



Mini Project Report On

RideShare

*Submitted in partial fulfillment of the requirements for the
award of the degree of*

Bachelor of Technology

in

Computer Science & Engineering

By

Dhiraj Bobby (U2103076)

Jerin Vincent (U2103112)

John Kurian (U2103116)

Kalidas Jayakumar (U2103122)

Under the guidance of

Dr. Saritha S

**Department of Computer Science & Engineering
Rajagiri School of Engineering & Technology (Autonomous)
(Affiliated to APJ Abdul Kalam Technological University)**

Rajagiri Valley, Kakkanad, Kochi, 682039

May 2024

CERTIFICATE

*This is to certify that the mini project report entitled "**RideShare**" is a bonafide record of the work done by **Dhiraj Bobby (U2103076)**, **Jerin Vincent (U2103112)**, **John Kurian (U2103116)**, **Kalidas Jayakumar (U2103122)**, submitted to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology (B. Tech.) in Computer Science and Engineering during the academic year 2023-2024.*

Dr. Saritha S
Professor
Dept. of CSE
RSET

Dr. Saritha S
Professor
Dept. of CSE
RSET

Dr. Preetha K G
Head of the Department
Professor
Dept. of CSE
RSET

ACKNOWLEDGEMENTS

I wish to express my sincere gratitude towards Dr P. S. Sreejith, Principal of RSET, and Dr. Preetha K.G., Head of the Department of Computer Science and Engineering for providing me with the opportunity to undertake my mini project, "RideShare".

I am highly indebted to my project coordinators, **Dr. Saritha S**, Professor, Department of Computer Science and Engineering for their valuable support.

It is indeed my pleasure and a moment of satisfaction for me to express my sincere gratitude to my project guide **Dr. Saritha S** for her patience and all the priceless advice and wisdom she has shared with me.

Last but not the least, I would like to express my sincere gratitude towards all other teachers and friends for their continuous support and constructive ideas.

Dhiraj Bobby

Jerin Vincent

John Kurian

Kalidas Jayakumar

Abstract

”RideShare”, a user-friendly mobile app built using Flutter for Android, is designed to connect individuals who want to share rides. RideShare facilitates carpooling by enabling users to both offer and search for rides, through an intuitive interface. Users can either offer rides by specifying their starting point, destination and available space, or search for existing rides matching their needs. This dual functionality allows users to either contribute by offering spare seats or find convenient, cost-effective rides. Built for simplicity, RideShare leverages Open Street Maps API and Open Route Services API alongside Firebase for seamless navigation and secure data management. By encouraging carpooling, RideShare aims to reduce traffic congestion, lower transportation costs for users, and contribute to a more environmentally friendly and connected community.

Contents

Acknowledgements	i
Abstract	ii
List of Figures	vii
1 Introduction	viii
1.1 Background	viii
1.2 Problem Definition	ix
1.3 Scope and Motivation	ix
1.4 Objectives	x
1.5 Challenges	x
1.6 Assumptions	x
1.7 Societal / Industrial Relevance	xi
1.8 Organization of the Report	xi
2 Software Requirements Specification	xiii
2.1 Introduction	xiii
2.1.1 Purpose	xiii
2.1.2 Product Scope	xiii
2.2 Overall Description	xiv
2.2.1 Product Perspective	xiv
2.2.2 Product Functions	xiv
2.2.3 Operating Environment	xv
2.2.4 Design and Implementation Constraints	xv
2.2.5 Assumptions and Dependencies	xv
2.3 External Interface Requirements	xvi
2.3.1 User Interfaces	xvi
2.3.2 Hardware Interfaces	xvii

2.3.3	Software Interfaces	xvii
2.3.4	Communications Interfaces	xvii
2.4	System Features	xviii
2.4.1	User Registration and Login	xviii
2.4.2	Search Carpools	xix
2.4.3	Create Carpools	xx
2.4.4	Join Carpools	xxi
2.4.5	Location Based Services	xxiii
2.4.6	Rating System	xxiv
2.4.7	Carpooled Distance and Carbon Emission Reduced	xxv
2.5	Other Non Functional Requirements	xxvi
2.5.1	Performance Requirements	xxvi
2.5.2	Safety Requirements	xxvi
2.5.3	Security Requirements	xxvii
2.5.4	Software Quality Attributes	xxvii
3	System Architecture and Design	xxix
3.1	System Overview	xxix
3.1.1	Carpool App: A Secure and Sustainable Ride-Sharing Solution . . .	xxix
3.2	Architectural Design	xxxi
3.3	Proposed Methodology/Algorithms	xxxvii
3.3.1	Algorithm for search engine module	xxxvii
3.3.2	Algorithm for carpool management module	xxxviii
3.4	User Interface Design	xli
3.5	Database Design	xlv
3.5.1	Database Choice: Firebase	xlv
3.5.2	Database Schema	xlvi
3.5.3	Schema Justification	xlvii
3.6	Description of Implementation Strategies	xlvii
3.7	Module Division	xlviii
3.8	Work Schedule - Gantt Chart	li

4 Results and Discussions	lii
4.1 Overview	lii
4.2 Testing	liii
4.3 Discussion	lxii
5 Conclusion	lxiii
5.1 Conclusion	lxiii
5.2 Future Scope	lxiii
Appendix A: Presentation	lxv
Appendix B: Vision, Mission, Programme Outcomes and Course Outcomes	xxii
Vision, Mission, POs, PSOs and COs	ii
Appendix C: CO-PO-PSO Mapping	vi

List of Figures

3.1	System Overview	xxix
3.2	Registration and Authentication module	xxxii
3.3	Carpool Management module	xxxiii
3.4	Carpool Management Sequence Diagram	xxxiv
3.5	Search Engine module	xxxv
3.6	Search Engine Sequence Diagram	xxxvi
3.7	Rideshare Analysis module	xxxvii
3.8	Use Case Diagram	xxxviii
3.9	Login Screen and Register Screen (Shows the app's initial screen with options to sign up or log in)	xli
3.10	Search Carpool and Publish Carpool (Depicts the screen for entering search criteria like origin, destination, date, etc and the screen for publishing ride along with necessary details.)	xlii
3.11	Search Engine and Rides Displayed (Shows how the search results pop up for riders based on their location radius)	xliii
3.12	User Profile with details such as reviews, personal information, distance travelled and level awarded is shown.	xliv
3.13	Gantt Chart	li
4.1	Splash Screen and Login Page	liii
4.2	Register Page and Edit Profile	liv
4.3	Ride Select	lv
4.4	User Profile	lvi
4.5	Search Page	lvii
4.6	Publish Page	lviii
4.7	My Activity Page	lix
4.8	Search By Location	lx

4.9	Route Navigation	lxi
-----	----------------------------	-----

Chapter 1

Introduction

1.1 Background

Urban areas face severe challenges like traffic congestion, environmental degradation, and inefficient transportation. To tackle these issues effectively, the development of a carpooling app has gained traction as a promising solution.

Current Scenarios

- Urban Congestion: Cities globally grapple with rising traffic congestion, leading to longer commute times and increased stress for commuters.
- Environmental Impact: Single-occupancy vehicles contribute significantly to carbon emissions and air pollution, posing threats to public health and exacerbating climate change.
- Transportation Inefficiencies: Traditional transportation systems suffer from underutilized vehicle capacity and limited accessibility, necessitating innovative solutions to optimize resources and enhance mobility.

Importance of the Carpooling App Project

- Mitigating Traffic Congestion: The app promotes ridesharing among commuters, reducing congestion and improving traffic flow.
- Promoting Environmental Sustainability: By encouraging shared transportation, the app helps minimize carbon emissions and environmental impact.
- Enhancing Community Connectivity: Carpooling fosters social interaction and community engagement, enhancing the commuting experience.

- Improving Accessibility and Affordability: The app provides a cost-effective alternative, making commuting more accessible to a broader population.
- Driving Innovation in Transportation: Leveraging technology, the app optimizes vehicle utilization, driving innovation in urban mobility solutions.

Conclusion: The carpooling app project offers a timely and impactful solution to urban transportation challenges. By promoting ride-sharing, reducing congestion, and fostering sustainability, it has the potential to transform urban mobility and improve quality of life

1.2 Problem Definition

Developing a mobile application (Android) to facilitate carpooling and reduce traffic congestion. This app will connect riders and drivers for cost-effective travel and promote a more sustainable transportation system.

1.3 Scope and Motivation

The project is designed to streamline the process of carpooling by connecting riders with drivers who have available seats, thus facilitating affordable travel options. Through a user-friendly platform, riders can easily search for carpools that match their desired routes and schedule, while drivers can offer their empty seats to offset travel costs. Key functionalities include seamless booking of carpools, sharing available seats, and suggesting efficient routes to optimize travel time. By focusing on these core features, the project aims to provide a straightforward and efficient carpooling experience for users. This initial scope deliberately excludes more advanced features like multi-stop rides and social networking, allowing for a focused development process with room for potential expansion in the future.

The primary motivation driving the project is the desire to address pressing transportation challenges such as traffic congestion, high travel costs, and environmental sustainability. By promoting carpooling as a viable alternative to single-occupancy vehicle trips, the platform seeks to reduce the number of vehicles on the road, thereby alleviating congestion and lowering carbon emissions. Additionally, by offering cost-effective travel options, the project aims to make transportation more accessible and affordable for a

wider range of individuals, particularly those facing financial constraints. Ultimately, the overarching goal is to foster a culture of sustainable commuting practices while providing tangible benefits to both commuters and the community at large.

1.4 Objectives

1. Allow users to register and login securely.
2. Enable users to search for available carpools based on their needs.
3. Enable users to publish carpools for others to join.
4. Integrate with mapping services for route navigation and visualization.
5. Incorporate a user review system to promote trust and accountability.
6. Facilitate user location services to streamline carpool searching.

1.5 Challenges

Our app will face challenges in efficiently matching riders and drivers with compatible schedules, locations, and car capacities. Building trust between strangers is also crucial, requiring features like driver verification and clear user guidelines.

1.6 Assumptions

1. User Behavior: We assume users will provide accurate and up-to-date information in their profiles and ride postings.
2. GPS Accuracy: We assume a base level of GPS accuracy for location services, acknowledging potential discrepancies that may require user adjustments during rides
3. Third-Party Integrations: The app heavily depends on successful integration with Open Street Maps API for functionalities like navigation, location services, and estimated travel times.

1.7 Societal / Industrial Relevance

The carpooling app offers a sustainable solution to alleviate traffic congestion and reduce carbon emissions by promoting ridesharing among commuters. In society, it fosters a sense of community by encouraging shared transportation, leading to enhanced social interaction and reduced isolation during travel. Industrially, the app revolutionizes the transportation sector by optimizing vehicle usage, thus lowering operational costs for individuals and companies while contributing to a more eco-friendly and efficient urban infrastructure.

1.8 Organization of the Report

- Chapter 1: This chapter provides an overview of the project, addressing urban transportation challenges and emphasizing the development of an Android carpooling app. It outlines project scope, motivation, objectives, challenges, and societal/industrial relevance.
- Chapter 2: This chapter focuses on Software Requirements Specification (SRS), covering system purpose, functions, constraints, external interface requirements, system features, and nonfunctional requirements.
- Chapter 3: This chapter starts with a comprehensive system overview, including detailed architecture diagrams. It then covers architectural design using tools like use case diagrams, ER diagrams and sequence diagrams, designs for user interface, database design rationale, module division, and a Gantt chart illustrating the project timeline and tasks.
- Chapter 4: This chapter summarizes the project's successful outcomes and evaluates the performance and effectiveness of the Rideshare app. It discusses the app's functionality, user experience, and areas for future improvement, supported by visual representations of the app's interface during testing.
- Chapter 5: This chapter concludes the project, affirming the achievement of its primary objectives and the app's effectiveness as a carpooling solution. It also

outlines potential enhancements to further improve the app's functionality and user satisfaction, ensuring its continued success and growth.

Chapter 2

Software Requirements Specification

2.1 Introduction



RideShare is a user-friendly mobile application designed for Android devices. It aims to revolutionize transportation by promoting carpooling and reducing traffic congestion.

2.1.1 Purpose

This Software Requirements Specification (SRS) defines the functional and non-functional requirements for the RideShare carpooling mobile application, Version 1.0. It outlines the core functionalities, user interactions, system integrations, and quality attributes that will guide the development process. This SRS focuses solely on the initial functionalities of the RideShare mobile application for Android devices.

2.1.2 Product Scope

RideShare is a user-friendly mobile application designed for Android devices. Its primary objective is to facilitate carpooling by connecting riders seeking cost-effective transportation with drivers who have available seats. The application aims to address traffic congestion, reduce transportation costs for users, and promote environmental sustainability.

ity by encouraging carpooling as a viable alternative. This version will focus on core functionalities of the carpooling service. Initial scope excludes advanced features like multi-stop rides and social networking. Future versions may incorporate additional features based on user feedback.

2.2 Overall Description

RideShare is a comprehensive carpooling application that aims provide a platform for users to share rides conveniently and cost-effectively. It offers a range of features designed to enhance user experience and promote environmental sustainability.

2.2.1 Product Perspective

RideShare is a carpooling app that provides a platform that connects drivers and passengers for ride-sharing, aiming to reduce the number of cars on the road, lessen commuting costs for passengers and drivers, and offer a convenient and potentially social alternative to single-occupancy vehicles. This app is a standalone product and is self-contained. However, it will integrate with existing mapping services like Open Street Maps for route navigation and geo-location functionalities.

2.2.2 Product Functions

- User registration and login: Users can create accounts and login securely.
- Account verification: Users can add essential information like profile picture, emergency contact, age to verify their account.
- Offer rides: Users can offer rides by specifying starting location, destination, date and available seats.
- Search for rides: Users can search for rides based on destination, date and other criteria.
- Request to join rides: Users can request to join rides offered by other users if it is suitable for them.
- Accept or Reject requests: Users can accept or reject join requests.

- Navigation and Tracking: The app integrates with mapping services for route navigation and potentially real-time ride tracking for safety and peace of mind.
- Rating and Review System: Users can rate and review other users, promoting accountability and trust within the carpooling community.
- Carpooling Rewards: A feature that promotes environmental awareness by rewarding users with levels based on their total carpooled distance.
- Safety Features: Additional features like emergency contact information can enhance user safety.

2.2.3 Operating Environment

The app will operate on mobile devices with Android operating system with Android version 7.0(nougat) or above.

2.2.4 Design and Implementation Constraints

- Regulatory Compliance: RideShare app must adhere to local transportation regulations and ride-sharing laws in all regions where it operates. This includes compliance with government regulations, and data protection laws such as DPDP.
- Compatibility with Third-party APIs: The app will rely on external APIs for services such as mapping. Developers must ensure compatibility with these APIs and adhere to their usage terms and conditions.
- Privacy Considerations: The app must respect user privacy and obtain explicit consent for obtaining their location and processing of personal data.

2.2.5 Assumptions and Dependencies

- User Behavior: We assume users will provide accurate and up-to-date information in their profiles and ride postings.
- GPS Accuracy: We assume a base level of GPS accuracy for location services, acknowledging potential discrepancies that may require user adjustments during rides.

- Third-Party Integrations: The app heavily depends on successful integration with Open Street Maps API for functionalities like navigation, location services, and estimated travel times.

2.3 External Interface Requirements

The RideShare app will interact with various hardware and software components to provide its functionalities. Here's an overview of the external interfaces it will utilize:

2.3.1 User Interfaces

The RideShare app will prioritize a user-friendly UI designed for ease of use on Android devices. Some of the key UI characteristics are:

- Clarity and Simplicity: Information will be presented hierarchically with clear labeling for optimal readability.
- Consistent Design Patterns: A consistent look and feel will be maintained throughout the app with standardized button styles, icons, and navigation elements. This will create a familiar and predictable user experience.
- Adaptability: The UI will be designed to adapt to different screen sizes effectively. This ensures optimal viewing experience across various Android device models.
- Core User Interfaces: This version will focus on essential functionalities with dedicated UI components for:
 - Login/Signup: Screens for user registration, login, and profile management.
 - Ride Offer: Interface for drivers to specify details like origin, destination, available seats, date, time, and price for their rides.
 - Ride Search: Functionality for riders to search for existing rides based on origin, destination, date, and time (with optional filtering by price or other criteria).
 - Ride Details: Detailed information screens for both drivers and riders showcasing ride specifics like origin, destination, time, driver/rider information (including profile picture and ratings).

- Route Navigation: Integration with Open Route Services API to provide navigation for drivers and an estimated route overview for riders during a carpool journey.
- Error Handling: User-friendly error messages will be displayed in a clear and concise manner, guiding users towards resolving any issues encountered within the app.

2.3.2 Hardware Interfaces

The RideShare app will primarily interact with the following hardware components on user devices:

- Touchscreen: The primary mode of user interaction will be through touchscreen gestures (taps, swipes) for navigation, selection, and data input.
- GPS : The app uses device GPS functionalities to determine the user's current location, potentially enhancing search results and navigation accuracy.

2.3.3 Software Interfaces

RideShare will interact with several external software components to deliver its core functionalities. Here's an outline of these interactions:

- Open Street Maps API (1.0.3): This API will provide location services (user location retrieval)and map data functionalities within the app.
- Open Route Services API (1.2.6): This API will provide route visualization from a start location to a destination location.
- Firebase: This backend-as-a-service platform will be used for user authentication, secure data storage (user information, ride details, potentially payment information if applicable), and real-time database functionalities for managing ride requests and confirmations.

2.3.4 Communications Interfaces

The RideShare app will rely on the following communication interfaces:

- Internet Connectivity: A stable internet connection will be required for all functionalities including user login, data exchange with backend services (Firebase), map loading, and potentially real-time communication features.
- Secure Communication Protocols: Secure protocols like HTTPS will be implemented to safeguard user data transmission between the app, user devices, and backend services. This ensures data privacy and security.
- API Communication Protocols: Each chosen external API will have its defined communication protocols for data exchange. The app will adhere to these protocols to ensure proper integration and functionality.

2.4 System Features

To ensure smooth user experience and data management, several core functionalities are outlined below.

2.4.1 User Registration and Login

To access and utilize the app's services, users must complete a registration and login process.

2.4.1.1 Description and Priority

- Users can add transportation routes by specifying start and end locations, date, and number of empty seats. Only a registered user can create and join a carpool. .
- This feature is of high priority.

2.4.1.2 Stimulus/Response Sequences

- User launches Ride-share app. .
- User taps on Sign-Up button
- User has to enter valid personal information by filling up registration form
- Upon successful validation: System creates a new user account. System logs the user in and directs them to the app's main dashboard

- Upon unsuccessful validation System displays error message for each invalid field
User can correct the invalid fields and resubmit the form

2.4.1.3 Functional Requirements

REQ-1	System shall display a signup form upon user selection.
REQ-2	The signup form shall include fields for name, email address, and password.
REQ-3	System shall validate the email address format.
REQ-4	System shall enforce minimum password length and complexity requirements.
REQ-5	System shall check for existing user accounts with the entered email address.
REQ-6	Upon successful registration, the system shall create a new user account.
REQ-7	System shall log the user in and redirect them to the main dashboard upon successful registration.
REQ-8	System shall display clear and specific error messages for invalid user input during registration.

