Taxi mobile application system

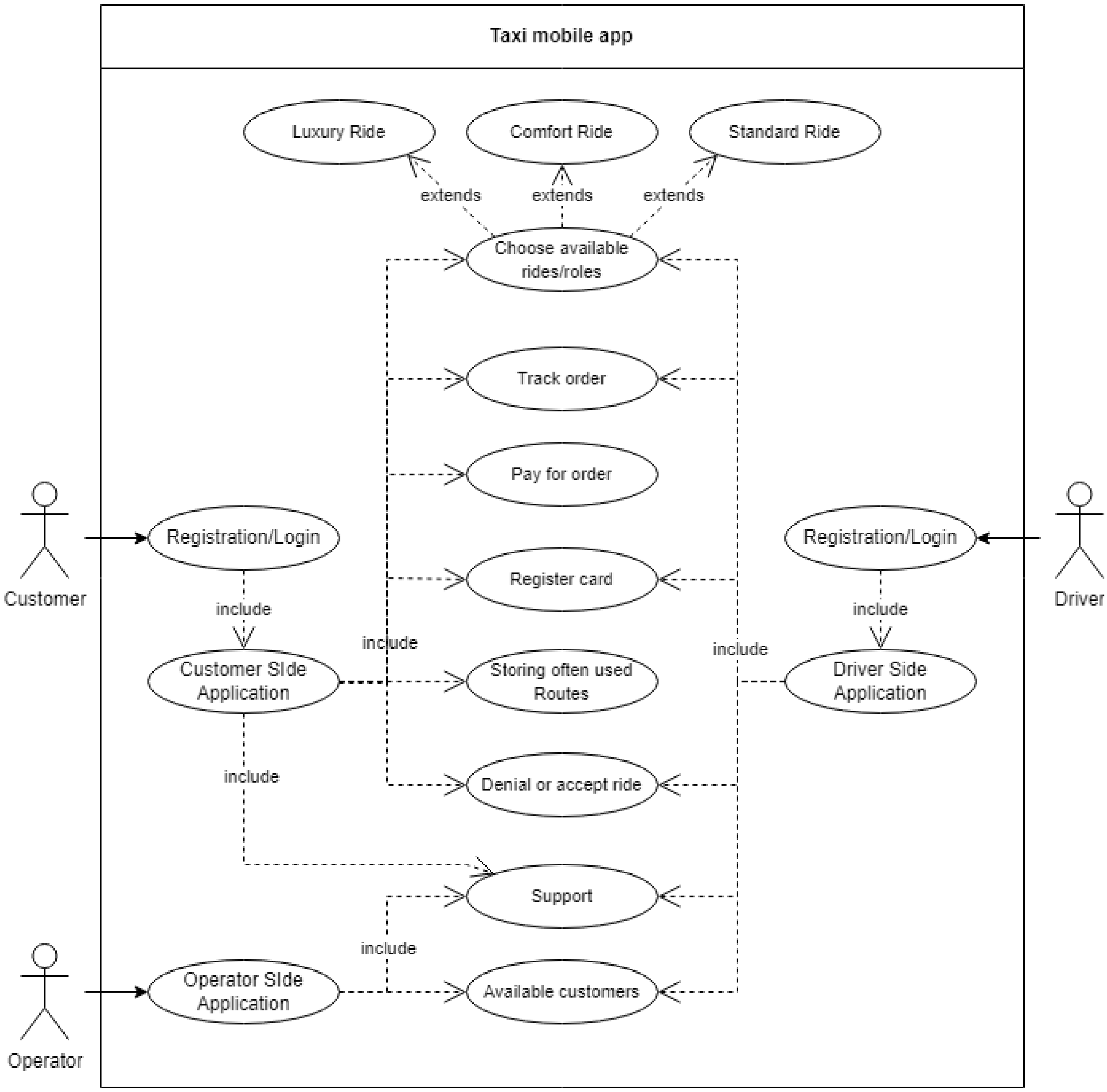

Figure 1. Use Case Diagram

Functional Requirements

Functional requirements specify the fundamental actions that the system must perform. In the context of the Taxi Mobile App, these requirements include:

1. User Registration and Login:

Allow customers and drivers to register and create their profiles using details such as name, contact information, and payment information.

Secure login mechanism for users to access their accounts.

2. Ride Options:
Provide different categories of rides such as Luxury, Comfort, and Standard, each with distinct features and pricing.

3. Booking and Management of Rides:

- Enable customers to choose and book available rides.
- Allow drivers to accept or decline ride requests.
- Provide functionality for customers and drivers to cancel the ride.

4. Payment Integration:

- Process payments through integrated secure payment systems.
- Support multiple forms of payment such as credit cards, debit cards, and mobile payments.

5. Route Management:

- Save and suggest frequently used routes to customers.
- Offer real-time GPS tracking of the ride for customers.

6. Support and Assistance:

Provide a help and support system for both drivers and customers to resolve issues related to rides, payments, or app usage.

7. Operator Interface:

Enable operators to view available customers and manage service quality.

Non-Functional Requirements

Usability:
The app should be user-friendly, with an intuitive interface that is easy to navigate for all user categories including the elderly.

Reliability:
The app should be reliable, with minimal downtime, ensuring that it is operational 24/7 without interruption.

Performance:
Response time should be quick, with actions like booking confirmation and payment processing completed in real time. The app should handle high volumes of user requests efficiently during peak times.

