GoTogether: Collaborative ride-sharing for college convenience

A Main Project submitted
in partial fulfillment of the requirements
for the award of the degree of

BACHELOR OF TECHNOLOGY
In
COMPUTER SCIENCE AND ENGINEERING

Submitted by

1. **P. Ganesh (21PA1A05D1)**
2. **J. L. V. S. Ganesh (22PA5A0510)**
3. **K. Vivek Vardhan (21PA1A0572)**
4. **K. Praneeth Varma (21PA1A0573)**

Under the esteemed guidance of
Mrs. B. Sridevi
Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
VISHNU INSTITUTE OF TECHNOLOGY
(Autonomous)
(Approved by AICTE, Accredited by NBA & NAAC and permanently affiliated to JNTU Kakinada)
BHIMAVARAM – 534 202
2024 – 2025

VISHNU INSTITUTE OF TECHNOLOGY

(Autonomous)

(Approved by AICTE, Accredited by NBA & NAAC and permanently affiliated to JNTU Kakinada)

BHIMAVARAM-534202

2024-2025

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project entitled "**GOTOGETHER: COLLABORATIVE RIDE-SHARING FOR COLLEGE CONVENIENCE**", is being submitted by ***P. GANESH, J. L. V. S. GANESH, K. VIVEK VARDHAN, K. PRANEETH VARMA*** bearing the **REGD.NOS: 21PA1A05D1, 22PA5A0510, 21PA1A0572, 21PA1A0573** submitted in fulfilment for the award of the degree of "**BACHELOR OF TECHNOLOGY"** in "**COMPUTER SCIENCE AND ENGINEERING"** is a record of work carried out by them under my guidance and supervision during the academic year 2024-2025 and it has been found worthy of acceptance according to the requirements of the university.

Internal Guide	**Head of the Department**
Mrs. B .Sridevi	Dr. R Srinivasa Raju

External Examiner

ACKNOWLEDGEMENT

It is natural and inevitable that the thoughts and ideas of other people tend to drift into the subconscious due to various human parameters, where one feels to acknowledge the help and guidance derived from others. We acknowledge each of those who have contributed to the fulfillment of this project.

We take the opportunity to express our sincere gratitude to **Dr. M. Venu,** Principal**,** VIT, Bhimavaram for allowing us to avail all the facilities available in the college.

We are very much thankful to **Dr. R. Srinivasa Raju**, Head of the Department, Department of Computer Science and Engineering for his continuous and unrelenting support and guidance. We thank and acknowledge our gratitude for the valuable guidance and support expended to us right from the conception of the idea to the completion of this project.

We are very much thankful to **Mrs. B. Sridevi**, Assistant Professor, our internal guide whose guidance from time to time helped us to complete this project successfully.

Project Associates

1. P. Ganesh (21PA1A05D1)
2. J. L. V. S. Ganesh (22PA5A0510)
3. K. Vivek Vardhan (21PA1A0572)
4. K. Praneeth Varma (21PA1A0573)

ABSTRACT

Efficient commuting remains a persistent challenge for college students, as traditional public transport systems often fail to address their specific needs. Many students struggle with unreliable transport schedules, safety concerns, and the high cost of private transportation, leading to daily commuting difficulties. To address this issue, our project introduces a mobile-based ride-sharing platform exclusively designed for college communities. The platform focuses on improving last-mile connectivity by providing an affordable, secure, and student-friendly solution.

Our project leverages modern cross-platform technologies, with Flutter used for the frontend and Firebase handling backend services, ensuring a seamless and responsive user experience. A key aspect of the platform is its gender-based ride matching feature, allowing users to opt for same-gender ride partners, thereby enhancing safety and comfort. Additionally, the system enforces college-specific access control, requiring users to verify their student identity through email authentication. This ensures that only legitimate students from a particular institution can participate in the ride-sharing network, creating a trusted environment for all users.

Another important feature is direct campus drop-offs, which help students reach their academic buildings more conveniently, minimizing the distance they need to walk from entry gates. By integrating these features, the platform significantly improves accessibility while reducing travel time and cost. The project also contributes to sustainability by encouraging shared rides, thereby reducing the number of individual vehicles on the road and lowering carbon emissions.

Through an extensive review of existing literature and a detailed analysis of current transportation challenges, our project identifies key gaps in existing solutions and demonstrates the effectiveness of a college-focused approach. Unlike general ride-sharing applications, our platform is exclusively designed for students, ensuring a safe and affordable commuting option tailored to their unique requirements. Experimental evaluations confirm that the proposed system enhances commuter safety, optimizes ride availability, and reduces travel expenses while fostering a sense of community within educational institutions.

