

RTC Bus Ticket Booking App

1. Introduction

The RTC Bus Ticket Booking App aims to provide a seamless digital ticketing experience for public transport buses, similar to metro train ticketing. The app will use QR codes to facilitate ticket booking, boarding verification, and fare calculation.

2. Features

User Features

- **QR Code Scanner:** Scan QR codes installed on buses to fetch bus details.
- **Route Selection:** Users select boarding and destination stops.
- **Fare Calculation:** The system calculates the fare based on distance.
- **Digital Ticket Generation:** Generate a digital ticket with a QR code.
- **Payment Integration (Future Scope):** Users can pay through UPI, wallets, or credit/debit cards.
- **Ticket History:** View past ticket bookings.

Admin Features

- **Bus Management:** Update bus details, routes, and schedules.
- **Fare Management:** Modify fare calculation logic.
- **Analytics Dashboard:** Track ticket sales and revenue.

3. Technology Stack

Frontend

- **Framework:** React Native (Expo)
- **Libraries:**
 - expo-barcode-scanner (QR scanning)
 - axios (API calls)
 - react-native-qrcode-svg (QR code generation)

Backend

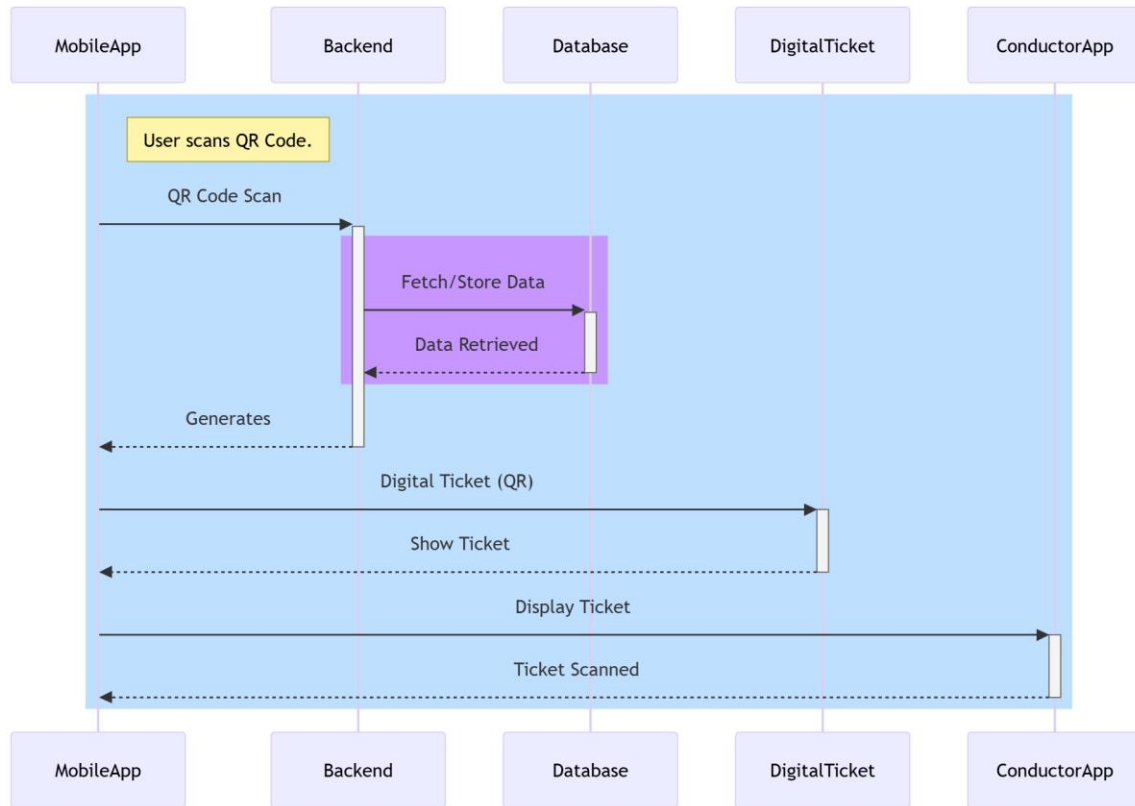
- **Technology:** Django ,
- **Database:** MongoDB / Firebase / PostgreSQL (TBD)
- **API Services:**
 - /fetch_bus_details - Get details of the scanned bus
 - /calculate_fare - Calculate fare from source to destination
 - /generate_ticket - Generate a digital ticket

4. Workflow

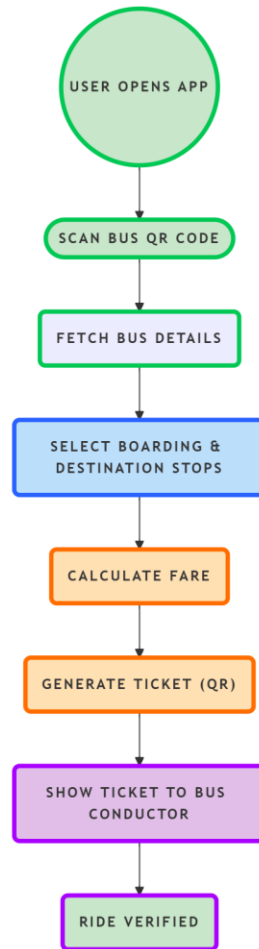
1. **User Scans QR Code:** Retrieves bus details.
2. **Selects Source & Destination:** User selects boarding and drop-off points.
3. **Fare Calculation:** System fetches the ticket price.
4. **Ticket Generation:** Generates a digital QR-based ticket.
5. **Verification on Boarding:** Conductor scans the ticket QR for validation.

5. Representation Of Diagrams

Sequence Diagram:



Data Flow Diagram :



5. Implementation Plan

Phase 1 (Basic Functionality)

- Set up React Native environment with Expo.
- Implement QR code scanning.
- Develop API to fetch bus details and fare calculation.
- Generate a basic digital ticket.

Phase 2 (Enhancements)

- Integrate payment gateways.
- Implement a user-friendly UI/UX.
- Add ticket history and admin panel.

Phase 3 (Final Testing & Deployment)

- Conduct end-to-end testing.
- Deploy backend on a cloud service (Heroku/AWS).
- Release the app on Play Store.

6. Challenges & Solutions

Challenges	Potential Solutions
QR Scanner Issues	Use Expo's barcode scanner for compatibility.
Offline Ticketing	Implement an offline mode for ticket storage.
Security Concerns	Encrypt ticket data and use secure API authentication.

7. Conclusion

This RTC Bus Ticket Booking App aims to digitize bus ticketing, improve efficiency, and enhance user experience. With future upgrades, it can serve as a model for smart public transportation systems.