2.4.2 Search Carpools

Users can easily find and join available carpools through the search functionality.

2.4.2.1 Description and Priority

- Search option is displayed on the homepage of the application that allows users to search for available carpools based on their current location or user's specified location. This feature is of high priority

2.4.2.2 Stimulus/Response Sequences

- User selects the "Search" option on the app.

- Users can search carpools by giving required origin and destination places. Users can select destination location given in the drop-down menu in the search option as well as set the destination by moving the icon on the screen.
- System searches for carpools matching the user's criteria.
- If matching carpools are found, the system displays a list of results to the user, showing details such as driver information, departure time, and available seats.
- If no matching pools are available, the system displays a message indicating that no carpools are available at the moment.

2.4.2.3 Functional Requirements

REQ-1	System shall provide a dedicated interface for searching carpools.
REQ-2	Users shall be able to enter origin and destination location
REQ-3	System should provide search filters based on user preferences like starting location, destination location, date and seats required.
REQ-4	System shall search the database for available carpools that match the user's specified criteria.
REQ-5	System shall display a list of matching carpools with relevant details including driver information, departure time and available seats.
REQ-6	If no matching carpools are found, the app shall display a message indicating that no carpools meeting the specified criteria were found.

2.4.3 Create Carpools

Registered users with cars can set up carpools by specifying ride details and available seats.

2.4.3.1 Description and Priority

- Only a registered user having a car can create a carpool by specifying the ride details and available seats. This feature has high priority

2.4.3.2 Stimulus/Response Sequences

- Users can create a carpool by clicking on the “publish ride” button on the app.
- App displays a form for carpool creation.
- User has to enter necessary information including origin, destination, date and available seats.
- User then submits the form for generating a carpool.
- System validates the provided information.
- Upon successful validation, the system displays the carpool information in the app.
- Upon unsuccessful validation, the system displays an error message

2.4.3.3 Functional Requirements

REQ-1	The app shall provide a form for users to input details such as start and end locations, departure date, and number of available seats.
REQ-2	The app shall validate the user's input to ensure that all required fields are filled out and that the route and departure time are in a valid format.
REQ-3	The app shall enforce constraints on the number of available seats, ensuring it is a positive integer
REQ-4	Upon successful creation of a carpool, the app shall store the carpool details in the database and associate them with the user's account.
REQ-5	If there are any errors during carpool creation, such as an invalid route or departure time in the past, the app shall display appropriate error messages and ask the user to correct them.

2.4.4 Join Carpools

This feature enables users to join available carpool options and manage their ride requests.

2.4.4.1 Description and Priority

- System allows registered users to request a ride from one of the many available options of carpools. If a user has already joined a carpool, the app shall allow them to leave that carpool. This is a high priority feature.

2.4.4.2 Stimulus/Response Sequences

- Once the user selects a carpool after searching for available carpools, they can select the “Ride Request” option.
- Upon requesting the ride, the app will send a ride request to the carpool driver and display request status.
- If the driver accepts the request, the system will confirm booking and notify both driver and user.
- If the request was not approved by the driver, the app shall notify the user about the unsuccessful request.

2.4.4.3 Functional Requirements

REQ-1	System shall allow users to view detailed information about car-pools displayed in search results.
REQ-2	System shall provide a way for users to request to join a car-pool. This shall be implemented by using a “Request Ride” button.
REQ-3	Upon user confirmation, the app shall notify the driver of the car-pool about the ride request. User should have the option to see the status of their request.
REQ-4	The driver should be able to accept or reject the ride request.
REQ-5	If the driver accepts the ride request, the app shall confirm the booking and notify both the user and the driver
REQ-6	If the driver rejects the ride request or does not respond within a specified time frame, the app shall notify the user that the request was unsuccessful.
REQ-7	The app shall handle ride requests in real-time, updating the status of requests and bookings accordingly.

2.4.5 Location Based Services

Enhancing the user experience by utilizing location data to streamline the search and display of nearby carpools.

2.4.5.1 Description and Priority

- The app can use the user’s location (with user’s permission) to make the searching process easier. It can automatically fill in your starting point and show you the nearest carpools first. This is of high priority

2.4.5.2 Stimulus/Response Sequences

- User opens the app and grants location access.
- System retrieves the user’s current location.

- System shall leverage user location data (when available) to auto-fill the origin field in the search interface
- System shall prioritize search results based on the user's location, displaying carpools with origins closest to the user first.
- System shall integrate with mapping services to display the planned carpool route on the app

2.4.5.3 Functional Requirements

REQ-1	The app shall request and obtain permission from the user to access their location.
REQ-2	The app shall retrieve the user's current location using GPS or other location services.
REQ-3	The app shall display nearby carpools, drivers, or passengers based on the user's location.
REQ-4	The app shall provide accurate and up-to-date location-based information to users
REQ-5	The app shall integrate Open Route Services API to display the planned route and should update data dynamically

2.4.6 Rating System

This feature allows users to evaluate their carpool experiences, helping to ensure driver accountability and enhance overall service standards.

2.4.6.1 Description and Priority

- Enable users to rate and provide feedback on their carpooling experience, facilitating accountability and improving service quality. This feature is of medium priority.

2.4.6.2 Stimulus/Response Sequences

- After completion of each ride the user will be asked to rate the driver.

- Users can provide an optional star rating and a text review about the ride
- System should store the review for the driver.
- System should calculate and update the driver's average rating.

2.4.6.3 Functional Requirements

REQ-1	System asks the user and driver to rate each other after each ride.
REQ-2	System allows users to rate each other by providing a star rating interface. Users can skip this step.
REQ-3	System also allows users to give a text review about the ride.
REQ-4	System stores the ratings and then calculates and updates the user's average rating
REQ-5	User's rating and reviews should be displayed in their profiles to help others to make informed decisions when selecting rides.

2.4.7 Carpoled Distance and Carbon Emission Reduced

This feature tracks and displays the total distance carpooled by users and their corresponding reward levels, promoting environmental awareness.

2.4.7.1 Description and Priority

- The system shall display the total distance traveled through carpooling and the corresponding level rewarded in the user's profile. This feature's priority is Low.

2.4.7.2 Stimulus/Response Sequences

- User navigates to their profile page
- The system will calculate the total distance carpooled and assign a level to the user based on this value.
- The system will display and update the metrics after each ride.

2.4.7.3 Functional Requirements

REQ-1	The system shall track and record the distance traveled by the user through carpooling.
REQ-2	The system will assign a level to the user based on where the value of total distance carpooled is placed in the rankings.
REQ-3	The system will display the total distance carpooled and the level assigned in the user's profile with clear and intuitive visualization.

2.5 Other Non Functional Requirements

These requirements ensure the system's efficiency, safety, security, and overall quality, enhancing user satisfaction and reliability.

2.5.1 Performance Requirements

A proper internet connection is needed for the users using this application and the user interface should be user-friendly and easy for Android mobile phone users. The database must function well in order for the user to have no delays in the program. Reading is more significant than writing since users are more likely to spend time browsing than posting trips. Hence a read operation should take less than 500ms and a write operation should approximately be 1s.

2.5.2 Safety Requirements

The app prioritizes user safety by mitigating risks of physical harm, data breaches, privacy violations, and fostering a false sense of security. Implemented safeguards include:

- User Verification: Mandatory account verification for both drivers and passengers is implemented to ensure safety.
- Data Security: Encryption of user data stored on servers, adherence to relevant data privacy regulations, and regular security audits and penetration testing.
- User Education: Promotion of responsible carpooling behavior through in-app tutorials and safety tips, along with encouraging users to verify driver information and

exercise caution during rides.

2.5.3 Security Requirements

The RideShare application will implement essential security measures to safeguard user data and ensure secure access to the platform. These measures include basic user authentication with password hashing to protect user credentials, encryption techniques for securing sensitive data during transmission and storage, and the implementation of role-based access control to differentiate between regular users and drivers, thereby controlling access to specific functionalities within the application. Additionally, the project will focus on minimal data collection to uphold user privacy. Adherence to secure development practices such as regular code reviews and following secure coding guidelines will be emphasized to mitigate common security vulnerabilities and ensure the integrity and reliability of the application.

2.5.4 Software Quality Attributes

- Maintainability

The system should be maintainable in the future. The system should be well documented such that new developers can quickly comprehend the system. We should be certain that the technology we choose will be supported for a long time.

- Scalability

As the number of users and operations grows, the server and database must be able to scale both vertically and horizontally. The system must be able to accommodate a user base from 100 users to 1 million users.

- Usability

The app should provide users with a user-friendly user interface for easy use of the app. Text, buttons and other graphic elements should be large enough to improve clarity and enhance user experience. Navigation through the app should be simple so that users can easily navigate through the app with ease.

- Portability

The initial version of the app shall be developed for the Android platform, ensuring compatibility with Android smartphones and tablets. The app architecture and codebase shall be designed with modularity and flexibility to facilitate future development for the iPhone platform. Targeting both Android and iOS allows the app to reach a broader user base, maximizing its potential reach and impact

Chapter 3

System Architecture and Design

3.1 System Overview

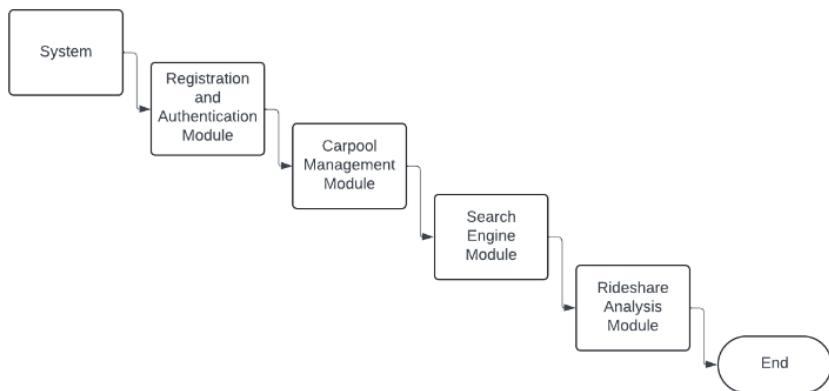


Figure 3.1: System Overview

RideShare offers secure user registration with options such as email/password or Google Sign-In, ensuring seamless access for existing users. Drivers can list carpools with details like origin, destination, date, time, and available seats, while riders can join based on their trip preferences. Location-based services help find nearby carpools, and search filters refine options for riders. The app leverages Open Street Maps API for autofill suggestions, streamlining origin and destination searches. RideShare calculates total distance traveled through carpooling, awarding badges for eco-friendly contributions and analyzing user ratings to highlight top-rated community members and encourage positive behavior.

3.1.1 Carpool App: A Secure and Sustainable Ride-Sharing Solution

This document outlines the comprehensive architecture and functionalities of a carpool app designed to foster a secure and eco-friendly ride-sharing community.

3.1.1.1 Secure User Management

The foundation of our carpool app lies in robust user management, ensuring a safe and trustworthy environment for all participants. Here's a breakdown of its key features:

- **Registration:** Users can create accounts using email addresses or social media logins (with proper permission control). Secure password hashing and storage are implemented to protect user credentials.
- **Verification:** Optional phone number or email verification can be introduced to enhance user authenticity and deter fraudulent accounts.
- **Profile Management:** Users can personalize their profiles by adding details like name, picture, and preferences (smoking, music, etc.).
- **User Roles:** The system can differentiate between riders and drivers, allowing them access to specific functionalities within the app.

3.1.1.2 Efficient Carpool Management

Connecting riders and drivers seamlessly is the core functionality of the carpool app. This section details how the app facilitates efficient carpool formation:

- **Trip Posting:** Drivers can create trip postings specifying details like origin, destination, departure time, available seats, and any preferences. Riders can browse these postings and filter them based on their needs (location, time, etc.).
- **Route Visualization:** Route visualization from the starting location to the destination location can be implemented using polylines.
- **Feedback System:** A post-trip rating and feedback system can be used to encourage responsible behavior from both riders and drivers, promoting a culture of courtesy and trust within the community.

3.1.1.3 Simplified RideShare Discovery

The app should prioritize user-friendliness and provide a smooth experience for finding carpool matches. Here's how the app simplifies carpool discovery:

- **Location-Based Search:** Leverage GPS technology to allow users to search for carpool options based on their current location or desired origin and destination.
- **Advanced Filters:** Implement filters for riders to refine their search based on specific criteria like departure time, car type, number of seats available, etc.

3.1.1.4 Promoting a Sustainable Future

Carpooling offers a significant environmental benefit by reducing the number of cars on the road. Our app actively promotes sustainability through the following features:

- **Rewards and Incentives:** Implement reward systems to incentivize eco-conscious behavior. This could involve awarding points or badges for frequent carpooling, offering discounts on carpool-related services (e.g., car maintenance), or participating in challenges that promote sustainable transportation.

3.2 Architectural Design

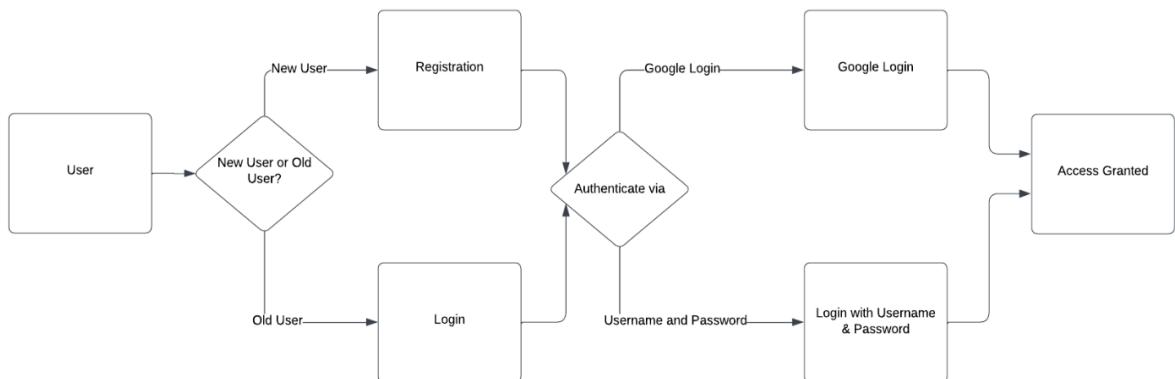


Figure 3.2: Registration and Authentication module

RideShare provides secure user registration with options like email/password or Google Sign-In, allowing existing users to access the app conveniently. The platform prioritizes user data privacy across all login methods.

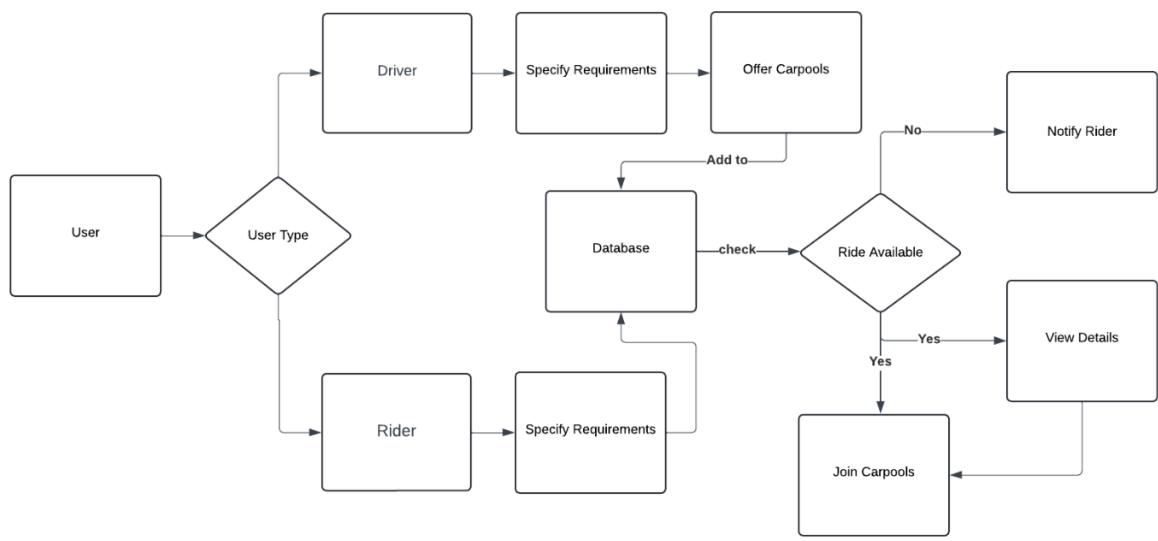


Figure 3.3: Carpool Management module

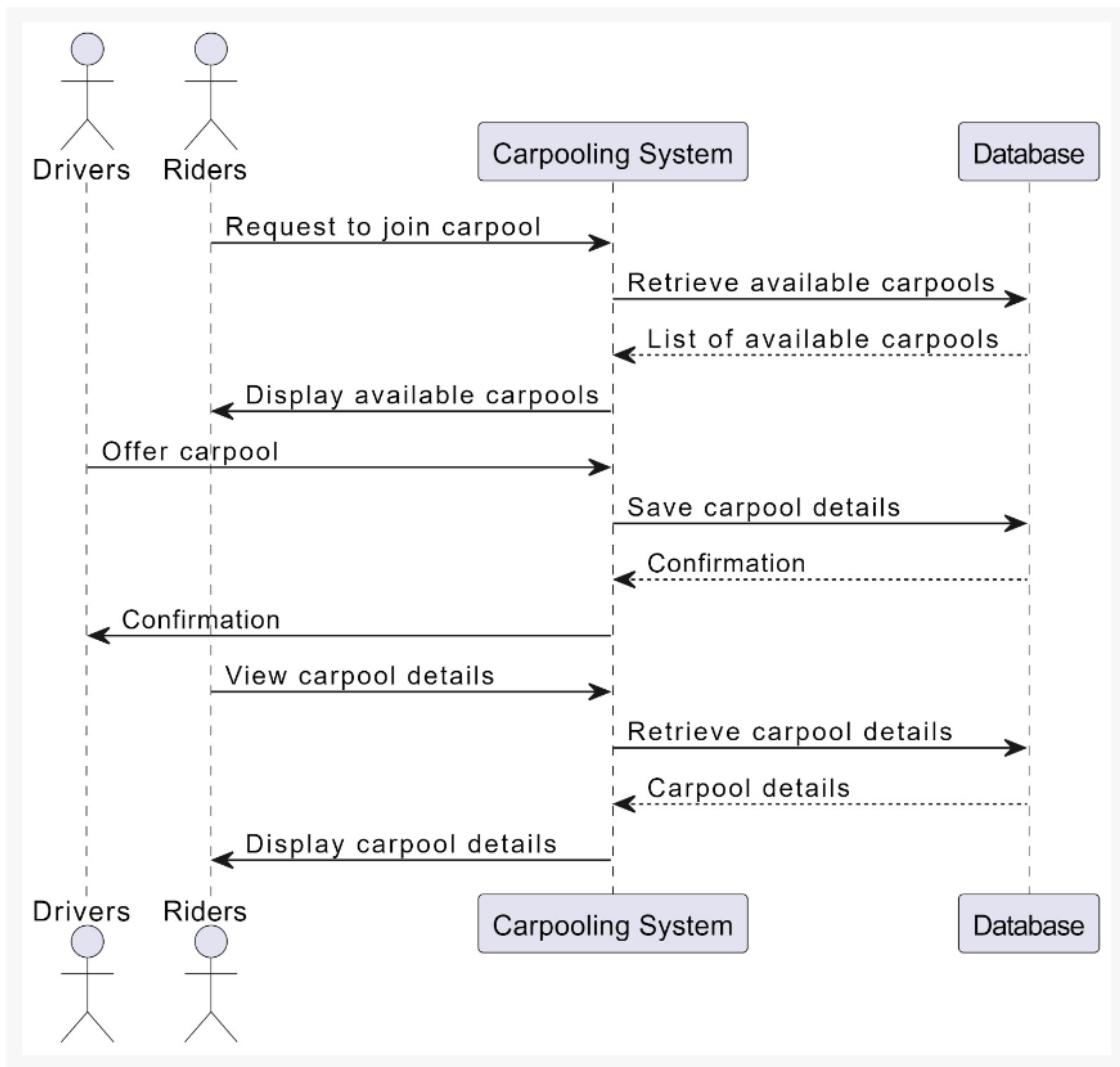


Figure 3.4: Carpool Management Sequence Diagram

Drivers can list carpools by providing origin, destination, date, time, and seat availability. Riders can join based on their preferred trip criteria and access carpool details for informed decisions.