TABLE OF CONTENTS

TABLE OF CONTENTS

LIST OF FIGURES

INTRODUCTION

1. INTRODUCTION

Commuting challenges among college students often result in delays, absenteeism, and inefficiencies in daily travel. Many students struggle with unreliable public transport, high travel costs, and safety concerns, while others with personal vehicles travel alone, leaving transportation resources underutilized. To address these issues, this project proposes a dedicated ride-sharing platform exclusively for college students, ensuring safe, affordable, and efficient commuting through a user-friendly mobile application.

1.1 Motivation

The motivation behind the “GoTogether: Collaborative ride-sharing for college convenience” is the challenge of commuting between home and college. Commutes are crucial aspect of student life. However, many students face difficulties due to inefficient public transportation systems and the unavailability of personal vehicles. Frequent delays, overcrowding, and high travel costs create significant obstacles to timely and convenient travel. In a country like India, where the student population in higher education is growing rapidly, these transportation issues become even more pressing. While some students own personal vehicles, they often travel alone, leading to underutilization of transportation resources. There is a need for an efficient, cost-effective, and safe ride-sharing solution specifically designed for college students.

1.2 Problem Statement

Public transportation inefficiencies significantly affect students' ability to attend classes regularly and on time. Research indicates that students traveling long distances often experience severe traffic congestion, high transit costs, and frequent delays, leading to increased absenteeism and academic disruptions. Additionally, conventional ride-sharing platforms do not address the unique safety concerns of students, particularly in terms of gender-specific preferences. There is a lack of a dedicated, student-centric ride-sharing system that ensures safety, affordability, and efficiency while addressing last-mile connectivity issues.

1.3 Objective of the Project

The primary objective of this project is to develop a mobile-based ride-sharing platform exclusively for college students, facilitating secure and efficient commuting. Our project focuses on 'student-centric last-mile' problem which refers to the difficulty of reaching a final destination from a transportation hub, such as a bus or railway station, despite easy access to central locations.

The key goals of the system include:

- Connecting students who own vehicles (riders) with students needing transportation (passengers).
- Implementing gender-based ride-matching to enhance safety and comfort.
- Providing an affordable and reliable alternative to public transportation.
- Reducing traffic congestion and promoting sustainable commuting practices.
- Ensuring security and trust by restricting access to verified college students.

1.4 Scope

The proposed platform will serve as a specialized ride-sharing application tailored for college students. The system will:

- Utilize Flutter for cross-platform mobile application development.
- Integrate Firebase for user authentication and database management.
- Offer a user-friendly interface to facilitate easy ride-matching.
- Prioritize safety through gender-based matching and verified student authentication.
- Improve last-mile connectivity by connecting students with similar commuting routes.

By leveraging modern technologies and user-centric design, this platform aims to transform student commuting into a more secure, efficient, and sustainable experience.

LITERATURE SURVEY

2. LITERATURE SURVEY

Literature survey is the most important step in the software development process. Before developing the tool, it is necessary to determine the time factor, economy, and company strength. Once these things are satisfied, the next steps are to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool, programmers need a lot of external support. This support can be obtained from senior programmers, books, or websites. Before building the system, the above considerations are considered for developing the proposed system.

2.1 RESEARCH PAPERS

Chatterjee et al., Urban areas in India are facing serious challenges, with rapid urbanization leading to heavy congestion, increased air pollution, and rising road fatalities. These issues are compounded by inefficiencies in public transport, which often leave commuters frustrated. A key part of this problem is last-mile connectivity (LMC), which refers to the journey people take from transit hubs, like metro stations, to their final destinations such as homes or workplaces. Poor infrastructure for this last segment discourages many from using public transit, forcing them to rely on private vehicles instead. This not only worsens congestion but also creates sustainability challenges that need to be addressed.

Meshrama et al. has highlighted significant safety challenges in ride-sharing services that affect women in India. High crime rates, coupled with deep-seated cultural fears, discourage women from sharing rides with male strangers leading to lower participation rates. Factors such as rides with male strangers and nighttime travel are consistently rated as unsafe. Women felt safer driving(control over the vehicle) than being passengers. The study highlights the cultural and systemic barriers to women's mobility in India. By addressing safety through verified formal systems, tech-driven safeguards, and gender-inclusive policies, ride-sharing can become a viable, sustainable option. The findings align with global trends but highlight unique challenges in developing economies, emphasizing the need for contextual solutions.

