Tawseela

Affordable rides

Team Members

Ahmed Abdul Aziz – Ahmed Khames

Youssef Altohamy – Mohamed Ibrahim

Mostafa Masoud – Mahmoud Mohamed

Ahmed Sobhi

Software Requirements Specification (SRS) for Tawseela

1. Introduction

1.1 Purpose

The purpose of this document is to specify the software requirements for the Tawseela application, a transportation app tailored specifically to support university students in Egypt. The app enables long-distance trips between governorates and short trips within cities.

1.2 Scope

This document is intended for the development team, project managers, and any other stakeholders interested in analyzing the system requirements.

Tawseela aims to provide a convenient and secure transportation solution for university students.

The system includes:

- User app for requesting and tracking rides.
- Driver app for receiving ride requests and managing ride status.
- Admin web dashboard for monitoring system activity.
- Firebase-based backend to store and update data in real-time.
- Google Maps API for location and route identification.

1.3 Overview

Tawseela is a transportation platform designed specifically to meet the needs of university students, providing a reliable and affordable solution for both short and long-distance rides. The app bridges the gap between students and drivers, ensuring a secure and user-friendly experience.

The project focuses on making travel easier for students who need to commute between Egyptian governorates or within cities. With its intuitive interface and real-time updates, Tawseela aims to become a trusted companion for students, helping them save time, money, and effort while ensuring their safety.

1.4 Business Context

Tawseela addresses a critical transportation gap faced by university students in Egypt, particularly those who live far from their campuses. Traditional public transportation often lacks reliability, safety, and convenience, creating a need for a service that caters specifically to this demographic.

The platform leverages technology to provide a smart transportation solution, offering:

- Convenience: Allowing users to request rides through a mobile app, track their trips in real-time, and manage bookings effortlessly.
- Safety: Ensuring secure rides by implementing verification processes for drivers and monitoring trips through a centralized system.
- Affordability: Offering cost-effective ride-sharing options tailored to students' budgets.

By integrating separate apps for users and drivers, along with an admin dashboard for operational oversight, Tawseela ensures a seamless experience for all stakeholders. With the

growing demand for reliable transportation solutions, Tawseela positions itself as a vital service provider in the student transportation sector.

The platform also holds potential for future scalability, with opportunities to expand services to other demographics, introduce new features, and enter additional markets across Egypt and beyond.

2. General Description

2.1 Product Perspective

Tawseela provides transportation services specifically for university students, offering features like:

- Ride requests for both long-distance and short trips.
- Real-time notifications to track ride status.
- Proximity-based driver matching to connect users with the nearest available drivers.
- Admin dashboard for monitoring rides and managing drivers.

2.2 Similar System Information

Tawseela operates in a market where multiple transportation systems exist, such as Uber and Careem, but it stands out by focusing on university students' specific needs. Unlike general ride-hailing apps, Tawseela offers tailored features such as shared rides for students traveling between cities, cost-effective pricing, and safety measures designed specifically for young adults. By addressing a niche market, Tawseela fills the gap left by larger competitors, providing a more personalized and efficient solution for student transportation.

2.3 User Characteristics

The primary users of Tawseela are university students aged between 18 and 25 years. They are tech-savvy, comfortable using mobile applications, and rely heavily on digital solutions for their daily needs. Most users seek affordable, reliable, and safe transportation options, especially for long-distance travel between governorates. Secondary users include drivers who are typically licensed professionals familiar with ride-sharing systems, and administrators managing the system's operations through a web dashboard.

2.4 User Problem Statement

University students often struggle with unreliable and costly transportation options, especially when commuting between cities for academic purposes. Public transport systems are frequently overcrowded, time-consuming, and lack proper safety measures. Additionally, existing ride-hailing services may not cater to student budgets or preferences, making them less accessible for daily use. These challenges highlight the need for a transportation system designed to address the unique requirements of students.

2.5 User Objectives

Tawseela aims to provide users with a seamless, affordable, and safe transportation solution that fits their specific needs. Students should be able to:
Quickly book rides using an intuitive mobile app.
Share rides with fellow students to reduce costs.
Travel safely with verified drivers and real-time trip tracking.

Access an efficient, reliable service for both short and long-distance journeys.
For drivers, the system ensures a steady flow of ride requests, efficient trip management, and transparent earnings tracking. Administrators benefit from centralized system management to ensure smooth operations.

2.6 System Features

- Ride Request: Allows users to request a new ride by specifying the destination.
- Nearby Driver Search: Uses GPS to locate available nearby drivers.
- User and Driver Management: Admin panel allows monitoring of activity and management of user and driver information.
- Ride Tracking: Track the current status of a ride from within the app. - Notifications: Alerts users and drivers on ride updates.

2.7 System Constraints

-The app must comply with intercity transportation regulations, including driver and vehicle permits.
-Emphasis on driver verification, secure payment methods, and real-time tracking to ensure a safe experience for students, especially those traveling long distances.
-Student commuting patterns may lead to high demand during weekends and holidays, requiring a scalable solution.
-Since affordability is crucial for students, pricing models should be competitive, with potential for discounts or partnerships with educational institutions.

3. System Documentation

3.1 *System Architecture*

The Tawseela app follows a multi-tier architecture to ensure scalability, maintainability, and security. The system includes three primary layers:

1. Frontend Layer (Client Side):

   This layer represents the interface through which users and drivers interact with the system. It includes:

   - **User Mobile Application:**
     Built specifically for students, this app enables users to book rides, track trips, communicate with drivers, and manage their ride history. The app will be developed using **Flutter** to ensure a seamless, cross-platform experience for both Android and iOS users.
   - **Driver Mobile Application:**
     A dedicated app for drivers to manage ride requests, update their availability, track trip status, and view earnings. Like the user app, it will be developed with **Flutter**, leveraging its efficiency in building interactive and responsive UIs.
   - **Admin Web Dashboard:**
     A web-based tool for administrators to oversee system operations, manage user

accounts, handle trip requests, monitor ride analytics, and generate reports. This dashboard will be developed using modern web technologies

2. Backend Layer (Server Side):

   The backend layer is responsible for handling all server-side processes, ensuring real-time communication and data synchronization. Key components include:

   - **Firebase:**
     1. Firebase Authentication: Manages secure user and driver authentication, including signup, login, and account management.
     2. Cloud Firestore: Serves as the primary NoSQL database for storing and managing data like user profiles, driver information, ride requests, and trip details.
     3. Firebase Cloud Functions: Used for processing backend logic such as calculating nearest drivers or sending notifications.
     4. Firebase Realtime Database: Handles real-time data like ride status, trip progress, and driver locations, ensuring updates are instantaneous for all stakeholders.

