**Software Requirements and Design Document**

**for**

<Motorverse>

**By:**

**Ahmed Ali 22i-0825**

**Ahmad 22i-1288**

**Qusai Mansoor 22i-0935**

**Table of Contents**

**[Introduction 1](#_heading=h.30j0zll)**

[1.1 Purpose 1](#_heading=h.1fob9te)

**[1.2 Product Scope 1](#_heading=)**

**[1.3 Title 1](#_heading=)**

**[1.4 Objectives 2](#_heading=)**

**[1. Create a unified platform for automotive services. 2](#_heading=)**

**[2. Streamline vehicle transactions with transparency and efficiency. 2](#_heading=)**

**[3. Offer a diverse selection of new and certified used cars. 2](#_heading=)**

**[4. Provide a comprehensive range of auto parts and accessories. 2](#_heading=)**

**[5. Deliver flexible, transparent financing and installment options. 2](#_heading=)**

**[6. Integrate tailored insurance solutions at the point of sale. 2](#_heading=)**

**[7. Ensure exceptional customer service and post-sale support. 2](#_heading=)**

**[1.5 Problem Statement 2](#_heading=)**

**[Overall Description 2](#_heading=)**

[1.6 Product Perspective 2](#_heading=h.4d34og8)

**[1.7 Product Functions 3](#_heading=)**

[1.8 List of Use Cases 4](#_heading=h.3rdcrjn)

[1.9 Extended Use Cases 4](#_heading=h.26in1rg)

[1.10 Use Case Diagram 30](#_heading=h.lnxbz9)

**[Other Nonfunctional Requirements 30](#_heading=)**

[1.11 Performance Requirements 30](#_heading=h.35nkun2)

**[1.12 Safety Requirements 30](#_heading=)**

**[1.13 Security Requirements 31](#_heading=)**

**[1.14 Software Quality Attributes 31](#_heading=)**

**[1.15 Business Rules 31](#_heading=)**

**[1.16 Operating Environment 32](#_heading=)**

**[1.17 User Interfaces 33](#_heading=)**

**[Domain Model 34](#_heading=h.2xcytpi)**

**[System Sequence Diagram 34](#_heading=)**

**[Sequence Diagram 35](#_heading=h.1ci93xb)**

**[Class Diagram 41](#_heading=)**

**[Component Diagram 42](#_heading=)**

**[Package Diagram 43](#_heading=)**

**[Deployment Diagram 44](#_heading=)**

# **Introduction**

## **Purpose**

Motorverse is an all-in-one automotive platform, offering new and certified used car sales, rentals, parts, insurance, and financing. We streamline the entire process, providing a seamless, transparent, and customer focused experience, ensuring convenience and trust in every transaction.

## **Product Scope**

Motorverse focuses on;

* New and used car sales
* Car rentals

Wide range of supplementary services like;

* Auto parts
* Insurance

Unlike typical car dealerships or rental services, Motorverse will integrate these services under one platform, making it easier for customers to explore multiple options simultaneously.

## **Title**

**Motorverse**

**Revolutionizing the Automotive Marketplace**

## **Objectives**

## Create a unified platform for automotive services.

## Streamline vehicle transactions with transparency and efficiency.

## Offer a diverse selection of new and certified used cars.

## Provide a comprehensive range of auto parts and accessories.

## Deliver flexible, transparent financing and installment options.

## Integrate tailored insurance solutions at the point of sale.

## Ensure exceptional customer service and post-sale support.

## **Problem Statement**

The automotive industry in Pakistan is highly fragmented, with separate entities providing sales, rentals, parts, insurance, and installment services. This creates significant challenges for customers looking to manage all their automotive needs efficiently.

**The current process often involves multiple interactions with different**

**service providers, each of whom operates independently. This fragmented**

**approach increases the time, effort, and potential confusion for consumers.**

**Solution :**

Motorverse addresses this problem by creating an all-in-one platform that integrates the various aspects of automotive services. By bringing **car sales, rentals, autoparts, and financial services like insurance** options under one roof, Motorverse will enhance the customer experience. The platform will also offer transparent pricing, detailed car listings, and easy comparison tools, reducing the complexity of automotive transactions.

# **Overall Description**

## **Product Perspective**

The **Motorverse** platform is a **new, self-contained product** designed to integrate various automotive services under one platform. It addresses inefficiencies in the current automotive market in Pakistan, where services such as car sales, rentals, insurance, and parts are fragmented, requiring customers to interact with multiple providers.

**Context and Origin:**

* The product aims to unify these services to provide convenience and transparency.
* Motorverse is **not a follow-on product** but a **comprehensive solution** tailored to the challenges of the local automotive industry.
* It serves as a standalone system but incorporates partnerships with manufacturers, insurers, and financial institutions to expand its functionality.

**Larger System Relationships:**

* The platform relies on APIs and partnerships with third-party providers for insurance, financing, and auto parts.
* It also facilitates interactions between individual buyers, sellers, and renters, creating a shared ecosystem.

**Diagram Summary:** The system can be visualized as a centralized platform where:

Buyers, renters, and sellers interact with a core system.

External services (insurance, financing, and parts manufacturers) are integrated via API connections.

## **Product Functions**

The **Motorverse platform** is designed to perform the following **major functions**:

1. **Car Sales:**
   * Offer both new and certified used cars with clear pricing and quality assurance.
   * Provide comparison tools for new and pre-owned options.
2. **Car Rentals:**
   * Allow users to browse, and book vehicles for short- or long-term rentals.
   * Enable test drives of cars available for purchase.
3. **Auto Parts:**
   * Present a wide catalog of autoparts.
   * Simplify the shopping process by centralizing sourcing and offering tailored recommendations.
4. **Insurance Services:**
   * Provide instant insurance options with customizable policies.
   * Offer bundled discounts for combined services.
5. **Customer Support:**
   * Deliver exceptional service through post-sale and rental support.
   * Maintain transparency in all transactions to build trust.

These functions align to create an integrated ecosystem for automotive transactions, aiming to revolutionize the user experience in the automotive sector.

## **List of Use Cases**

## **Extended Use Cases**

**Ahmad:**

**Use Case 1: Admin Dashboard Management**

**Name:** Admin Dashboard Management  
**Primary Actor:** Admin  
**Scope:** Motorverse System  
**Level:** User goal

**Stakeholders and Interests:**

· Admin: Wants to manage users, car listings, and transactions efficiently.

· System: Should provide a user-friendly interface for managing key aspects of the platform, ensuring smooth operations.

**Preconditions:**

· Admin must be logged into the system with appropriate permissions.

· Admin dashboard must be functional and accessible.

**Success Guarantee (Postconditions):**

· Admin can successfully view, modify, or delete users, listings, and transactions.

· The system updates and reflects changes in real-time.

**Main Success Scenario (Basic Flow):**

|  |  |
| --- | --- |
| **Actions and Intentions** | **System Responsibility** |
| 1. Admin logs into the system. | 2. System authenticates the admin and grants access to the dashboard. |
| 3. Admin navigates to the dashboard. | 4. System displays the dashboard with all management options (users, listings, transactions). |
| 5. Admin selects the desired management function (e.g., user management). | 6. System presents the corresponding management interface for the selected function. |
| 7. Admin makes the necessary changes (e.g., updates user status, modifies listings). | 8. System saves the changes and logs the actions taken by the admin. |

**Extensions (Alternate Flows):**

**· 4a. Invalid Login:**

1. If admin credentials are incorrect, the system notifies the admin and prompts them to re-enter the credentials.

**· 7a. Error when updating user details:**

1. If the system encounters an error, it will display an error message and log the issue for technical support.

**Special Requirements:**

· The system should support bulk actions (e.g., updating or deleting multiple listings/users at once).

**Frequency of Occurrence:** Daily, whenever the admin needs to manage system components.

**-------------------------------------------------------------------------------------------------------------------------------**

**Use Case 2: Progress Report**

**Name:** Generate Progress Report  
**Primary Actor:** Admin  
**Scope:** Motorverse System  
**Level:** User goal

**Stakeholders and Interests:**

· Admin: Wants to generate reports about business outcomes to analyze performance and make strategic decisions.

· Management: Expects accurate, up-to-date reports on transactions, sales, and rentals.

**Preconditions:**

· Admin must have reporting access rights.

· The system must have enough data to generate a meaningful report.

**Success Guarantee (Postconditions):**

· A detailed report is generated, summarizing sales, rentals, and financial performance.

· Report is available for download or print.

**Main Success Scenario (Basic Flow):**

|  |  |
| --- | --- |
| **Actions and Intentions** | **System Responsibility** |
| 1. Admin navigates to the "Reports" section on the dashboard. | 2. System displays available reporting options (sales, rentals, etc.). |
| 3. Admin selects the type of report they wish to generate. | 4. System generates a progress report based on the selected criteria. |
| 5. Admin views the report and downloads it for further review. | 6. System provides a downloadable PDF or CSV format. |

**Extensions (Alternate Flows):**

**· 3a. Report generation fails due to incomplete data:**

1. System displays an error message indicating missing or incomplete data.

**Special Requirements:**

· The system should support customizable report formats (e.g., daily, weekly, or monthly reports).