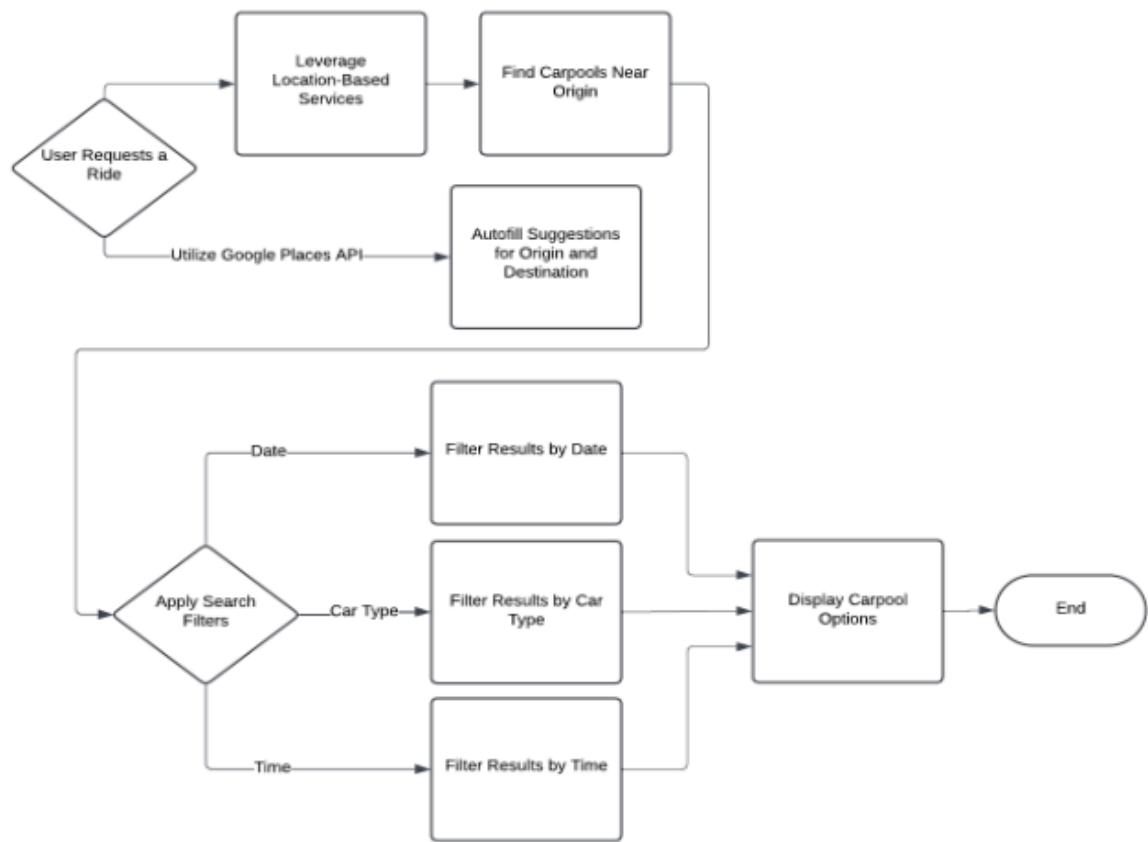


Figure 3.5: Search Engine module

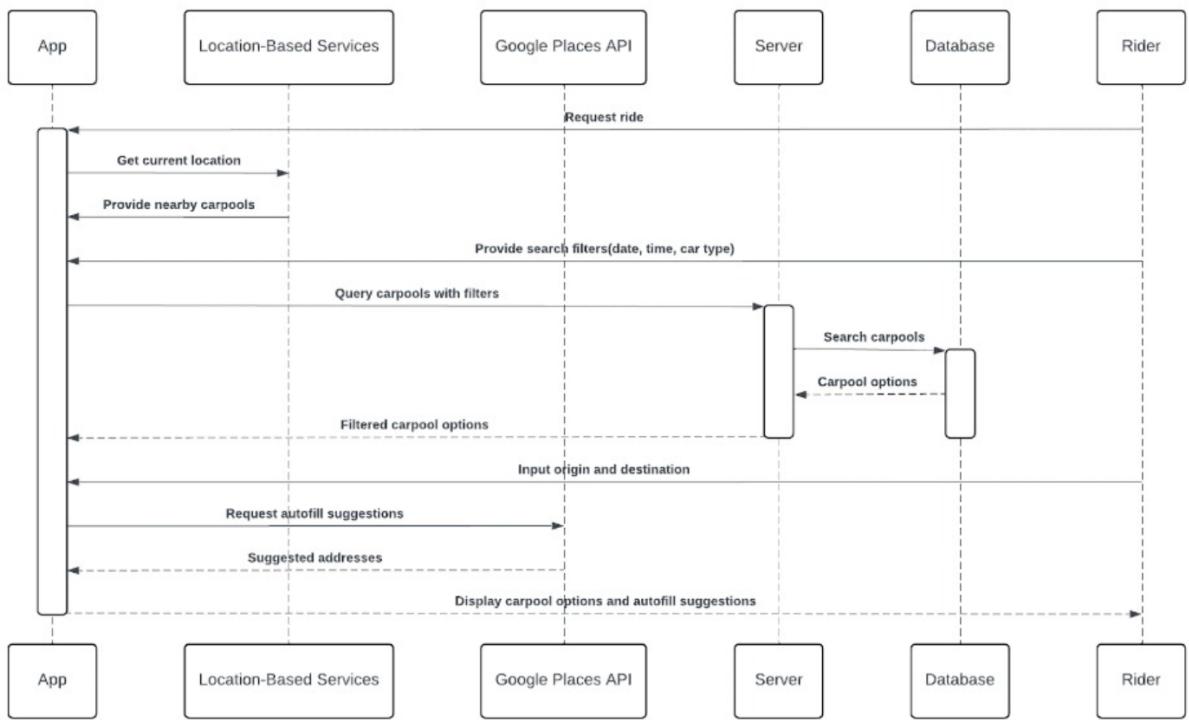


Figure 3.6: Search Engine Sequence Diagram

RideShare uses location-based services to find nearby carpools and offers search filters for refining trips by date, time, and car type. It utilizes Open Street Maps API for autofill suggestions to streamline origin and destination searches.

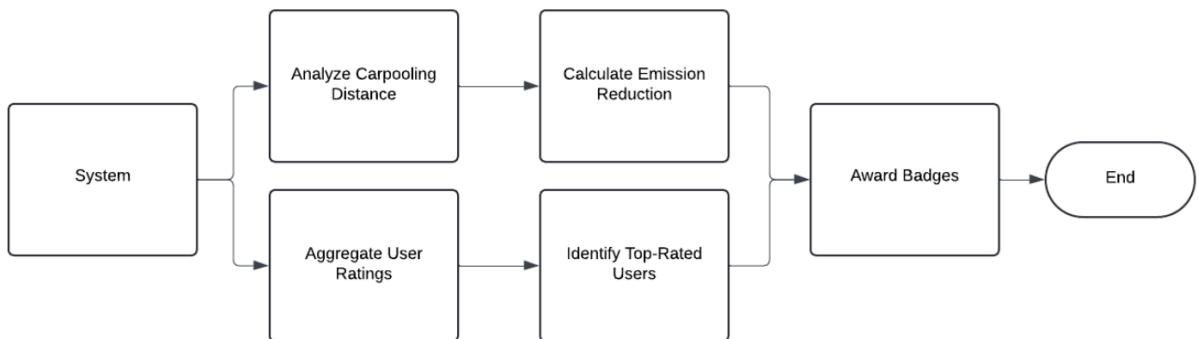


Figure 3.7: Rideshare Analysis module

RideShare tracks the total distance carpooled to demonstrate the collective reduction in individual car use and estimated carbon emissions. It rewards users for eco-friendly

contributions with badges and analyzes user ratings to highlight top-rated community members and promote positive behavior.

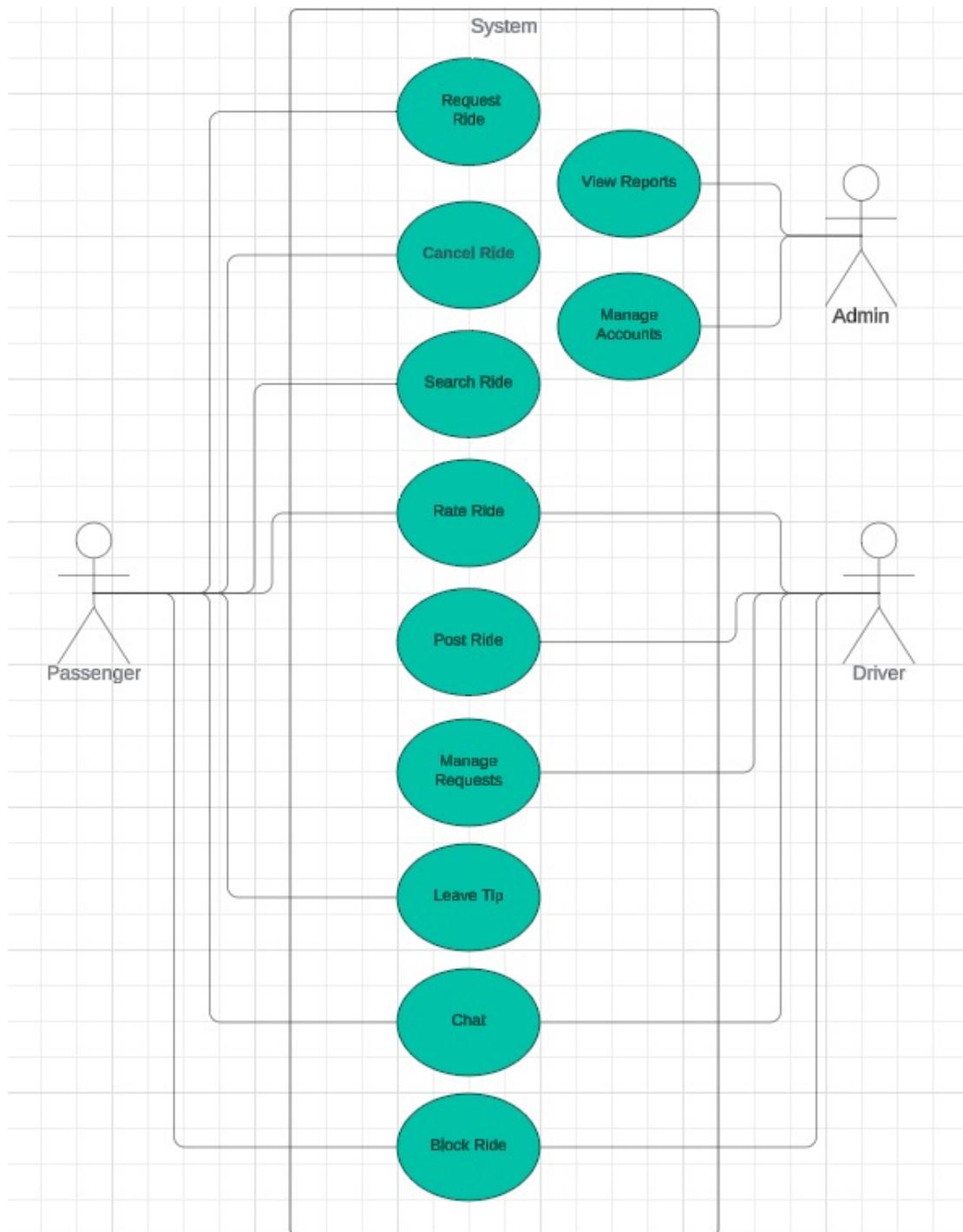


Figure 3.8: Use Case Diagram

3.3 Proposed Methodology/Algorithms

Following the proposed methodologies and algorithms, the system ensures efficient and accurate matching of carpool offers and requests, enhancing user experience and operational reliability.

3.3.1 Algorithm for search engine module

The algorithm for the search engine module efficiently matches users with potential carpools by converting addresses to coordinates, calculating distances and presenting detailed carpool options.

3.3.1.1 Inputs:

User enters origin, destination (text), number of seats, and date.

3.3.1.2 Outputs

App displays a list of potential carpool matches with details.

3.3.1.3 STEPS

1. Convert Addresses to Coordinates:

- (a) Use Open Street Maps API to convert origin and destination addresses into latitude and longitude coordinates.

2. Calculate Distance and Travel Time:

- (a) For each potential carpool:

- i. Use Open Street Maps to calculate the distance between the user's origin and the carpool location.

3. Present Matching Carpools:

- (a) Show the user a list of potential carpool options. Each carpool entry should include:

- i. Driver and car information

- ii. Estimated travel time
- iii. Starting and destination location
- iv. Seats remaining

3.3.2 Algorithm for carpool management module

The algorithm for the carpool management module ensures seamless creation and joining of carpools by drivers and riders, providing efficient matching based on origin, destination, date, and seat availability.

3.3.2.1 Inputs:

- Driver offers a carpool with details including:
 - Origin (text)
 - Destination (text)
 - Date
 - Available Seats (integer)
- Rider searches for carpools with details including:
 - Origin (text)
 - Destination (text)
 - Date
 - Seats needed

3.3.2.2 Outputs

- Driver successfully creates a carpool offer.
- Rider successfully joins a carpool or receives a message indicating no matches found.

3.3.2.3 STEPS

1. Driver Creates Carpool Offer:

- **1.1** Convert Driver's origin and destination addresses to latitude and longitude coordinates using Open Street Maps API.
- **1.2** Store carpool offer details in the database (including origin coordinates, destination coordinates, date, available seats, driver information, car details - optional).

2. Rider Searches for Carpools:

- **2.1** Convert Rider's origin and destination addresses to latitude and longitude coordinates using Google Places API.
- **2.2** Search the database for carpool offers that match the Rider's search criteria:
 - **Match Origin:** Carpool origin coordinates should be within a certain radius (configurable distance) of the Rider's origin (or Rider's entered origin coordinates if provided).
 - **Match Destination:** Carpool destination should exactly match the Rider's destination.
 - **Match Date:** Carpool date should match the Rider's desired date.

3. Present Matching Carpools:

- **3.1** If matching carpools are found:
 - For each matching carpool, calculate the distance between the Rider's origin (or entered origin coordinates) and the carpool origin using the Open Street Maps API.
 - Estimate the travel time for each carpool using the Open Street Maps API.
 - Display a list of matching carpool options to the Rider. Each entry should include:
 - * Driver information (name, rating - optional)
 - * Starting and Destination location

- * Available seats

- **3.2** If no matching carpools are found, display a message to the Rider indicating no carpools match their search criteria.

4. Rider Joins Carpool:

- **4.1** If the Rider chooses to join a carpool:

- Update the carpool entry in the database to reflect one less available seat.
- Send a notification to the Driver informing them that a Rider has joined their carpool.
- Display a confirmation message to the Rider indicating they have successfully joined the carpool.

