**Notes on Use Case and Sequence Diagrams for the Event Car Service App**

**1. Use Case Diagrams**

A use case diagram helps you visualize the different types of users (actors) interacting with the system and the functions (use cases) that the system provides. For the Event Car Service App, the focus will be on three actors:

**Customer:** The user booking the rides.

**Driver**: Service providers who drive customers.

**Admin:** The system administrator who manages the app and oversees operations.

**Payment Gateway**: External system for processing payments.

Each actor will interact with different functionalities in the app. The use case diagram will show how each actor performs their respective tasks.

**Use Cases for Customers:**

**Register/Login**: Customers and drivers create accounts or log into the App.

**Book Ride**: Customers request a ride and schedule a booking.

**Track Service Ride:** Customers track their ride’s progress and it’s arrival.

**Make Payment:** Handle payments for rides through the App.

**Rate and Review Service Provider:** Provide feedback on the service.

**Manage Rides**: Admins view and manage all rides.

**Use Cases for Service Providers:**

**Create Profile:** Register and update service offerings and personal details.

**Manage Availability:** Set working hours and availability for ride bookings.

**Accept or Reject Jobs:** Review job requests and decide to accept or reject based on availability.

**View Booking Details:** View specific event information after acceptance.

**Receive Payments:** Track event deposits and receive payments for completed ride services provided.

**Use Cases for Admin:**

**Manage Users:** Approve or reject service provider registrations.

**Monitor Transactions:** Oversee payments and earnings on the platform.

**Resolve Issues:** Handle complaints, refunds, and disputes.

**Manage Content:** Update and manage service categories, prices, and system settings.

**2. Sequence Diagrams**

Sequence diagrams show the interaction between users and the system over time. For the Event Car Service App, here, the focus will be on the main booking workflows.

**Sequence Diagram for Booking a ride.**

This diagram will show the interaction between a **customer**, the **system**, and a **driver** as a customer books the event car.

**Steps;**

The **Customer** requests a ride through the app.

The **System** displays available cars and drivers for the selected ride service.

The **Customer** chooses a ride and initiates the booking process.

The **system** sends a booking request to the **driver** to accept the ride.

The **system** confirms the booking and sends notifications to the **customer**.

On the scheduled date, the **customer** can track the **service provider**'s location.

The **customer** makes the payment to confirm the transaction.

The **system** notifies the **driver** and **customer** of the successful booking.

**Sequence Diagram for Service Provider Registration**

This diagram will show the process of a **service provider** registering on the app.

Steps:

The **service provider** opens the App and selects the registration option.

The **system** prompts for details (Name, Location, ID verification, Request Ride).

The **service provider** submits the registration form.

The **system** sends the registration request to the **admin** for approval.

The **admin** verifies the information and either approves or rejects the booking.