Natarajan et al. present a compelling work on the sexual victimization of female college students during their commutes, using a “whole journey” framework that examines risks during walking, waiting, and riding phases. Their research, based on a survey of female students at John Jay College, reveals that up to 77% of respondents have experienced some form of harassment from staring and verbal abuse to stalking and physical assault underscoring a critical safety issue in public transport environments. Drawing on crime opportunity theory and situational crime prevention, the study highlights how factors such as male presence and nighttime travel exacerbate vulnerability, while also noting that existing measures like women-only autos and panic buttons have not fully addressed the problem. This work provides valuable insights into the intersection of urban mobility and gendered safety concerns, and its recommendations offer a framework for developing more secure public transport systems.

Chowdhury et al. , presents a practical approach to addressing the acute transportation challenges faced by students in developing countries, particularly in Dhaka. The authors explained how severe congestion, costly and overcrowded public transport, and pressing safety concerns, especially among female commuters can be mitigated through a dedicated ride-sharing platform for academic institutions. Drawing on a survey from North South University, the paper highlights that a significant number of seats in daily commutes remain unused, and students are keen on sharing rides if proper security measures are in place. The proposed system emphasizes strict verification using university email registration and document checks to ensure that only authenticated students participate. Overall, the platform aims to optimize existing transportation resources, reduce traffic congestion and fuel consumption, and promote environmental sustainability, while also offering improved community engagement and safer commutes.

Mehnaj, Raida, and Murshed’s study on mode choice preferences among female university students in Dhaka reveals that safety, comfort, and affordability are paramount in shaping transportation decisions. Their survey of 360 BUET students found that rickshaws are favored for their perceived safety and accessibility, while traditional modes like auto-rickshaws see reduced usage when ride-sharing options are available. Notably, the research demonstrates that integrating gender-sensitive measures such as secure, ride- sharing services can significantly enhance mobility for women, suggesting that formalized ride- sharing systems have the potential to alleviate prevalent safety concerns. These findings reinforce the rationale behind our proposed application, which is designed to offer a secure, gender-sensitive ride-sharing platform tailored for college communities.

Kamaruddin et al. developed a campus ride-sharing mobile application to address parking limitations and reduce vehicular emissions at Universiti Teknologi MARA. The study employed rigorous requirements validation including expert audits and user walkthroughs to refine system artifacts such as use case models and domain diagrams. The resulting Android prototype, built on a three-tier architecture with Firebase and Java, demonstrates a viable solution for managing ride-sharing interactions on campus.

Several other studies have explored ride-sharing and carpooling systems to address transportation challenges. In Survey-based research, Gallo et al. indicates that students are generally inclined to share rides with known peers, suggesting that trust is a crucial factor in ride-sharing systems. Furthermore, research on Android-based ride-sharing solutions points out limitations in current applications, including a lack of cross-platform functionality and security issues. Research paper provided valuable insights into the design and implementation of various application modules in ride-sharing application using flutter, significantly influencing the development of our app Collectively, these studies highlight both the strengths and gaps in existing ride-sharing systems. While significant progress has been made in enhancing efficiency, reducing costs, and promoting environmental sustainability. Mainly emphasizing the specific need for gender-sensitive ride matching within a college community. Our proposed mobile-based platform aims to fill this gap by offering a secure, cost-effective, and user-friendly solution to the unique commuting challenges faced by college students.

SYSTEM ANALYSIS

3. SYSTEM ANALYSIS

System analysis is the process of studying and evaluating a system to understand its requirements, structure, and functionality. It involves identifying problems in the existing system, analyzing user needs, and defining the best approach for development.

3.1 Existing System

Prior research has explored integrating ridesharing with academic resource sharing systems, often through web based platforms. While the studies highlight some work, they predominantly prioritize web-based applications over mobile solutions, despite the widespread adoption and convenience for users. Furthermore, many platforms rely on technologies that lack cross-platform compatibility, restricting accessibility for users on diverse operating systems.

Real-World Transportation Challenges: Traditional commuting options, such as buses and auto- rickshaws, present persistent challenges for students. Bus services frequently suffer from delays and inconsistent schedules, rendering them impractical for students adhering to fixed academic timetables. Auto-rickshaws, though flexible, are often prohibitively expensive and predominantly operate during peak college hours, leaving students stranded during off-peak times.

