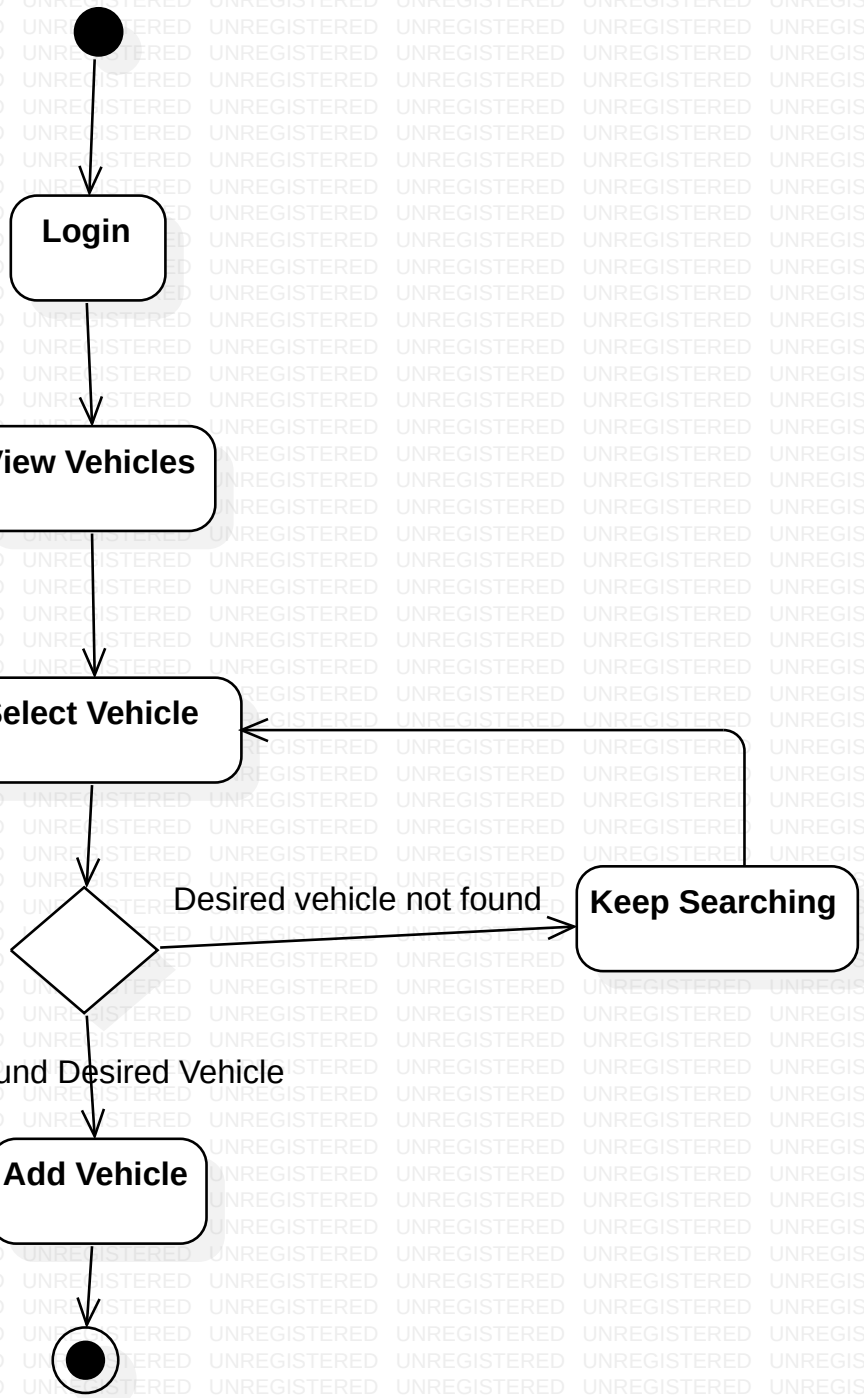
- Updated list of verified client payments**.**