3.4 User Interface Design

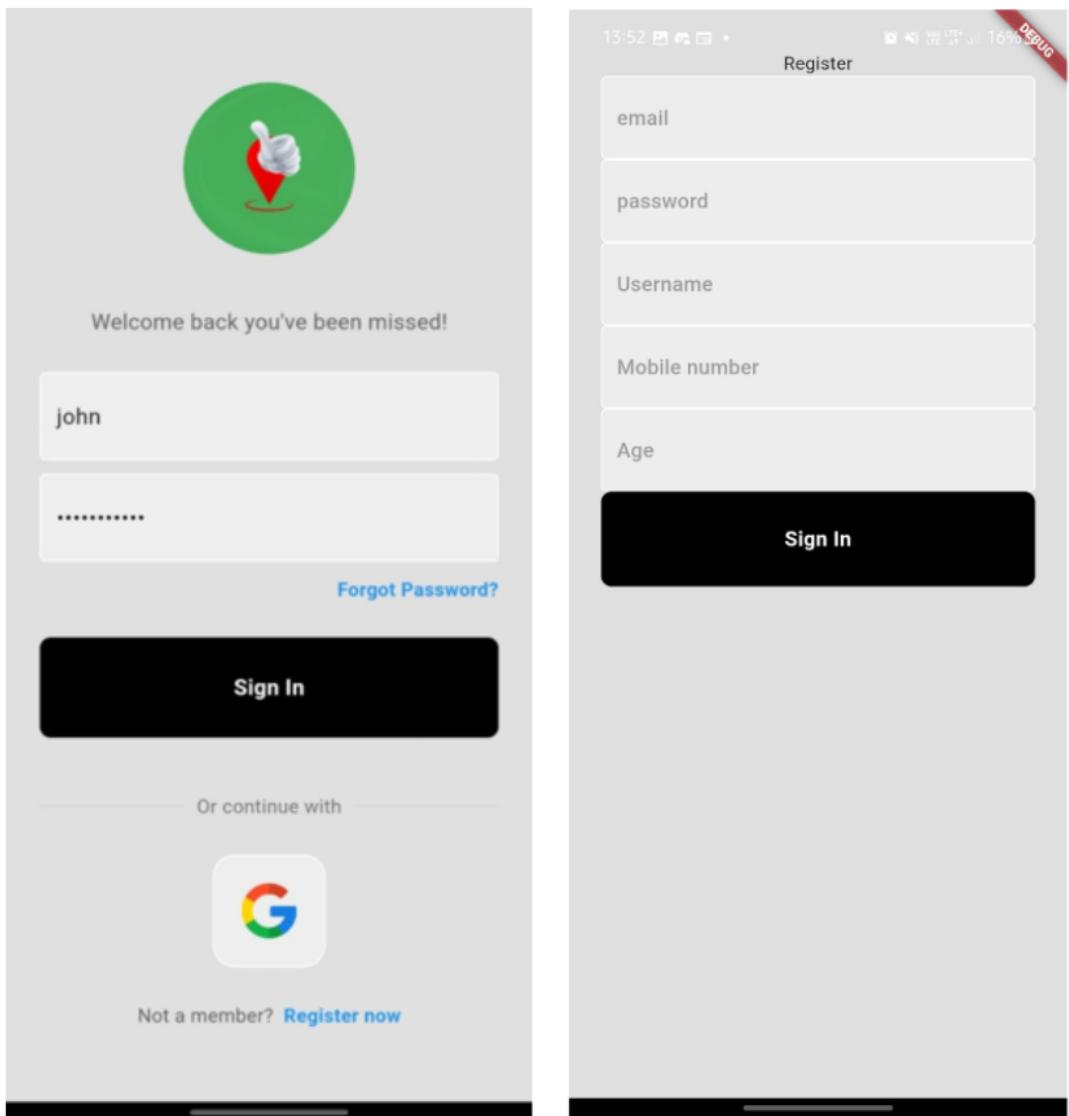


Figure 3.9: Login Screen and Register Screen (Shows the app's initial screen with options to sign up or log in)



Find a ride

Where

Leaving from... x ↑↓

Going to... x

When

Pick a date x

Seats needed

- 1 +

Search



Publish a ride

Where

Leaving from... x ↑↓

Going to... x

When

Pick a date x

Seats available

- 1 +

Publish

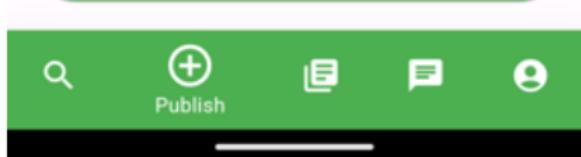


Figure 3.10: Search Carpool and Publish Carpool (Depicts the screen for entering search criteria like origin, destination, date, etc and the screen for publishing ride along with necessary details.)

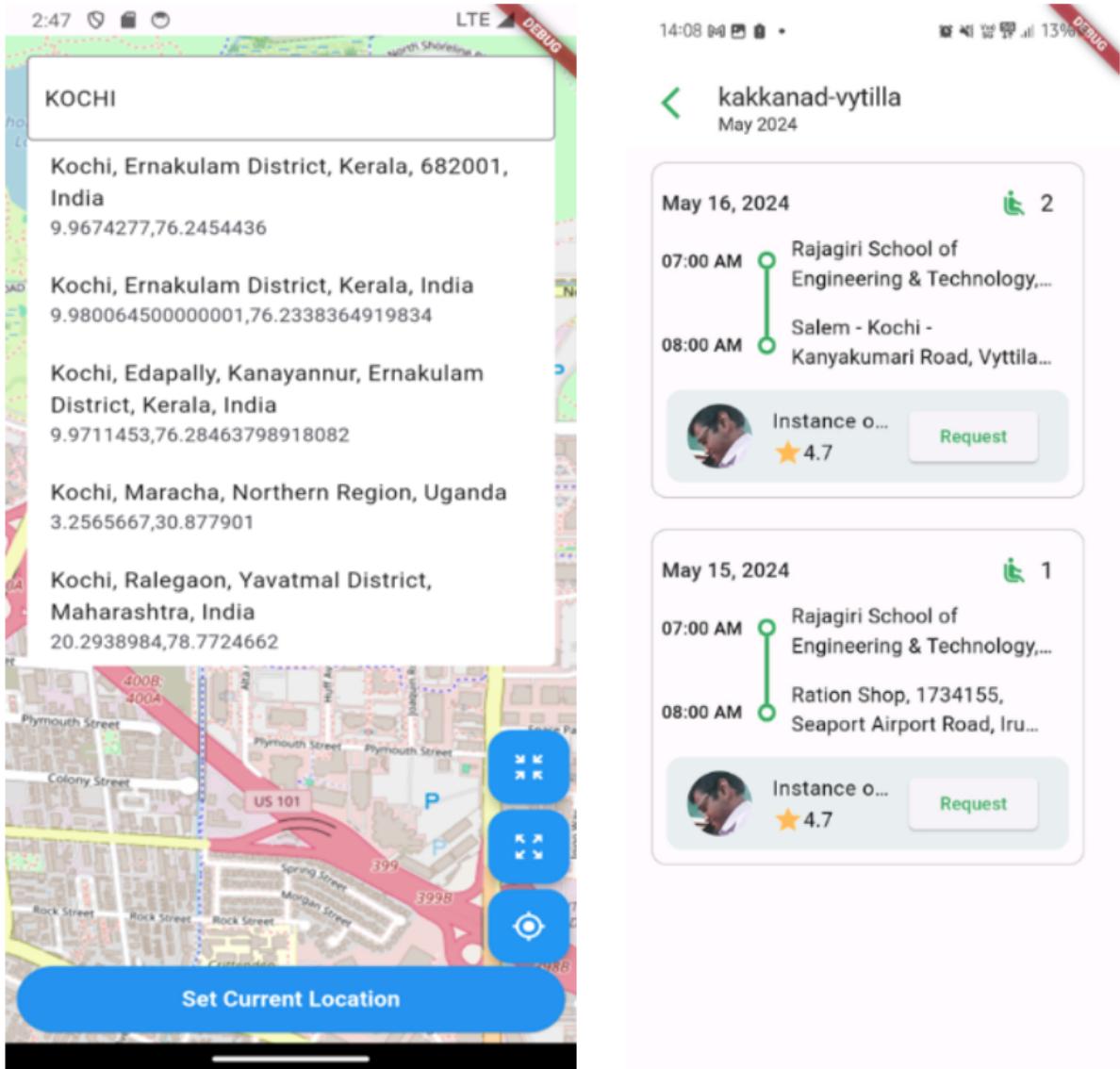


Figure 3.11: Search Engine and Rides Displayed (Shows how the search results pop up for riders based on their location radius.

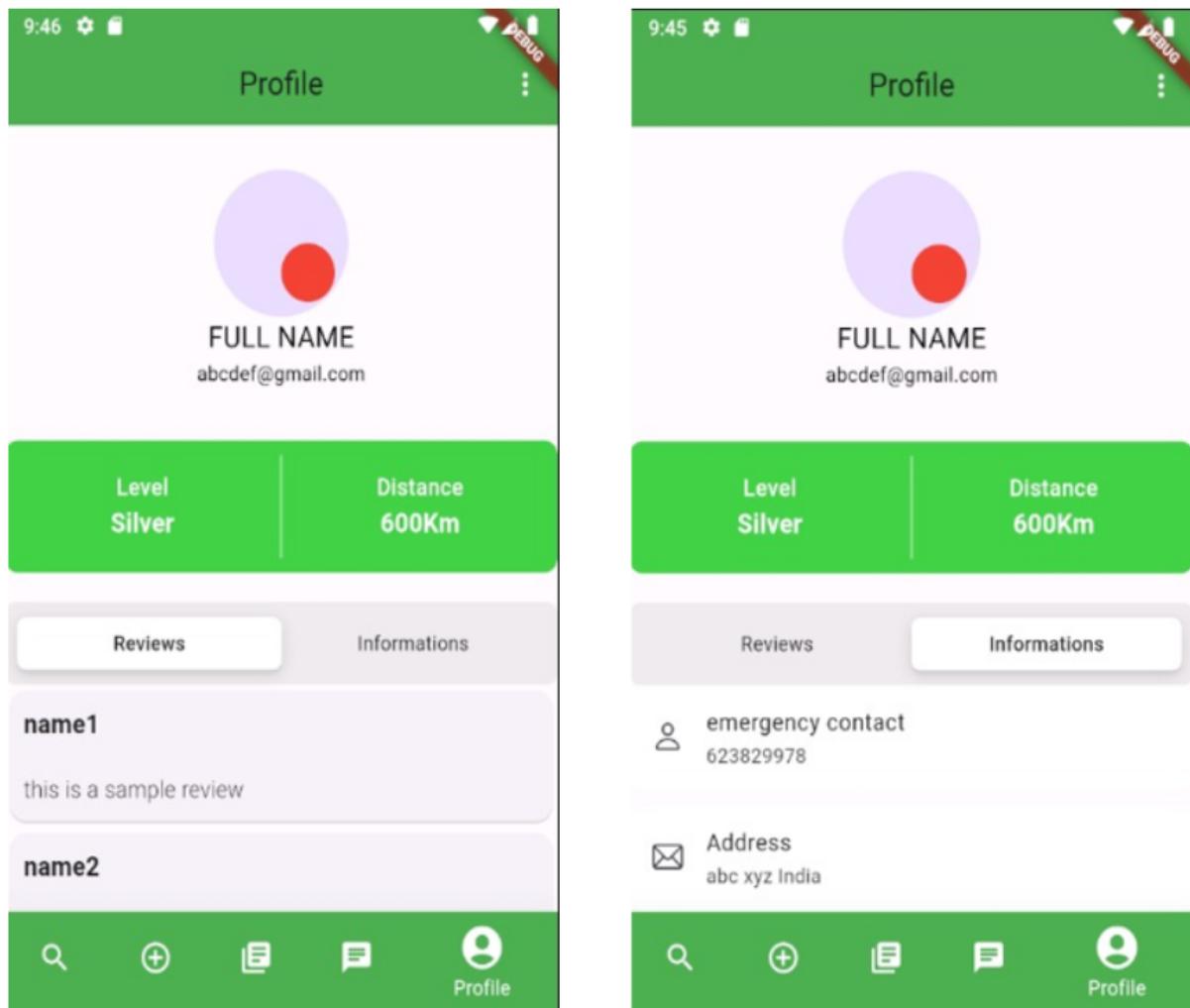
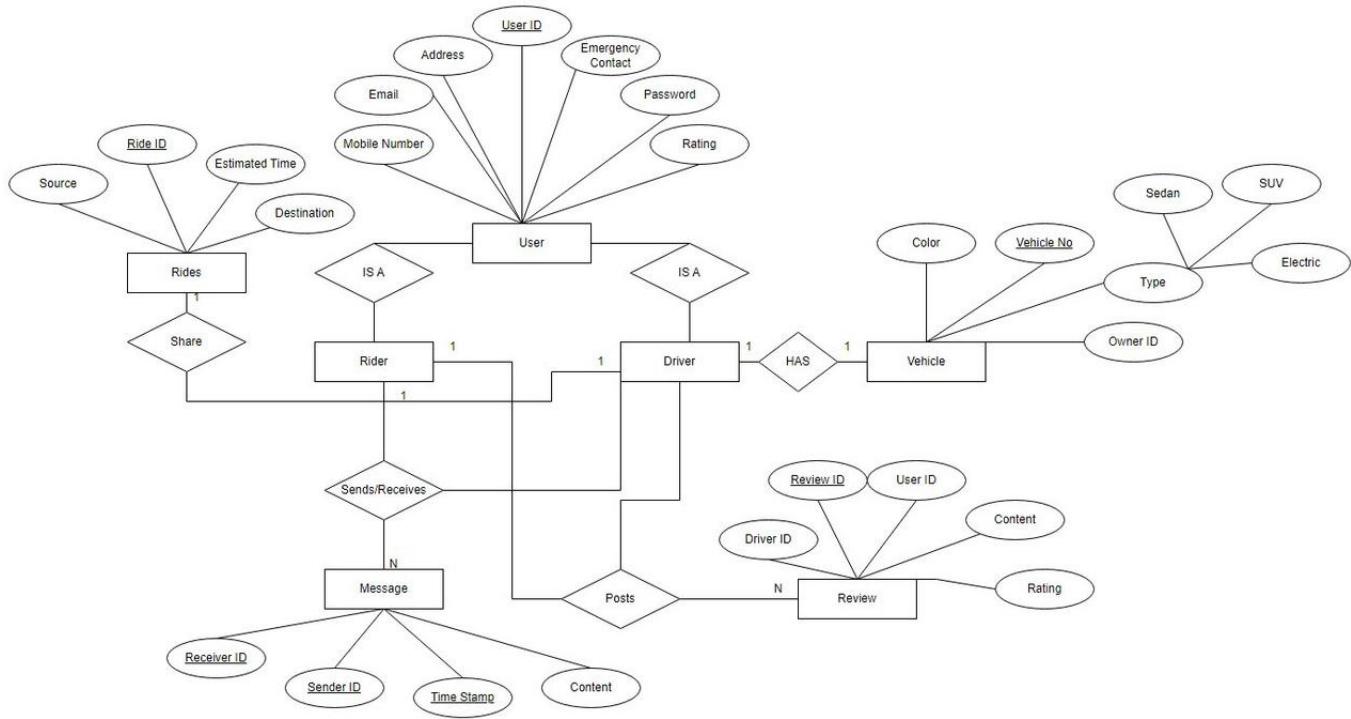


Figure 3.12: User Profile with details such as reviews, personal information, distance travelled and level awarded is shown.

3.5 Database Design



3.5.1 Database Choice: Firebase

This carpool app utilizes Firebase as its primary database solution. Firebase offers several advantages that make it well-suited for this application:

- **Scalability:** Firebase scales efficiently to accommodate a growing user base and increasing carpool listings.
- **Real-time Updates:** Firebase provides real-time data synchronization, ensuring users see the latest carpool information and updates instantly.
- **Security:** Firebase integrates robust security features to protect user data and ensure secure carpool management.
- **Cloud-based:** Firebase is a managed cloud service, eliminating the need for server maintenance on your end.

3.5.2 Database Schema

The carpool app leverages Firebase Firestore Database, a NoSQL document database, to store and manage carpool data. The database schema utilizes a hierarchical structure with the following collections:

- **Users:** This collection stores user profiles with information like:

- User ID (Document ID) (Unique Identifier)
- Name
- Age
- Bio
- Email address
- Mobile number
- Level
- Distance Travelled
- Rides as driver
- Rides as passenger
- Emergency contact

- **Rides:** This collection stores carpool listings with details like:

- Ride ID (Document ID) (unique identifier)
- Destination name
- Destination geopoint
- Start location name
- Start location geopoint
- Available seats
- Date
- Time
- Email

3.5.3 Schema Justification

- **User:** The User entity forms the core of the carpool system, storing user information for profiles, car details (drivers), and ratings.
- **Ride:** The Ride entity (or referenced connections) manages the many-to-many relationship between carpools and riders, tracking participation and ensuring accurate available seat information

3.6 Description of Implementation Strategies

These implementation strategies ensure that RideShare will be robust, user-friendly, and capable of delivering a seamless carpooling experience across multiple platforms.

3.6.0.1 Development Framework

Flutter: RideShare will be developed using Flutter, a popular open-source framework by Google for building beautiful, native-looking mobile applications for both iOS and Android platforms with a single codebase.

3.6.0.2 APIs and Services

- **Open Street Maps API:**
 - This API will be utilized to visually display carpool locations and routes within the app. Riders can view carpool locations on a map relative to their own position, and drivers can share their planned routes with riders using open-source map data.
- **Open Route Service API**
 - This API will provide routing and navigation functionalities within the app. It enables the calculation of optimal routes for carpool trips, allowing riders and drivers to see detailed route information, including distance, travel time, and step-by-step directions.
- **Firebase:** This suite of Google cloud services will be utilized for several key functionalities:

- **User Authentication:** Firebase Authentication provides a secure and reliable way for users to register and log in to the RideShare app. It supports various login methods like email/password and social login (e.g., Google Sign-In), offering flexibility and convenience to users.
- **Database:** Firebase Realtime Database or Cloud Firestore can be used to store user data (profiles, car details, etc.), carpool details (origin, destination, timings, etc.), and carpool activity data. This data will be essential for various functionalities within the app, including user profile management, carpool creation and searching, and rideshare analysis.

3.7 Module Division

The RideShare carpooling application functions through a cohesive network of four core modules: Registration and Authentication, Carpool Management, Rideshare Analysis, and Search Engine

1. Registration and Authentication Module This module serves as the entry point for users and ensures secure access to the RideShare app. It provides users with various methods for creating accounts and logging in. Users can choose traditional methods like username/email and password for registration, or leverage the convenience of social login options like Google Sign-In (if applicable). Existing users can seamlessly access the app using their chosen credentials (username/password or Google account), ensuring a smooth and efficient login experience. Regardless of the chosen login method, RideShare prioritizes user data privacy by implementing robust security measures to protect user information. This module is assigned to John Kurian.

Key Features:

- Secure user registration with various methods.
- Login functionality with chosen credentials.
- Prioritizes user data privacy regardless of the chosen login method.

2. Carpool Management Module This module is the heart of the RideShare app, empowering users to create and search for carpool opportunities. Drivers can create carpool offers by specifying details like origin, destination, date and the number of available

seats in their car. Riders, on the other hand, can leverage the search functionality to find carpools that match their desired trip details (origin, destination, date) and any specific requirements they may have. The module provides riders with the ability to filter search results based on various criteria, allowing them to find carpools that best suit their needs. To aid informed decision-making, the module also allows riders to view carpool details such as driver information, car details and any additional information included by the driver in their offer. This module is assigned to Kalidas Jayakumar.

Key Features:

- Drivers can create carpool offers with details (origin, destination, date, available seats).
- Riders can search for carpools based on trip details and preferences.
- Functionality to view carpool details (driver information) for informed decision making.

3. Rideshare Analysis Module This module goes beyond simply facilitating carpooling by measuring its positive impact on the environment and user behavior. It tracks carpool activity and calculates the total distance traveled through carpooling. By aggregating this data, the module can showcase the collective reduction in individual car usage achieved by the RideShare community. The Rideshare Analysis Module also plays a role in fostering a positive and trustworthy community within the app. It aggregates and analyzes user ratings for both riders and drivers. This data can be used to identify top-rated users and incentivize positive behavior within the community. For instance, users who consistently receive high ratings could be rewarded with badges or other perks, motivating users to maintain a positive carpooling experience for everyone. This module is assigned to Jerin Vincent.