Safety Concerns in Ride-hailing apps: Commercial ride-hailing applications, while addressing last- mile connectivity (e.g., linking transportation hubs to campuses), do not properly implement critical safety considerations. The absence of gender-sensitive matching mechanisms has led to documented risks, particularly for female passengers. For instance, in Bengaluru and Neelangarai, incidents in-volving Rapido bike-taxi drivers harassing passengers shows some systemic vulnerabilities and exploitative behavior of riders . Such cases highlight an urgent need for platforms that prioritize user safety without compromising accessibility.

Synthesis of Needs: These gaps in both research and real-world systems emphasize the necessity for a ride-sharing solution tailored to student commuters. An ideal platform must balance convenience (e.g., mobile-first design, cross-platform functionality) with proactive safety measures, such as gender-based ride matching and robust user verification protocols, to mitigate risks observed in existing services.

3.2 Disadvantages of Existing System

- **Unreliable Public Transport:** Buses face delays, inconsistent schedules, and overcrowding, while auto-rickshaws are expensive and unavailable during off-peak hours.
- **Safety Concerns:** Ride-hailing apps do not offer gender-sensitive ride matching, leading to security risks, especially for female students.
- **Lack of Student-Centric Features:** Existing platforms do not ensure affordability, trust, or secure verification for college students.

3.3 Proposed System

Our proposed solution is a mobile application designed to address the gaps in existing ride-sharing systems through a student-centric, safety-first approach. The system consists of the following key features:

Same Gender Rides: To prioritize user safety and comfort, the platform employs a feature, which ensures that riders receive requests from passengers of the same gender. This feature is implemented to enhance the comfort and safety of the users during their commute.

College-Specific Access Control: Registration is restricted to users with a valid institutional email domain (e.g., @university.edu.in), ensuring exclusivity to verified college students. This domain-based authentication minimizes risks associated with unauthorized external users and reinforces platform accountability.

Identity Verification: Users undergo a verification process via their college email to confirm their identity on the platform.

Direct Campus Drop-Off/Pick-Up: Since our college campus is expansive and the distance between the college gate and academic buildings can be significant, our system allows riders to drop off or pick up passengers near the academic buildings, as the riders are part of the college community. Because outsiders are not permitted, this arrangement eliminates the need for passengers to walk from the gate to the college.

Cross-Platform Accessibility: Leveraging Flutter's framework, the app delivers native performance on both Android and iOS devices, eliminating accessibility limitations.

3.4 Advantages of proposed system

- **Accessibility:** Mobile-first design ensures ease of use.
- **Compatibility:** Cross-platform support for wider reach.
- **Reliability:** Reduces delays with a structured ride-sharing system.
- **Affordability:** Cost-effective alternative to public transport.
- **Safety:** Gender-based ride matching enhances security.
- **Exclusivity:** Restricted access for verified college students.
- **Sustainability:** Reduces traffic congestion and carbon footprint.

3.5 System Requirements

The appropriation of requirements and implementation constraints gives the general overview of the project in regard to what the areas of strength and deficit are and how to tackle them.

3.5.1 Software Requirements

- Operating System: Windows, macOS, or Linux (for development)
- Mobile OS: Android 8.0+ and iOS 12+ (for app usage)
- Development Framework: Flutter (for cross-platform mobile development)
- Backend & Database: Firebase (for authentication, database, and cloud storage)
- Programming Languages: Dart (Flutter)
- IDE: Android Studio, Visual Studio Code, or Xcode (for development)

3.5.2 Hardware Requirements

- Processor: Intel i5/i7 or AMD Ryzen 5/7 (for development)
- RAM: Minimum 8GB (16GB recommended for smooth development)
- Storage: At least 50GB free space (SSD recommended for better performance)
- Mobile Devices: Android & iOS smartphones for testing
- Internet Connection: Stable connection for cloud-based operations and real-time updates

3.6 Functional Requirements

Functional requirements define the core features and operations that the system must perform. These specify what the system should do to fulfill user needs. Functional requirements describe the interactions between users and the system, ensuring that all intended functionalities are met.

1. User Authentication
2. Profile Management
3. Ride Creation & Search
4. Gender-Based Matching
5. Payment Management
6. Review & Feedback
7. Admin Dashboard

3.7 Non-Functional Requirements

Non-functional requirements define the system's quality attributes and constraints. These focus on how the system performs rather than what it does. They ensure aspects such as security, scalability, performance, and usability.

