

A dark blue vertical bar on the left side of the page. A blue arrow points to the right from this bar, containing the date 1/4/2024.

1/4/2024

IQRA UNIVERSITY

COURSE: MAD

PROJECT: CAR POOLING
SYSTEM

SRS DOCUMENT

Several thin, curved lines in dark blue and light grey originate from the bottom of the vertical bar and sweep upwards and to the right.

Talha Ahmed

Software Requirements Specification for Carpooling System

1. Introduction

1.1 Purpose

The primary goal of this document is to outline the requirements for the development of an extensive Carpooling System mobile application. The system aims to address the increasing expenses associated with fuel and oil, coupled with the prevalent issue of traffic congestion. By providing a platform for shared rides, the Carpooling System aims to significantly reduce commuting costs for users, alleviate the financial burden of individual transportation, and contribute to the mitigation of traffic congestion. The system further offers the flexibility of negotiating fare and extends its inclusivity to two-wheelers, expanding the benefits to a broader range of commuters.

1.2 Scope

The Carpooling System is designed to present a dynamic and user-friendly interface tailored for Android devices. It caters to users seeking a practical solution to combat the rising costs of petrol and oil, as well as the challenges posed by heavy traffic on roads. The application empowers users to effortlessly register, log in, and specify their role as either a driver or a passenger. The system facilitates the creation of user profiles, encourages ride-sharing with options for negotiating fares, and extends its support to two-wheelers. By prioritizing cost-saving and traffic reduction, the Carpooling System aspires to promote sustainable and economical commuting practices.

Software Uses for Application:

- 1) UI Design Illustrator
- 2) Frontend implemented on Flutter
- 3) Database Mongo DB
- 4) Backend Node JS

2. System Overview

2.1 System Description

The Carpooling System will integrate advanced functionalities, such as user authentication, profile creation, ride-sharing with fare negotiation, and options for two-wheelers. The system will prioritize a seamless user experience and aesthetically pleasing UI/UX design.

2.2 System Features

User Registration and Authentication

Role Selection (Driver/User) Screen

Profile Creation for Drivers and Users

Driver Information Entry (Name, Contact, Email, Car Model, Car Number)

User Information Entry (Name, Contact, Email)

Location/Route Sharing via Google Maps Integration

Contact Mechanism (Phone, WhatsApp, Email)

Fare Negotiation

Support for Two-Wheelers

3. Functional Requirements

3.1 User Registration and Authentication

Users must register with a unique username, password, and basic contact information.

Authentication ensures the security of user accounts.

3.2 Role Selection Screen

After login, users are prompted to select their role as either a driver or a user.

3.3 Profile Creation for Drivers and Users

Drivers and users can create and update their profiles with personal details.

Profiles include essential information such as name, contact, and email.

3.4 Driver Information Entry

Drivers enter additional details such as car model and car number.

3.5 User Information Entry

Users enter their details excluding car information.

3.6 Location/Route Sharing

Users can share their location or route using list for route selection from list.

3.7 Contact Mechanism

Users and drivers can contact each other through phone.

.

3.8 Fare Negotiation

Users and drivers can negotiate on fare within the application.

3.9 Support for Two-Wheelers

Users and drivers with two-wheelers can register and share rides.

3.12 Use Cases for Users and Drivers

User

- |-- Register
- |-- Log In
- |-- View Profile
- |-- Edit Profile
- |-- Share Location/Route
- |-- Contact Driver
- |-- Negotiate Fare
- |-- Provide Feedback

Driver

- |-- Register
- |-- Log In
- |-- View Profile

- |-- Edit Profile
- |-- Share Car Details
- |-- Share Location/Route
- |-- Contact User
- |-- Negotiate Fare

4. Non-Functional Requirements

4.1 Performance

The system should support concurrent logins for a minimum of 1000 users.

Response time for user interactions should be below 2 seconds.

4.2 Security

User passwords must be securely stored using encryption.

Communication between the app and server must use HTTPS.

4.3 Usability

The user interface should be intuitive, adaptable, and aesthetically pleasing.

Support for multiple languages to enhance user accessibility.

4.4 UI/UX Design Recommendations

Utilize a color scheme that promotes a sense of safety and trust.

Implement clear and concise buttons and navigation for ease of use.

Ensure that the design is responsive across various screen sizes.

4.5 Error Handling and Recovery

The system should implement robust error-handling mechanisms to manage unexpected events and recover gracefully. Detailed error messages and logging should aid in diagnosing issues during troubleshooting.

5. Constraints

The application is developed for Android devices (version 7.0 and above).

Development is constrained to a six-week-timeline.

6. Testing Requirement

The testing approach should include comprehensive testing, covering unit testing, integration testing, and user acceptance testing. A detailed test plan should be developed to ensure the reliability and stability of the Carpooling System.

7. Developer Documentation

A comprehensive technical documentation guide should be provided for developers. This guide includes detailed information on the system architecture, APIs, and development guidelines to facilitate the development and maintenance of the Carpooling System.

9. Approval

This SRS document is approved by:

Name: Talha Ahmed Qureshi

Date: [4/01/2024]