**Frequency of Occurrence:** Admin may generate reports monthly or quarterly.

**-------------------------------------------------------------------------------------------------------------------------------**

**Use Case 3: Register User**

**Name**: Register User  
**Primary Actor**: User  
**Scope**: Motorverse System  
**Level**: User goal

**Stakeholders and Interests**:

**· User**: Wants to create an account quickly to access all platform features.

**· System**: Needs to validate user credentials and ensure successful registration.

**Preconditions**:

· User must have access to the registration page and a valid e-mail address.

**Success Guarantee (Postconditions)**:

· User account is successfully created.

· User receives confirmation and can now access platform features.

**Main Success Scenario (Basic Flow)**:

|  |  |
| --- | --- |
| **Actions and Intentions** | **System Responsibility** |
| 1. User navigates to the registration page. | 2. System presents a registration form with required fields. |
| 3. User fills in the form with necessary details (e.g., email, password). | 4. System validates the data and checks for existing accounts. |
| 5. User submits the registration form. | 6. System creates the user account and sends a confirmation email. |

**Extensions (Alternate Flows):**

**· 3a. Invalid or missing input:**

1. System highlights the fields with errors and prompts the user to correct them.

**· 5a. User already exists:**

2. System informs the user that the account already exists and suggests logging in.

**Special Requirements:**

· System should support email verification as part of the registration process.

**Frequency of Occurrence:** Whenever a new user wants to join the platform.

**-------------------------------------------------------------------------------------------------------------------------------**

**Use Case 4: User Profile Management**

**Name:** User Profile Management  
**Primary Actor:** User  
**Scope:** Motorverse System  
**Level:** User goal

**Stakeholders and Interests:**

**· User:** Wants to manage their profile details easily, including updating information like address and payment methods.

**· System:** Ensures user data is kept accurate and secure.

**Preconditions:**

· User must be logged into the system to access profile management.

**Success Guarantee (Postconditions):**

· User’s profile is updated successfully, and changes are saved.

· System keeps user data secure and updated.

**Main Success Scenario (Basic Flow):**

|  |  |
| --- | --- |
| **Actions and Intentions** | **System Responsibility** |
| 1. User navigates to the profile section. | 2. System displays the current user profile details. |
| 3. User selects the option to edit their profile. | 4. System provides an editable form with current profile details. |
| 5. User updates their information (e.g., email, phone number, etc.). | 6. System validates the changes and updates the user’s profile. |

**Extensions (Alternate Flows):**

**· 6a. Invalid Input**:

1. If the updated information is invalid (e.g., email already in use), the system notifies the user and requests corrections.

**Frequency of Occurrence:** Whenever the user needs to update their profile information.

**-------------------------------------------------------------------------------------------------------------------------------**

**Use Case 5: Purchase Vehicle Insurance**

**Name:** Purchase Vehicle Insurance  
**Primary Actor:** User  
**Scope:** Motorverse System  
**Level:** User goal

**Stakeholders and Interests:**

**· User**: Wants to purchase vehicle insurance seamlessly and receive immediate confirmation.

**· Insurance** **Provider**: Wants to ensure users can easily purchase valid policies.

**· System**: Facilitates easy insurance purchases and ensures accurate record-keeping.

**Preconditions:**

· User must have a vehicle in their profile to insure.

· Insurance providers must be integrated into the platform.

**Success Guarantee (Postconditions):**

· Insurance is purchased successfully, and the user receives confirmation.

· The system updates the insurance information for the vehicle.

**Main Success Scenario (Basic Flow):**

|  |  |
| --- | --- |
| **Actions and Intentions** | **System Responsibility** |
| 1. User selects the vehicle they wish to insure. | 2. System displays available insurance options for that vehicle. |
| 3. User selects an insurance package and proceeds to payment. | 4. System processes the payment and confirms the insurance purchase. |
| 5. User receives a confirmation of the insurance purchase. | 6. System updates the vehicle’s insurance status in the user’s profile. |

**Extensions (Alternate Flows):**

**· 3a. Insurance provider unavailable:**

1. System informs the user that no insurance options are available and suggests trying later.

**Special Requirements:**

· The system must support integration with multiple insurance providers.

· The system must ensure secure payment processing.

**Frequency of Occurrence:** Whenever users purchase new insurance for their vehicles.

**-------------------------------------------------------------------------------------------------------------------------------**

**-------------------------------------------------------------------------------------------------------------------------------**

**Use Case 7: Contact Customer Support**

**Name:** Contact Customer Support  
**Primary Actor:** User  
**Scope:** Motoverse System  
**Level:** User Goal

**Stakeholders and Interests:**

· User: Wants quick and effective help with an issue.

· System: Ensures smooth user experience by providing timely support.

**Preconditions:**

· User must be logged in to access personalized customer support.

**Success Guarantee (Postconditions):**

· User successfully submits an issue or receives assistance.

· System tracks the request for future reference

**Main Success Scenario (Basic Flow)**:

|  |  |
| --- | --- |
| **Actions and Intentions** | **System Responsibility** |
| 1. User navigates to the "Customer Support" section. | 2. System presents option to issue a support ticket. |
| 3. User selects the option to either start a live chat or submit a support ticket. |  |
| 4. User describes their issue and submits the request. | 5. System assigns the request to a customer service agent. |
| 6. User receives confirmation that their request has been received. | 7. The support agent resolves the issue and sends a response to the user. |
| 8. User acknowledges the response, closing the support case. |  |

**Extensions (Alternate Flows):**

**· 1a. No representative for Live chat:**

2. If no representative is available for live chat, the customer is prompted to use other methods (e.g., FAQ).

**Technology and Data Variations List**

· Support can be provided via chat, email, or a ticketing system.

**Frequency of Occurrence:** Whenever the user encounters issues.

**-------------------------------------------------------------------------------------------------------------------------------**

**Use Case 8: Request a Test Drive**

**Name:** Request a Test Drive  
**Primary Actor:** Buyer  
**Scope:** Motoverse System  
**Level:** User Goal

**Stakeholders and Interests:**

· Buyer: Wants to test a vehicle before purchase.

· Seller: Wants to ensure buyers have a convenient way to schedule test drives.

· System: Manages the test drive scheduling process.

**Preconditions:**

· Buyer must be logged in to schedule a test drive.

· The selected vehicle must be available for test driving.

**Success Guarantee (Postconditions):**

· Test drive is scheduled, and both buyer and seller receive confirmation.

**Main Success Scenario (Basic Flow):**

|  |  |
| --- | --- |
| **Actions and Intentions** | **System Responsibility** |
| 1. Buyer navigates to the vehicle listing they are interested in. | 2. System displays an option to "Request a Test Drive." |
| 3. Buyer selects a date and time for the test drive. | 4. System confirms the availability of the selected time slot. |
| 5. Buyer confirms the request. | 6. System sends a confirmation to both the buyer and seller. |
| 7. Seller prepares the vehicle for the scheduled test drive. |  |
| 8. Buyer receives a reminder prior to the test drive. |  |

**Extensions (Alternate Flows):**

**· 4a. Time Slot Unavailable:**

1. If the selected time slot is unavailable, the system prompts the buyer to choose another time.

**Special Requirements:**

· The system must support real-time scheduling to avoid double-booking.

· Notifications should be sent via email.

**Technology and Data Variations List:**

· Time slots can be managed using a calendar system.

**Frequency of Occurrence:** Whenever a buyer is interested in a vehicle.

**-------------------------------------------------------------------------------------------------------------------------------**

**Use Case 9: Buy Vehicle (extends Vehicle Insurance)**

**Name:** Buy Vehicle  
**Primary Actor:** Buyer  
**Scope:** Motoverse System  
**Level:** User Goal

**Stakeholders and Interests:**

· Buyer: Wants topurchase a vehicle with clear details and no hidden costs.

· Seller: Wants to ensure the transaction is completed smoothly.

· Insurance Provider: Wants to offer insurance as part of the purchase process.

**Preconditions:**

· Buyer must have an active account.

· Vehicle must be listed for sale.

**Success Guarantee (Postconditions):**

· Vehicle purchase is completed, and the buyer receives confirmation.

· If applicable, insurance is purchased alongside the vehicle.