1. Security
2. Scalability
3. Performance
4. Usability
5. Reliability
6. Cross-Platform Compatibility
7. Maintainability
8. Availability
9. Data Integrity

3.8 Feasibility Study

The feasibility of the project is analyzed in this phase and the business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be conducted. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are:

1. Economic feasibility
2. Technical feasibility
3. Social feasibility

Economic Feasibility

Economic feasibility evaluates whether the proposed system is cost-effective and sustainable in the long run. The primary costs involved in developing this ride-sharing platform include cloud storage, server maintenance, and mobile application development. To keep expenses minimal, cloud services such as Firebase are used for authentication, database storage, and hosting, ensuring a scalable and cost-efficient infrastructure. An initial investment of ₹1000 has been made to cover cloud services, ensuring smooth operation. Since the platform is designed specifically for college students, it does not require high marketing costs. Additionally, potential monetization strategies, such as premium features or minimal service fees, could make the system self-sustaining. Given these factors, the project is economically viable with manageable costs and long-term benefits.

Technical Feasibility

Technical feasibility assesses whether the required technology and expertise are available to develop the proposed system. The ride-sharing platform is built using Flutter, which enables cross-platform mobile app development, ensuring compatibility with both Android and iOS devices. Firebase is chosen for backend services, including authentication, database storage, and real-time notifications, eliminating the need for a complex server setup. Since all the required technologies are readily available and align well with the project's needs, the system is technically feasible. Furthermore, modern smartphones and internet connectivity among students ensure that users can efficiently access and utilize the platform.

Social Feasibility

Social feasibility evaluates how well the proposed system will be accepted by users and its impact on society. The platform is designed to address the commuting challenges faced by college students, offering a safe, reliable, and cost-effective ride-sharing solution. The inclusion of gender-based ride matching enhances security and comfort, making it more socially acceptable, especially in college environments. By encouraging students to share rides, the system fosters a sense of community while also promoting eco-friendly commuting, reducing traffic congestion and carbon emissions. Given the increasing reliance on digital platforms for convenience, students are likely to adopt the system, making it socially feasible and beneficial to the target audience. This project's outcome is simple to use with very little effort making it handy for all.

SYSTEM DESIGN

4. SYSTEM DESIGN

System design is the process of defining the architecture, components, modules, interfaces, and data flow of a system to meet specified requirements. It transforms analyzed requirements into a structured framework for implementation. This section depicts the design of our application using various diagrams.

4.1 System Architecture

The architecture diagram provides an overview of the system's high-level components, their interactions, and how data flows between them.