3. Mapping and GPS Layer:

   This layer facilitates location-based services and navigation, ensuring accurate trip tracking and efficient routing:

   Google Maps API:
   1. Provides real-time GPS tracking for users and drivers.
   2. Enables route optimization for quicker trips.
   3. Displays nearby available drivers to users and navigational routes to drivers for efficient trip execution.

3.2 *Components*

1. User Mobile App (Frontend):
   - Features: Users can register, view available drivers, request rides, track rides, rate drivers, and make payments.
   - Technologies Used: Flutter for mobile development, Firebase for authentication and real-time updates, Google Maps API for route tracking.
2. Driver Mobile App (Frontend):
   - Features: Drivers can register, set their availability, accept or reject ride requests, track trips, and earn rewards.
   - Technologies Used: Flutter for mobile development, Firebase for authentication and real-time updates, Google Maps API for navigation.
3. Admin Web Dashboard (Frontend):
   - Features: Admin can monitor users and drivers, view ride history, manage data, generate reports, and handle customer support. o Technologies Used: React or Angular for front-end development, Firebase for backend, and Google Maps API for location-based services.

4. Backend (Server):
    - Features: Firebase serves as the backend to store all app data, handle user authentication, and deliver real-time updates.
    - Technologies Used: Firebase Firestore, Firebase Cloud Functions, Firebase Realtime Database.

3.3 *Data Flow*

1. Ride Request Process:
    - A user initiates a ride request from the mobile app, providing their current location and destination.
    - The system identifies nearby available drivers using location data (via Google Maps API).
    - Drivers are notified of the request. If a driver accepts, the status is updated in real-time using Firebase Realtime Database.
    - The user and driver track the trip in real-time via Google Maps, with status updates stored in Firestore.
2. Booking Confirmation & Notifications:
    - Once a ride is confirmed, both the user and the driver receive real-time notifications.
    - The system sends push notifications for trip updates (ride accepted, driver on the way, trip completed, etc.).
3. Payment Process:
    - After completing the ride, the user can make a payment (via integrated payment gateways).
    - Payment status and transaction details are stored in Firestore and sent to the admin dashboard.

3.4 *Database Design*

The database design for Tawseela is structured around multiple collections in Firestore to handle various entities:

1. Users Collection:
    - Fields: User ID, name, email, phone number, governorate , ride history, payment methods.
    - Purpose: Stores personal information about users and their interaction history with the app.
2. Drivers Collection:
    - Fields: Driver ID, name, email, phone number, car details (type,color, license plate),status, rating.

- Purpose: Stores driver-related data including their status (available or not), vehicle details, and completed trips.

3. Ride Requests Collection:
    - Fields: Ride ID, user ID, driver ID, pickup location, destination location, status (pending, accepted, completed, canceled), ride duration, payment status.
    - Purpose: Tracks all the ride requests made by users, including the status and associated driver details.
4. Ratings Collection:
    - Fields: Rating ID, ride ID, user ID, driver ID, rating value, feedback comment.
    - Purpose: Stores ratings provided by users for drivers after completing a ride.
5. Transactions Collection:
    - Fields: Transaction ID, user ID, ride ID, payment amount, payment status, transaction date.
    - Purpose: Stores payment data related to completed rides.

3.5 *Security Considerations*

1. Authentication: Firebase Authentication handles user and driver login processes. Authentication mechanisms like email/password, Google sign-in, and phone number verification are supported.
2. Authorization: Access control policies ensure users and drivers can only access relevant data (e.g., users can't access other users' ride data).
3. Data Privacy: All sensitive data, including payment information and personal details, is encrypted and stored securely.

4.Functional Requirements

Client:

1. Create an account with personal and contact details.
2. View and update personal information and payment methods.
3. Enter pick-up and drop-off locations.
4. Track the driver's location in real-time.
5. Choose and confirm a payment method.
6. View history of previous rides and their details.
7. Receive notifications for ride confirmations, arrivals, and payments.
8. Cancel a ride before the driver starts the trip.

Driver:

1. Create an account and provide required documents like license and vehicle information.

2. Update personal information, vehicle details, and contact information.
3. Receive notifications for new ride requests.
4. View ride details, such as pickup/drop-off points and fare estimates, and accept or reject requests.
5. Update ride status (On the way, Picked Up, Completed) to keep the user informed.
6. Communicate with the user through calling in case of any issues during the ride.
7. View daily and weekly earnings and receive notifications for completed payments.

Admin:

1. View, approve, edit, and deactivate user and driver accounts.
2. Verify driver documents and approve only verified drivers.
3. View payment history and transaction records.
4. Review and manage user and driver feedback, ratings, and complaints.
5. Address issues reported by students or drivers and follow up as needed.
6. Monitor driver earnings and handle payment disputes.
7. Send system notifications to users and drivers about updates, promotions, or alerts.

5. Non-Functional Requirements

5.1 Scalability

- The system must scale to handle high demand during peak hours, weekends, and holidays when student travel increases.
- The architecture should support the addition of more cities or universities without impacting existing performance.

5.2 Usability

- Design a simple, intuitive interface that is easy for students and drivers to navigate, with minimal training required.
- Ensure a user-friendly admin panel for managing operations efficiently.

5.3 Maintainability

- Code should be modular and well-documented, allowing for easy updates and feature additions.
- Perform regular system maintenance with minimal downtime for bug fixes and enhancements.

5.4 Data Integrity

- Ensure all rides, payments, and user actions are accurately recorded without data loss.
- Implement checks to prevent data duplication or corruption in transactions and ride histories.

5.5 Logging and Monitoring

- All significant events should be logged to aid in error tracking.

5.6 Performance Requirements

Response Time: Ride status update notifications should be near-instantaneous (within 3 seconds).

Error Rate: Database queries should have a failure rate of less than 1%.

5.7 Security Requirements

Authentication: Firebase Authentication will be used to verify the identity of users and drivers.

Authorization: Users and drivers should have restricted access to authorized data only. Data Encryption: All data must be encrypted during transfer via HTTPS.

5.8 Compatibility Requirements

Platform Compatibility: The system should work on Android and iOS.

Browser Compatibility: The admin dashboard should be compatible with the latest versions of Chrome, Firefox, and Safari.

5.9 User Interface Requirements

User App Interface: A simple, intuitive interface for ride requests and status tracking.

Driver App Interface: Includes ride details and allows drivers to update ride status. Admin Dashboard: Provides an overview of active rides, users, and drivers.