**Main Success Scenario (Basic Flow):**

|  |  |
| --- | --- |
| **Actions and Intentions** | **System Responsibility** |
| 1. Buyer browses vehicles and selects one they wish to purchase. | 2. System displays the vehicle’s details, price, and available options (e.g., insurance). |
| 3. Buyer confirms their intent to purchase the vehicle. |  |
| 4. Buyer is prompted to review optional insurance packages. |  |
| 5. Buyer selects an insurance package or skips the option. | 6. System processes payment for both the vehicle and insurance (if selected). |
| 7. Buyer receives a confirmation for the vehicle purchase (and insurance if applicable). |  |
| 8. Seller is notified of the sale and prepares the vehicle for delivery. |  |

**Extensions (Alternate Flows):**

**· 5a. No Insurance Selected:**

1. If the buyer skips the insurance option, the system finalizes the vehicle purchase without insurance.

**· 6a. Payment Failure:**

2. If the payment fails, the system notifies the user and prompts them to retry or select a different payment method.

.

**Technology and Data Variations List:**

· Payment can be processed via credit/debit cards, or other supported payment methods.

**Frequency of Occurrence:** Whenever a buyer completes a vehicle purchase.

**-------------------------------------------------------------------------------------------------------------------------------**

**Use Case 10: Buy Auto Parts and Accessories**

**Name:** Buy Auto Parts and Accessories  
**Primary Actor:** Buyer  
**Scope:** Motoverse System  
**Level:** User Goal

**Stakeholders and Interests:**

**· Buyer**: Wants to purchase parts or accessories easily.

**· System**: Facilitates the transaction smoothly.

**Preconditions:**

· Buyer must be logged in to make a purchase.

· The part or accessory must be available in stock.

**Success Guarantee (Postconditions):**

· The purchase is completed, and the buyer receives a confirmation.

**Main Success Scenario (Basic Flow):**

|  |  |
| --- | --- |
| **Actions and Intentions** | **System Responsibility** |
| 1. Buyer navigates to the auto parts and section. |  |
| 2. Buyer searches for and selects a part. | 3. System displays product details and price. |
| 4. Buyer adds the item to their cart. |  |
| 5. Buyer proceeds to checkout. | 6. System processes payment. |
| 7. Buyer receives a confirmation of the purchase. | 8. Seller is notified to ship the item. |

**Extensions (Alternate Flows):**

**· 5a. Out of Stock:**

o If the item is out of stock, the system notifies the buyer and allows them to set a reminder.

**· 6a. Payment Failure:**

o If the payment fails, the system notifies the user and prompts them to retry or select a different payment method.

**Special Requirements:**

· System must integrate with inventory management for real-time stock updates.

· The system should support multiple payment methods.

**Technology and Data Variations List:**

· Payment options can vary based on location and method.

**Frequency of Occurrence:** Whenever a buyer purchases parts or accessories.

**-------------------------------------------------------------------------------------------------------------------------------**

**Ahmed Ali:**

**Use Case 11: Post vehicle for Sale**

**Name:** Post vehicle for Sale

**Primary Actor:** Seller

**Scope:** Motorverse System

**Level:** User goal

**Stakeholders and Interests:**

**· Seller:** Wants to easily list their vehicle for sale, reach a wider audience, and receive inquiries.

**· Buyer:** Wants to find relevant vehicles for sale and contact sellers.

**· System:** Ensure that the listing’s information is correct, listing is correctly published, handle any dispute between seller and buyer and seamless flow for both seller and buyer.

**Preconditions:**

· Seller has an account

· Seller is logged in

**Success Guarantee (Postconditions):**

· The seller’s vehicle listing is published on the marketplace.

· Buyers can view the listing and contact the seller.

**Main success scenario:**

|  |  |
| --- | --- |
| **Actor Action (or Intention)** | **System Responsibility** |
| 1. Seller logs into their account. |  |
|  | 2. System verifies the seller's credentials. |
| 3. Seller navigates to the "Post Vehicle for Sale" section. |  |
|  | 4. System displays a form for entering vehicle details. |
| 5. Seller enters vehicle information (e.g., make, model, year, mileage, price, photos). |  |
|  | 6. System validates the entered information. |
| 7. Seller adds a description of the vehicle. |  |
|  | 8. System allows the seller to enter a detailed description. |
| 9. Seller submits the listing. |  |
|  | 10. System reviews the listing for compliance with marketplace guidelines. |
|  | 11. System publishes the listing. |
|  | 12. System displays the published listing to buyers. |

**Extensions:**

**4a. If user has exceeded posting limit:**

1. Don’t show the form

2. Tell the user to remove any previous listing to post this listing

**5a. Invalid input format:**

1. Ask the user to repeat the input

**6a. Information not validated correctly:**

1. Ask the user to repeat the input

**10a. Listing not approved:**

1. Cancel the listing

2. Inform User to resubmit listing

**-------------------------------------------------------------------------------------------------------------------------------**

**Use Case 12: Manage Vehicle Listing**

**Name:** Manage Vehicle Listing

**Primary Actor:** Seller

**Scope:** Motorverse System

**Level:** User goal

**Stakeholders and Interests:**

**· Seller**: Wants to easily manage vehicle listings, update/edit information, and receive inquiries

**· System**: Wants to update the seller listing correctly

**Preconditions:**

· Seller has an account

· Seller is logged in

· Seller has published a listing

**Success Guarantee (Postconditions):**

· Seller's vehicle listing is updated on the marketplace.

· Buyers can view the listing and contact the seller.

**Main success scenario:**

|  |  |
| --- | --- |
| **Actor Action (or Intention)** | **System Responsibility** |
| 1. Seller logs into their account. |  |
|  | 2. System verifies the seller's credentials. |
| 3. Seller navigates to their vehicle listings. |  |
|  | 4. System displays the seller's active listings. |
| 5. Seller selects a listing to manage. |  |
|  | 6. System displays the details of the selected listing. |
| 7. Seller updates listing information (e.g., price, description, photos). |  |
|  | 8. System saves the updated information. |
| 9. Seller views inquiries and messages. |  |
|  | 10. System displays any inquiries or messages related to the listing. |
| 11. Seller responds to inquiries. |  |
|  | 12. System sends the seller's response to the buyer. |

**Extensions:**

**2a. Failed to verify user:**

1. Ask user to renter credentials

**7a. Error in updating information:**

1. Ask user to renter information

**9a. No inquiries:**

1. Tell user there are no inquiries/messages.

**-------------------------------------------------------------------------------------------------------------------------------**

**Use Case 13: Rent a Vehicle**

**Name:** Rent a Vehicle

**Primary Actor:** Renter

**Scope:** Motorverse System

**Level:** User goal

**Stakeholders and Interests:**

**· Renter:** Wants to rent any available vehicle with easy booking and fair price

**· System:** Wants efficient rental management, increased revenue and customer satisfaction

**Preconditions:**

· Renter must be logged in to rent

· Renter must have a valid payment method

· Chosen Vehicle must be available

**Success Guarantee (Postconditions):**

· The purchase is completed, and the renter receives a confirmation, and vehicle is successfully available for renter**.**

**Main Success Scenario (Basic Flow):**

|  |  |
| --- | --- |
| **Actor Action (or intention)** | **System Responsibility** |
|  |  |
| 1. Renter browses available vehicles. |  |
|  | 2. System displays available vehicles |
| 3. Renter selects a vehicle and desired rental period. |  |
|  | 4. System checks vehicle availability for that period |
| 5. Renter provides personal Information and payment detail. |  |
|  | 6. System verifies renter's information and processes payment. |
| 7. System generates a rental agreement. |  |
|  | 8. System sends the rental agreement to the renter's email. |
| 9. Renter arrives at the rental location to pick up the vehicle. |  |
| 10. Renter inspects the vehicle and signs the rental agreement. |  |
| 11. Renter drives away with the rented vehicle. |  |
|  | 12. System updates the vehicle's rental status. |

**Extensions (Alternate Flows):**

**4a. Vehicle not available for that period:**

1. Tell the user vehicle is unavailable.

2. Recommend other vehicles

**6a. Payment failure:**

1. System prompts the renter to try a different payment method.

**10a. Renter doesn’t approve rental agreement:**

1. Cancel the transaction.

**-------------------------------------------------------------------------------------------------------------------------------**

**Use Case 14: Return rented vehicle**

**Name:** Return rented vehicle

**Primary Actor:** Renter

**Scope:** Motorverse System

**Level:** User goal

**Stakeholders and Interests:**

**Renter:** Wants efficient return process, accurate billing, timely refund.

**System**: Wants timely vehicle return, accurately calculate any damage and insurance cost, and preparation for the next rental**.**

**Preconditions:**

· Renter has a valid rental agreement.

· Vehicle is rented.

