Project Plan

[VOY – Ride Sharing App]

Contents

1. Introduction

The objective of this document is to provide a detailed plan for developing the "Voy" rideshare app, including specific methods and technologies to be used for implementing the features.

2. System Design

The system design phase involves creating a detailed architecture for both the frontend (mobile app) and backend (server), including database design and API specifications.

3. Scope of Work

3.1. App Architecture

- Frontend: Use React Native for developing a cross-platform mobile app for both Android and iOS.
- Backend: Use Node.js for server-side logic, and MongoDB for the database.

3.2. UI/UX Design

- Create detailed wireframe and user flows using tools like Figma.
- Ensure a consistent and intuitive user experience.

3.3. Database Design

- Design schemas for users, rides, chat messages, vehicle details, etc.
- Ensure relationships and indexing for optimal performance.

3.4. API Specifications

- Define RESTful APIs for user registration, authentication, profile management, ride requests, selection, chat, notifications, ratings, and bidding.

4. Development

Development will be divided into frontend and backend components, with regular integration points to ensure seamless functionality.

Technologies: Node.js with express.js, MongoDB, Mongoose (for MongoDB ORM), JWT (for authentication), WebSocket (for real-time communication).

Tasks to be accomplished :

4.1. User Registration and Authentication

- Implement API endpoints for user registration, login, and verification.
- Store encrypted passwords and secure token generation using JWT.

4.2. Profile Management

- Develop API endpoints for viewing and updating user profiles.
- Ensure secure storage of payment details using encryption.

4.3. Ride Request and Acceptance

- Create endpoints for ride requests and driver acceptance.
- Implement logic for finding nearest drivers and updating ride status.

4.4. Seat Selection and Availability

- Develop endpoints for seat selection and real-time availability updates.

4.5. Vehicle Details

- Implement endpoints for uploading and fetching vehicle details.

4.6. In-App Chat

- Develop WebSocket server for handling real-time chat messages.
- Ensure message history is stored securely.

4.7. Notifications

- Implement push notification service like Firebase Cloud Messaging (FCM) or Amazon SNS.

4.8. Ride Ratings

- Create endpoints for submitting and retrieving ride ratings.
- Booking for Known Individuals: Implement API for booking rides for saved contacts.

4.9. Ride Request and Bidding

- Develop endpoints for submitting ride requests and handling driver bids.
- Implement logic for calculating fare based on distance and fare range.

5. Testing

We will conduct rigorous testing to ensure the app meets all functional and non-functional requirements.

5.1. Unit Testing

- Write unit tests for individual components and functions using Jest.

5.2. Integration Testing

- Test interactions between frontend and backend components.
- Ensure real-time features work seamlessly.

5.3. User Acceptance Testing

- Conduct testing sessions with a group of users to gather feedback.
- Address any identified issues or bugs.

5.4. Performance Testing

- Test app performance under different conditions.
- Ensure the app loads within specified times and handles concurrent users efficiently.

6. Deployment

- The final stage involves deploying the app to production and making it available to users.

6.1. Server Setup

- Set up server infrastructure on a cloud platform (e.g., AWS, Azure).
- Deploy backend services and database.

6.2. App Publishing

- Publish the mobile app on Google Play Store and Apple App Store.
- Ensure compliance with app store guidelines.

6.3. Documentation

- Provide detailed documentation for potential collaborators and users.

6.4. Monitoring and Support

- Monitor app performance and user feedback.
- Provide ongoing support and maintenance.