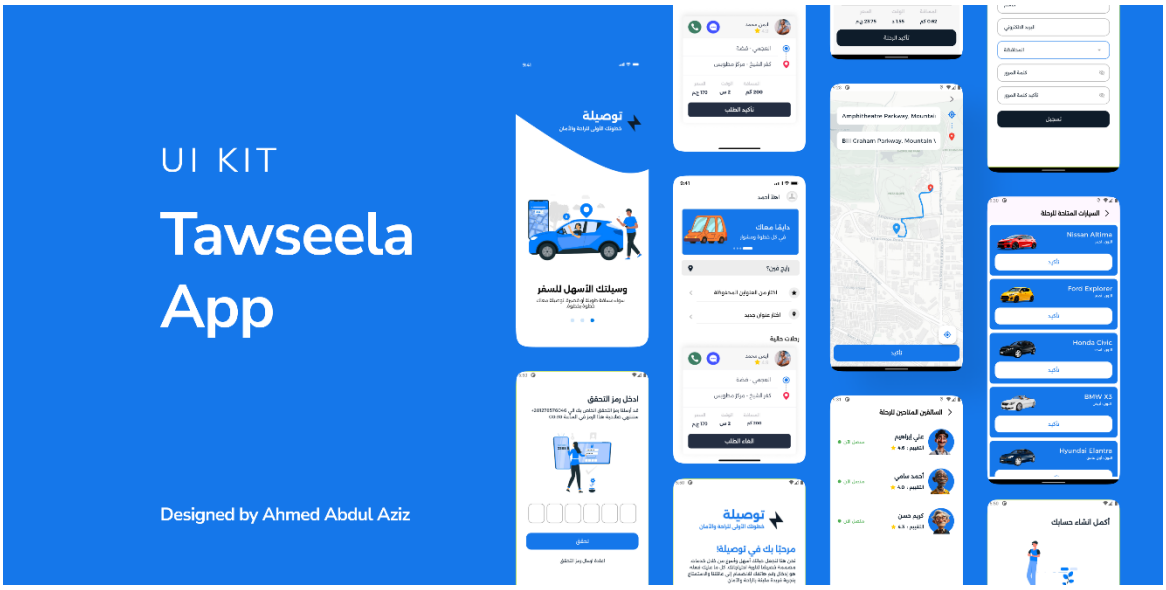
6. Requirements Traceability

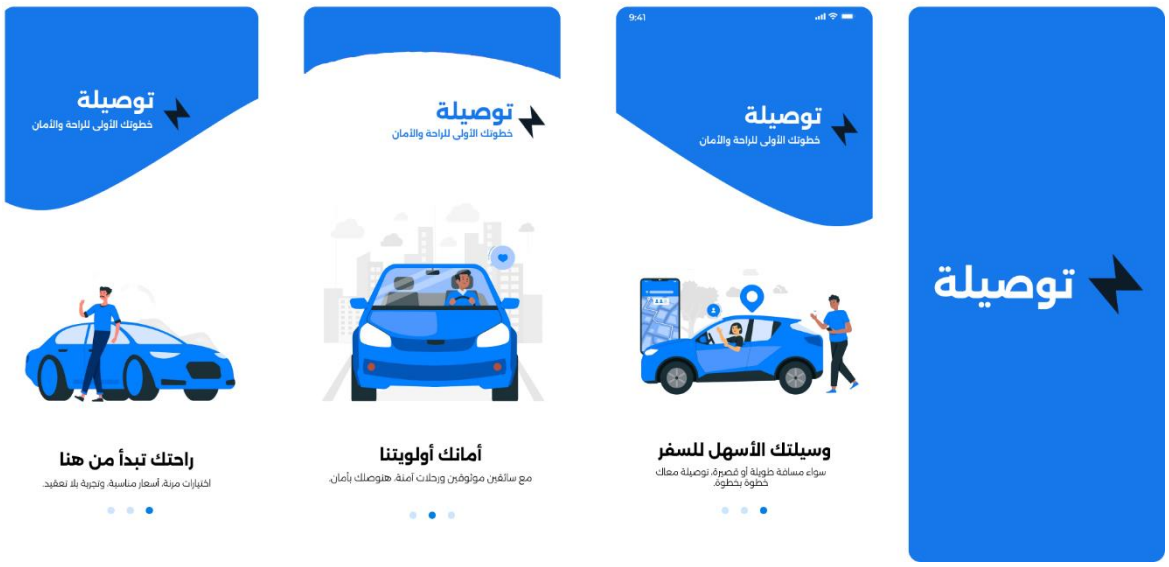
This section ensures that all requirements are traceable and linked to specific functionality.

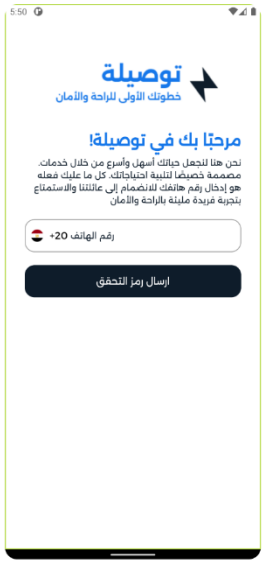
6.1 Traceability Table

Requirement ID	Feature	Priority	Status
1	User Management	High	In Progress
2	Ride Request	High	In Progress
3	Ride Notifications	Medium	Planning Phase
4	Admin Dashboard	High	Planning Phase

7. User Interface Design



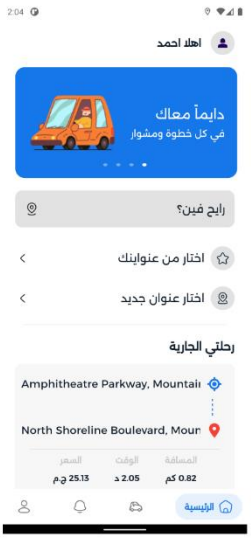


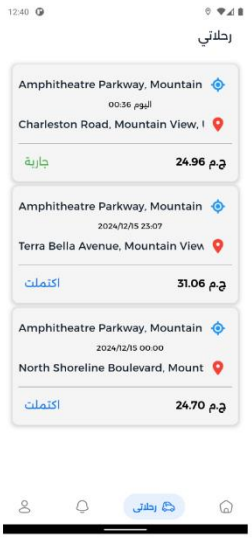

توصيلة
خطوتك الأولى للراحة والأمان
مرحبًا بك في توصيلة!
نحن هنا لنجعل حياتك أسهل وأسرع من خلال خدمات مصممة خصيصًا لتلبية احتياجاتك. كل ما عليك فعله هو إدخال رقم هاتفك للانضمام إلى عائلتنا والاستمتاع بتجربة فريدة مليئة بالراحة والأمان
رقم الهاتف +20
ارسال رمز التحقق