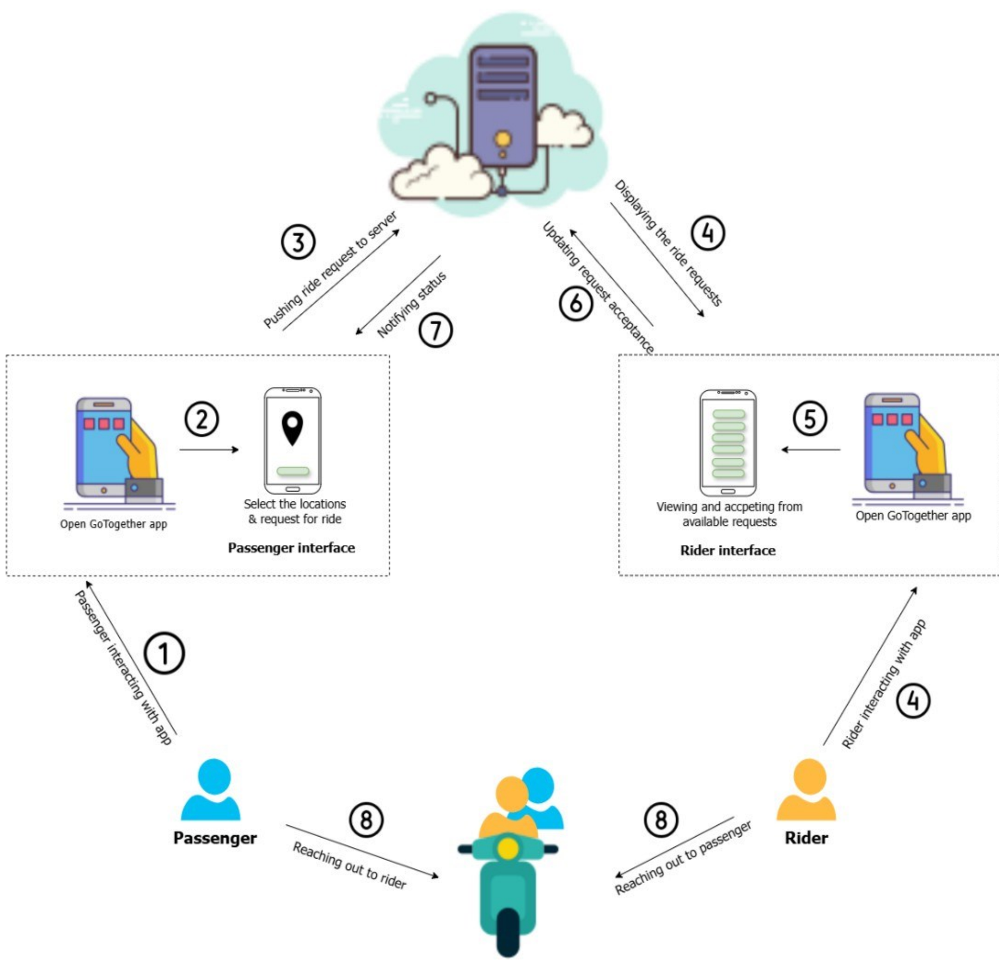

Figure 4.1 System Architecture

The figure 4.1 illustrates the workflow of a college-centric ride-sharing app designed to address last-mile connectivity challenges. It begins with a passenger logging into the platform (Step 1), authenticated via their college email to ensure exclusivity and security. The passenger then selects pickup/drop-off locations (Step 2), such as campus gates or academic buildings, and submits a ride request, which is pushed to the Firebase backend (Step 3). Riders, restricted to verified students, view filtered requests (Step 5) that prioritize same-gender matches for safety and comfort. Upon accepting a request (Step 4), real-time updates notify both parties, enabling direct coordination (Step 8) for campus drop-offs near academic buildings, eliminating long walks from gates. Built with Flutter, the app ensures seamless cross-platform interaction for riders and passengers, while Firebase handles data synchronization and authentication. This workflow aligns with the platform's goals of enhancing safety, reducing commute inefficiencies, and leveraging existing student resources for sustainable mobility.

4.2 Usecase diagram

The use case diagram illustrates the different user roles and their interactions with the system, highlighting the core functionalities.