Security:
Implement strong security measures to protect user data, including encryption for data transmission and secure storage of personal and financial information.
Compliance with data protection regulations (e.g., GDPR, CCPA).

Scalability:
The system should be scalable to accommodate growth in the number of users and simultaneous ride requests without degradation in performance.

Accessibility:
The app should be accessible to users with disabilities, incorporating features such as voice commands, high-contrast visual options, and screen readers.

Maintainability:
The app should be easy to update and maintain, with the ability to quickly deploy new features or patches.

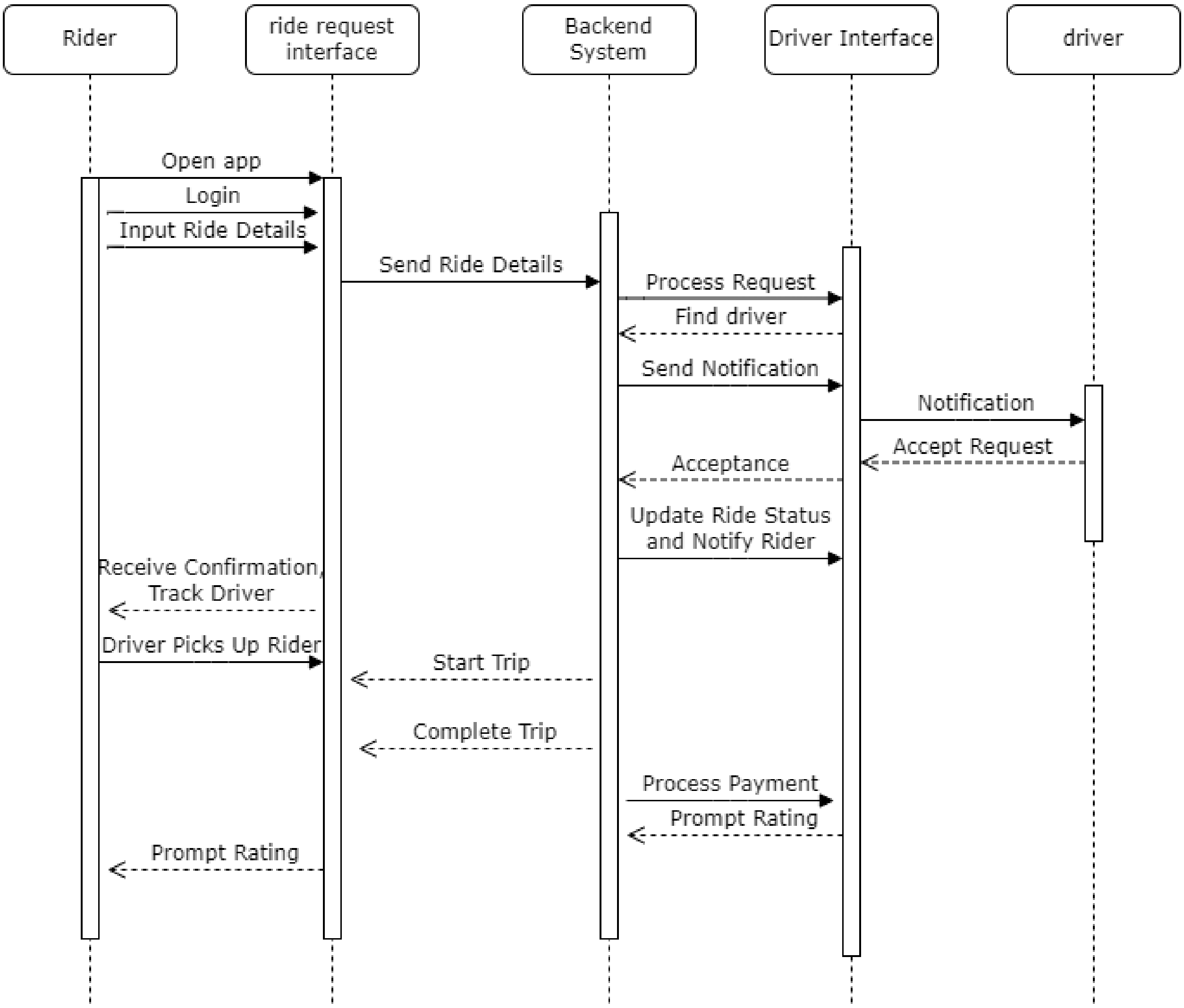

Figure 2. Sequence Diagram

Description of the sequence diagram

Rider: Opens the app and logs in.

Inputs ride details (pickup and drop-off locations) into the Ride Request Interface.

Ride Request Interface: Sends the ride details to the Backend System.

Backend System: Processes the request, finds a matching Driver, and sends a ride request notification to the Driver Interface.

Driver Interface: Receives the notification and the Driver accepts the ride request.

Sends acceptance confirmation back to the Backend System.

Backend System :Updates the ride status and notifies the Rider via the Ride Request Interface.

Rider: Receives confirmation and can track the Driver arriving.

Driver: Picks up the Rider, starts the trip in the app, completes the trip, and ends the ride in the app.

Backend System: Processes the payment and prompts the Rider to rate the ride.This sequence diagram provides a clear and detailed view of the interactions between all participants in the ride-hailing process, highlighting the flow of information and actions from start to finish.