**Success Guarantee (Postconditions):**

· Vehicle is returned to company office

· Renter pays any damage cost if applicable

**Main Success Scenario (Basic Flow):**

|  |  |
| --- | --- |
| **Actions and Intentions** | **System Responsibility** |
| 1. Renter goes to company office and initiates return process | 2. System verifies the vehicle's return. |
|  | 3. System records any damage or issues. |
| 4. Renter pays any damage cost if applicable | . |
|  | 5. System sends a final invoice and receipt to the renter. |
|  | 6. System updates the vehicle's rental status. |

**Extensions (Alternate Flows):**

4a. Renter unable to pay damage cost at this moment:

1. Send a payment invoice to renter’s email with 2 days deadline.

**-------------------------------------------------------------------------------------------------------------------------------**

**Use Case 15: View rental history**

**Name:** View rental history

**Primary Actor:** Renter

**Scope:** Motorverse System

**Level:** User goal

**Stakeholders and Interests:**

**Renter**: Wants to Access and review their past rental transactions for reference.

**System:** Wants to correctly retrieve and show the rental history and provide transparency and customer service.

**Preconditions:**

· Renter has a rental history

**Success Guarantee (Postconditions):**

· Renter can view their rental history

**Main Success scenario:**

|  |  |
| --- | --- |
| **Actor Action (or Intention)** | **System Responsibility** |
| 1. Renter logs into their account. |  |
|  | 2. System verifies the renter's credentials. |
| 3. Renter navigates to their rental history. |  |
|  | 4. System displays a list of the renter's past rentals. |
| 5. Renter selects a rental to view details. |  |
|  | 6. System displays the details of the selected rental, including dates, vehicle information, and costs. |
|  |  |
|  |  |

**Extensions (Alternate Flows):**

**\* System error:**

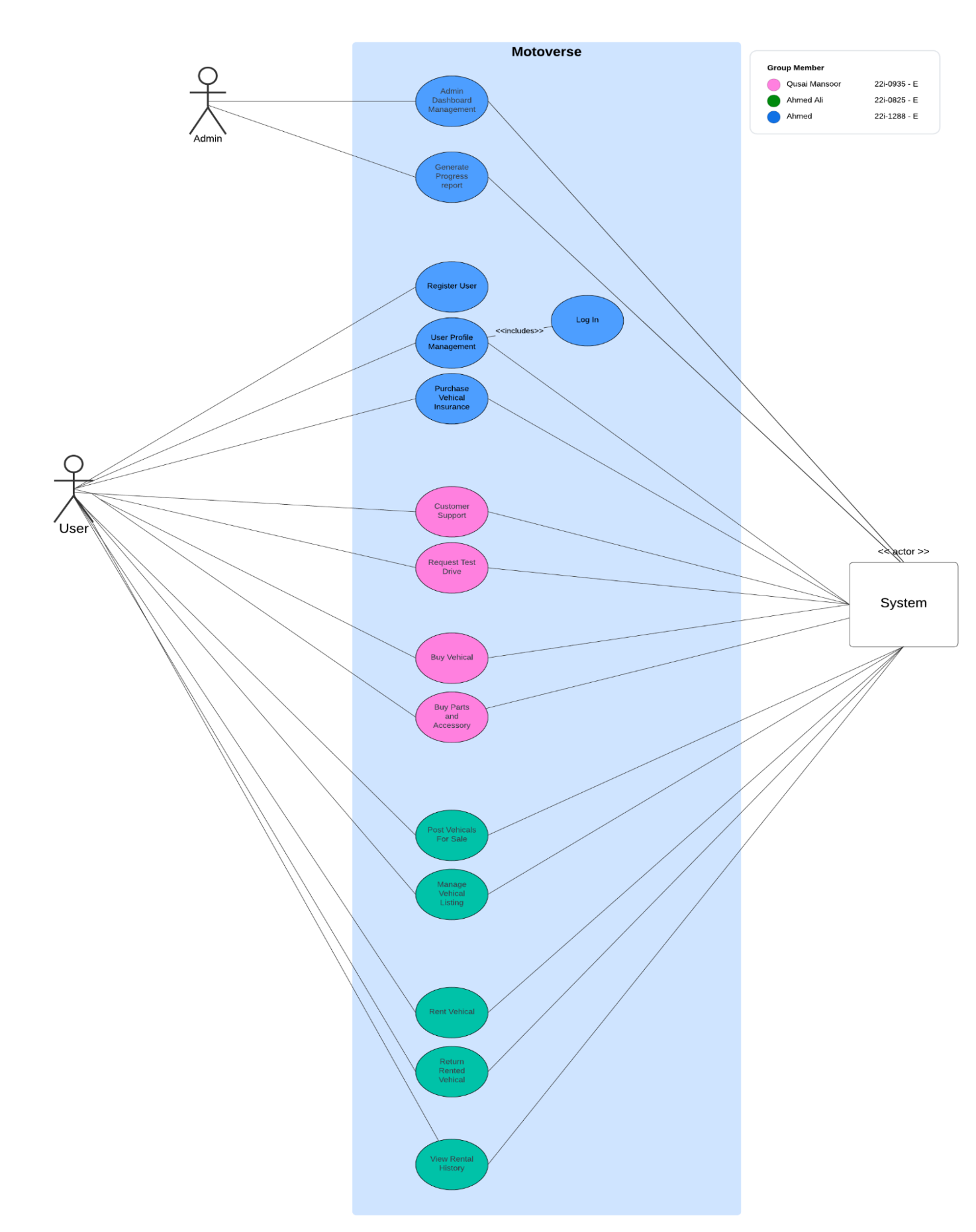
1. System displays an error message

2. Provides options for the renter to contact customer support.

**3a. No rental history found:**

1. System displays a message indicating that the renter has no rental history.

## **Use Case Diagram**



# **Other Nonfunctional Requirements**

## **Performance Requirements**

#### 

**Responsiveness**:

The application must load the dashboard, listings, and other major UI components within **2 seconds** under normal operating conditions.

Database queries (e.g., fetching vehicle listings, rental agreements) should complete within **500 milliseconds** for up to 10,000 records.

**Scalability**:

The application should efficiently handle up to **100 simultaneous user operations** on the same machine.

**Processing Speed**:

Payment processing through the PaymentContext should not exceed **1 second**, including validation and updates to the database.

**Rationale**:

These requirements ensure a smooth and fast user experience, critical for maintaining user satisfaction and avoiding frustration during time-sensitive actions like payments.

## **Safety Requirements**

**Error Prevention**:

Prevent user actions that could lead to invalid system states (e.g., double payments, conflicting rentals) through validation checks at both the UI and database levels.

**User Warnings**:

Provide clear warnings or confirmation dialogs for irreversible actions (e.g., deleting a listing, finalizing a payment).

## **Security Requirements**

***Authentication****:*

*Users must authenticate using a* ***username and password****.*

## **Software Quality Attributes**

**Adaptability**:

The application should support future integration with cloud databases (e.g., AWS RDS) without significant changes to the existing architecture.

**Availability**:

Ensure 99.9% availability in typical usage conditions (excluding system maintenance).

**Correctness**:

The application should accurately perform core operations (e.g., payment processing, rental agreement management) without logical or arithmetic errors.

**Flexibility**:

Adding new features (e.g., additional payment methods) should require minimal changes to the existing structure due to the use of the **Strategy Pattern** in Payment Context.

## **Business Rules**

**Role-Based Access**:

Only **Admins** can:

* + View and modify all user accounts.
  + Access system-wide reports.

**Payment Policies**:

Payments are non-refundable once processed, and this must be explicitly agreed upon by users during checkout.

**Listing Validity**:

Vehicle listings must include all required fields (e.g., make, model, price, and image) before being published.

## **Operating Environment**

#### Hardware Platform:

* **Minimum Requirements**:
  + **Processor**: Dual-core 2.0 GHz or better.
  + **RAM**: 4 GB minimum (8 GB recommended).
  + **Storage**: 500 MB of free disk space for application files and database.
  + **Display**: 1280x720 resolution or higher.

#### Operating System:

* The application is designed to run on the following operating systems:
  + **Windows**: Windows 10 or later.
  + **MacOS**: macOS 10.13 (High Sierra) or later.
  + **Linux**: Any distribution with Java Runtime Environment (JRE) support (e.g., Ubuntu 20.04, Fedora).

#### Software Components:

* **Java Runtime Environment (JRE)**:
  + The application requires **JRE 11** or later to run.
* **Database**:
  + **MySQL 8.0** is used as the database engine, installed locally on the same machine.
* **JavaFX Framework**:
  + JavaFX libraries (javafx.controls, javafx.fxml) are used for the graphical user interface.
* **MySQL Connector/J**:
  + The application uses the **mysql-connector-java** library for database connectivity.
* **File System Access**:
  + The application must have read/write access to the file system for saving images (e.g., vehicle photos uploaded by the user).

#### 

#### 

#### Networking:

* **Offline Operation**:
  + The application operates entirely offline with a local database and does not require internet connectivity.
* **Optional Future Support**:
  + Networking capabilities may be added later to enable cloud storage or remote database access.