Key Features:

- Analyzes total distance traveled through carpooling.
- Awards badges to users based on their contribution to carbon emission reduction.
- Aggregates and analyzes user ratings for riders and drivers.

4. Search Engine Module This module plays a critical role in connecting riders with suitable carpool options. It leverages location-based services to find carpools operating near the rider's origin, ensuring that riders are matched with carpools that are convenient and accessible. To further refine search results and improve matching accuracy, the module provides riders with search filters. These filters allow riders to narrow down their search based on criteria like date. By applying these filters, riders can find carpool options that perfectly align with their specific trip needs and preferences. The Search Engine Module also integrates with the Open Street Maps API to offer autofill suggestions for origin and destination searches. This streamlines the search process for riders by providing them with accurate and convenient location suggestions as they enter their trip details. By leveraging location-based services, search filters, and Open Street Maps integration, the Search Engine Module ensures a fast, efficient, and user-friendly carpool search experience for riders. This module is assigned to Dhiraj Bobby.

Key Features:

- Location-based search for carpools near the rider's origin.
- Search filters by date.
- Displays relevant carpool options based on search criteria and proximity.
- Integrates with Open Street Maps API for autofill suggestions on origin and destination.

3.8 Work Schedule - Gantt Chart

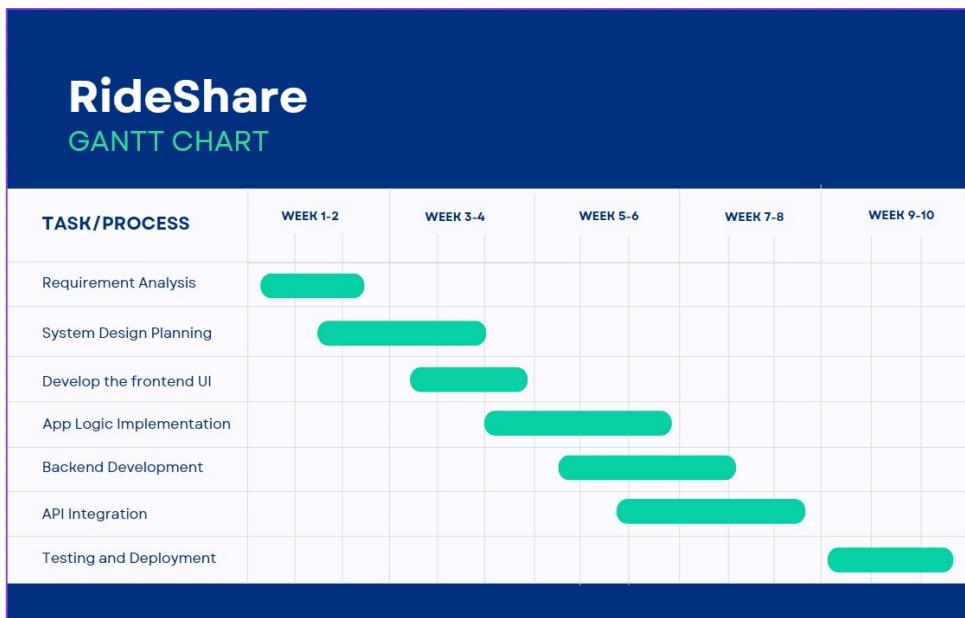


Figure 3.13: Gantt Chart

Chapter 4

Results and Discussions

4.1 Overview

The Rideshare project successfully achieved its primary objectives, resulting in a robust carpooling application that facilitates seamless interactions between riders and drivers. The app enables riders to search for available rides, send join requests, and view detailed route maps from their source to destination. On the driver's side, the app allows for the publication of rides, the ability to accept or reject rider requests, and comprehensive management of riders. Additionally, all users can manage their profiles effectively within the application. These functionalities have been implemented with a user-friendly interface, ensuring a smooth and efficient carpooling experience.

4.2 Testing

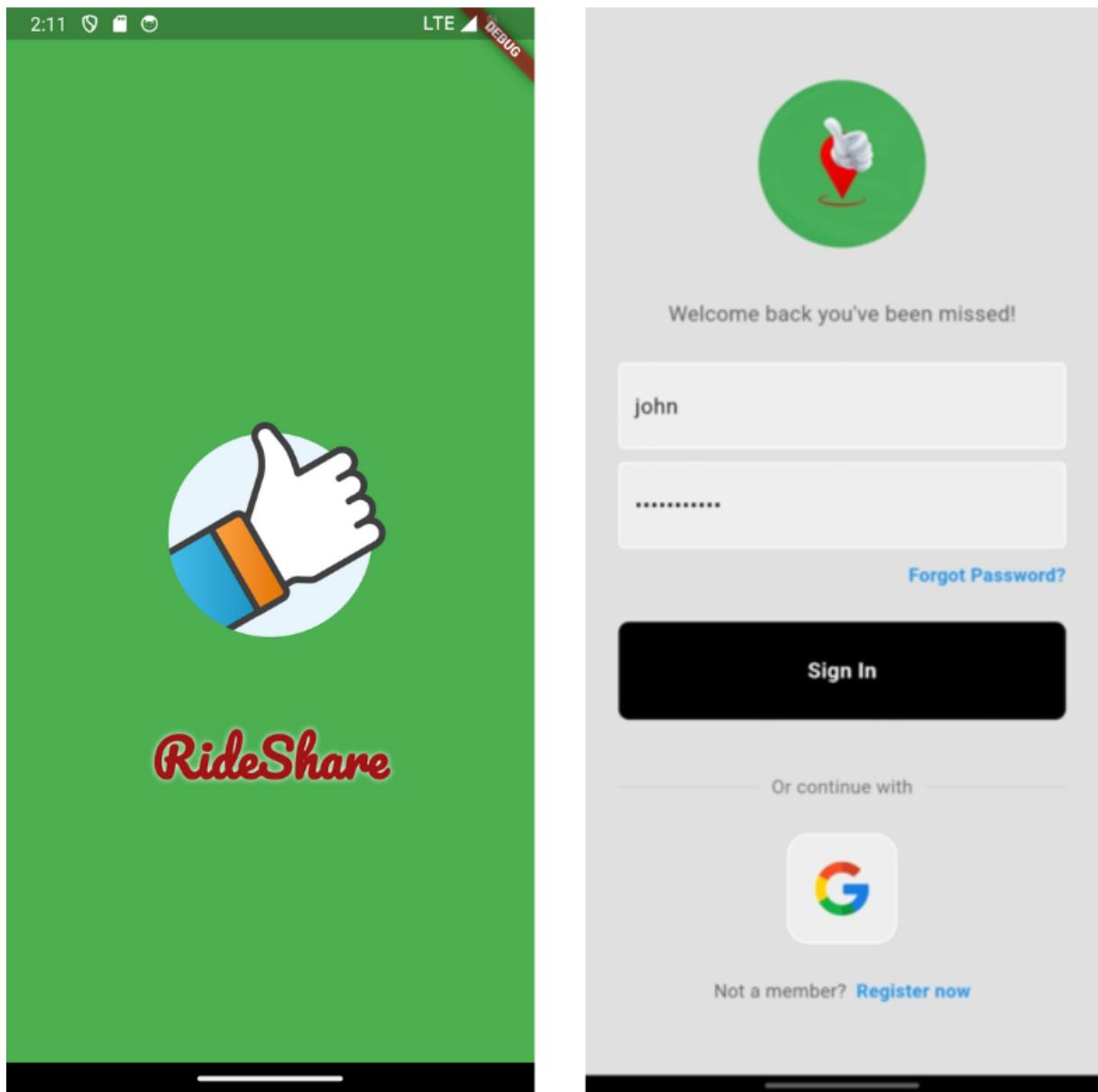


Figure 4.1: Splash Screen and Login Page

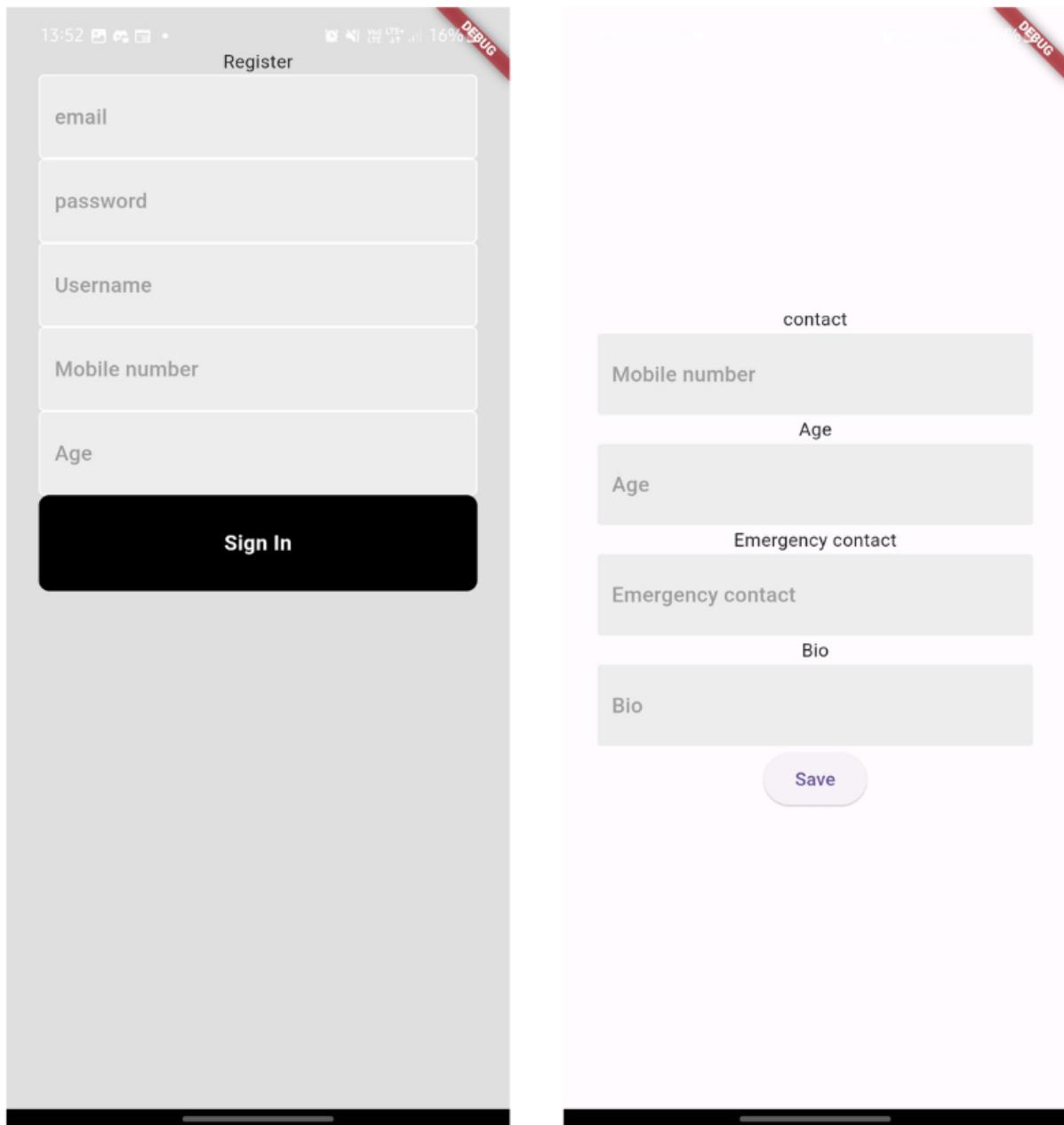


Figure 4.2: Register Page and Edit Profile

14:08 M ☀ 13% DEBUG

◀ kakkanad-vytilla
May 2024

May 16, 2024 2

07:00 AM Rajagiri School of
Engineering & Technology,...
08:00 AM Salem - Kochi -
Kanyakumari Road, Vytila...



Instance o...

★ 4.7

Request

May 15, 2024 1

07:00 AM Rajagiri School of
Engineering & Technology,...
08:00 AM Ration Shop, 1734155,
Seaport Airport Road, Iru...



Instance o...

★ 4.7

Request

Figure 4.3: Ride Select

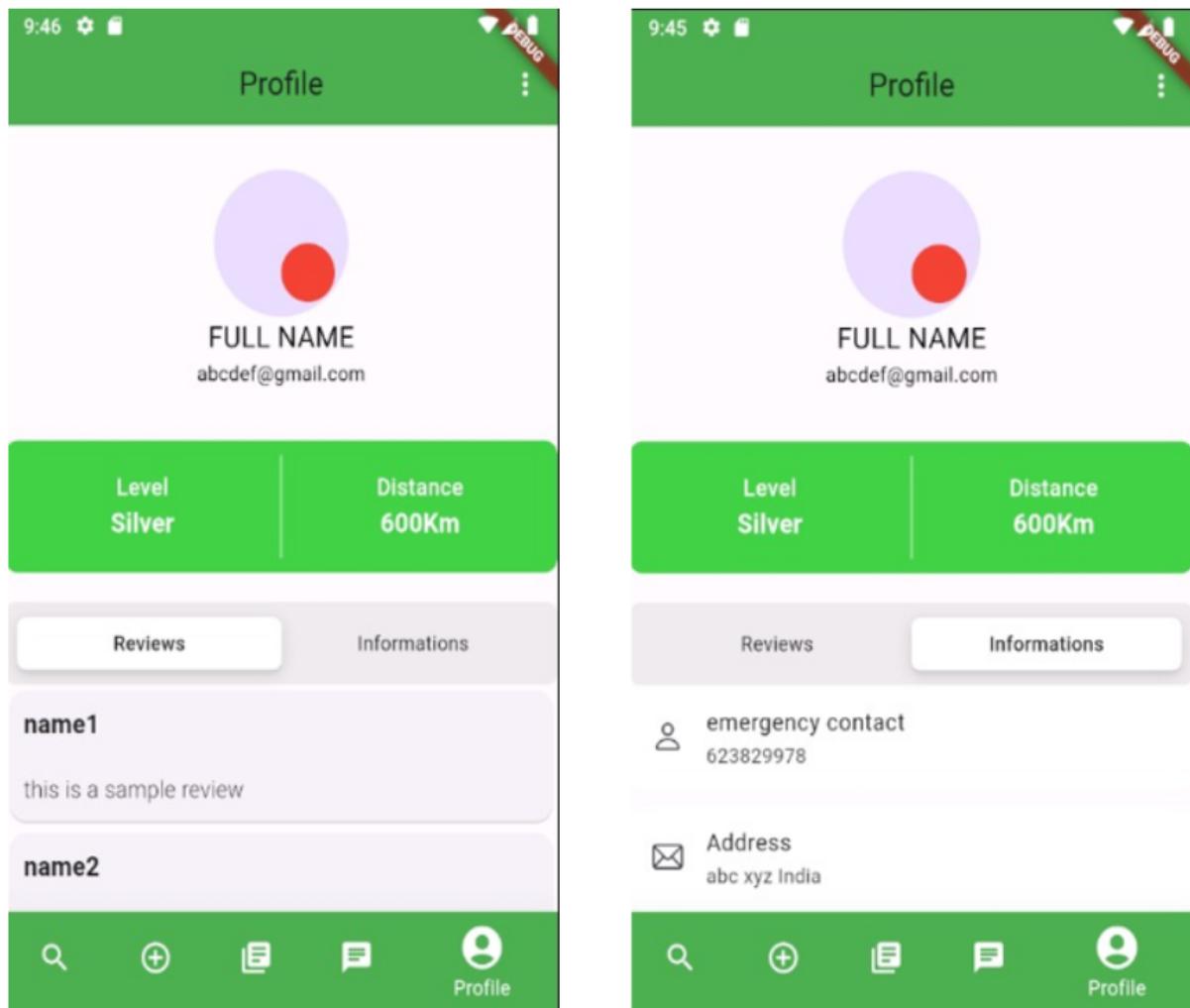


Figure 4.4: User Profile

8:26 PM | 6.9KB/s S



8:26 PM | 62.6KB/s S



Find a ride

Where

Leaving from... x ↑↓

Going to... x

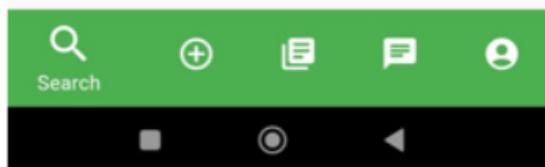
When

Pick a date x

Seats needed

- 1 +

Search



Find a ride

Where

Kakkanad x ↑↓

Vytilla x

When

Tuesday, April 9 x

Seats needed

- 3 +

Search

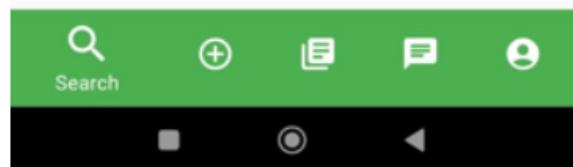


Figure 4.5: Search Page

2:30

LTE DEBUG

2:31

LTE DEBUG

Publish a ride

Where

Leaving from... X ↑↓

Going to... X

When

Pick a date X

Seats available

- 1 +

Publish



Publish a ride

Where

Abad Pepper Route, KB Jacc X ↑↓

Vytilla Flyover, Vytila, Ernakulam X

When

Monday, May 13 X

Seats available

- 2 +

Publish



Figure 4.6: Publish Page

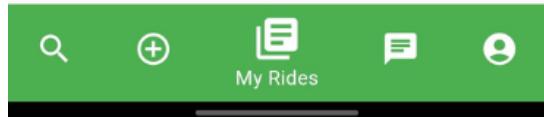
10:41 9% DEBUG

< My Activity



**Your future travel
plans will appear
here.**

Find the perfect ride from
thousands of destinations, or
publish to share your ride.



9:12 AM | 13.2KB/s S DEBUG

< My Activity

April 28, 2024

3

07:00 AM Bharananganam -
Pravithanam Road, ...

08:00 AM Lalam church road,
lalam Palam, Kottar...

Passengers

No passenger yet.

Requests:

No ride requests.