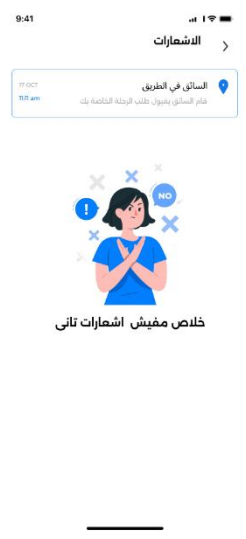

ادخل رمز التحقق
قد أرسلنا رمز التحقق الخاص بك الى +201278576046
ستنتهي صلاحية هذا الرمز في الساعة 00:30
تحقق
اعادة ارسال رمز التحقق


أكمل انشاء حسابك
الاسم
البريد الالكتروني
المحافظة
كلمة المرور
تأكيد كلمة المرور
تسجيل

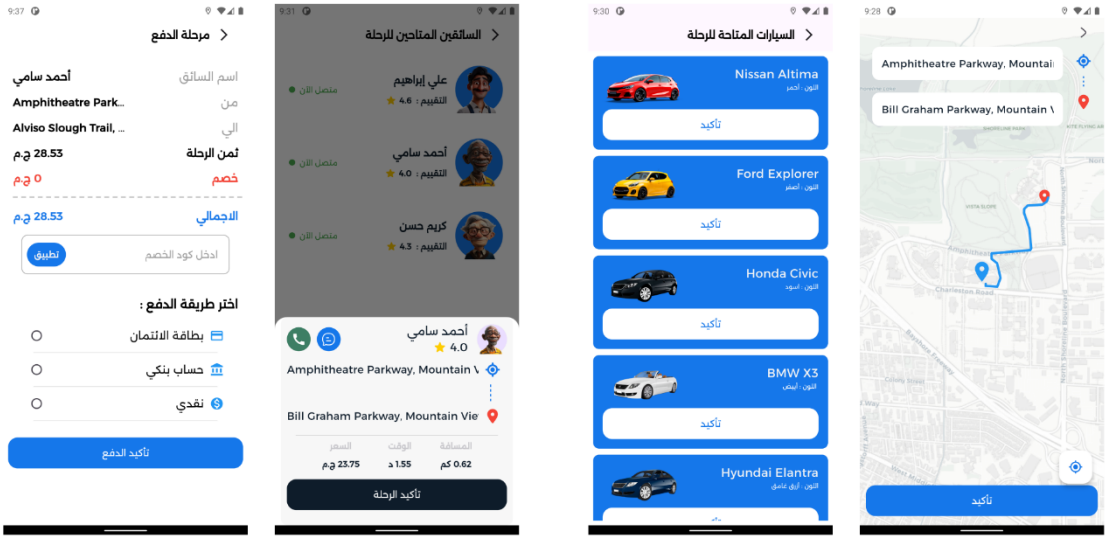

مرحبًا بك مرة أخرى!
قم بتسجيل الدخول للمتابعة واستكشاف خدماتنا
البريد الالكتروني
كلمة المرور
تسجيل الدخول
انشاء حساب


اهلا احمد
دايماً معاك
في كل خطوة ومشوار
رايح فين؟
اختار من عناوينك
اختار عنوان جديد
رحلتي الجارية
Amphitheatre Parkway, Mountai
North Shoreline Boulevard, Mour
المسافة 0.82 كم
الوقت 2.05 د
السعر 25.13 ج.م
الرئيسية

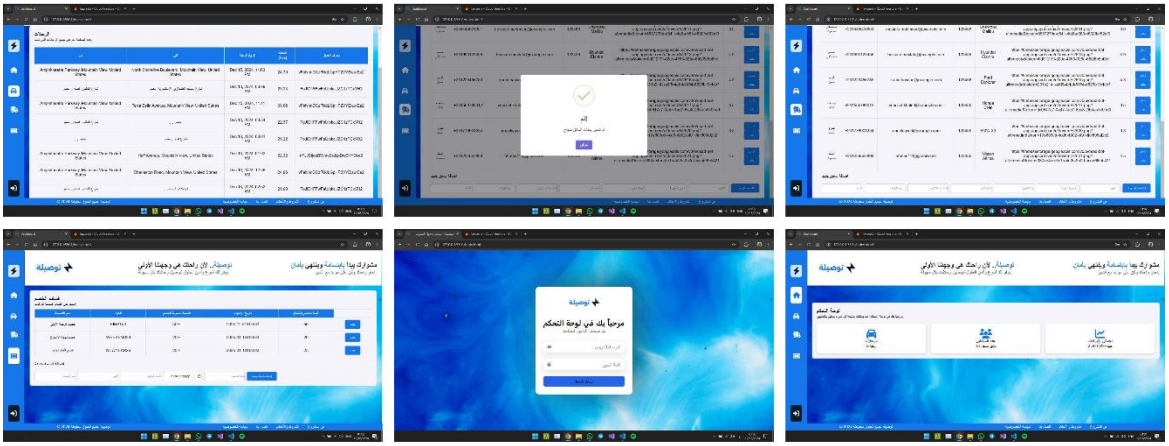

رحلاتي
Amphitheatre Parkway, Mountain
اليوم 00:36
Charleston Road, Mountain View, I
جارية
24.96 ج.م
Amphitheatre Parkway, Mountain
2024/12/15 23:07
Terra Bella Avenue, Mountain View
اكتملت
31.06 ج.م
Amphitheatre Parkway, Mountain
2024/12/15 00:00
North Shoreline Boulevard, Mount
اكتملت
24.70 ج.م
رحلاتي


الاشعارات
السائق في الطريق
قام السائق بقبول طلب الرحلة الخاصة بك
خلاص مفيش اشعارات تانى