Alternative flow: None

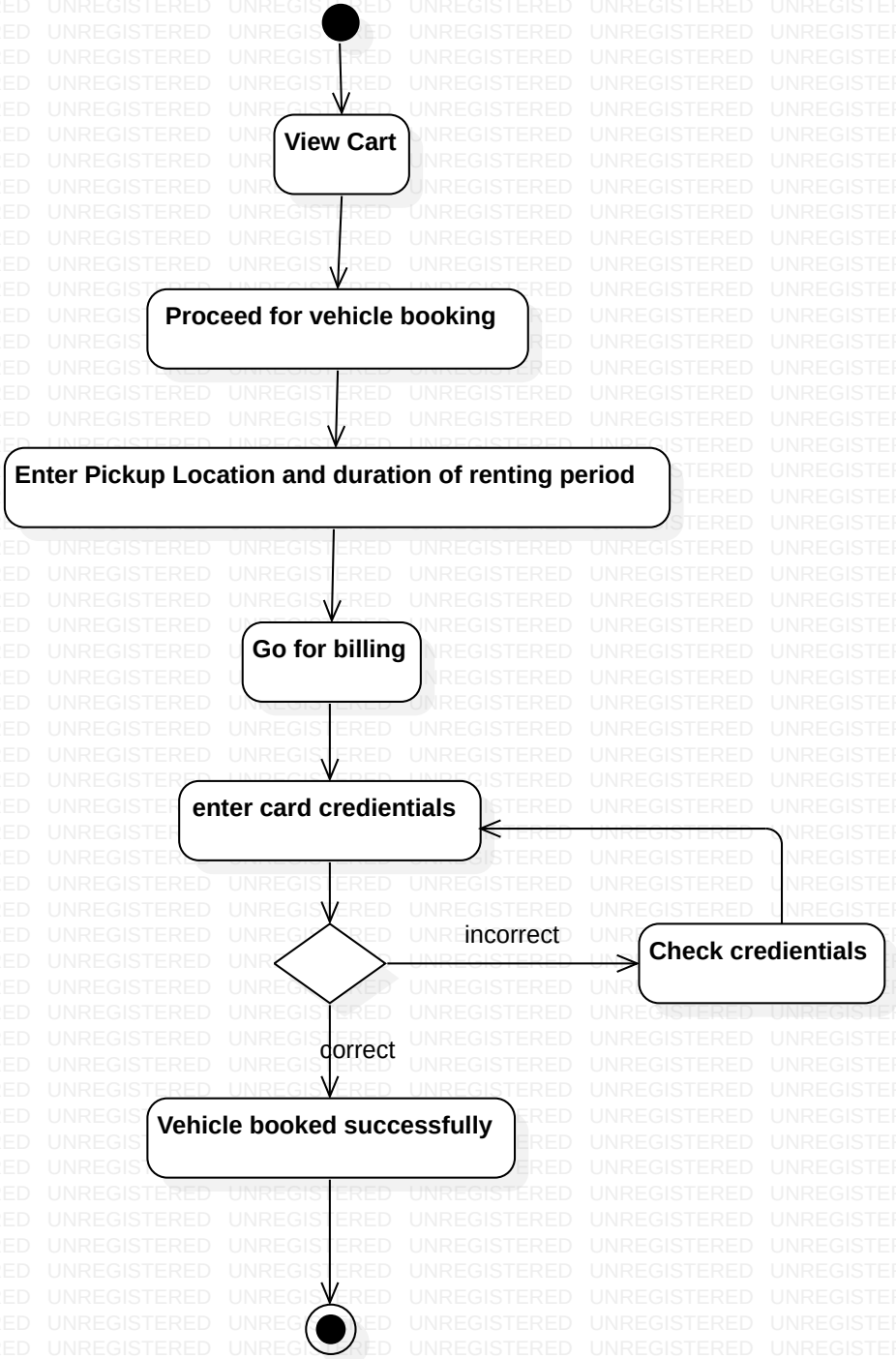
Activity1::ActivityDiagram - Login



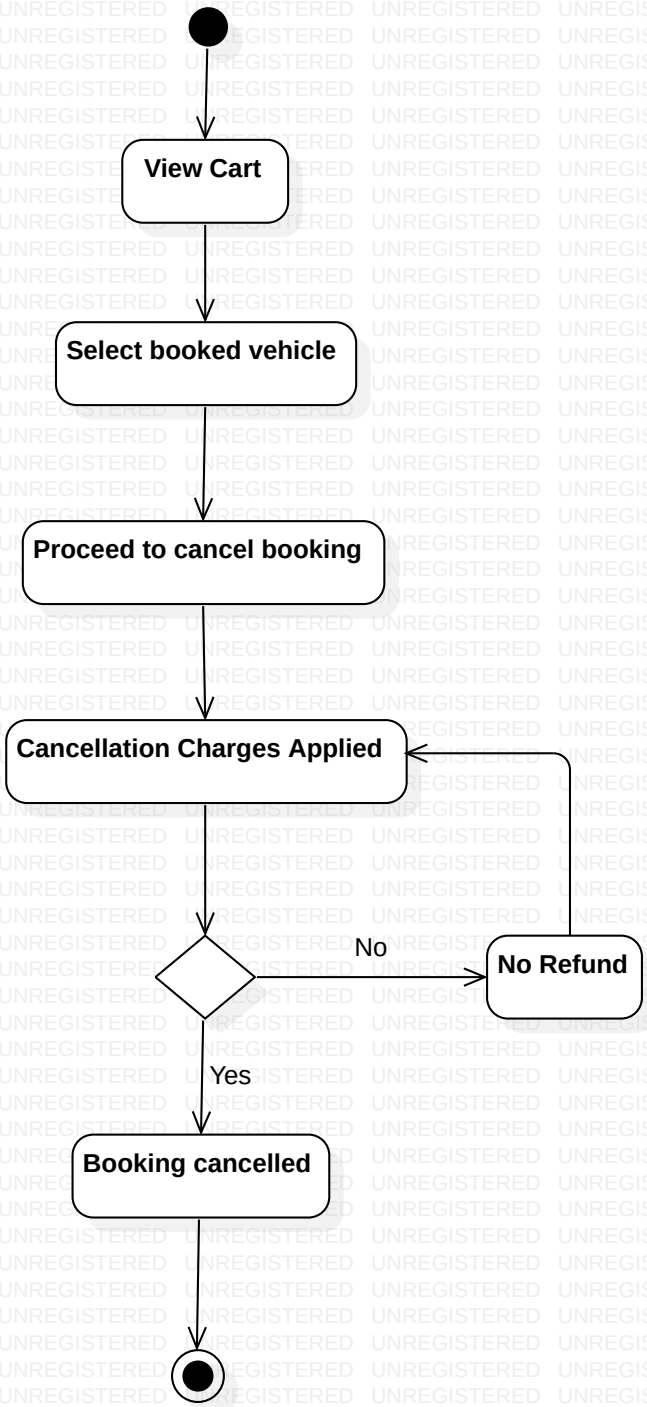
Activity1::ActivityDiagram - Add Vehicle



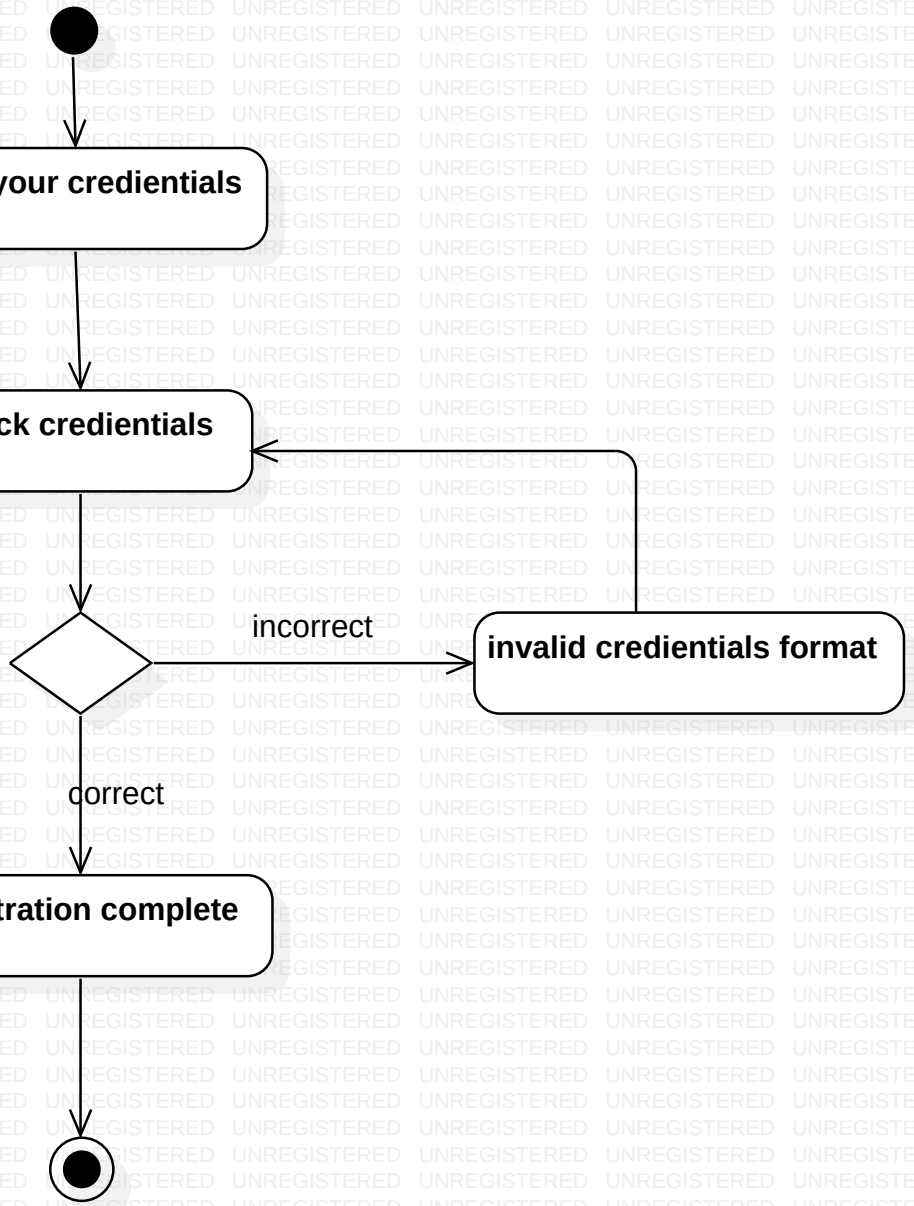
Activity1::ActivityDiagram - Booking



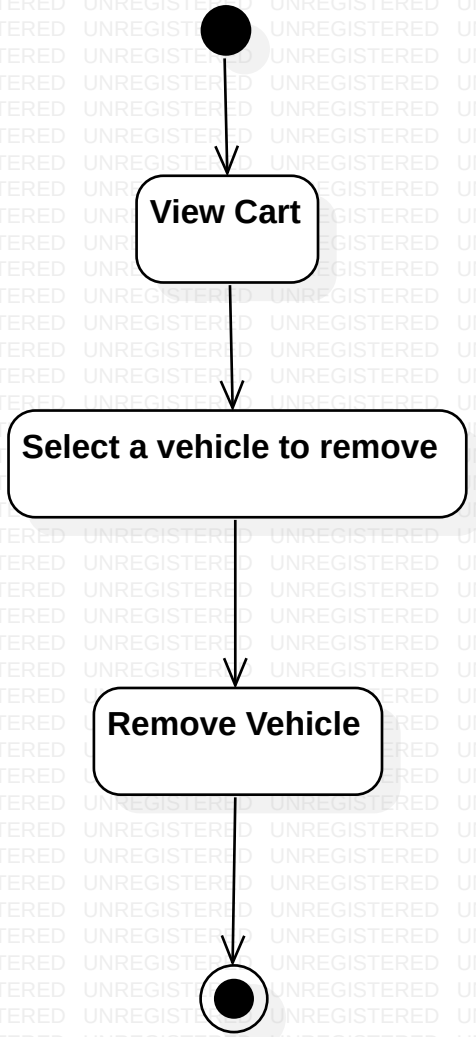
Activity1::ActivityDiagram - Cancel Booking