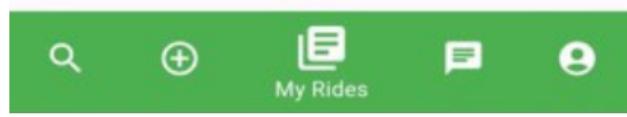


Figure 4.7: My Activity Page

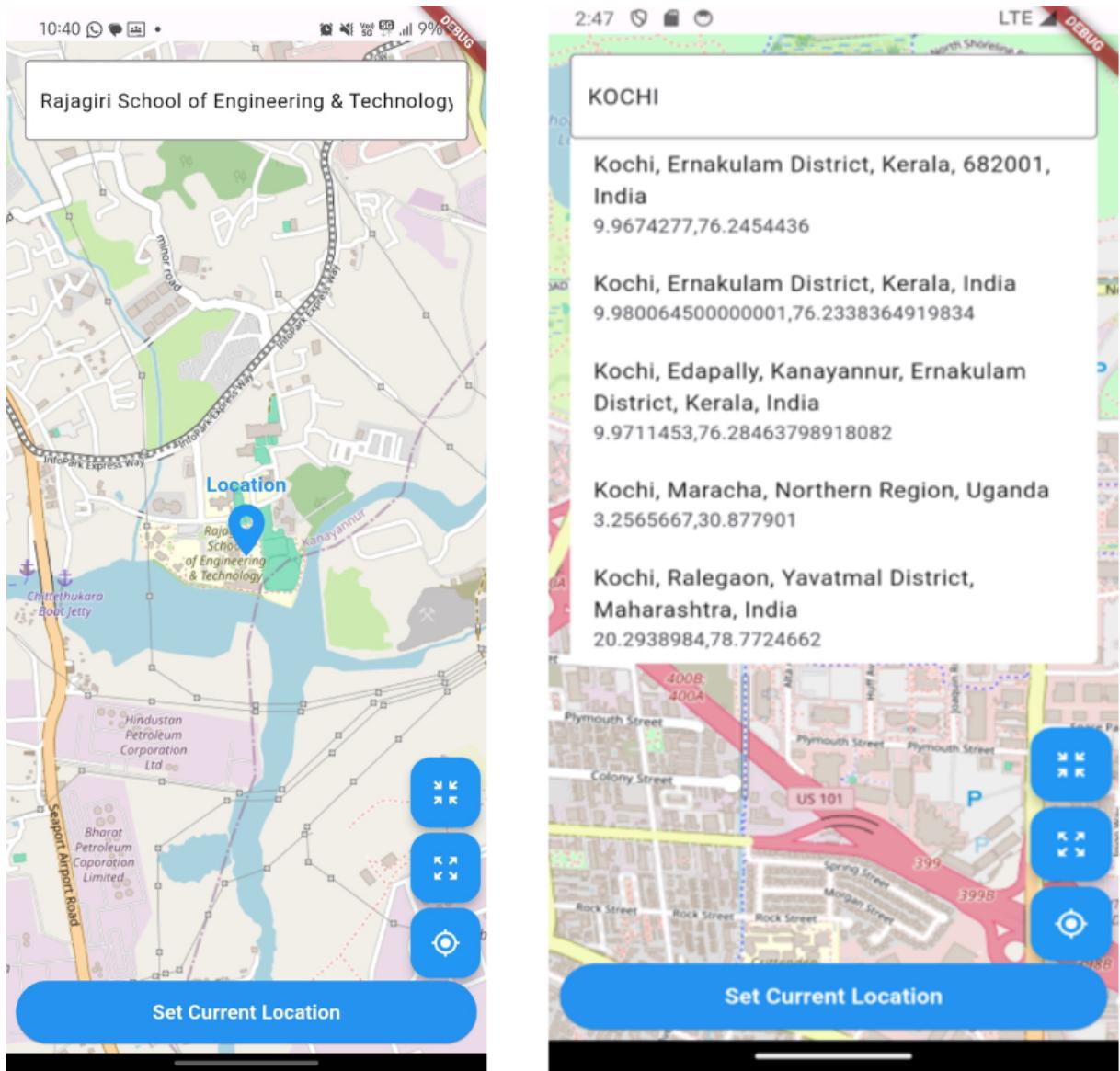


Figure 4.8: Search By Location

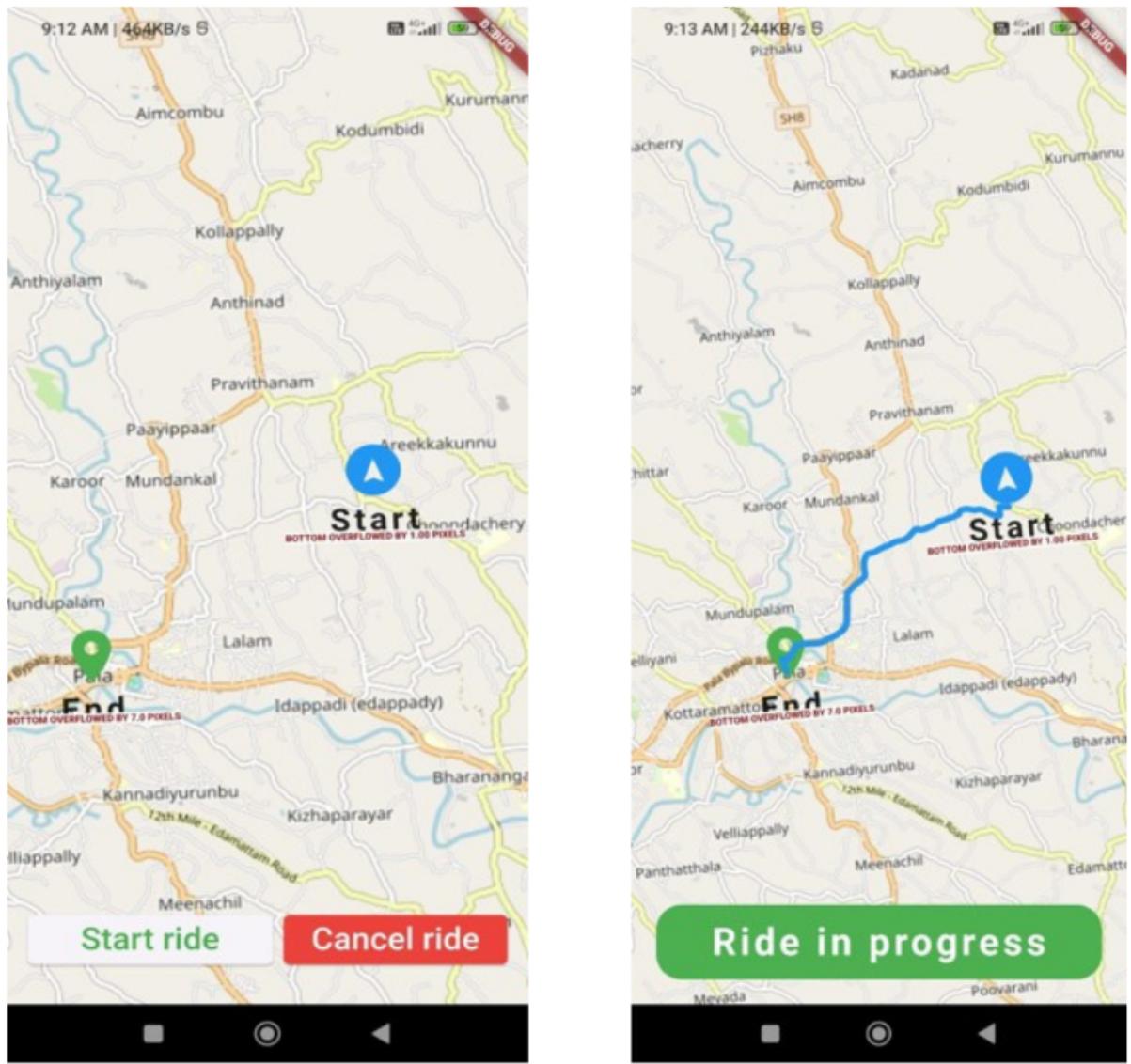


Figure 4.9: Route Navigation

4.3 Discussion

The comprehensive set of features implemented in the Rideshare app effectively meets the diverse needs of its users, resulting in a practical and user-friendly carpooling solution. The advanced filtering option, which limits ride searches to those within a 5-kilometer relative distance, ensures that riders find the most convenient and relevant options. This not only enhances user satisfaction but also increases the efficiency of the search process. The restriction for drivers to publish only one ride at a time, with the ability to post a new ride only after the current one concludes, helps manage ride availability and ensures that drivers focus on fulfilling their commitments, thereby maintaining service quality.

The capability for drivers to accept or reject ride requests and manage riders adds a layer of control and security, which is crucial for building trust in the service. The route map feature provides clear navigation and enhances transparency, making it easier for both parties to plan their journey. Profile management functionalities further personalize the user experience, allowing for better interaction and communication within the app.

While the implementation has been largely successful, minor deviations were observed, such as occasional delays in ride request processing and the need for further optimization in route mapping algorithms. These issues are being actively addressed in subsequent updates to improve overall performance and user satisfaction. Despite these minor challenges, the Rideshare app stands out as a highly effective tool for modern carpooling, fulfilling its intended goals and providing significant value to its users.

Chapter 5

Conclusion

5.1 Conclusion

The project has achieved its primary objectives by creating a platform that is both practical and user-friendly. It addresses the common challenges of carpooling with features that promote security, reliability, and convenience. While minor issues such as occasional delays in ride request processing have been identified, these are being systematically addressed through ongoing updates. Overall, the Rideshare app has proven to be an effective and efficient tool for facilitating carpooling, offering a sustainable and convenient alternative to traditional commuting methods. With ongoing improvements and user feedback, the app is well-equipped to continuously meet the evolving needs of its users.

5.2 Future Scope

The future scope of the Rideshare project includes several potential enhancements to further improve user experience and service quality. Implementing real-time messaging and notifications will facilitate seamless communication between riders and drivers during rides. Integrating secure in-app payment solutions for tipping drivers will streamline the gratuity process, fostering a positive experience for both parties. Furthermore, introducing a system where drivers can publish a daily route, rather than having to publish each ride individually everyday, will enhance convenience and efficiency for regular commuters. These extensions will significantly enhance the app's functionality and user satisfaction, positioning it for continued success and growth.

Bibliography

- [1] Napoli, M.L., 2019. *Beginning Flutter: A Hands-on Guide to App Development*. John Wiley and Sons.
- [2] Tashildar, A., Shah, N., Gala, R., Giri, T., & Chavhan, P., 2020. *Application Development Using Flutter*. International Research Journal of Modernization in Engineering Technology and Science, 2(8), 1262-1266.
- [3] Dagne, L., 2019. *Flutter for Cross-Platform App and SDK Development*.
- [4] Pawar, S., Vishwakarma, A., Desai, J., Kashyap, S., & Pawar, T., 2021. *Development of Mobile App for Cab Booking and Rental Using Flutter SDK*.

Appendix A: Presentation

RideShare

DESIGN PRESENTATION

Dr. Saritha S

Dhiraj Bobby
Jerin Vincent
John Kurian
Kalidas Jayakumar

Contents

- Introduction
- Problem Definition
- Objectives
- System Overview
- System Architecture
- UI Design
- Database Design
- Work Division – Gantt Chart
- Software/Hardware Requirements
- Conclusion
- References

Introduction

- RideShare is an Android app facilitating carpooling to reduce traffic, offer cost-effective travel, and promote environmental responsibility by minimizing your carbon footprint.
- *Fewer Cars, Less Traffic:* RideShare carpooling reduces the number of vehicles on the road, leading to smoother commutes and a greener environment.
- *Budget-Friendly Travel:* Offers a cost-effective alternative to taxis and ride-hailing services, saving you money on every trip.

Problem Definition

- Developing a mobile application (Android) to facilitate carpooling and reduce traffic congestion.
- This app will connect riders and drivers for cost-effective travel and promote a more sustainable transportation system.

Objectives

1. Allow users to register and login securely.
2. Enable users to search for available carpools and create their own.
3. Integrate with mapping services for route navigation and visualization.
4. Implement a communication system for riders and drivers to coordinate pick-up/drop-off details.
5. Implement features to enhance user safety during carpooling.
6. Incorporate a user rating system to promote trust and accountability.
7. Facilitate user location services to streamline carpool searching.

Scope and Relevance

- Connects riders & drivers: Matches riders seeking affordable travel with drivers offering empty seats.
- Seamless carpooling: Enables booking carpools, sharing available seats, and suggesting efficient routes.
- The main focuses are to reduce traffic congestion, offer cost-effective travel, and promote carpooling.
- Initial scope excludes advanced features like multi-stop rides, social networking, and public transport integration.
- Focuses on core functionalities: Prioritizes core carpooling experience with potential for future expansion.

System Overview

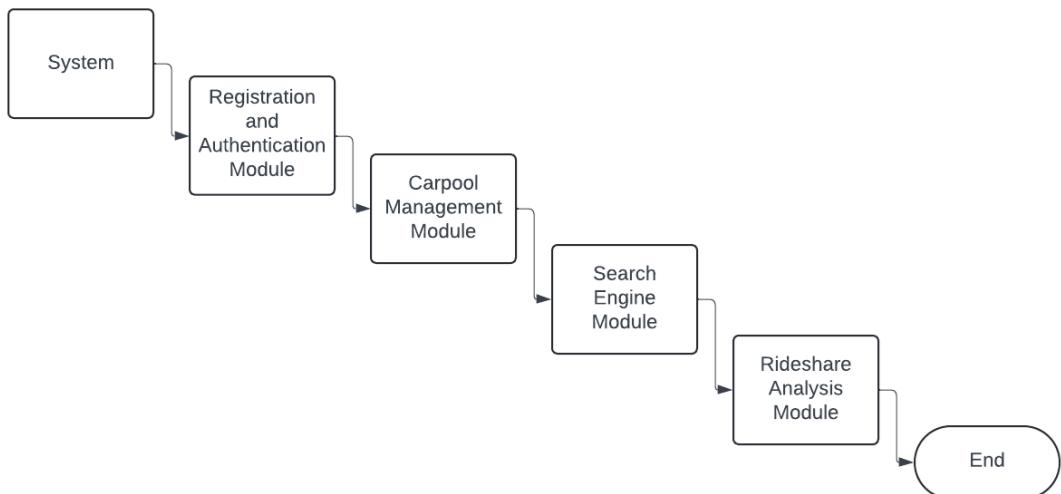
1. Secure User Management handles user registration, login, and profile management, ensuring a secure foundation for the carpooling community.
2. Carpool Management is carried out efficiently to connects riders and drivers for shared journeys.
3. RideShare simplifies carpool discovery with location-based search, filters, and smart suggestions.
4. RideShare tracks carpool impact, rewarding eco-conscious users and promoting a positive community through data analysis.

14/5/2024

RideShare

7

System Overview



14/5/2024

RideShare

8

Registration and Authentication Module

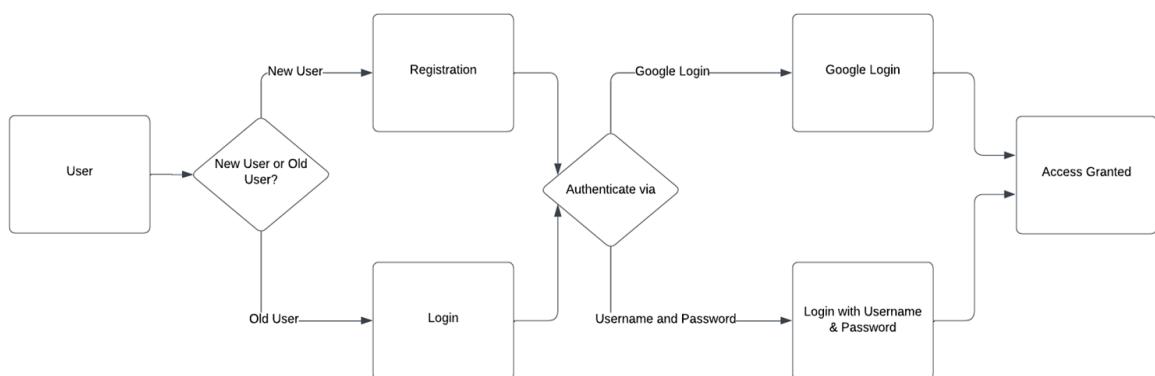
- RideShare offers secure user registration with options like username/email/password and Google Sign-In (if applicable), ensuring user convenience.
- Existing users can seamlessly access the app using their chosen method (login credentials or Google Sign-In), promoting a smooth and efficient login experience.
- RideShare prioritizes user data privacy regardless of the chosen login method.

14/5/2024

RideShare

9

Registration and Authentication Module



14/5/2024

RideShare

10

Carpool management module

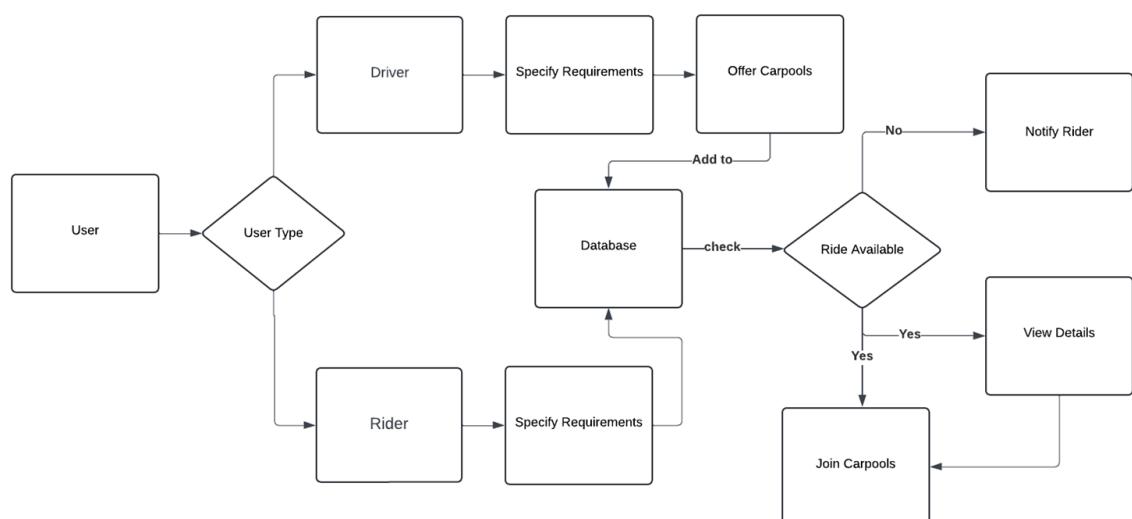
- Drivers can offer carpools by specifying origin, destination, date, time, and available seats.
- Riders can join carpools based on their desired trip details and requirements(origin, destination, date, time).
- Functionality to view carpool details (driver information, car details, etc.) for informed decisions.