## **User Interfaces**

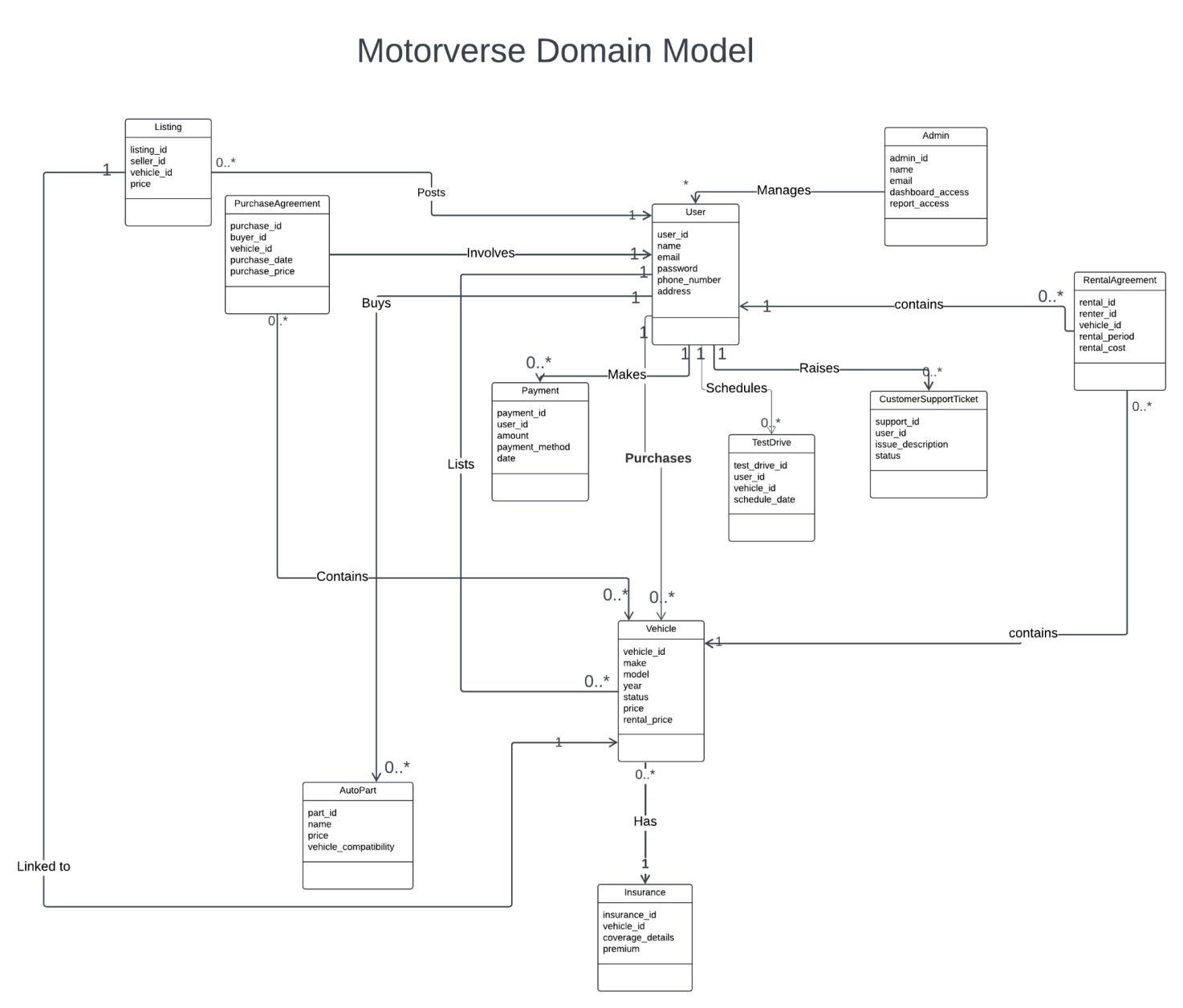
#### Error Display Standards:

* All error messages:
  + Use modal pop-ups to ensure visibility.
  + Provide actionable advice (e.g., “Password must be at least 8 characters long.”).
  + Avoid technical jargon and focus on user-friendly language.

#### Logical Characteristics:

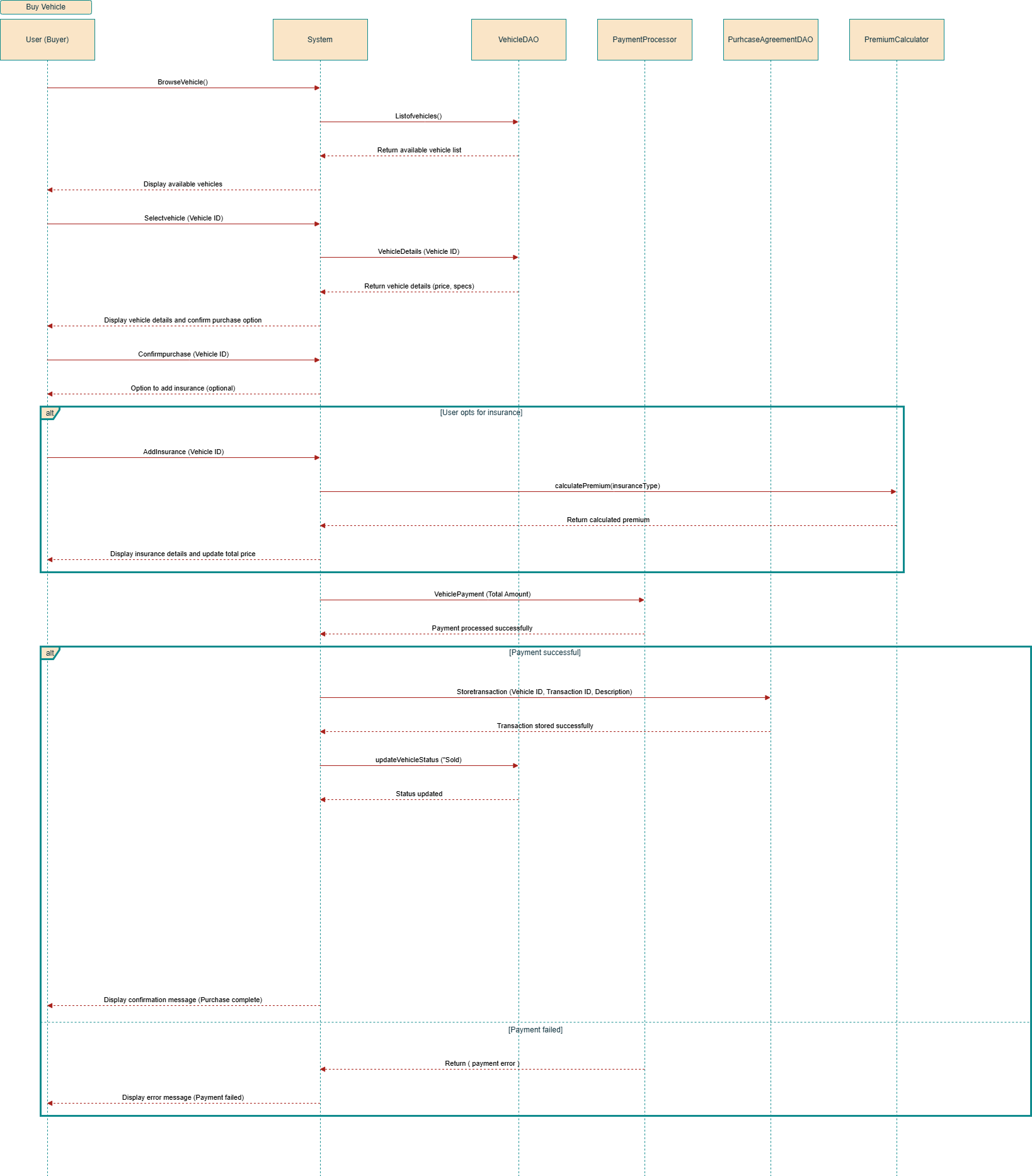
* **Screen Layouts**:
  + The application follows a consistent layout structure for all screens:
    - **Header**: Includes the application logo and navigation menu.
    - **Main Content Area**: Displays dynamic content based on user actions (e.g., vehicle listings, payment forms).
* **Standard Buttons and Functions**:
  + Common buttons:
    - **Submit**: For forms (e.g., login, payments, listing creation).
    - **Cancel**: Allows users to return to the previous screen without saving changes.