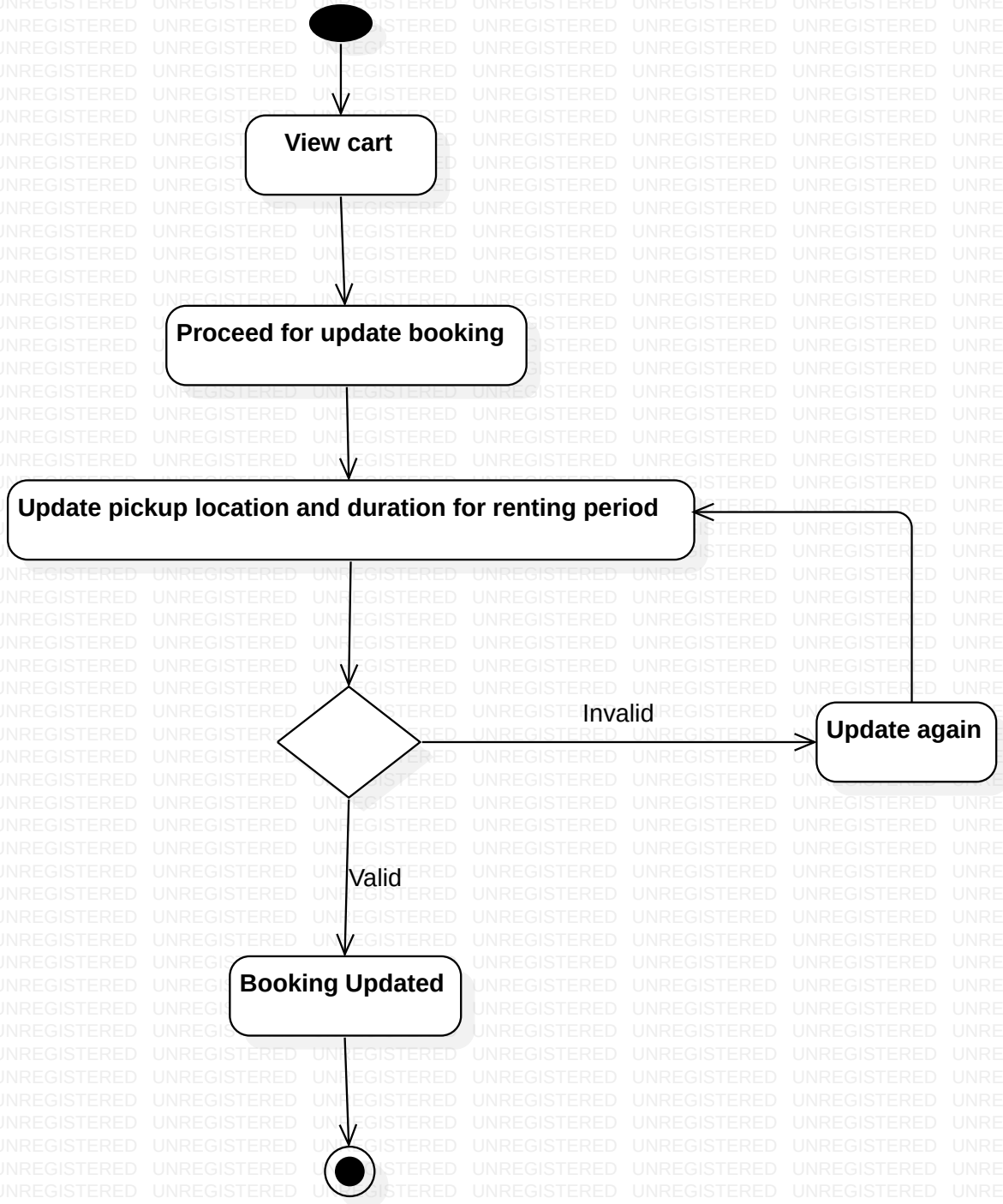
Activity1::ActivityDiagram - Registration



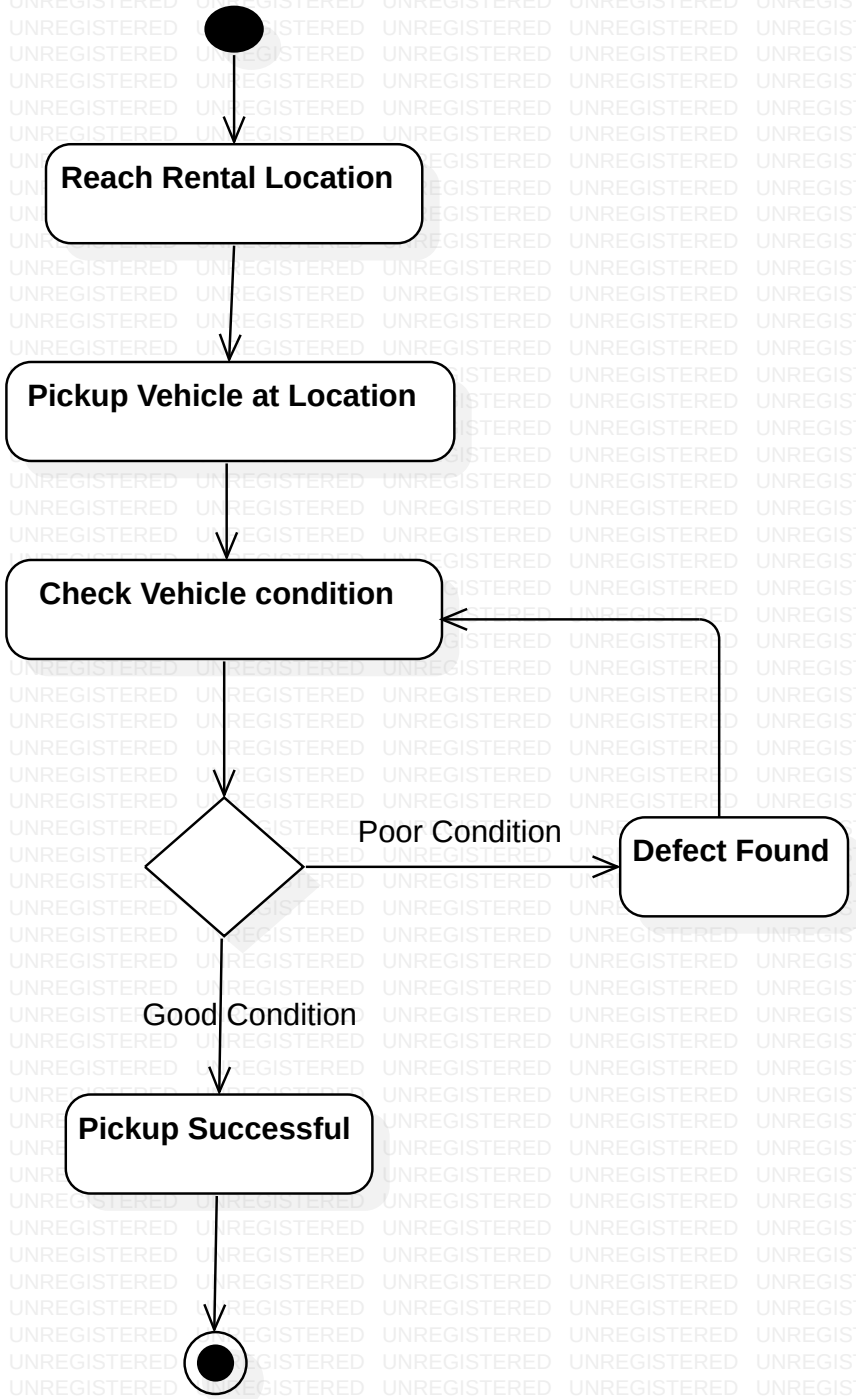
Activity1::ActivityDiagram - Remove Vehicle