14/5/2024

RideShare

11

Carpool management module

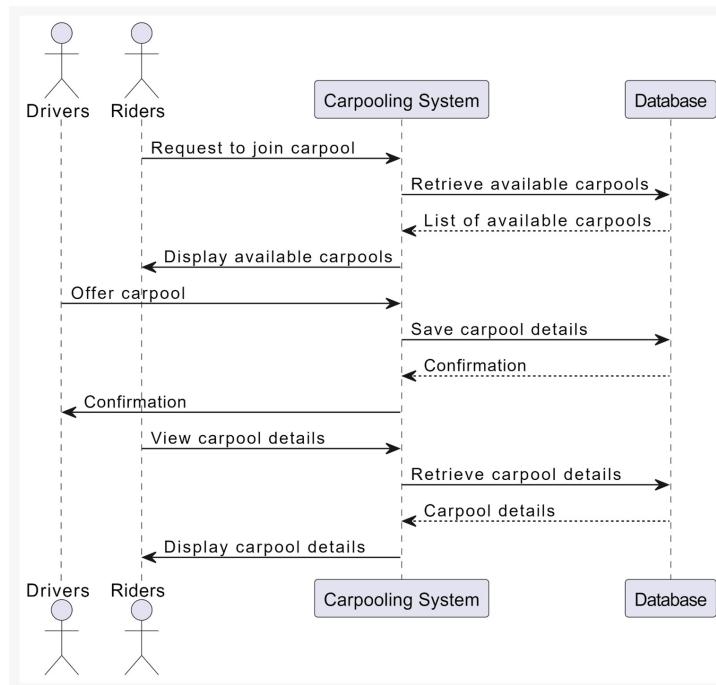


14/5/2024

RideShare

12

Carpool management module



14/5/2024

RideShare

13

Search engine module

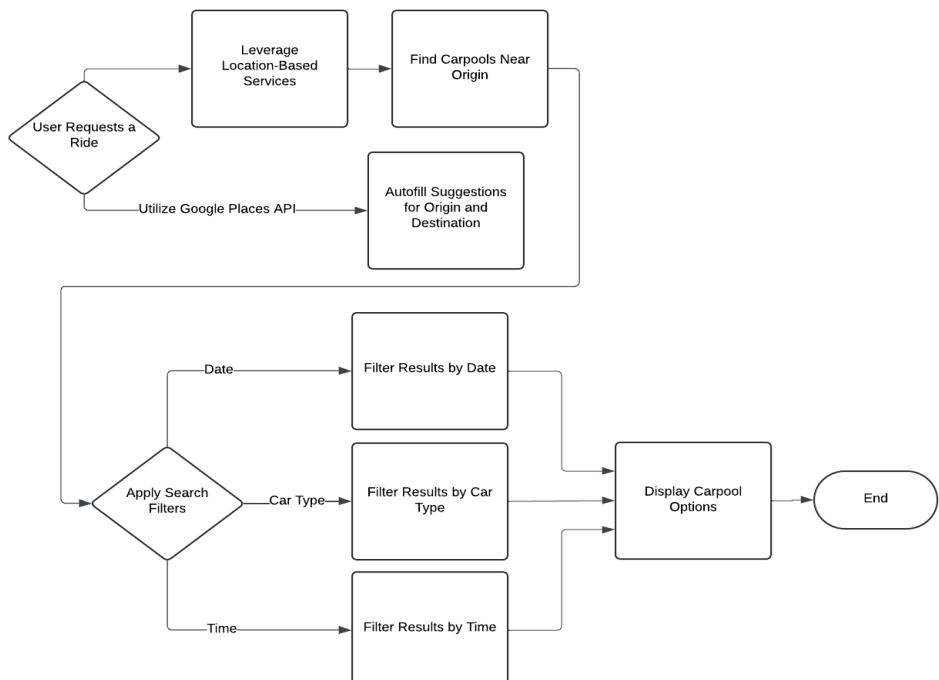
- Leverages location-based services to find carpools near the rider's origin.
- Search filters allow riders to refine their search by date, time, and car type (if applicable).
- Displays relevant carpool options based on search criteria and proximity.
- RideShare leverages Google Places API for autofill suggestions, streamlining origin and destination searches for riders.

14/5/2024

RideShare

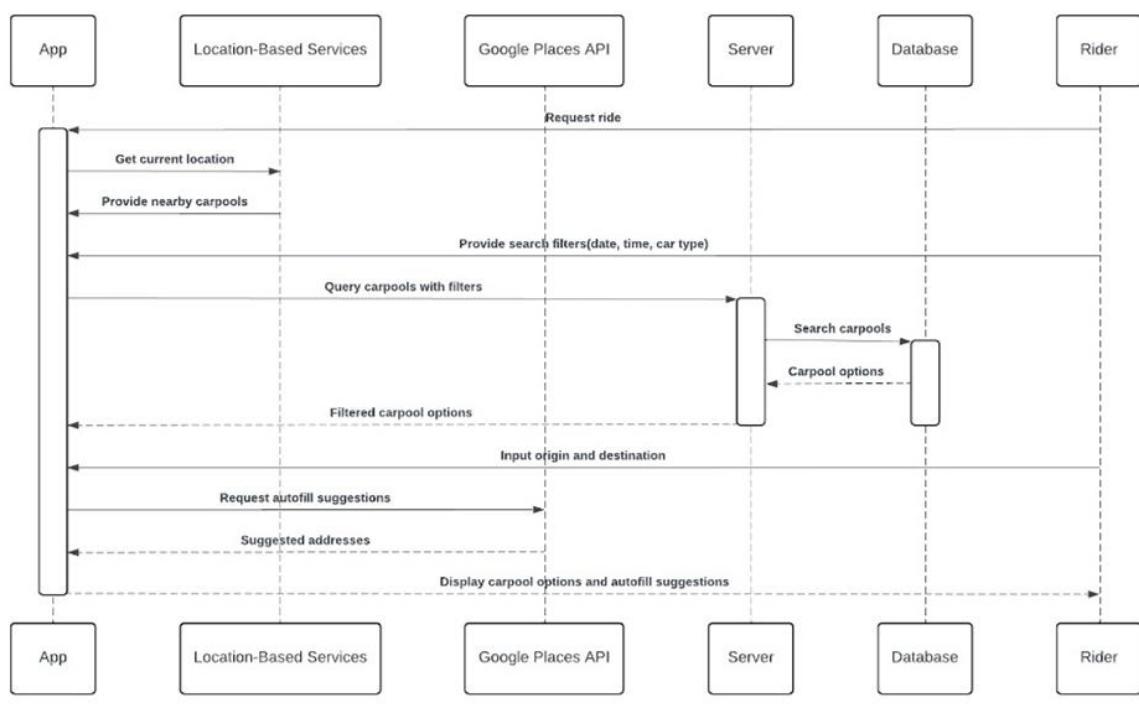
14

Search engine module



1

Search engine module



Rideshare Analysis module

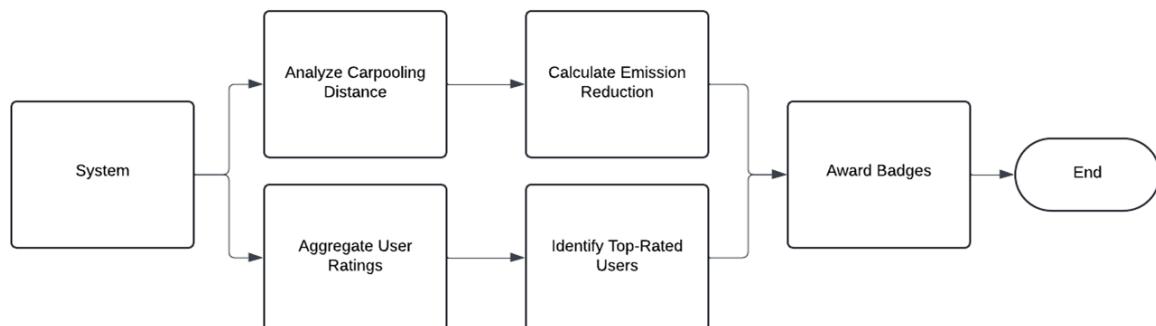
- Analyzes the total distance traveled through carpooling, showcasing the collective impact on reducing individual car usage.
- Awards badges to users based on their contribution to carbon emission reduction through total carpooled distance.
- Aggregates and analyzes user ratings for both riders and drivers. This data can be used to identify top-rated users and incentivize positive behavior within the community.

