Use Case Diagram: Car Rental Management System

Name: Aditya Agrahari

Registration Number: 23BRS1148

This document outlines the Use Case Diagram for the Car Rental Management System, incorporating detailed actor roles, system boundaries, and use case relationships like <<include>> and <<extend>>.

1. System Boundary

The system boundary defines the scope of the Car Rental Management System. Everything inside the boundary is part of the system being developed, including all the processes for booking, user management, and car inventory. The actors (User, Admin, Payment Gateway) operate outside of this boundary but interact with the system's functionalities. In the diagram, the boundary is represented by the rectangle that encloses all the use cases.

2. Actors and Roles

a. User (Traveler)

The primary user of the system.

Roles and Responsibilities:

• Account Management: The User can register for a new account, log in to their existing account, and update their profile information.

• Car Rental: The User's main role is to search for available cars, check their details, book a car for a specific period, and cancel a booking if needed.

• Transaction: The User provides payment information to finalize a booking and views their booking history.

b. Admin

The system administrator who manages the platform.

Roles and Responsibilities:

• Inventory Management: The Admin is responsible for managing the car database. This includes adding new cars to the system, updating the details of existing cars (e.g., availability, pricing), and removing cars that are no longer in service.

• User and Booking Oversight: The Admin can view and manage the database of all registered users and has access to all booking records within the system for monitoring and support purposes.

c. Bank / Payment Gateway

An external, third-party system responsible for handling financial transactions.

Roles and Responsibilities:

• Payment Processing: Securely processes the payment request initiated by the User during the booking process.

• Payment Validation: Verifies the User's payment details (e.g., credit card information) and sends a confirmation or failure status back to the system.

3. Use Cases and Relationships

Use cases represent the functionalities of the system. They are linked to actors and to each other through relationships.

Relationships: <<include>> vs. <<extend>>

• <<include>> (Mandatory): This relationship signifies that a base use case must incorporate the functionality of another use case. For example, Book Car must always include Make Payment. The booking process is not complete without payment.

• <<extend>> (Optional): This relationship signifies that a use case can optionally include additional functionality. For example, a User can choose to Apply Promo Code when they Book Car, but it is not a required step.

Key Relationships in this System:

• Book Car includes Make Payment.

• Make Payment includes Process Transaction.

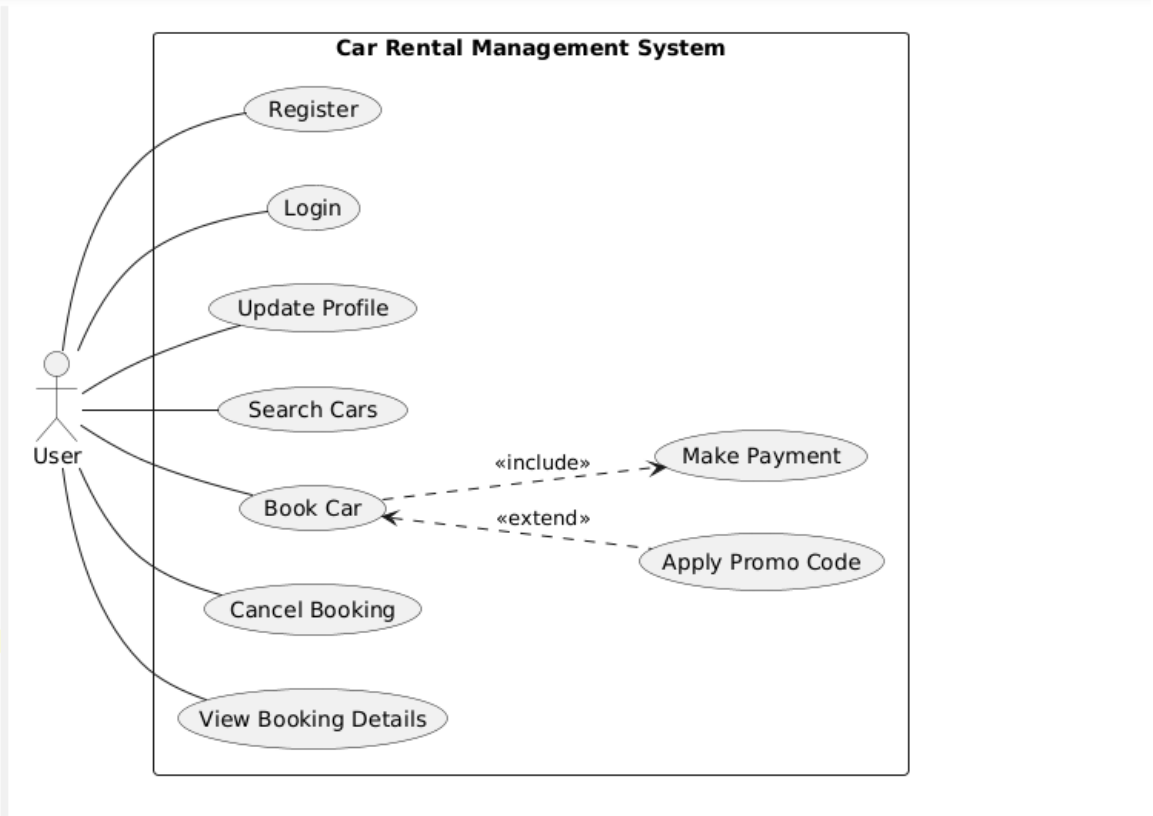
• Book Car can be extended by Apply Promo Code.