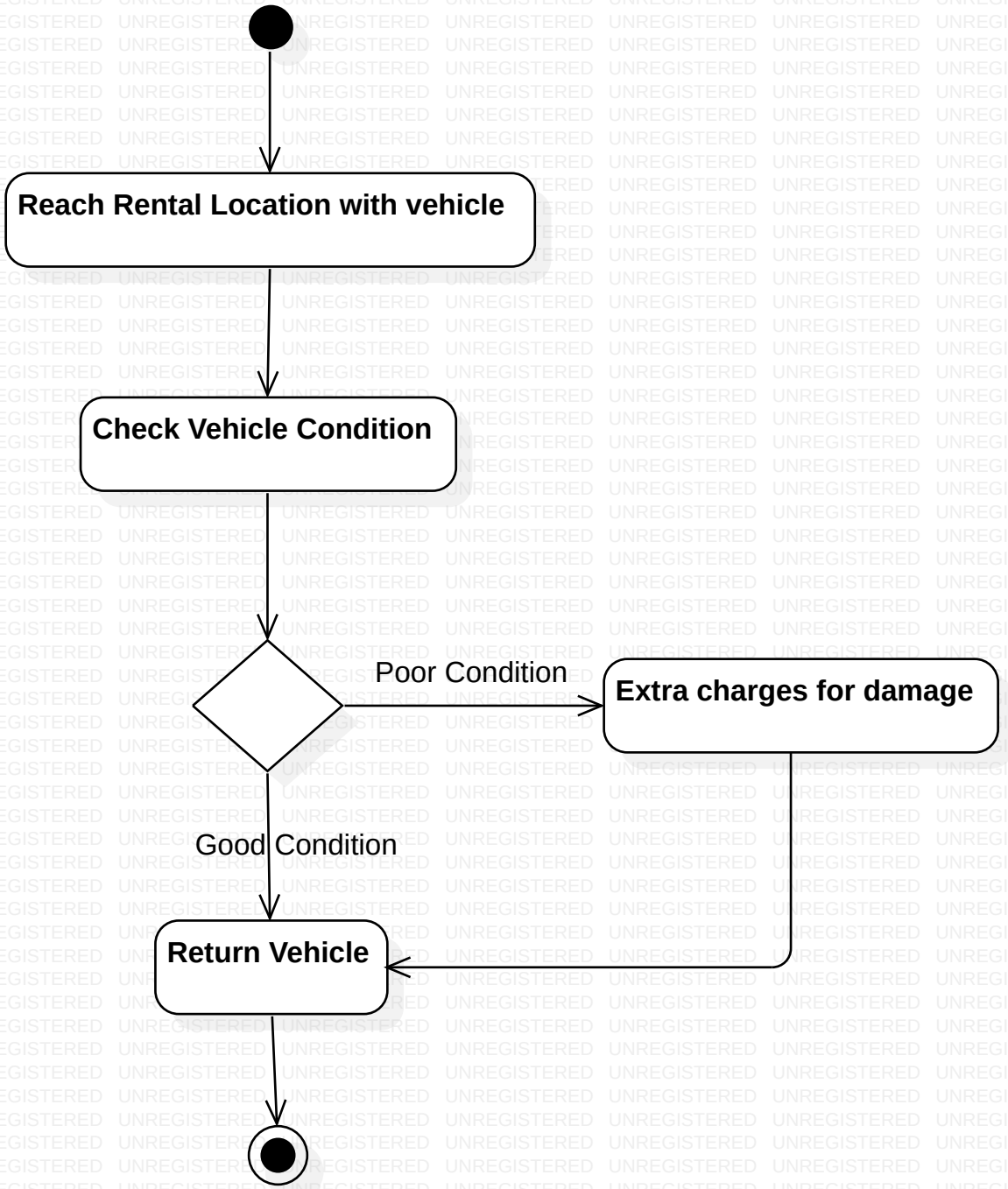
Activity1::ActivityDiagram - Update booking



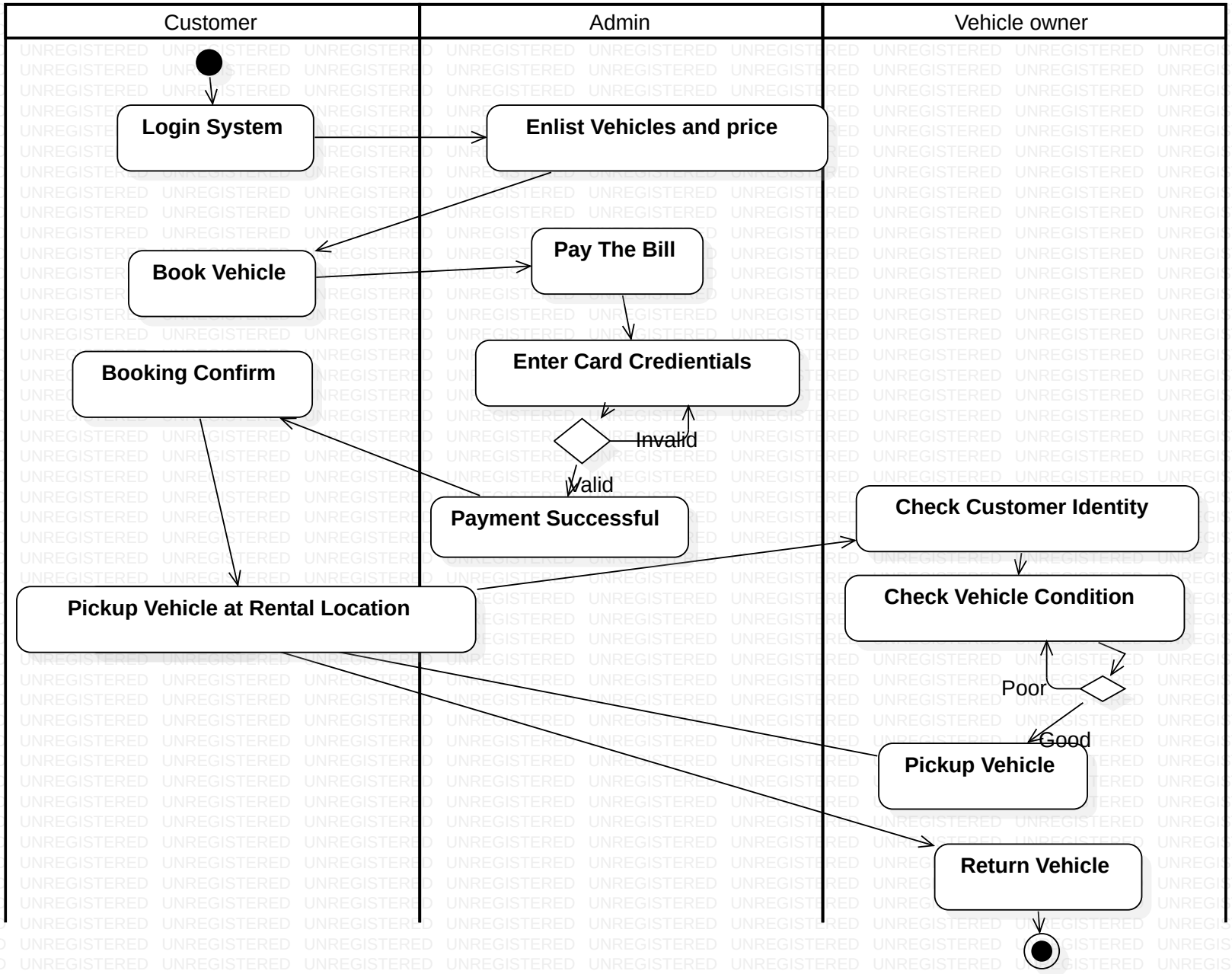
Activity1::ActivityDiagram - Pickup Vehicle



Activity1::ActivityDiagram - Return Vehicle



Activity Diagram With Swimlanes



CRC Cards

1. Identify classes of the system and design CRC cards.

Ans- The various classes are as follow -

1. Registration
2. Admin
3. Vehicle
4. User
5. Vehicle Owner
6. Login
7. Customer
8. Rental Record
9. Database
10. Booking
11. Cancel Booking
12. Payment
13. Cash
14. Online Mode

ClassName – Registration	
Responsibilities user login	**Collaboration** Server

ClassName – Admin	
Responsibilities add vehicle remove vehicle modify vehicle	**Collaboration** Booking Server

ClassName - Vehicle	
calculate bills check bank details	**Collaboration** Login Server

ClassName - User	
Subclass - Admin, Customer, Vehicle owner	
Responsibilities calculate bills check bank details	**Collaboration** Login Server

ClassName - Vehicle Owner	
Responsibilities new user modify user	**Collaboration** Login Server

ClassName - Login	
Responsibilities authorization check balance payment authorization	**Collaboration** Login Server

ClassName - Customer	
Responsibilities check vehicle availability check user pickup point check account validity	**Collaboration** server

ClassName - Rental Record	
Responsibilities cancel booking of vehicle	**Collaboration** Login Booking Server

ClassName - Database	
Responsibilities new user modify user	**Collaboration** Login Server