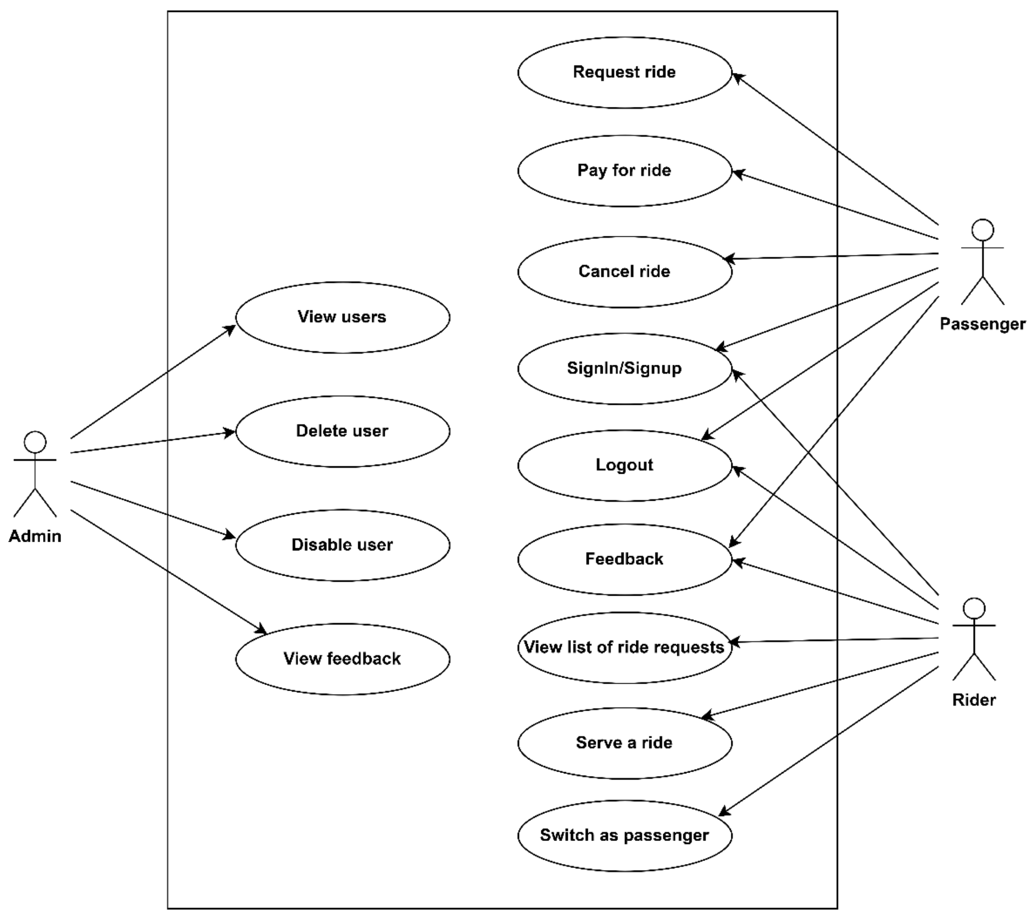

Figure 4.2 Use case diagram

The use case diagram figure 4.2 illustrates the various user roles and their interactions with the system. It provides a visual representation of the functionalities available to different users, helping in identifying system requirements and user expectations.

4.3 Flow Chart

A flowchart is a type of diagram that represents a workflow or process. A flowchart can also be defined as a diagrammatic representation of an algorithm, a step-by-step approach to solving a task.

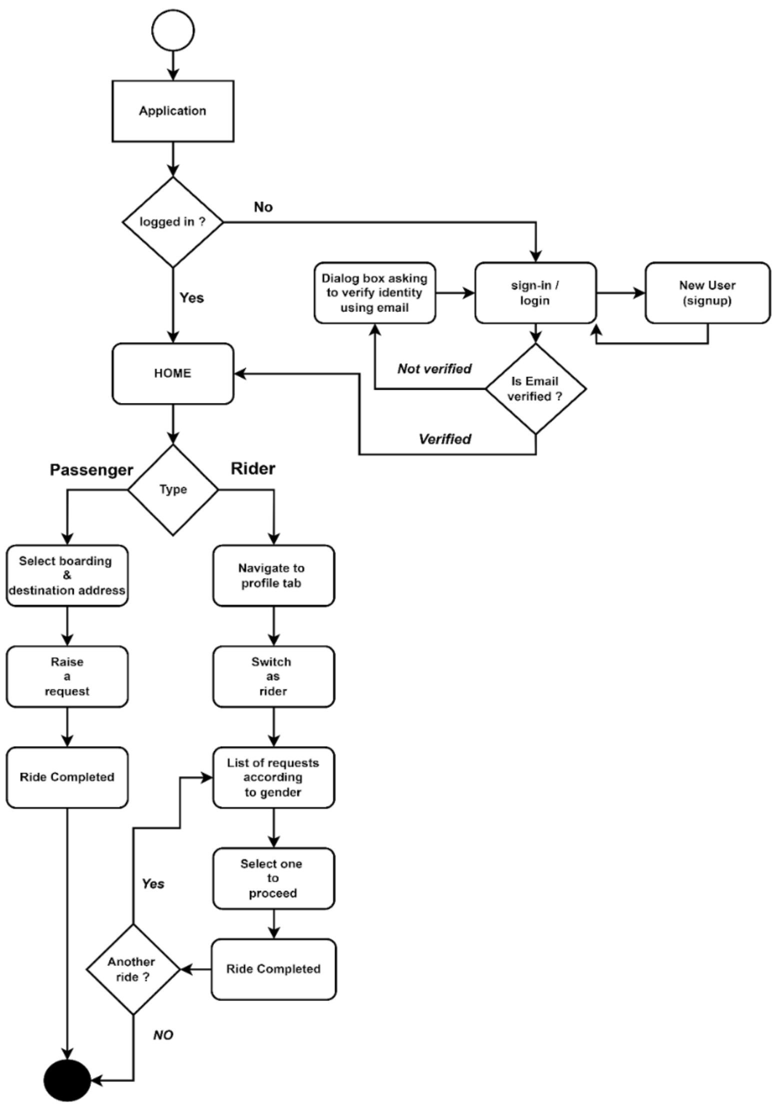

Figure 4.3 Flow Chart

The flow chart figure 4.3 depicts the sequential flow of actions in the application. It clearly depicts the authentication workflow, rider and passenger sequential action steps for a ride.

4.4 Sequence diagram

The sequence diagram details the sequence of interactions between system components, showing how data is exchanged over time to accomplish specific tasks.

