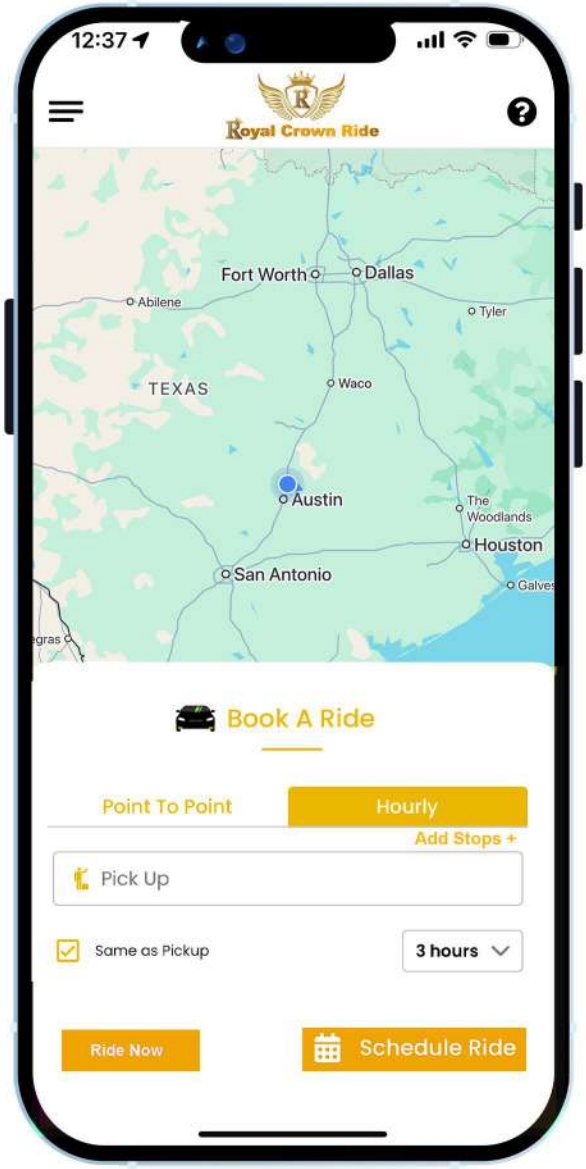
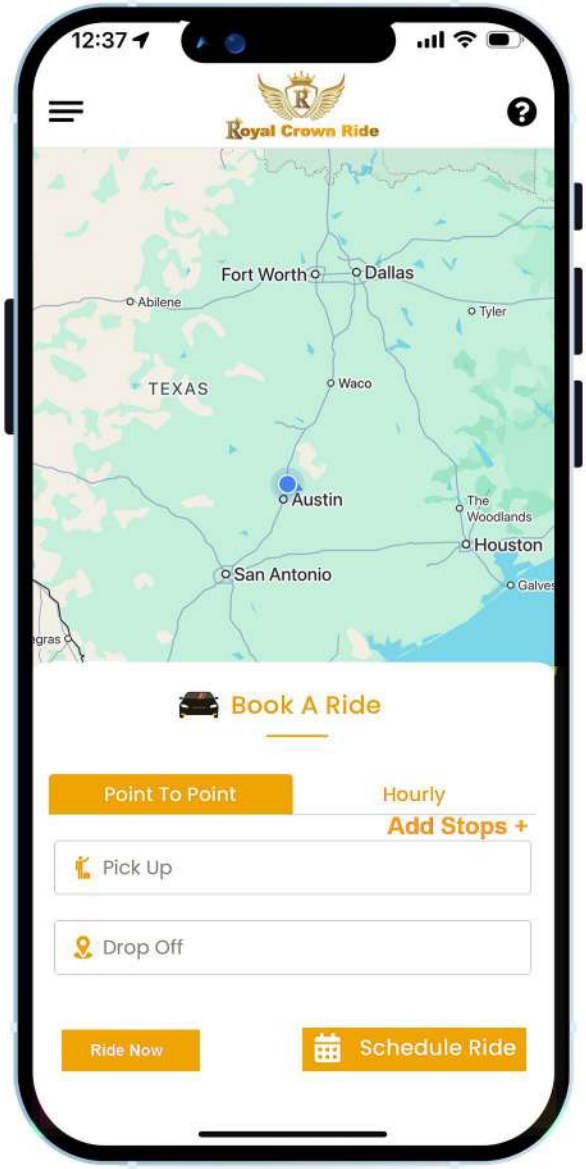
Royal Ride
Passnger App



Royal Ride
Passnger App



Passenger Features for the Rideshare App

1. Book Ride on Demand

Goal: Allow passengers to book a ride instantly based on their current location.
Functionality:
Passenger selects pickup location (auto-detected or manually entered).
Choose drop-off location.
Ride is booked with immediate availability of nearby drivers.
Technology:
Google Maps API for location services.
Firebase for real-time data handling and location tracking.
Real-Time GPS integration for driver location updates.