ClassName -Booking

Responsibilities	**Collaboration**
new user modify user	Login Server

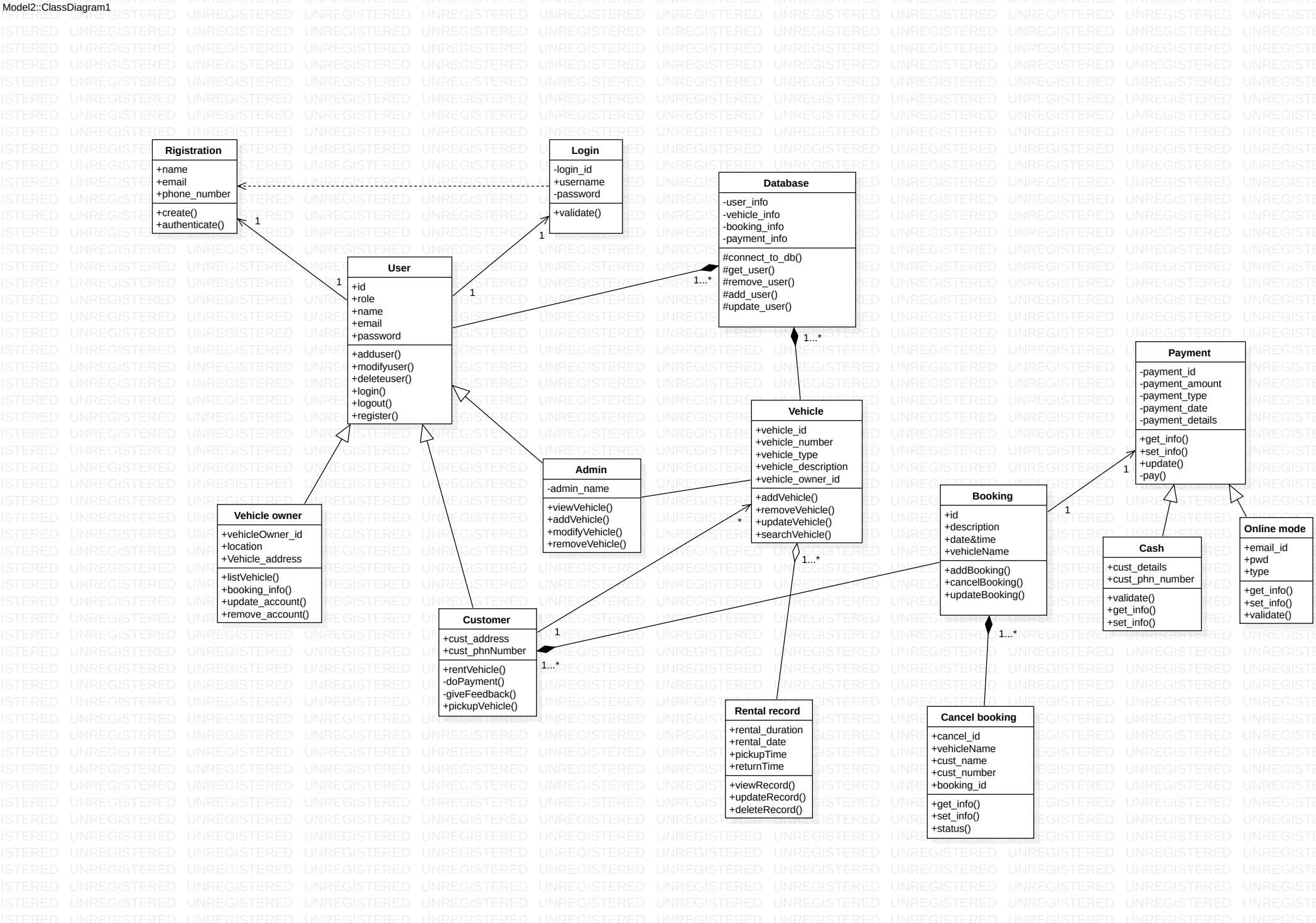
ClassName -Cancel Booking	
Responsibilities	**Collaboration**
new user modify user	Login Server

ClassName - Payment	
Subclass - Cash, Online mode	
Responsibilities	**Collaboration**
new user modify user	Login Server

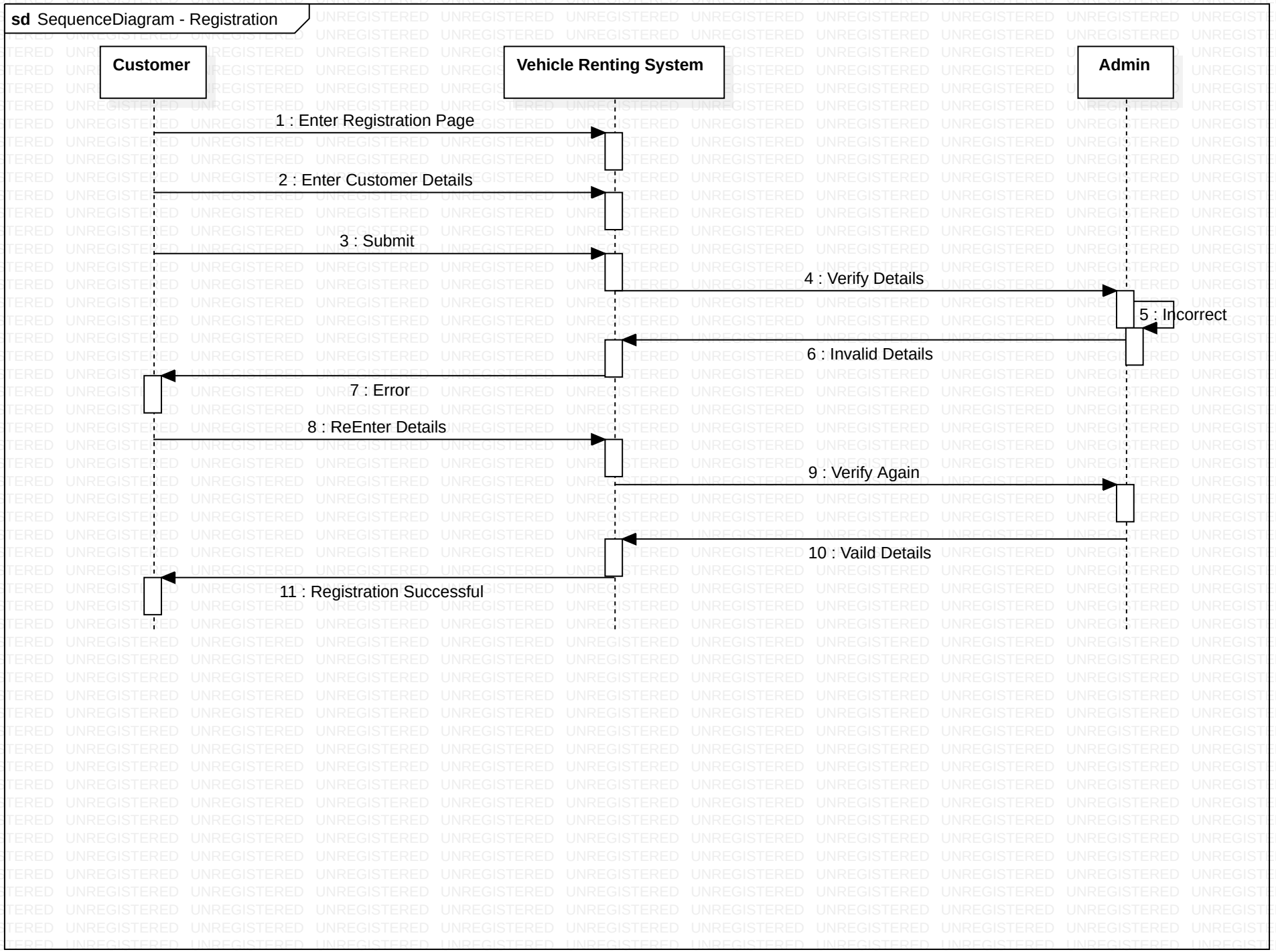
ClassName – Cash	
Responsibilities	**Collaboration**
new user modify user	Login Server

ClassName – Online Mode	
Responsibilities	**Collaboration**
new user modify user	Login Server