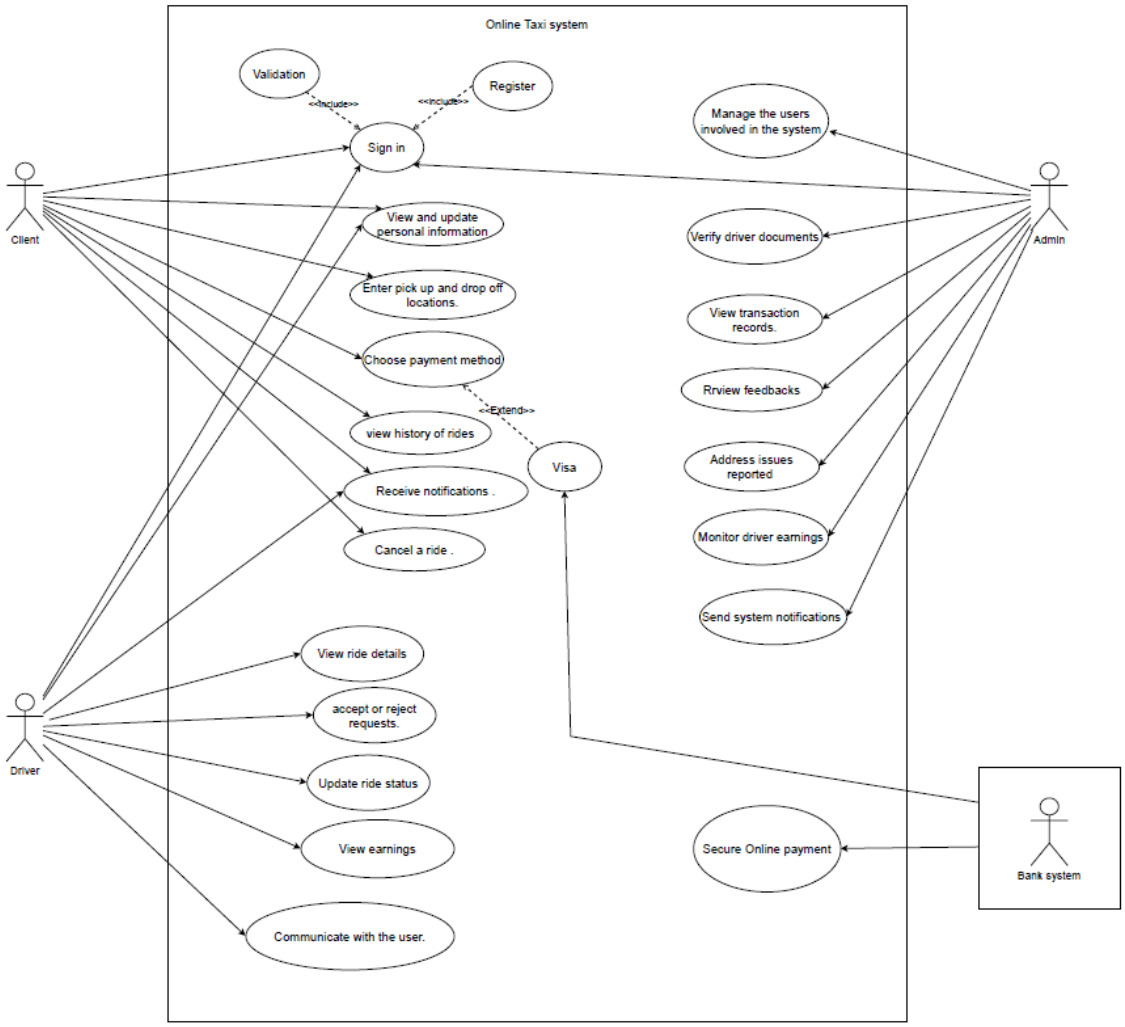

ملفي
احمد عبدالعزيز
01278576046
تعديل الملف
العناوين
الفساتم
العروض
عن التطبيق
تسجيل خروج
حذف الحساب
ملفي



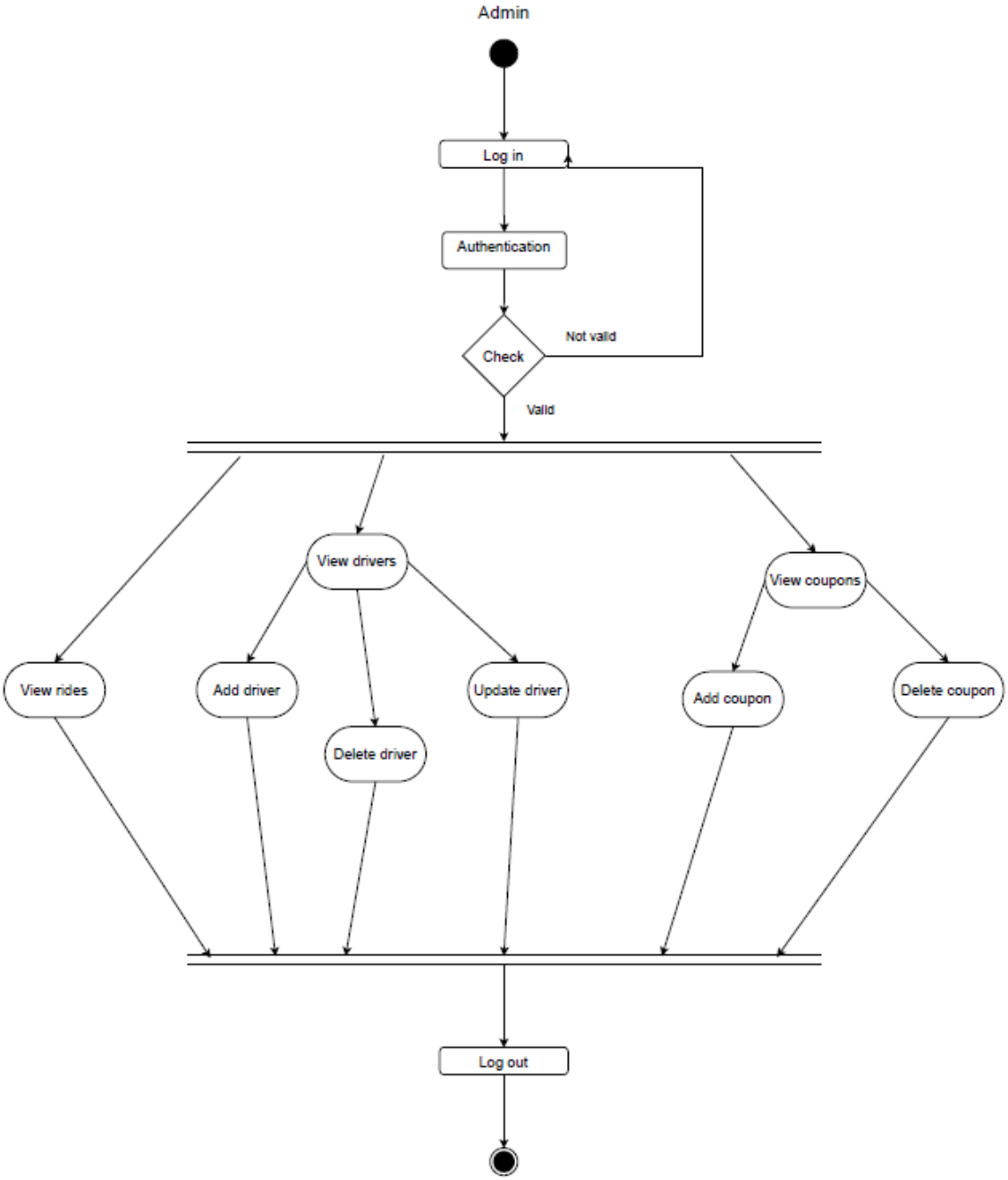
Admin Dashboard:

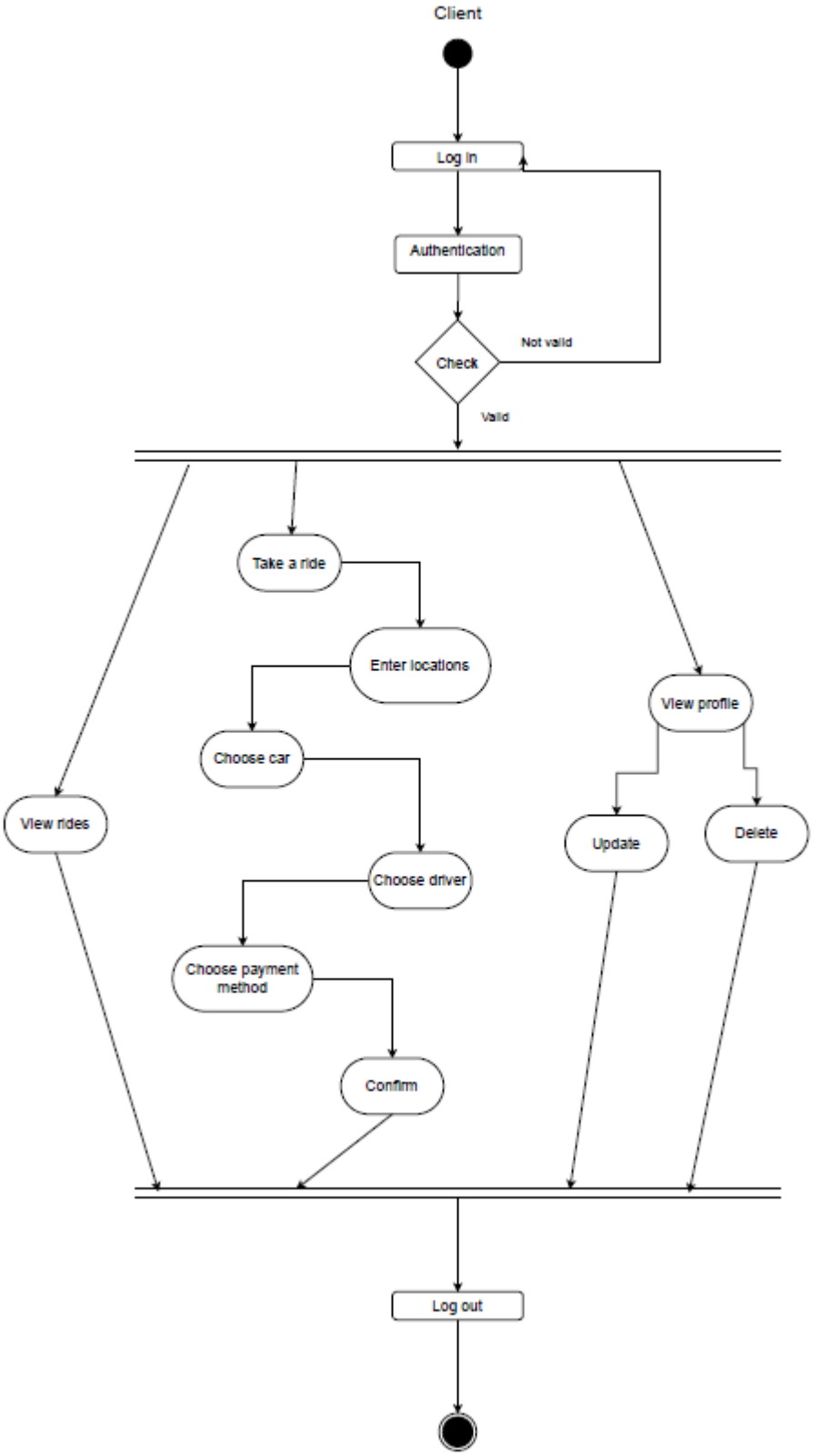


8.Use Case Diagram

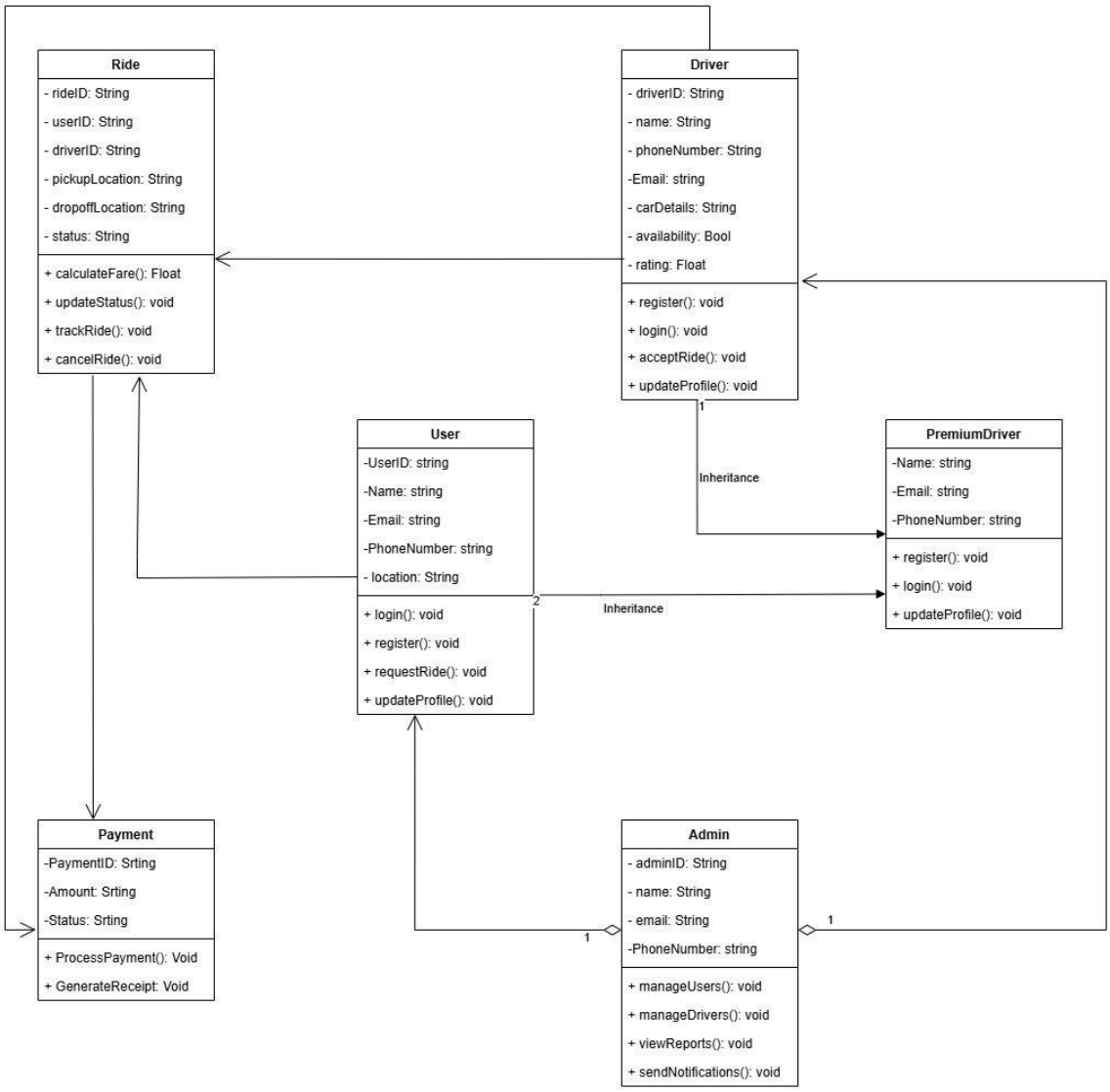


9.Activity Diagram

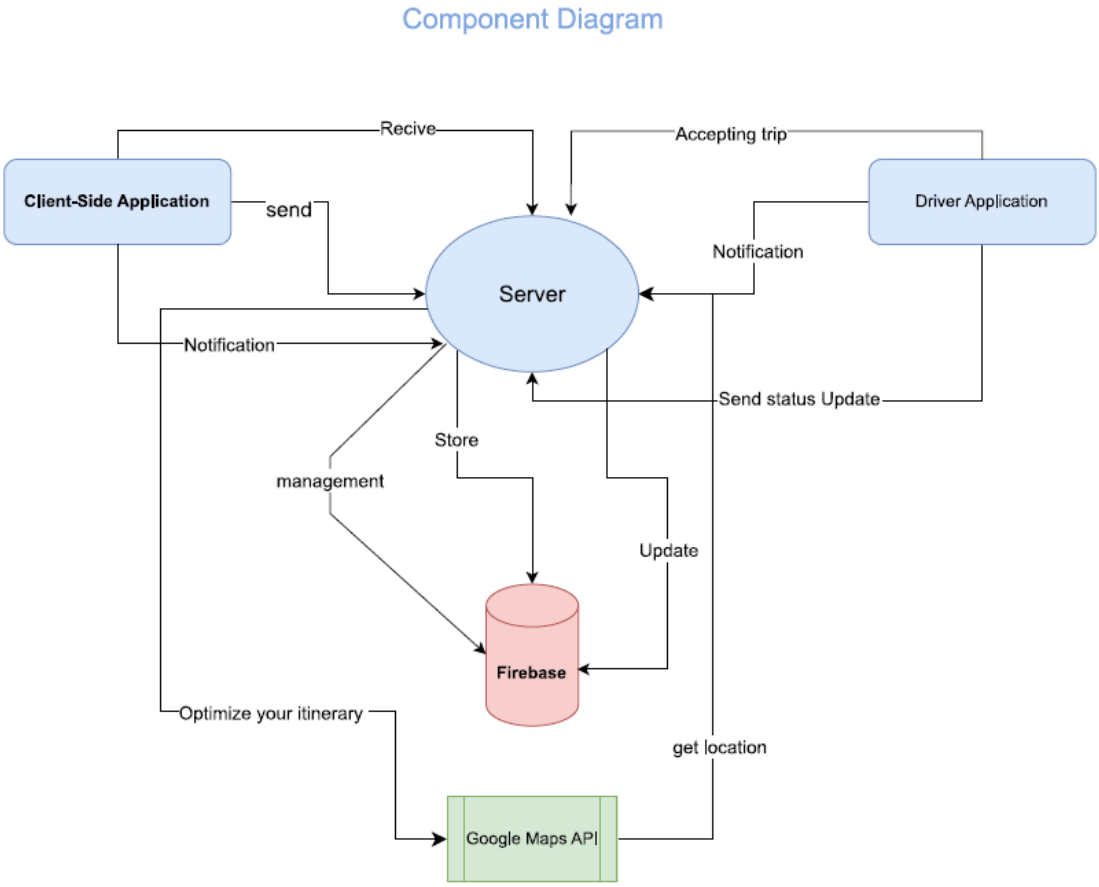



Client
Log in
Authentication
Check
Not valid
Valid
Take a ride
Enter locations
Choose car
Choose driver
Choose payment method
Confirm
View rides
View profile
Update
Delete
Log out