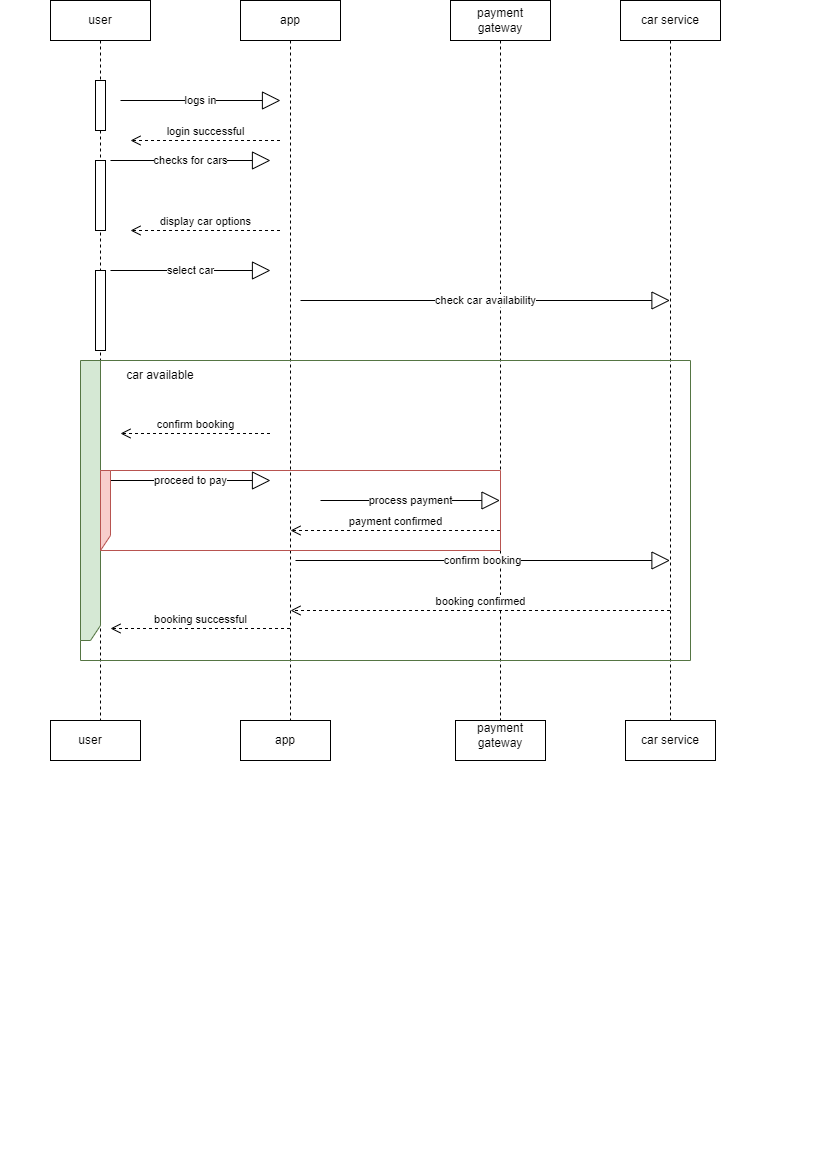
The **system** notifies the **service provider** to make payment.

The **system** confirms the payment via payment gateway.

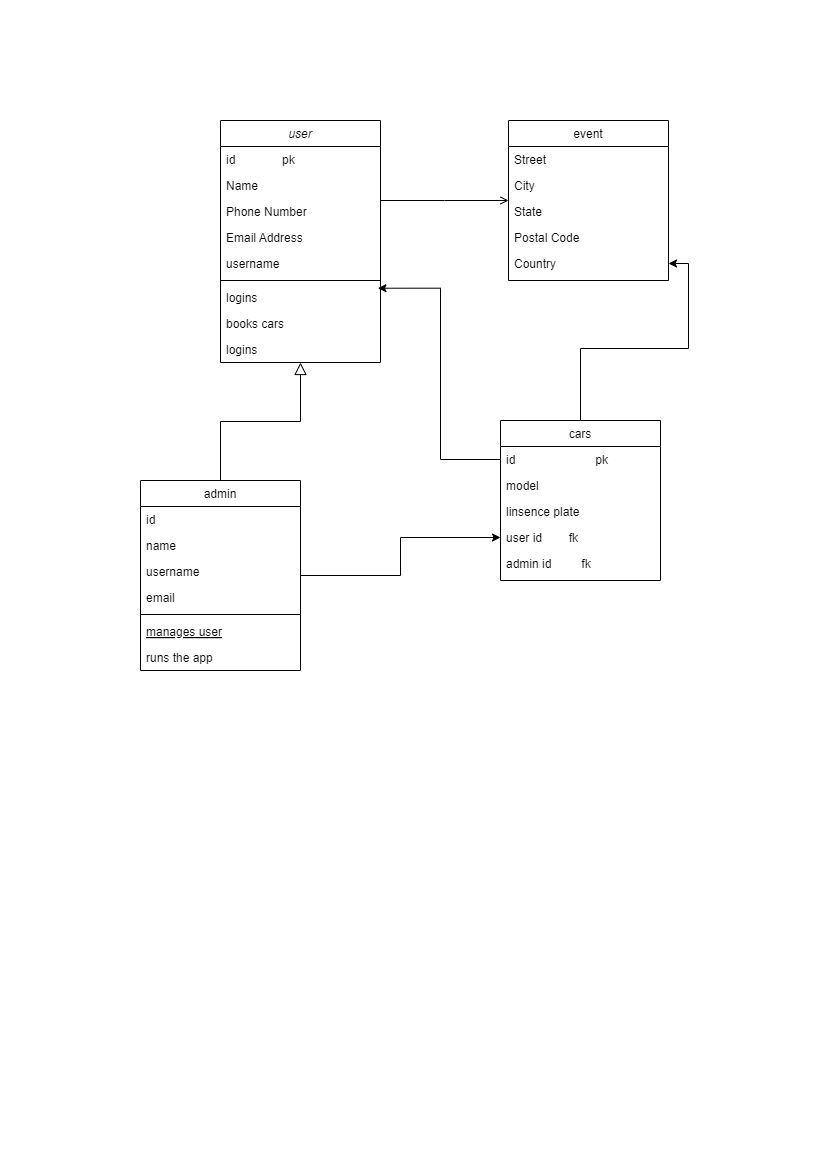
The **system** notifies both the **customer** and the **driver** a successful booking.