2. Book Scheduled Trip

Goal: Allow passengers to schedule a ride for a future time and date.
Functionality:
The passenger selects a date and time for pickup.
System ensures drivers are available for the scheduled time.
Passenger receives a reminder before the ride.
Technology:
Firebase Firestore or AWS DynamoDB to store scheduled trips.
Cloud Functions for sending reminders and notifications.
Google Calendar API integration for scheduling functionality.

3. User Tracking Driver on Live Map

Goal: Allow passengers to track the assigned driver's real-time location on the map.
Functionality:
After booking, passengers can see the driver's location on the map.
Estimated Time of Arrival (ETA) is displayed.
Technology:
Google Maps SDK or Mapbox for real-time map updates.
Firebase Realtime Database for location updates.
Socket.IO or WebSockets for real-time communication.

4. Passenger Can Store, Modify, and Cancel Scheduled Trips

Goal: Let passengers store their scheduled trips, modify them, or cancel if needed.
Functionality:
Passengers can save trips for future reference.
Modify trip details (pickup/drop-off locations, time) and receive confirmation.
Cancel a ride with a notification.
Technology:
Firebase Firestore for storing trips.
Cloud Functions for modifying/cancelling trips and triggering notifications.
Push Notifications for ride status updates.

5. Passenger Can Book Ride for Someone in a Different Location

Goal: Enable passengers to book a ride for someone else at a different location.
Functionality:
Enter the recipient's name, phone number, and address.
Specify the ride details and ensure the driver picks up at the given location.
Technology:
Google Maps API for location input and validation.
Firebase Firestore to store ride information for the person being booked for.

6. Passenger Can Add Stop or Edit Drop-Off Location After the Trip Starts

Goal: Allow passengers to modify the route during the trip
(e.g., adding a stop or changing the drop-off location).
Functionality:
Passenger can add a stop or modify the drop-off point in the app during the ride.
System recalculates the fare and provides an updated ETA.

Technology:
Google Maps Directions API for recalculating routes.
Firebase Realtime Database for instant updates and communication with the driver.

7. Passenger App Must Pay and Collect Money First

Goal: Ensure passengers pay upfront before the ride begins.
Functionality:
The passenger must provide payment details before confirming the ride.
Payment is processed immediately after booking.
Technology:
Stripe or Braintree for payment processing.
Apple Pay/Google Pay SDK for easy mobile payments.
Firebase for storing payment details securely (using encryption).

8. Driver Can Add Notes/Instructions (e.g., Gate Code, Road Navigation)

Goal: Allow drivers to add instructions for passengers (e.g., gate codes, specific entry points).
Functionality:
Driver can add notes that will be visible to the passenger before or during the ride.
Technology:
Firebase Firestore or Realtime Database for storing and syncing notes.
Push Notifications to alert the passenger about special instructions.

9. Driver Can Add Pet, Child Car Seat, and Booster Seat Options

Goal: Allow passengers to request specific accommodations, such as pets, child car seats, or booster seats.
Functionality:
During the ride booking, passengers can choose to add a pet or select a car seat/booster seat.
The driver is notified of these preferences.
Technology:
Firebase Firestore for storing these preferences.
Push Notifications for notifying the driver of special requests.

10. Passenger Can Add Luggage

Goal: Allow passengers to indicate if they have luggage that requires additional space.
Functionality:
Passengers can select the luggage option when booking the ride.
Technology:
Firebase Firestore to track this preference.
User Interface Design for luggage options in the booking flow.

11. Passenger Can Save Payment Card on File

Goal: Allow passengers to securely store their payment methods for future use.
Functionality:
Passengers can securely add their card information, which is stored for future rides.
Technology:
Stripe or Braintree for secure payment storage.
Tokenization to store card details securely, complying with PCI DSS standards.

12. Passenger Can Call Dispatch, Emergency Phone, or the Driver

Goal: Provide an option for passengers to call emergency services, dispatch, or the driver directly through the app.
Functionality:
A button that connects the passenger to emergency services or the dispatch center.
In-app call feature to contact the driver or dispatch team.
Technology:
Twilio API for in-app calling or messaging.
Firebase Cloud Messaging for notifications and emergency triggers.

13. Passenger Can Choose Vehicle Type

Goal: Allow passengers to choose from a variety of vehicle types (e.g., economy, premium, SUV).
Functionality:
Display available vehicle options (e.g., Sedan, SUV, XL, etc.) with details about the vehicle and fare estimates.
Technology:
Firebase Firestore to store vehicle types and availability.
User Interface (UI) to display vehicle options clearly.

Summary of Key Technologies for Development:

Google Maps API/SDK: For real-time location tracking, route planning, and map integration.
Firebase Firestore/Realtime Database: For storing and syncing trip data, user preferences, and ride statuses in real-time.
Stripe/Braintree: For secure payment processing and card tokenization.
Twilio API: For in-app calling and SMS functionalities.
Push Notifications (Firebase Cloud Messaging): For real-time alerts, reminders, and special notifications.
AWS Cloud Functions: For server-side logic, like trip reminders, cancellations, and payment validation.
Mapbox (alternative to Google Maps): For custom map styling and enhanced map functionalities.