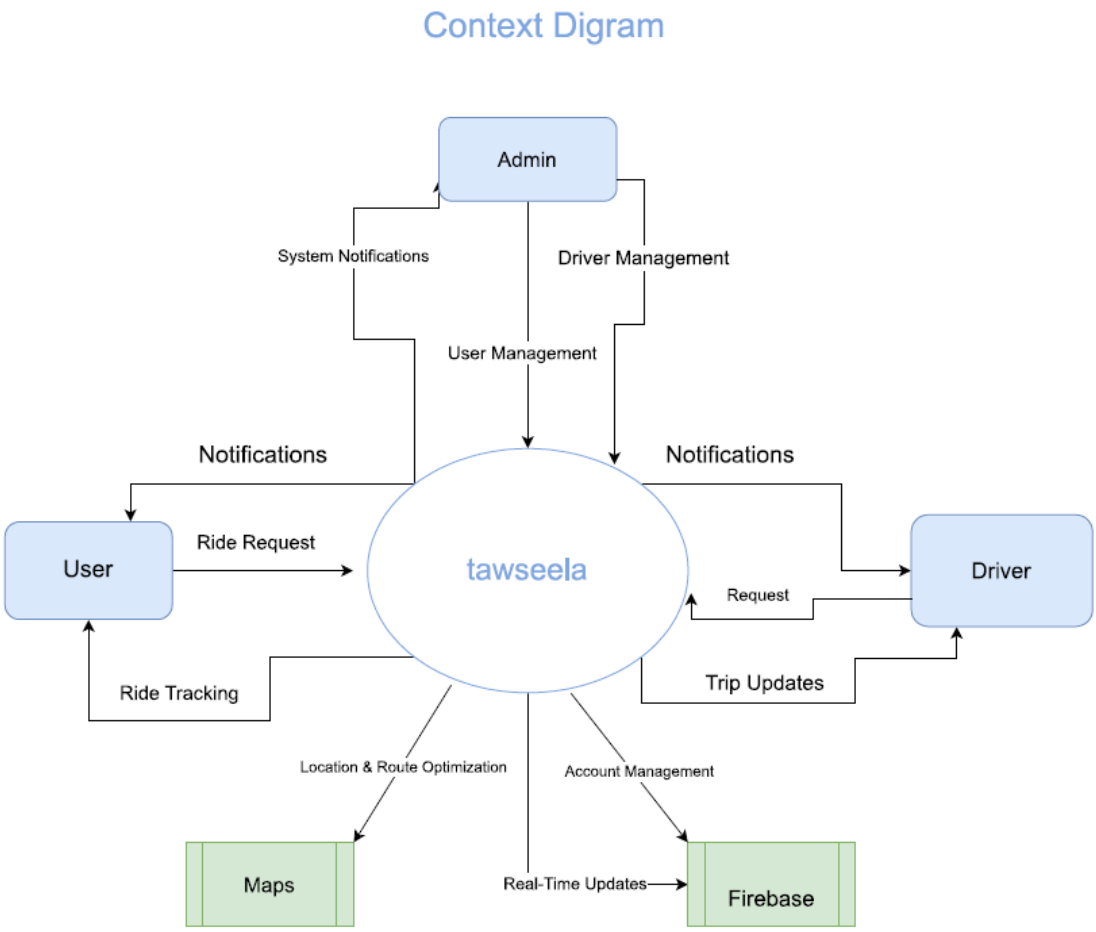
10.Class Diagram



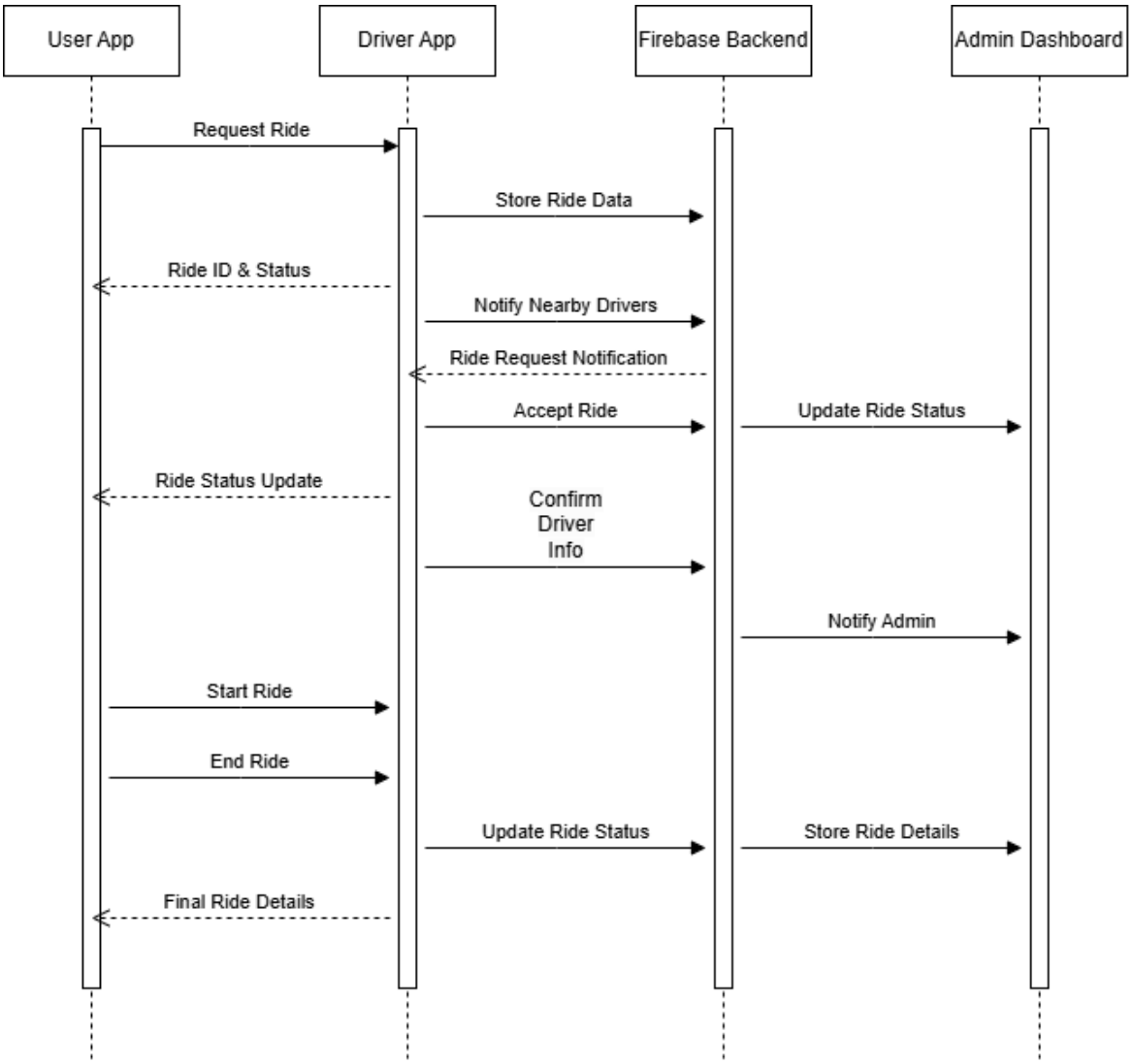
11.Component Diagram



12.Context Diagram



13.Sequence Diagram



14.GitHub

App repo :

https://github.com/interesta22/Tawseela-App.git

Admin Dashboard repo :

https://github.com/Mostafamasoud55/Tawsela