Project Documentation for Centralized Taxi App

Project Overview

The objective of this project is to develop a centralized taxi service application. The app will allow taxi owners to list their taxis, manage bookings, and track revenues. It will also provide users with the ability to book taxis, view live locations, and get estimated arrival times. The project will be built using Flutter.

Key Features

1. **Taxi Listing**: Taxi owners can list their taxis on the app.

2. **Owner Dashboard**: A dashboard for each taxi owner to view total revenue.

3. **Premium Subscription**: Users can opt for a premium subscription at INR 500 per month or INR 5000 per year.

4. **Live Location Tracking**: Users can see available taxis within a 5 km radius.

5. **Booking Options**: Users can choose between multiple seat booking or booking the entire taxi.

6. **Automated Location Fetching**: The app fetches user location automatically upon booking confirmation.

7. **ETA Display**: The app shows the estimated time of arrival for the taxi.

Requirements

Functional Requirements

1. **User Registration and Login**

   - Users (both passengers and taxi owners) should be able to register and log in.

2. **Taxi Listing**

- Taxi owners can list their taxis with details such as vehicle number, type, capacity, and contact information.

3. **Owner Dashboard**

- Dashboard for taxi owners to view total revenue, manage bookings, and update taxi details.

4. **Subscription Management**

- Users can subscribe to premium features.

- Payment gateway integration for managing subscription payments.

5. **Live Location Tracking**

- Integration with GPS to track live location of taxis.

- Display available taxis within a 5 km radius from the user.

6. **Booking System**

- Option for users to book multiple seats or the entire taxi.

- Notification system for booking requests and confirmations.

7. **Automated Location Fetching and ETA**

- Fetch user location automatically upon booking confirmation.

- Calculate and display the estimated time of arrival (ETA) for the taxi.

Non-Functional Requirements

1. **Scalability**: The app should be scalable to handle a growing number of users and taxis.

2. **Performance**: The app should be responsive and provide real-time updates.

3. **Security**: User data and payment information should be securely stored and processed.

4. **Usability**: The app should have an intuitive and user-friendly interface.

Technology Stack

- **Frontend**: Flutter
- **Backend**: Node.js/Express.js
- **Database**: MongoDB
- **Payment Gateway**: Razorpay/Stripe
- **Location Services**: Google Maps API
- **Authentication**: Firebase Auth

Application Architecture

1. User Flow

1. **Registration/Login**
   - User registers or logs in.
2. **Profile Setup**
   - Taxi owners set up their profile and list their taxis.
3. **Dashboard**
   - Taxi owners view their dashboard to manage listings and view revenues.
4. **Subscription**
   - Users subscribe to premium features.
5. **Taxi Search**
   - Users search for available taxis within a 5 km radius.
6. **Booking**

- Users book a taxi (multiple seats or entire taxi).

7. **Booking Confirmation**

   - Taxi drivers receive booking requests and confirm them.

8. **Live Tracking**

   - Users can track the live location of the booked taxi and view ETA.

2. System Components

1. **Frontend (Flutter)**

   - User Interface for registration, login, profile setup, booking, and tracking.

2. **Backend (Node.js/Express.js)**

   - RESTful API for handling user requests, taxi listings, bookings, and subscription management.

3. **Database (MongoDB)**

   - Storing user data, taxi details, booking information, and subscription status.

4. **Payment Gateway Integration (Razorpay/Stripe)**

   - Handling subscription payments securely.

5. **Location Services (Google Maps API)**

   - Fetching and displaying live locations and calculating ETA.

6. **Authentication (Firebase Auth)**

   - Secure user authentication and authorization.

Development Plan

Phase 1: Initial Setup

1. **Requirement Analysis and Documentation**
2. **Design Database Schema**
3. **Set up Flutter Project**
4. **Set up Backend with Node.js and Express.js**

Phase 2: Core Features Development

1. **User Registration and Login**
2. **Taxi Listing and Owner Dashboard**
3. **Live Location Tracking**

Phase 3: Advanced Features

1. **Subscription Management**
2. **Booking System**
3. **Automated Location Fetching and ETA Calculation**

Phase 4: Integration and Testing

1. **Integration of Frontend and Backend**
2. **Testing (Unit, Integration, and User Acceptance Testing)**

Phase 5: Deployment and Maintenance

1. **Deployment to App Stores**
2. **Monitoring and Maintenance**

Documentation

1. User Guide

- **Registration and Login**
- **Using the Owner Dashboard**
- **Listing a Taxi**
- **Subscribing to Premium Features**
- **Booking a Taxi**

2. Developer Guide

- **Setting up the Development Environment**
- **Frontend Setup and Configuration**
- **Backend Setup and Configuration**
- **Database Schema**
- **API Endpoints and Integration**

3. Testing Guide

- **Unit Testing Procedures**
- **Integration Testing Procedures**
- **User Acceptance Testing Procedures**