14/5/2024

RideShare

17

Rideshare Analysis module

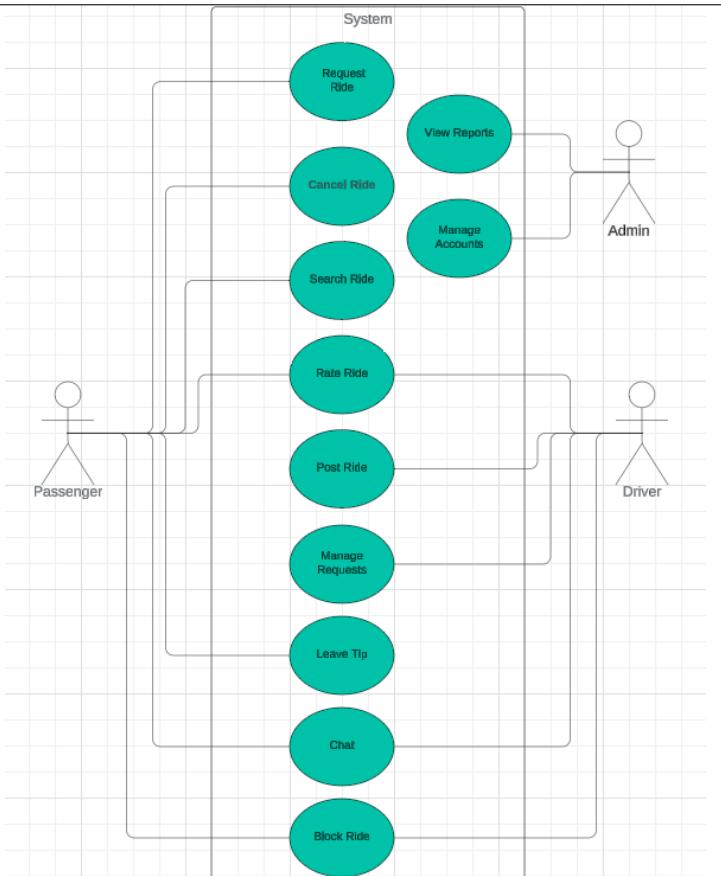


14/5/2024

RideShare

18

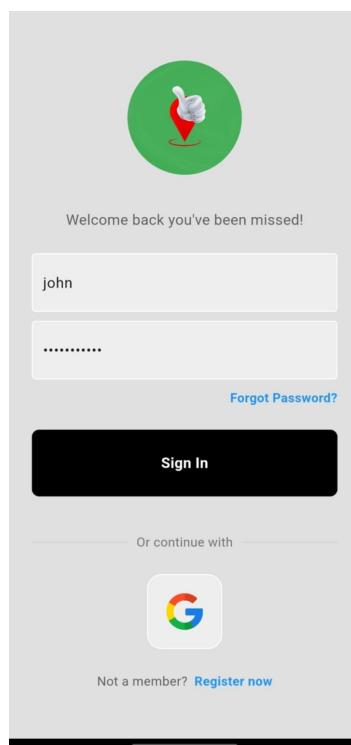
USE CASE DIAGRAM



14/5/2024

19

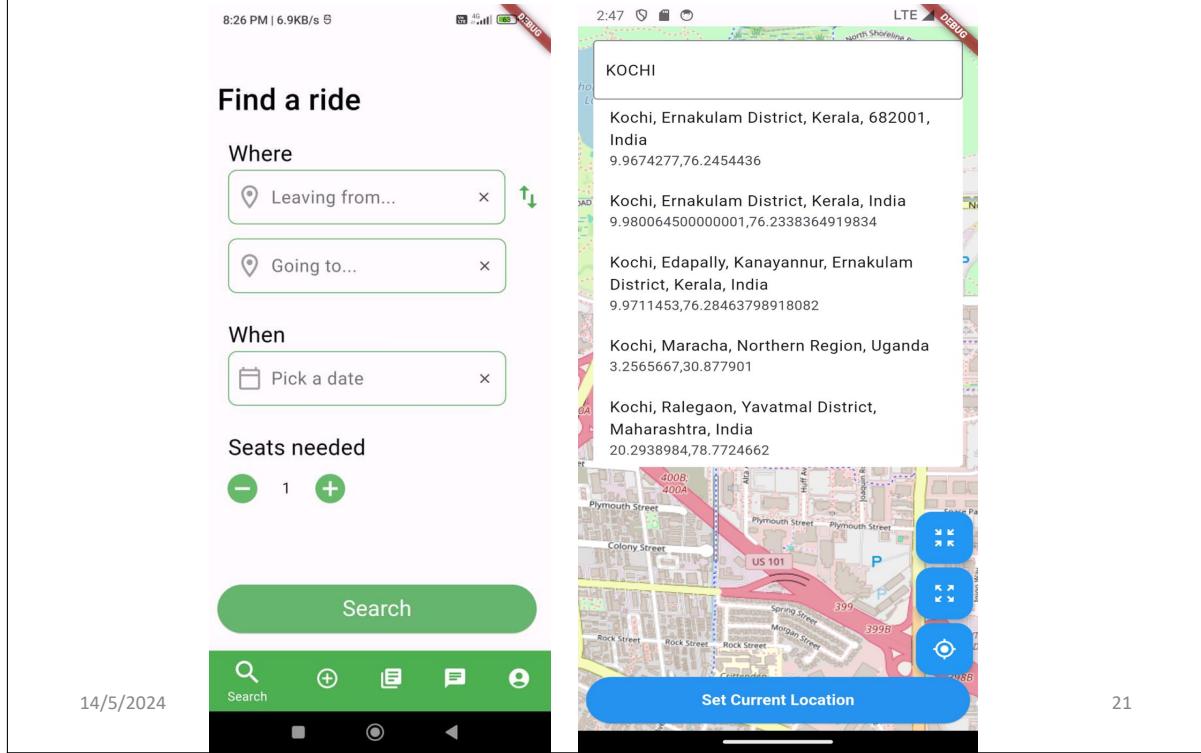
LOGIN PAGE UI



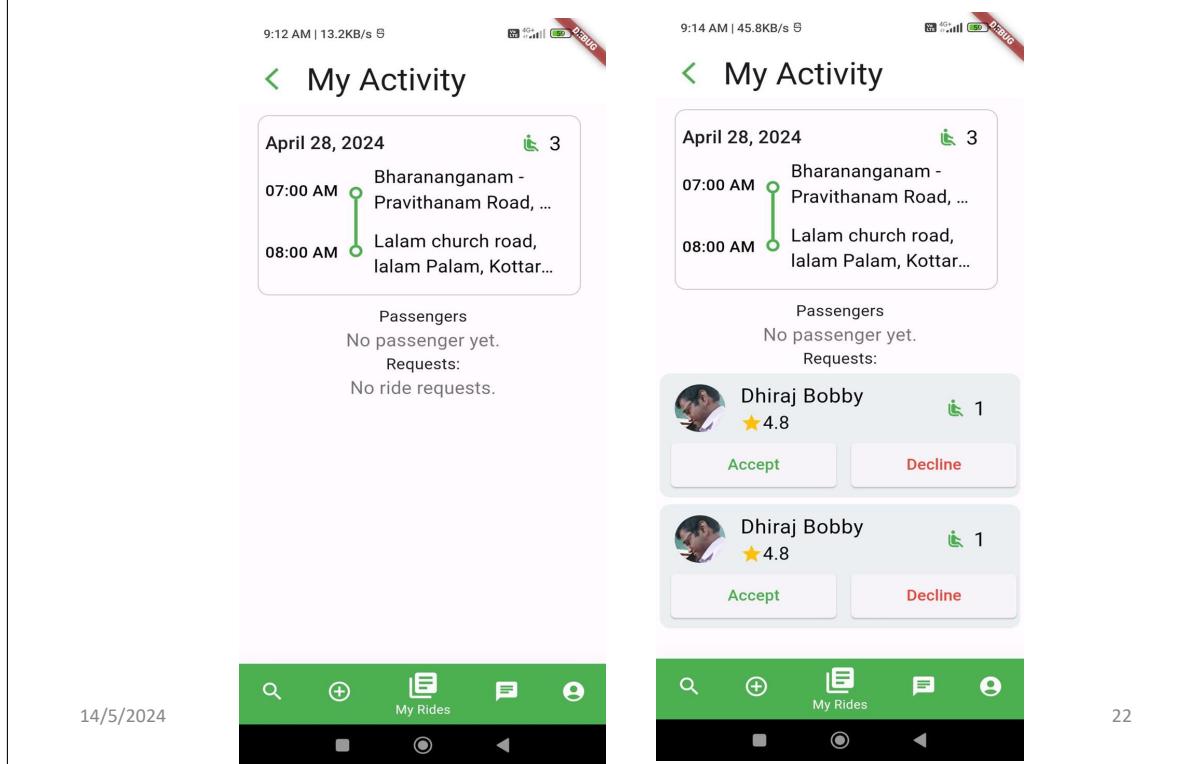
14/5/2024

20

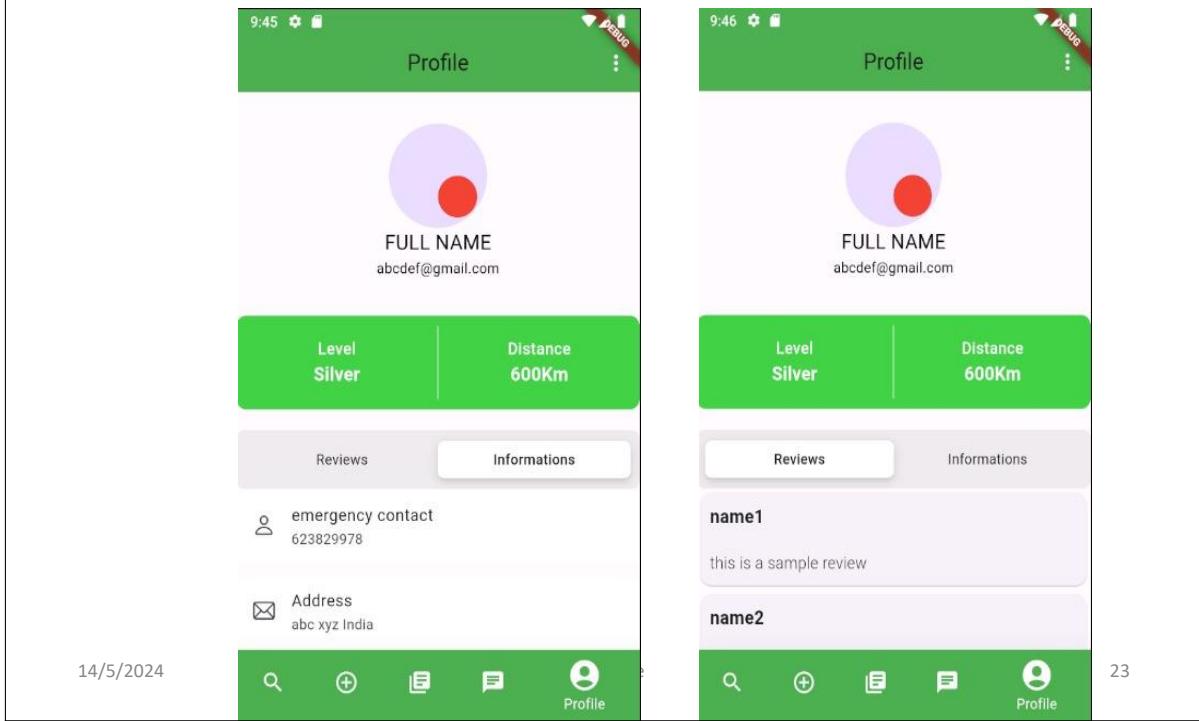
SEARCH PAGE UI



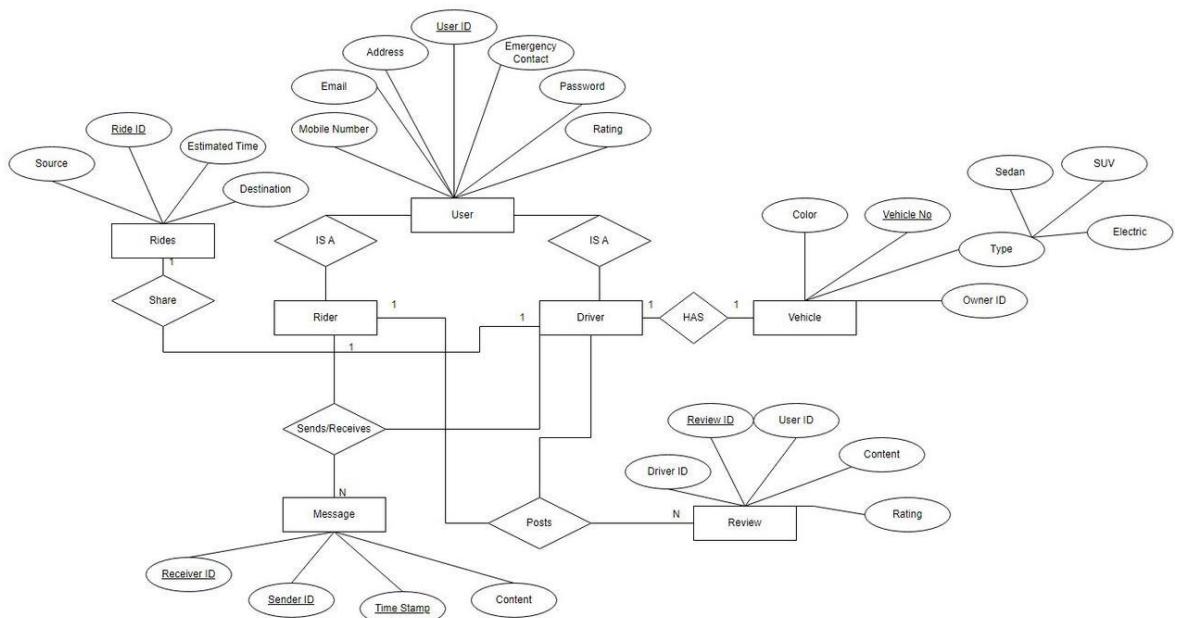
MY ACTIVITY UI



PROFILE PAGE UI



Database Design



Work Division



Software/ Hardware Requirements

Software Requirements:

- Flutter
- Dart Programming Language
- Firebase
- Google Maps API
- Location-based APIs (Geolocation API, Places API, etc.)
- Flutter Packages
- Payment Gateway API

Software/ Hardware Requirements

Hardware Requirements:

- Android OS 7.0+
- 4GB RAM
- 8 GB storage
- GPS
- Wi-Fi
- Mobile data
- Snapdragon 600 series or equivalent processor
- 720x1280 display

Conclusion

- Streamline your commute with RideShare. Our user-friendly carpooling app connects riders and drivers, letting you join or create carpools for a faster, smoother journey.
- Find carpools that match your needs, with real-time notifications, clear route visualization, and easy tracking for a seamless experience.
- Reduce traffic, save money, and contribute to a greener environment by carpooling with RideShare.

Future Enhancements

Enhanced Communication: Enable real-time messaging and notifications between riders and drivers, facilitating seamless interaction during the ride.

Integrated Payment Solutions: Implement a secure in-app payment service for tipping drivers, streamlining the gratuity process and fostering a positive experience for both riders and drivers.

Future Enhancements

User Feedback and Review System: Create an intuitive feedback mechanism allowing users to easily leave reviews for riders and drivers.

Utilize rating systems, comment sections, and incentivized feedback to maintain service quality and cultivate a trustworthy community.

References

- Napoli, M.L., 2019. Beginning flutter: a hands on guide to app development. John Wiley & Sons.
- Tashildar, A., Shah, N., Gala, R., Giri, T. and Chavhan, P., 2020. Application development using flutter. International Research Journal of Modernization in Engineering Technology and Science, 2(8), pp.1262-1266.
- Dagne, L., 2019. Flutter for cross-platform App and SDK development.
- Pawar, S., Vishwakarma, A., Desai, J., Kashyap, S. and Pawar, T., 2021. Development of Mobile App for Cab Booking and Rental using Flutter SDK.

Appendix B: Vision, Mission, Programme Outcomes and Course Outcomes

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
RAJAGIRI SCHOOL OF ENGINEERING & TECHNOLOGY (AUTONOMOUS)
RAJAGIRI VALLEY, KAKKANAD, KOCHI, 682039
(Affiliated to APJ Abdul Kalam Technological University)**



Vision, Mission, Programme Outcomes and Course Outcomes

Institute Vision

To evolve into a premier technological institution, moulding eminent professionals with creative minds, innovative ideas and sound practical skill, and to shape a future where technology works for the enrichment of mankind.

Institute Mission

To impart state-of-the-art knowledge to individuals in various technological disciplines and to inculcate in them a high degree of social consciousness and human values, thereby enabling them to face the challenges of life with courage and conviction.

Department Vision

To become a centre of excellence in Computer Science and Engineering, moulding professionals catering to the research and professional needs of national and international organizations.

Department Mission

To inspire and nurture students, with up-to-date knowledge in Computer Science and Engineering, ethics, team spirit, leadership abilities, innovation and creativity to come out with solutions meeting societal needs.

Programme Outcomes (PO)

Engineering Graduates will be able to:

- 1. Engineering Knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
- 2. Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
- 3. Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
- 4. Conduct investigations of complex problems:** Use research-based knowledge including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
- 5. Modern Tool Usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
- 6. The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal, and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
- 7. Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
- 8. Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
- 9. Individual and Team work:** Function effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings.

10. Communication: Communicate effectively with the engineering community and with society at large. Be able to comprehend and write effective reports documentation. Make effective presentations, and give and receive clear instructions.

11. Project management and finance: Demonstrate knowledge and understanding of engineering and management principles and apply these to one's own work, as a member and leader in a team. Manage projects in multidisciplinary environments.

12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change.

Programme Specific Outcomes (PSO)

A graduate of the Computer Science and Engineering Program will demonstrate:

PSO1: Computer Science Specific Skills

The ability to identify, analyze and design solutions for complex engineering problems in multidisciplinary areas by understanding the core principles and concepts of computer science and thereby engage in national grand challenges.

PSO2: Programming and Software Development Skills

The ability to acquire programming efficiency by designing algorithms and applying standard practices in software project development to deliver quality software products meeting the demands of the industry.

PSO3: Professional Skills

The ability to apply the fundamentals of computer science in competitive research and to develop innovative products to meet the societal needs thereby evolving as an eminent researcher and entrepreneur.

Course Outcomes

After the completion of the course the student will be able to:

CO1:

Identify technically and economically feasible problems (Cognitive Knowledge Level: Apply)

CO2:

Identify and survey the relevant literature for getting exposed to related solutions and get familiarized with software development processes (Cognitive Knowledge Level: Apply)

CO3:

Perform requirement analysis, identify design methodologies and develop adaptable & reusable solutions of minimal complexity by using modern tools & advanced programming techniques (Cognitive Knowledge Level: Apply)

CO4:

Prepare technical report and deliver presentation (Cognitive Knowledge Level: Apply)

CO5:

Apply engineering and management principles to achieve the goal of the project (Cognitive Knowledge Level: Apply)

Appendix C: CO-PO-PSO Mapping

COURSE OUTCOMES:

After completion of the course the student will be able to

SL. NO	DESCRIPTION	Blooms' Taxonomy Level
CO1	Identify technically and economically feasible problems (Cognitive Knowledge Level: Apply)	Level 3: Apply
CO2	Identify and survey the relevant literature for getting exposed to related solutions and get familiarized with software development processes (Cognitive Knowledge Level: Apply)	Level 3: Apply
CO3	Perform requirement analysis, identify design methodologies and develop adaptable & reusable solutions of minimal complexity by using modern tools & advanced programming techniques (Cognitive Knowledge Level: Apply)	Level 3: Apply
CO4	Prepare technical report and deliver presentation (Cognitive Knowledge Level: Apply)	Level 3: Apply
CO5	Apply engineering and management principles to achieve the goal of the project (Cognitive Knowledge Level: Apply)	Level 3: Apply

CO-PO AND CO-PSO MAPPING

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PS O3
C O1	3	3	3	3		2	2	3	2	2	2	3	2	2	2
C O2	3	3	3	3	3	2		3	2	3	2	3	2	2	2
C O3	3	3	3	3	3	2	2	3	2	2	2	3			2
C O4	2	3	2	2	2			3	3	3	2	3	2	2	2
C O5	3	3	3	2	2	2	2	3	2		2	3	2	2	2

3/2/1: high/medium/low

JUSTIFICATIONS FOR CO-PO MAPPING

MAPPING	LOW/ MEDIUM/ HIGH	JUSTIFICATION
101003/CS6 22T.1-PO1	HIGH	Identify technically and economically feasible problems by applying the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
101003/CS6 22T.1-PO2	HIGH	Identify technically and economically feasible problems by analysing complex engineering problems reaching substantiated conclusions using first principles of mathematics.
101003/CS6 22T.1-PO3	HIGH	Design solutions for complex engineering problems by identifying technically and economically feasible problems.
101003/CS6 22T.1-PO4	HIGH	Identify technically and economically feasible problems by analysis and interpretation of data.
101003/CS6 22T.1-PO6	MEDIUM	Responsibilities relevant to the professional engineering practice by identifying the problem.
101003/CS6 22T.1-PO7	MEDIUM	Identify technically and economically feasible problems by understanding the impact of the professional engineering solutions.
101003/CS6 22T.1-PO8	HIGH	Apply ethical principles and commit to professional ethics to identify technically and economically feasible problems.
101003/CS6 22T.1-PO9	MEDIUM	Identify technically and economically feasible problems by working as a team.
101003/CS6 22T.1-PO10	MEDIUM	Communicate effectively with the engineering community by identifying technically and economically feasible problems.
101003/CS6 22T.1-P011	MEDIUM	Demonstrate knowledge and understanding of engineering and management principles by selecting the technically and economically feasible problems.
101003/CS6 22T.1-PO12	HIGH	Identify technically and economically feasible problems for long term learning.
101003/CS6 22T.1-PSO1	MEDIUM	Ability to identify, analyze and design solutions to identify technically and economically feasible problems.
101003/CS6 22T.1-PSO2	MEDIUM	By designing algorithms and applying standard practices in software project development and Identifying technically and economically feasible problems.
101003/CS6 22T.1-PSO3	MEDIUM	Fundamentals of computer science in competitive research can be applied to Identify technically and economically feasible problems.
101003/CS6 22T.2-PO1	HIGH	Identify and survey the relevant by applying the knowledge of mathematics, science, engineering fundamentals.

101003/CS6 22T.2-PO2	HIGH	Identify, formulate, review research literature, and analyze complex engineering problems get familiarized with software development processes.
101003/CS6 22T.2-PO3	HIGH	Design solutions for complex engineering problems and design based on the relevant literature.
101003/CS6 22T.2-PO4	HIGH	Use research-based knowledge including design of experiments based on relevant literature.
101003/CS6 22T.2-PO5	HIGH	Identify and survey the relevant literature for getting exposed to related solutions and get familiarized with software development processes by using modern tools.
101003/CS6 22T.2-PO6	MEDIUM	Create, select, and apply appropriate techniques, resources, by identifying and surveying the relevant literature.
101003/CS6 22T.2-PO8	HIGH	Apply ethical principles and commit to professional ethics based on the relevant literature.
101003/CS6 22T.2-PO9	MEDIUM	Identify and survey the relevant literature as a team.
101003/CS6 22T.2-PO10	HIGH	Identify and survey the relevant literature for a good communication to the engineering fraternity.
101003/CS6 22T.2-PO11	MEDIUM	Identify and survey the relevant literature to demonstrate knowledge and understanding of engineering and management principles.
101003/CS6 22T.2-PO12	HIGH	Identify and survey the relevant literature for independent and lifelong learning.
101003/CS6 22T.2-PSO1	MEDIUM	Design solutions for complex engineering problems by Identifying and survey the relevant literature.
101003/CS6 22T.2-PSO2	MEDIUM	Identify and survey the relevant literature for acquiring programming efficiency by designing algorithms and applying standard practices.
101003/CS6 22T.2-PSO3	MEDIUM	Identify and survey the relevant literature to apply the fundamentals of computer science in competitive research.
101003/CS6 22T.3-PO1	HIGH	Perform requirement analysis, identify design methodologies by using modern tools & advanced programming techniques and by applying the knowledge of mathematics, science, engineering fundamentals.
101003/CS6 22T.3-PO2	HIGH	Identify, formulate, review research literature for requirement analysis, identify design methodologies and develop adaptable & reusable solutions.

101003/CS6 22T.3-PO3	HIGH	Design solutions for complex engineering problems and perform requirement analysis, identify design methodologies.
101003/CS6 22T.3-PO4	HIGH	Use research-based knowledge including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
101003/CS6 22T.3-PO5	HIGH	Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools.
101003/CS6 22T.3-PO6	MEDIUM	Perform requirement analysis, identify design methodologies and assess societal, health, safety, legal, and cultural issues.
101003/CS6 22T.3-PO7	MEDIUM	Understand the impact of the professional engineering solutions in societal and environmental contexts and Perform requirement analysis, identify design methodologies and develop adaptable & reusable solutions.
101003/CS6 22T.3-PO8	HIGH	Perform requirement analysis, identify design methodologies and develop adaptable & reusable solutions by applying ethical principles and commit to professional ethics.
101003/CS6 22T.3-PO9	MEDIUM	Function effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings.
101003/CS6 22T.3-PO10	MEDIUM	Communicate effectively with the engineering community and with society at large to perform requirement analysis, identify design methodologies.
101003/CS6 22T.3-PO11	MEDIUM	Demonstrate knowledge and understanding of engineering requirement analysis by identifying design methodologies.
101003/CS6 22T.3-PO12	HIGH	Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change by analysis, identify design methodologies and develop adaptable & reusable solutions.
101003/CS6 22T.3-PSO3	MEDIUM	The ability to apply the fundamentals of computer science in competitive research and prior to that perform requirement analysis, identify design methodologies.
101003/CS6 22T.4-PO1	MEDIUM	Prepare technical report and deliver presentation by applying the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
101003/CS6 22T.4-PO2	HIGH	Identify, formulate, review research literature, and analyze complex engineering problems by preparing technical report and deliver presentation.

101003/CS6 22T.4-PO3	MEDIUM	Prepare Design solutions for complex engineering problems and create technical report and deliver presentation.
101003/CS6 22T.4-PO4	MEDIUM	Use research-based knowledge including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions and prepare technical report and deliver presentation.
101003/CS6 22T.4-PO5	MEDIUM	Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools and Prepare technical report and deliver presentation.
101003/CS6 22T.4-PO8	HIGH	Prepare technical report and deliver presentation by applying ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
101003/CS6 22T.4-PO9	HIGH	Prepare technical report and deliver presentation effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings.
101003/CS6 22T.4-PO10	HIGH	Communicate effectively with the engineering community and with society at large by prepare technical report and deliver presentation.
101003/CS6 22T.4-PO11	MEDIUM	Demonstrate knowledge and understanding of engineering and management principles and apply these to one's own work by prepare technical report and deliver presentation.
101003/CS6 22T.4-PO12	HIGH	Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change by prepare technical report and deliver presentation.
101003/CS6 22T.4-PSO1	MEDIUM	Prepare a technical report and deliver presentation to identify, analyze and design solutions for complex engineering problems in multidisciplinary areas.
101003/CS6 22T.4-PSO2	MEDIUM	To acquire programming efficiency by designing algorithms and applying standard practices in software project development and to prepare technical report and deliver presentation.
101003/CS6 22T.4-PSO3	MEDIUM	To apply the fundamentals of computer science in competitive research and to develop innovative products to meet the societal needs by preparing technical report and deliver presentation.
101003/CS6 22T.5-PO1	HIGH	Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
101003/CS6 22T.5-PO2	HIGH	Identify, formulate, review research literature, and analyze complex engineering problems by applying engineering and management principles to achieve the goal of the project.

101003/CS6 22T.5-PO3	HIGH	Apply engineering and management principles to achieve the goal of the project and to design solutions for complex engineering problems and design system components or processes that meet the specified needs.
101003/CS6 22T.5-PO4	MEDIUM	Apply engineering and management principles to achieve the goal of the project and use research-based knowledge including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
101003/CS6 22T.5-PO5	MEDIUM	Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools and to apply engineering and management principles to achieve the goal of the project.
101003/CS6 22T.5-PO6	MEDIUM	Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal, and cultural issues and the consequent responsibilities by applying engineering and management principles to achieve the goal of the project.
101003/CS6 22T.5-PO7	MEDIUM	Understand the impact of the professional engineering solutions in societal and environmental contexts, and apply engineering and management principles to achieve the goal of the project.
101003/CS6 22T.5-PO8	HIGH	Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice and to use the engineering and management principles to achieve the goal of the project.
101003/CS6 22T.5-PO9	MEDIUM	Function effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings and to apply engineering and management principles to achieve the goal of the project.
101003/CS6 22T.5-PO11	MEDIUM	Demonstrate knowledge and understanding of engineering and management principles and apply these to one's own work, as a member and leader in a team. Manage projects in multidisciplinary environments and to apply engineering and management principles to achieve the goal of the project.
101003/CS6 22T.5-PO12	HIGH	Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change and to apply engineering and management principles to achieve the goal of the project.
101003/CS6 22T.5-PSO1	MEDIUM	The ability to identify, analyze and design solutions for complex engineering problems in multidisciplinary areas. Apply engineering and management principles to achieve the goal of the project.

101003/CS6 22T.5-PSO2	MEDIUM	The ability to acquire programming efficiency by designing algorithms and applying standard practices in software project development to deliver quality software products meeting the demands of the industry and to apply engineering and management principles to achieve the goal of the project.
101003/CS6 22T.5-PSO3	MEDIUM	The ability to apply the fundamentals of computer science in competitive research and to develop innovative products to meet the societal needs thereby evolving as an eminent researcher and entrepreneur and apply engineering and management principles to achieve the goal of the project.