# **Domain Model**

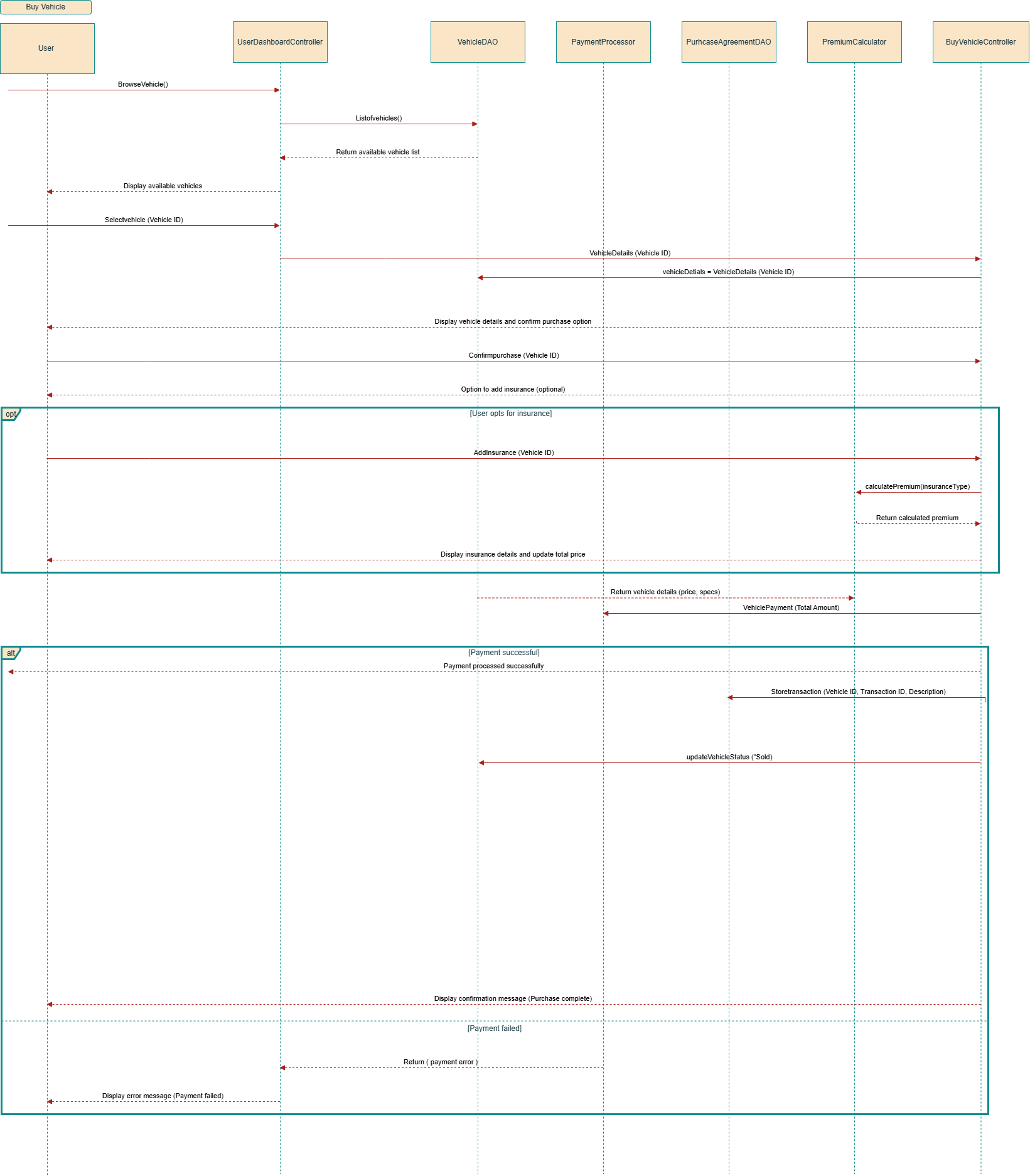
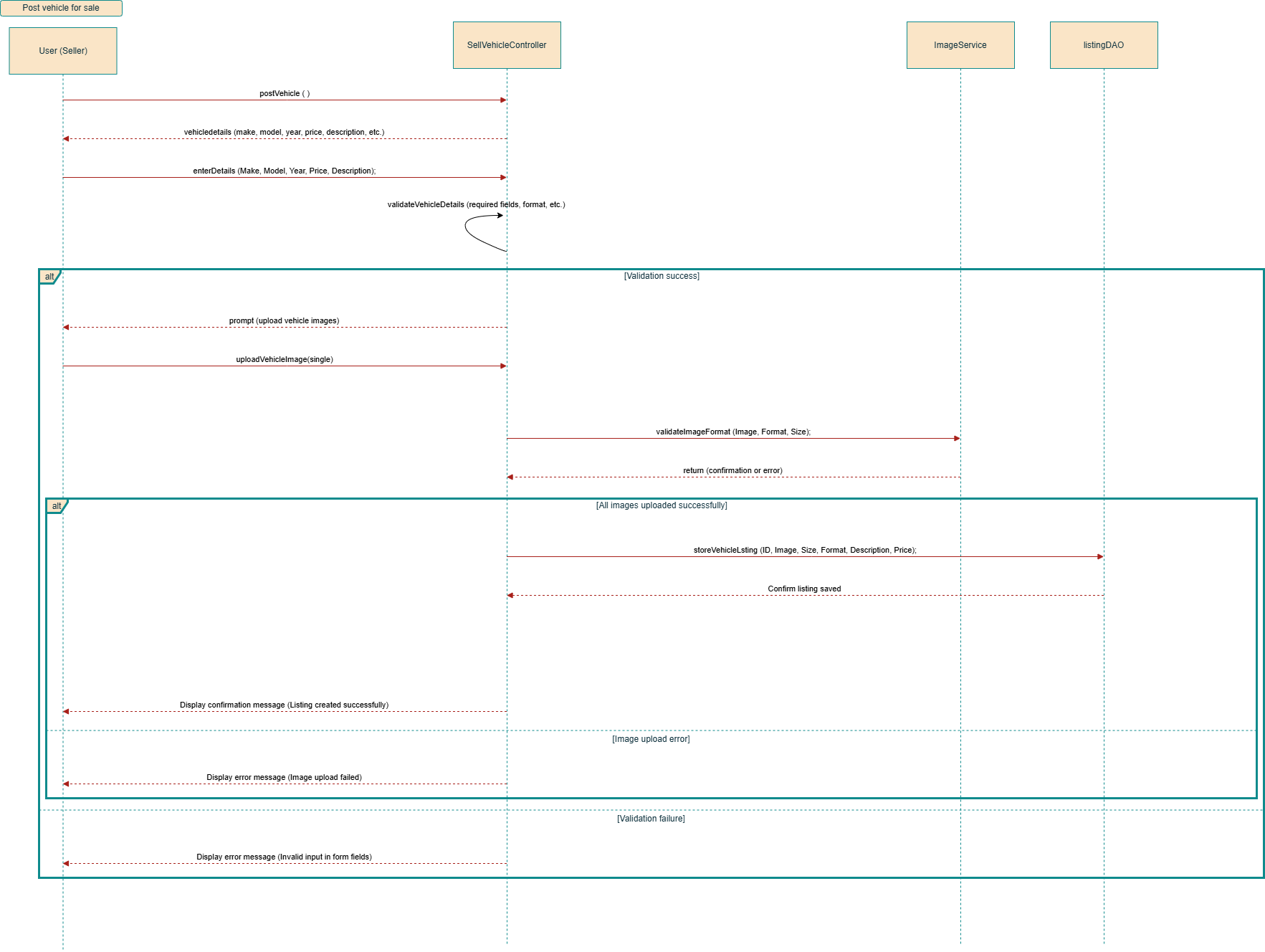
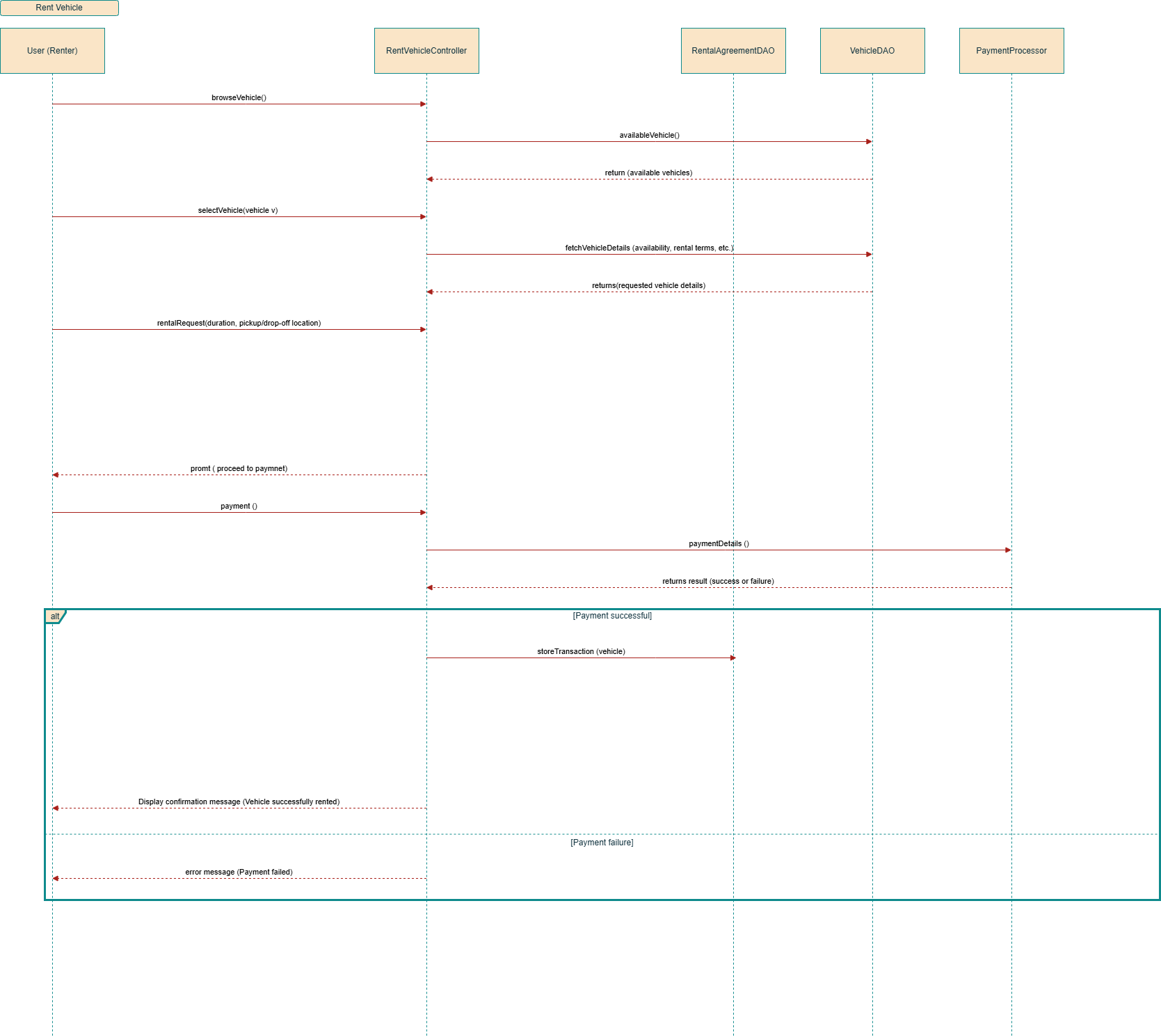
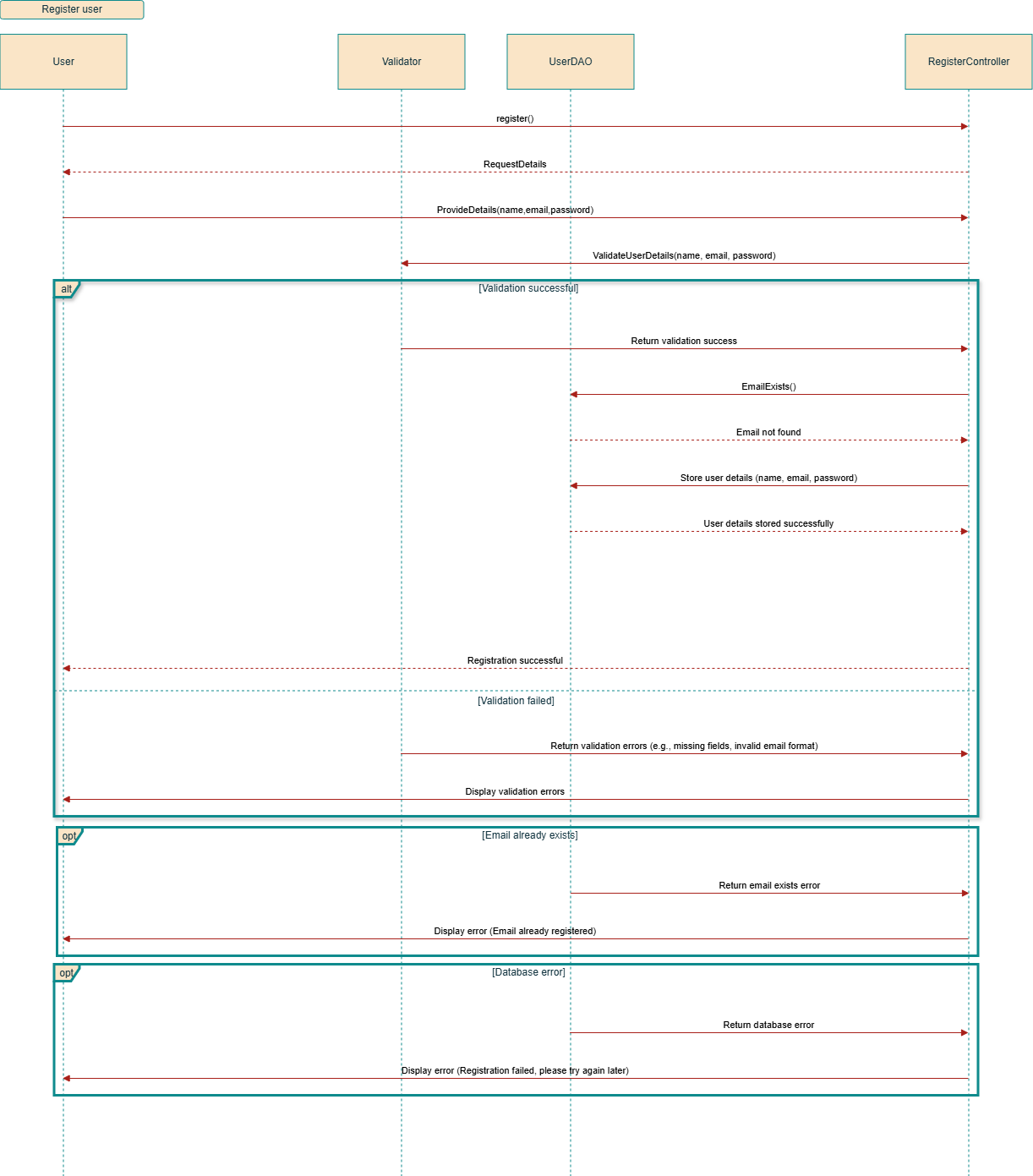