**SEQUENCE DIAGRAM AND FUNCTIONAL COMPONENTS FOLLOW**

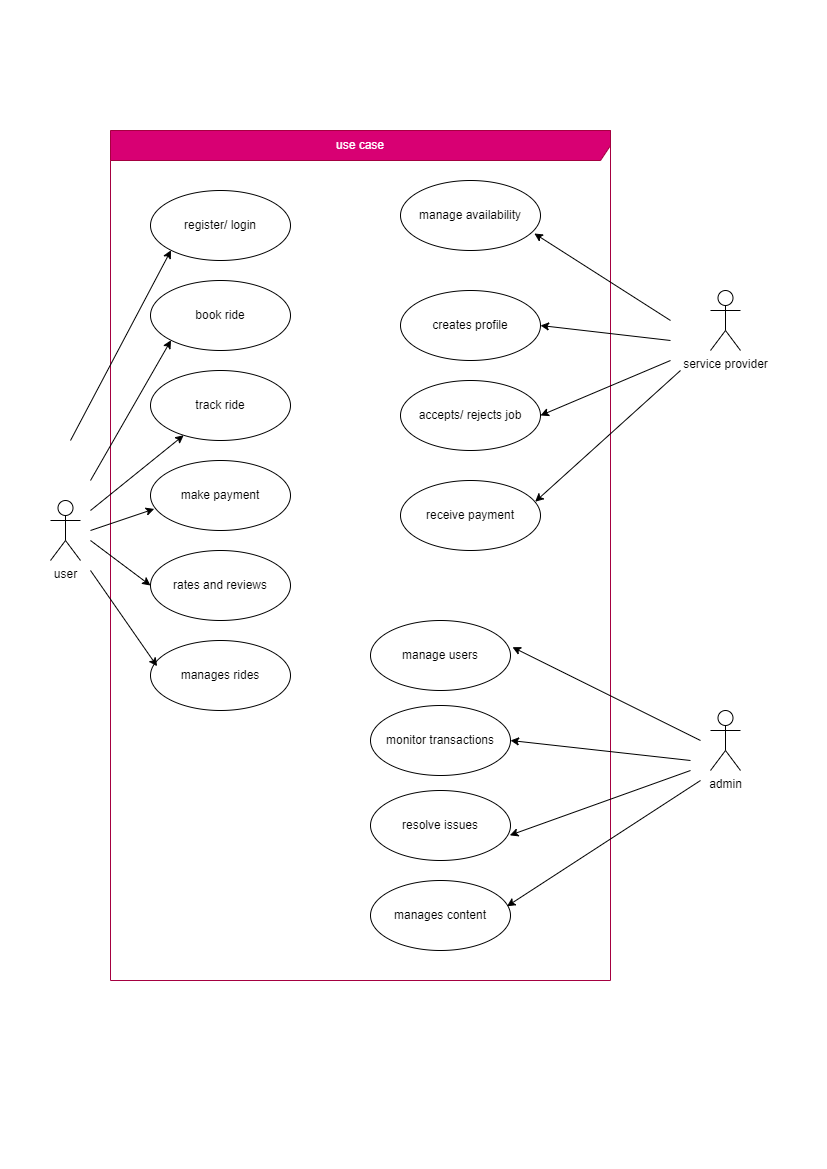
Customer requests a ride through the app.



**CLASS DIAGRAM**

****

**USE CASE.**

****