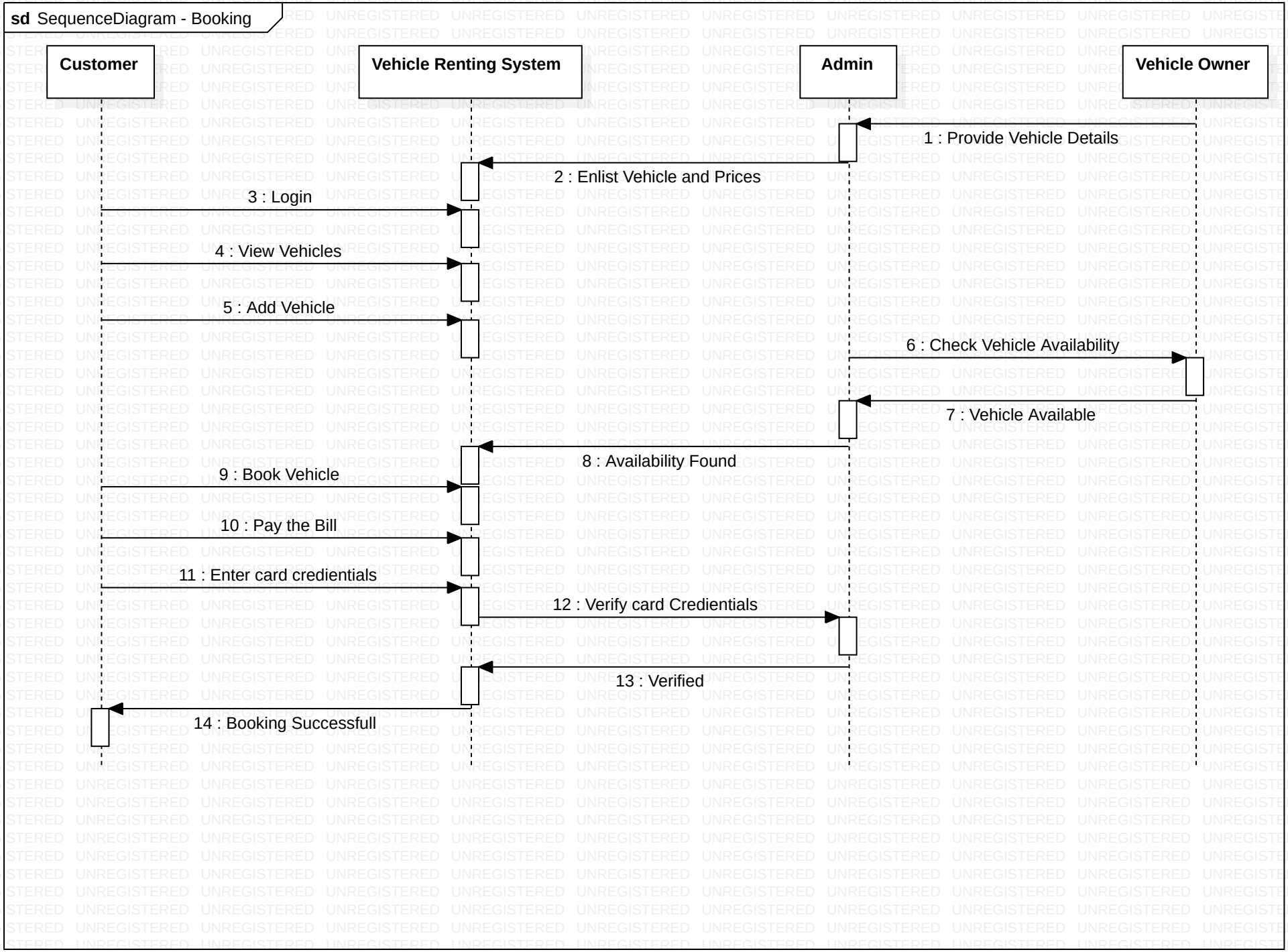
Model2::ClassDiagram1



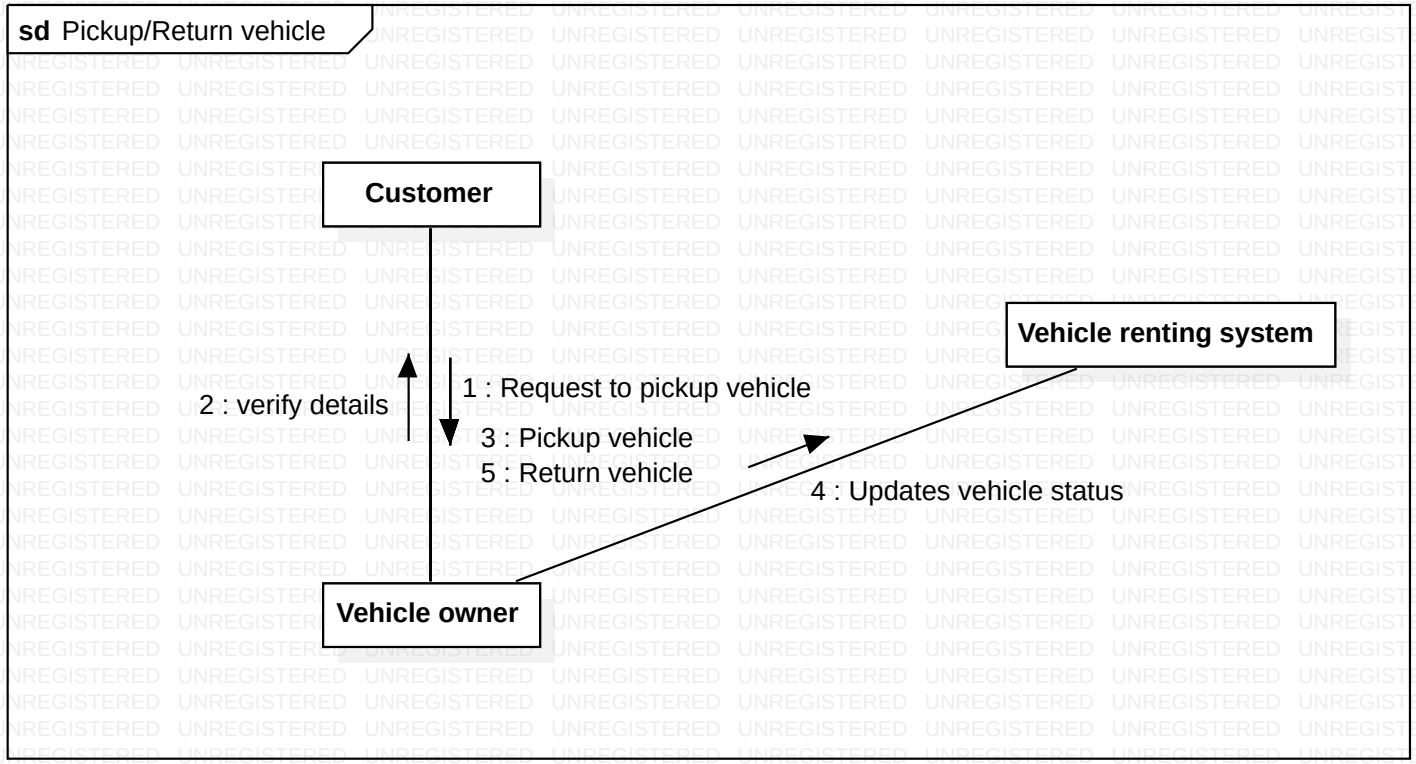
Collaboration1::Interaction1::SequenceDiagram - Registration