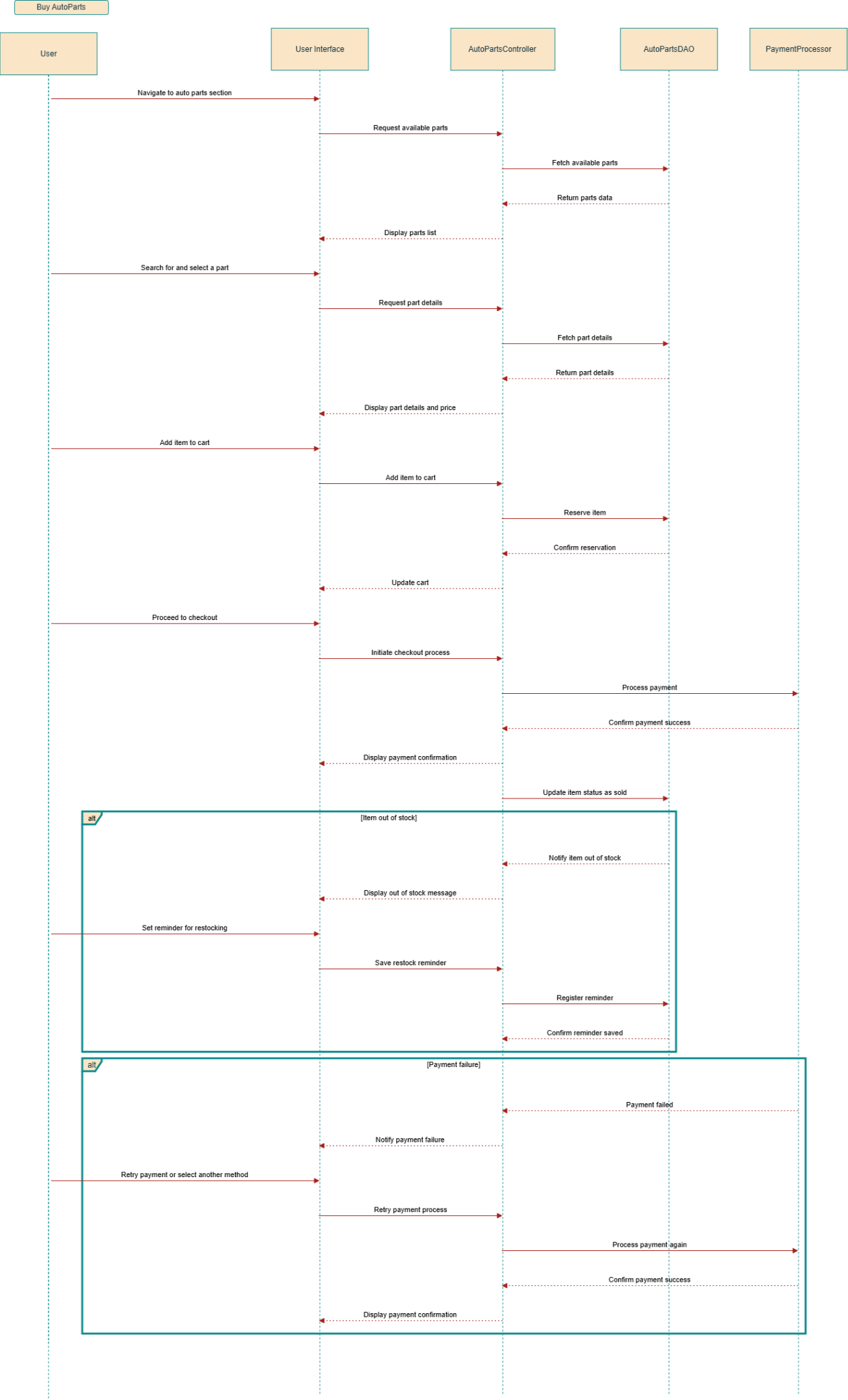
# **System Sequence Diagram**

(For the main Use Case of Buy Vehicle)