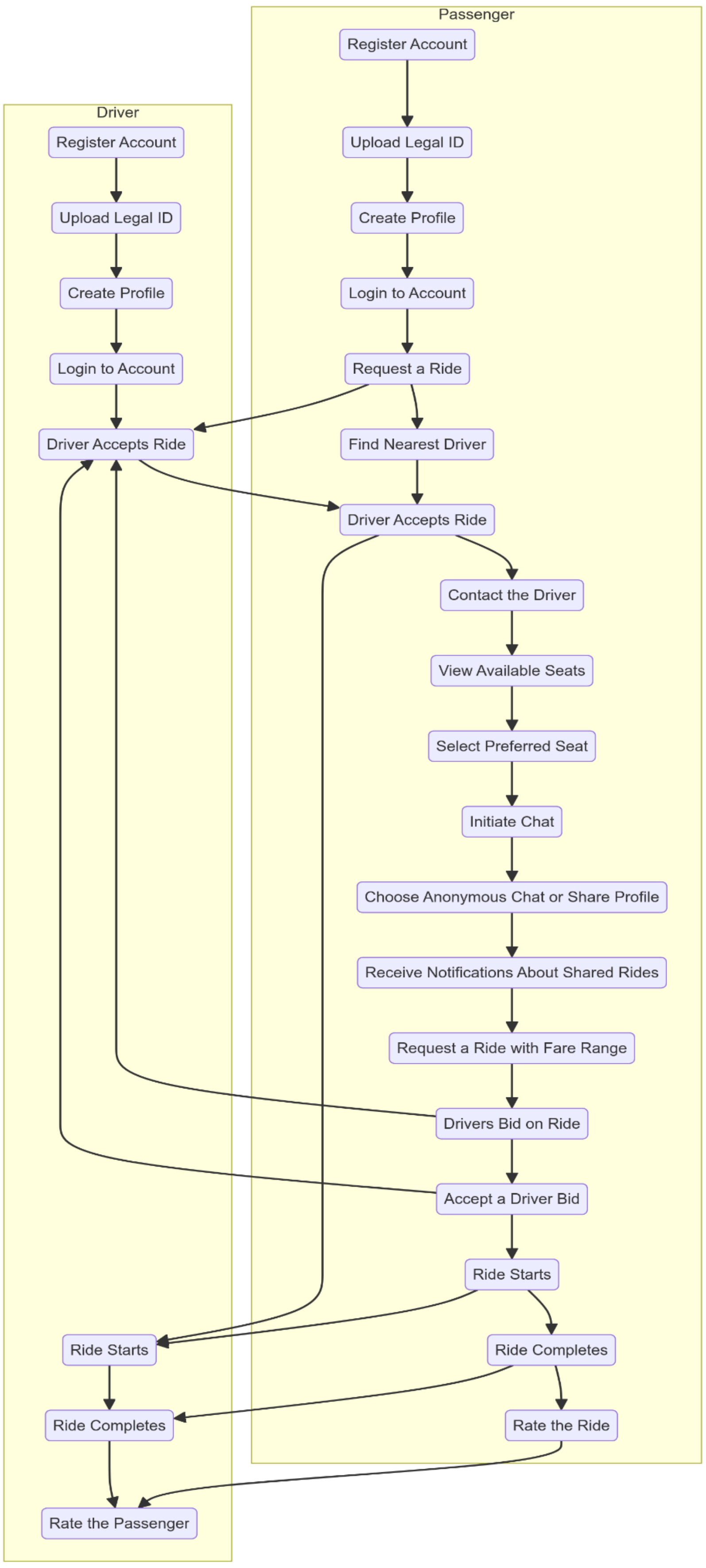
7. Maintenance and Updates

- Post-deployment, we will provide regular updates to fix bugs and improve performance.

8. Timeline

Task	**Days**
Code Setup	3-4 Days
UI/UX Design	10 – 15 Days
App & API Development	15 – 20 Days
Third Party Integrations	7-10 Days
Deployment	4-5 Days
Testing and Final Delivery	10-15 Days
Total : 40-50 Days	

9. Component Interaction


Passenger
Register Account
Upload Legal ID
Create Profile
Login to Account
Request a Ride
Find Nearest Driver
Driver Accepts Ride
Contact the Driver
View Available Seats
Select Preferred Seat
Initiate Chat
Choose Anonymous Chat or Share Profile
Receive Notifications About Shared Rides
Request a Ride with Fare Range
Drivers Bid on Ride
Accept a Driver Bid
Ride Starts
Ride Completes
Rate the Ride
Driver
Register Account
Upload Legal ID
Create Profile
Login to Account
Driver Accepts Ride
Ride Starts
Ride Completes
Rate the Passenger