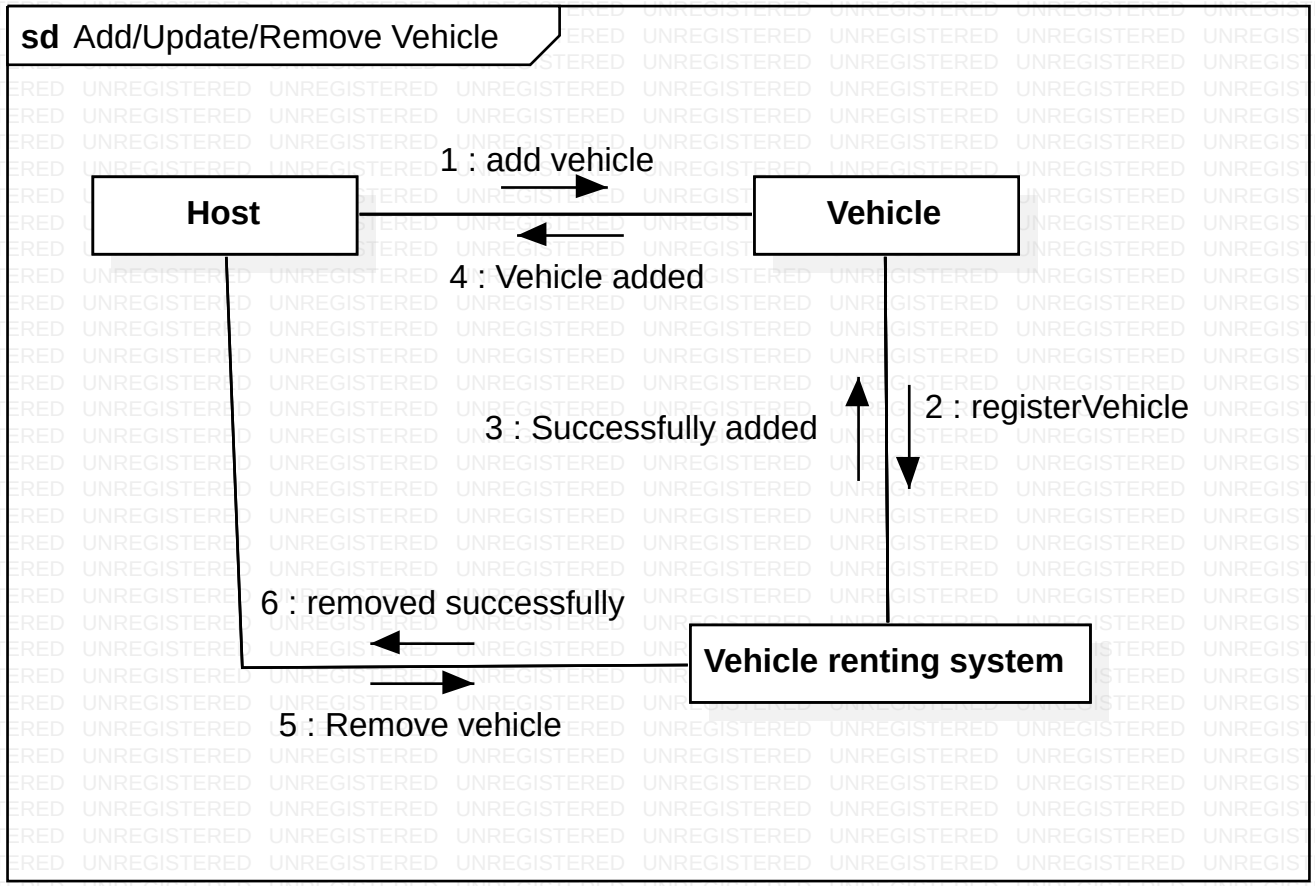
Collaboration2::Interaction1::SequenceDiagram - Booking



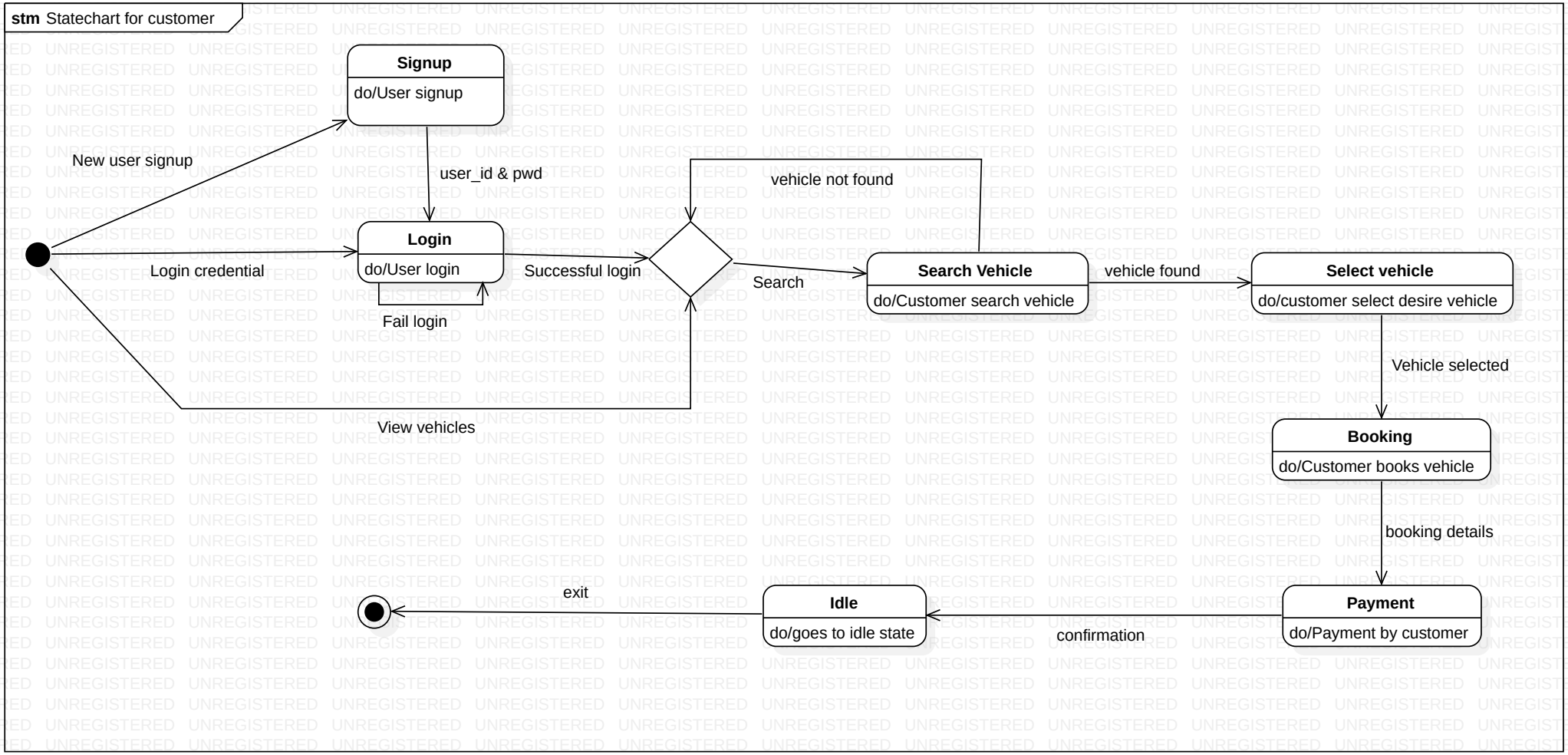
Collaboration2::Interaction1::Pickup/Return vehicle



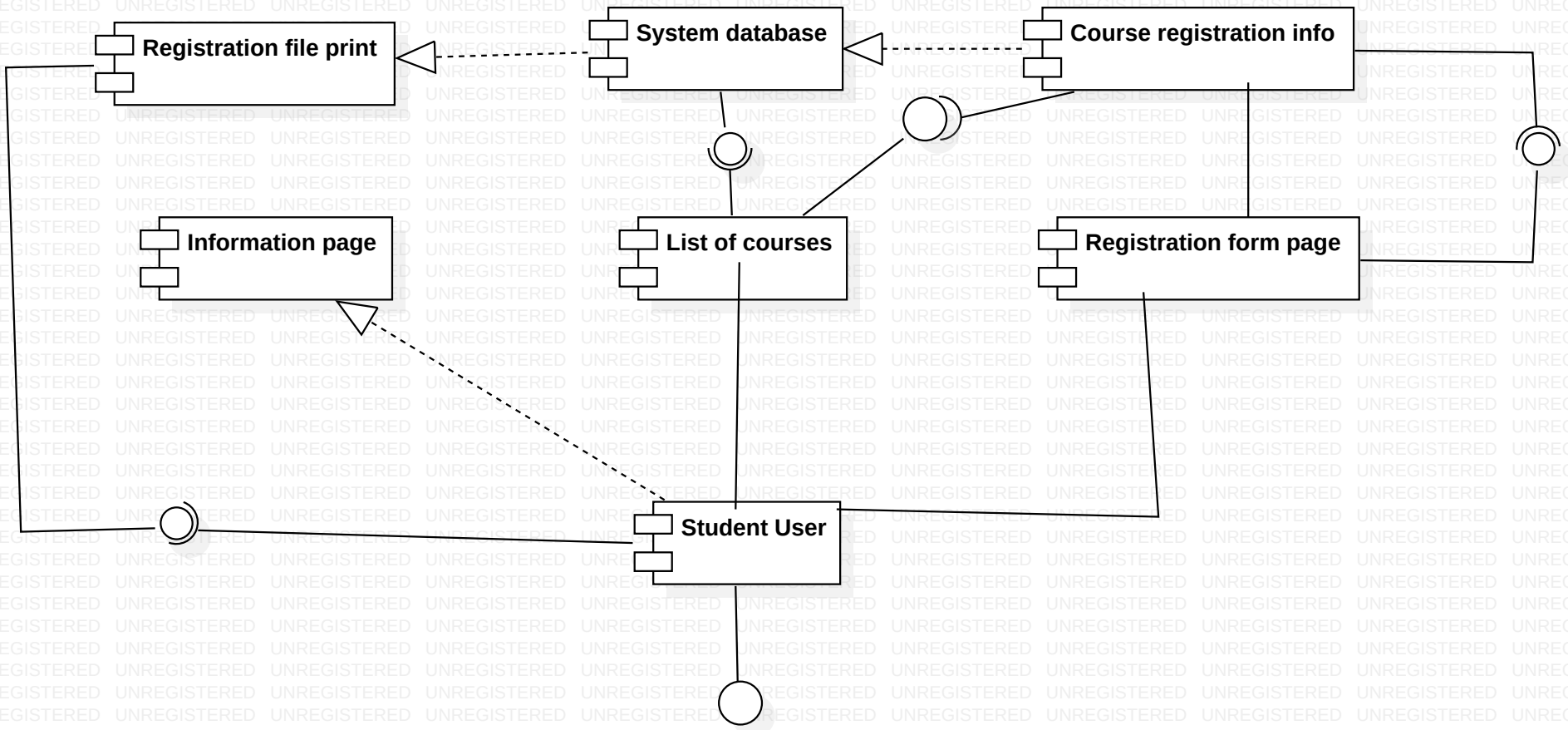
Collaboration1::Interaction1::Add/Update/Remove Vehicle