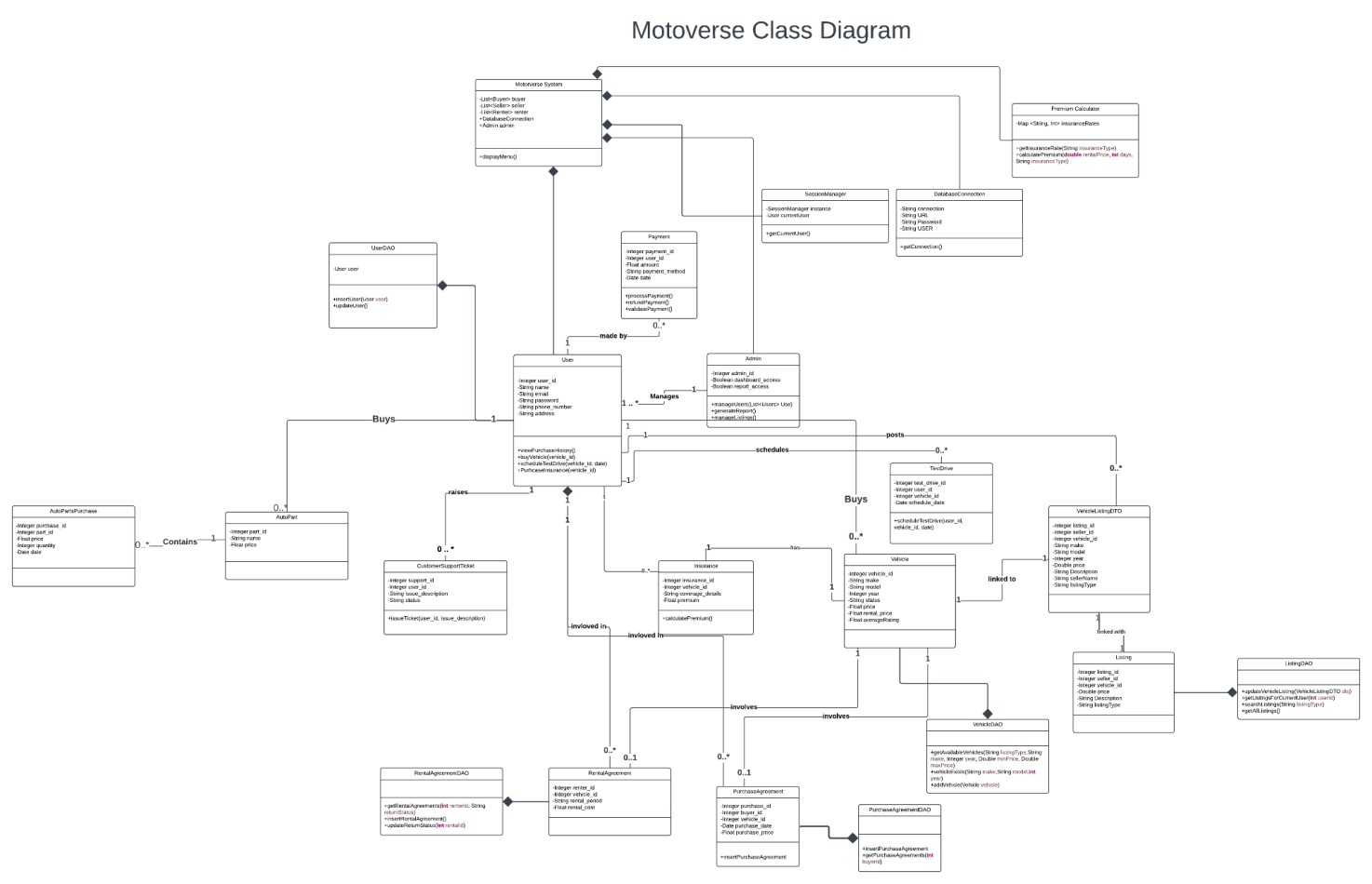


# **Sequence Diagram**

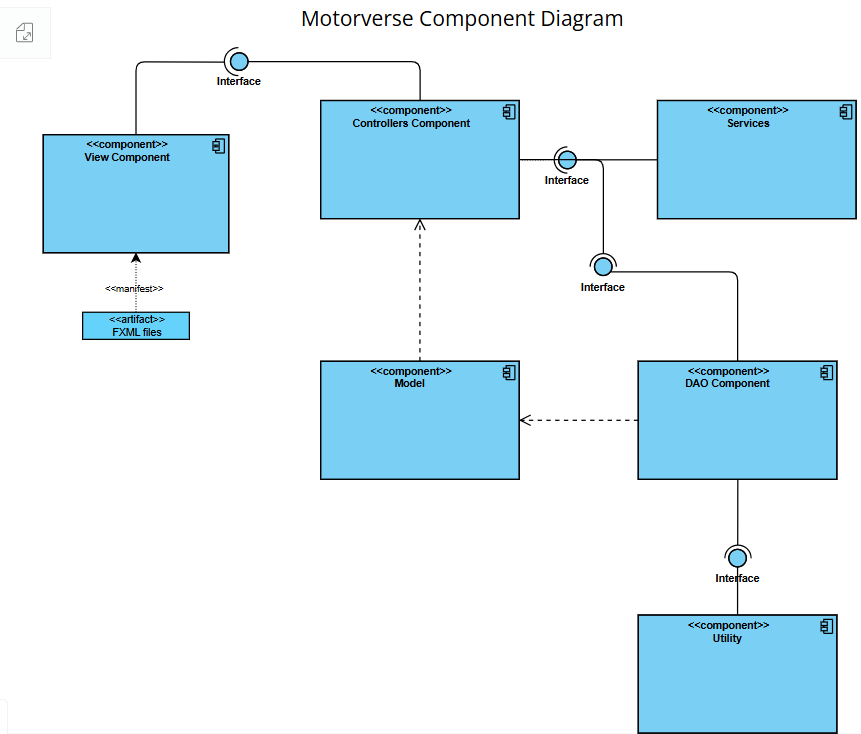




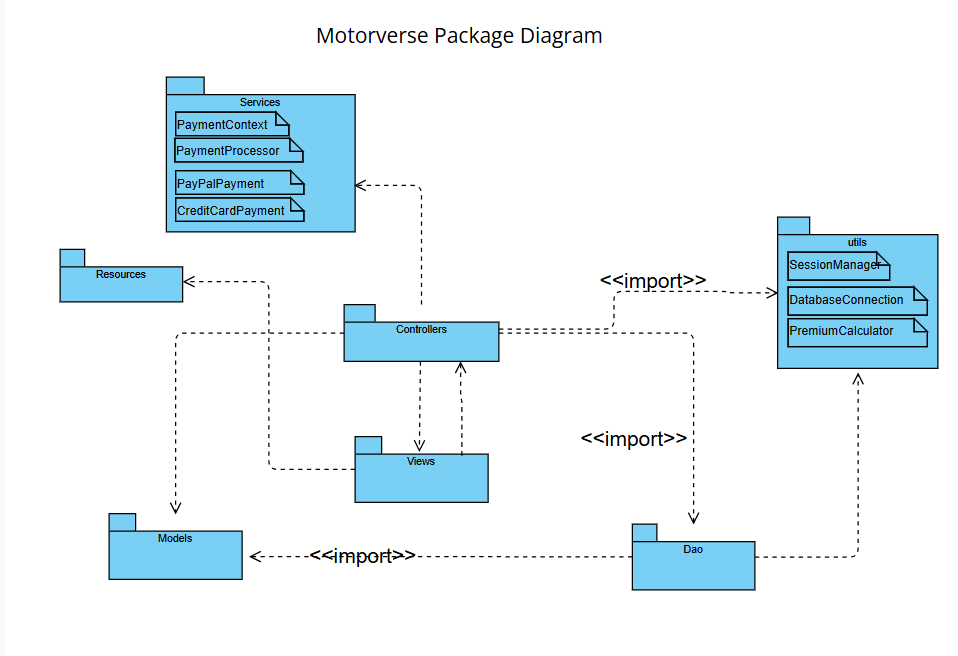
# **Class Diagram**



# **Component Diagram**



# **Package Diagram**



# **Deployment Diagram**