• User and Admin are specializations of a general System User who must Login.

4. Individual Use Case Diagrams by Actor

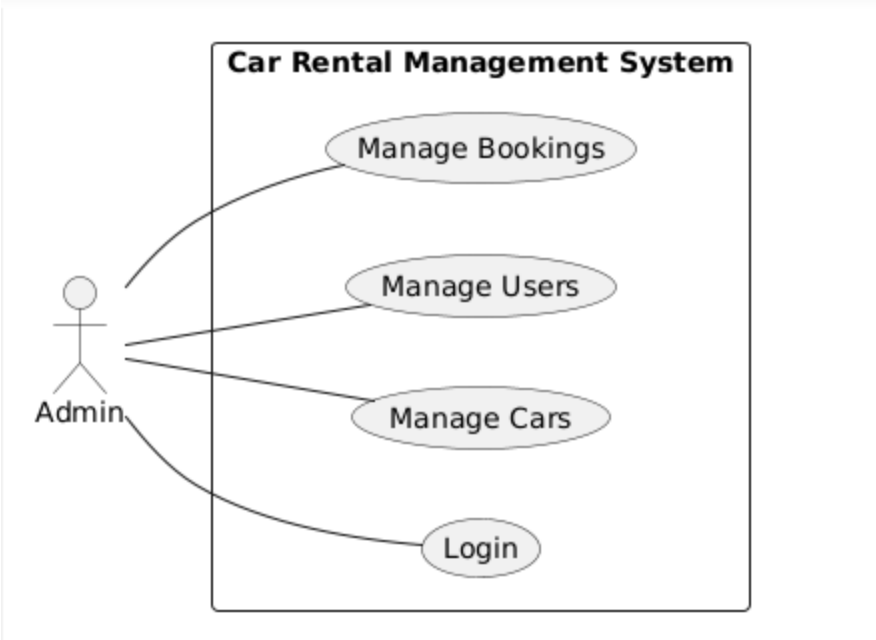
1. User (Traveler) Diagram

This diagram shows all the functionalities available to a User. It focuses on account management, car searching, booking, and managing their reservations.



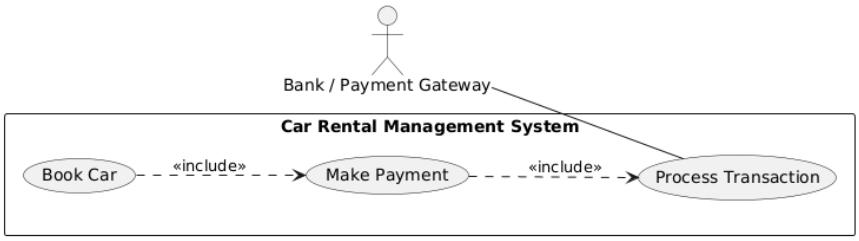
2. Admin Diagram

This diagram illustrates the backend management tasks performed by the Admin. The focus is on system oversight, including inventory control and management of users and bookings.



3. Bank / Payment Gateway Diagram

This diagram shows the focused interaction of the external Bank / Payment Gateway system. Its sole responsibility is to process financial transactions securely when a user makes a payment.



4. Combined Use Case Diagram