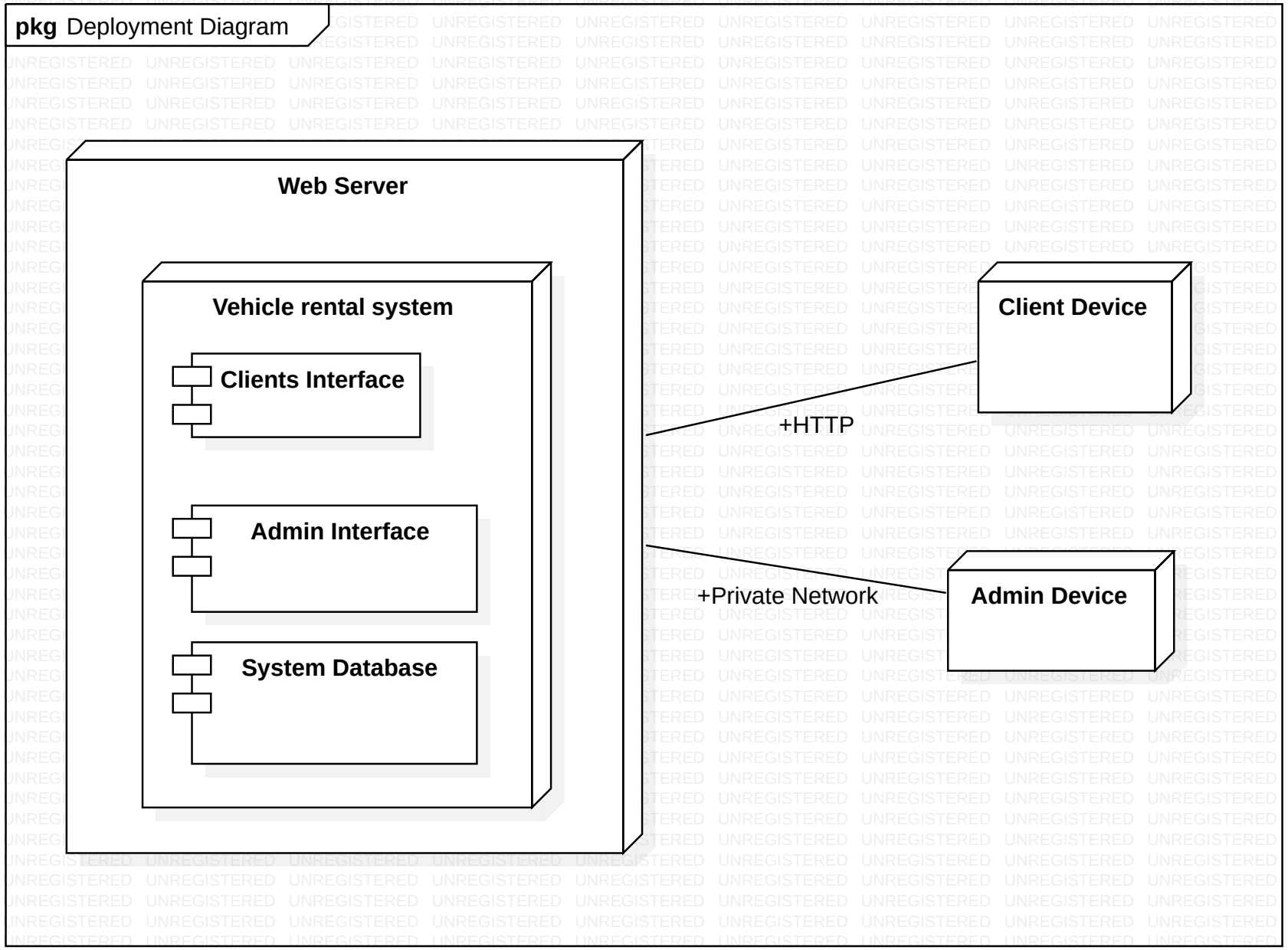
Statechart for customer::StatechartDiagram1



Model1::ComponentDiagram1



Model1::DeploymentDiagram1



Test Cases :

S. No.	Test Case Name	Input	Actual Output	Expected Output	Pass/Fail
1	Login (with correct credentials)	User enters correct email and password	User successfully logs in and is redirected to home page	User should be able to log successfully and is redirected to home page	Passed
2	Login (with incorrect credentials)	User enters correct email but incorrect password	Error message appears including that the passwordis incorrect	User should receive an error message indicating that the password is incorrect	Passed
3	Vehicle Selection (available vehicle)	User selectsa vehicle from the available list	Vehicle selected and user further proceed for booking.	User shouldbe able to proceed for booking of selected vehicle	Passed
4	Vehicle Selection (unavailable vehicle)	User selectsa vehicle from the unavailable vehicle list	Message appears saying selected vehicle not available	User should receive a message saying vehicle not available.	Passed
5	Booking (with valid details)	User enters valid details	Booking is further proceeded for confirmation	User shouldbe able to confirm booking	Passed
6	Booking (with invalid details)	User enters invalid details	Error message appears indicating that details are invalid	User should receive an error message saying invaliddetails	Passed

7	Payment (with correct credentials)	User enters correct card credentials	Payment successful and Booking confirmed	User should receive message of payment success and booking confirmed	Passed
8	Payment (with incorrect credentials)	User enters incorrect credentials	Message appears indicating Payment failed	User should Receive message indicating payment failed	Passed
9	Booking Cancellation (valid details)	User enters Valid bookingID	Booking is successfully cancelled	Booking should be cancelled and messageshould be displayed	Passed
10	Booking Cancellation (invalid details)	User enters invalid booking ID	Booking is not cancelled	Booking should not becancelled and user should not receive any message	Passed
11	Vehicle Listing (correct vehicle details)	Vehicle Owner entersvalid vehicle details	Vehicles listed successfully and status of vehicles visible	Vehicle owner shouldbe able to see status of the vehicles.	Passed
12	Vehicle Listing (incorrect vehicle details	Vehicle Owner entersInvalid vehicle details	Vehicles not listed	Vehicle Owner should not beable to see vehicle status	Passed
13	Pickup Vehicle(Good Vehicle Condition)	User checks vehicle and find its in a good condition	User pickup the vehicle and vehicle owner confirms pickup	User shouldbe able to pickup vehicle and use it	Passed

14	Pickup Vehicle(Poor Vehicle Condition)	User checks vehicle and find its in a poor condition	User complains vehicle ownerfor poor vehicle condition and pickup is unsuccessful	User shouldbe able to complain forvehicle not accepting pickup	Passed
15	Return Vehicle (Good Vehicle Condition)	Vehicle owner checksvehicle and find its in a good condition	Vehicle successfully returned and confirmed by vehicle owner	Vehicle owner should be able to confirm that vehicle is properly returned	Passed
16	Return Vehicle(Poor Vehicle Condition)	Vehicle owner checksvehicle and find its in a poor condition	Vehicle owner asks for damage cost from user and return of vehicle failed	Vehicle Owner should be able to askfor damagecost	Passed