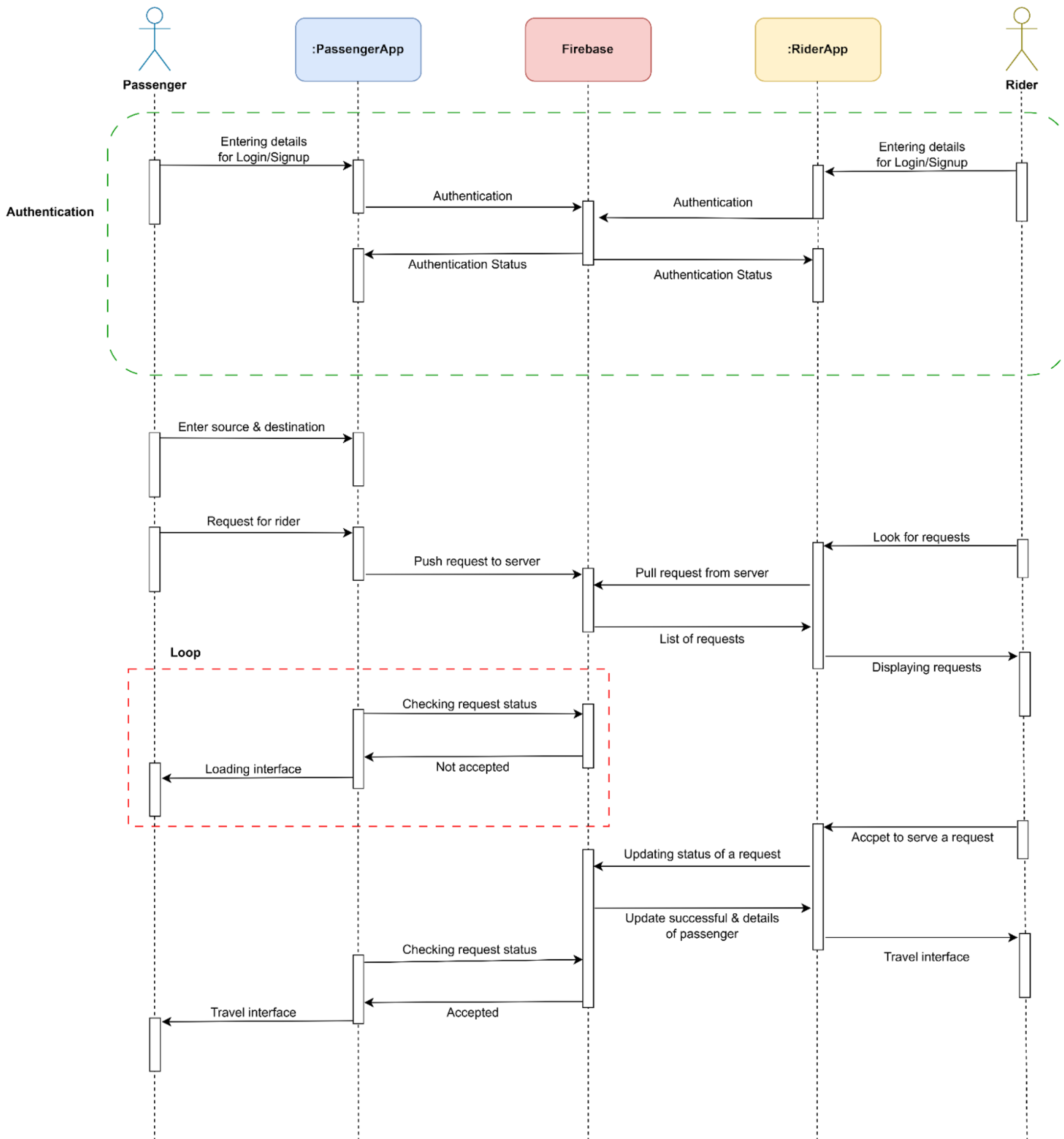

Figure 4.4 Sequence Diagram

The figure 4.4 depicts the sequence diagram , which shows the authentication of user using server(firebase) and interaction of rider and passenger using the the application and server.

4.5 Class diagram

A class diagram represents the structure of a system by showing its classes, attributes, methods, and relationships between objects. It defines how different entities interact within the system and helps in understanding object-oriented design. In our project, the class diagram illustrates key components like users (riders and passengers), ride requests, and authentication mechanisms, showing how they relate to each other.

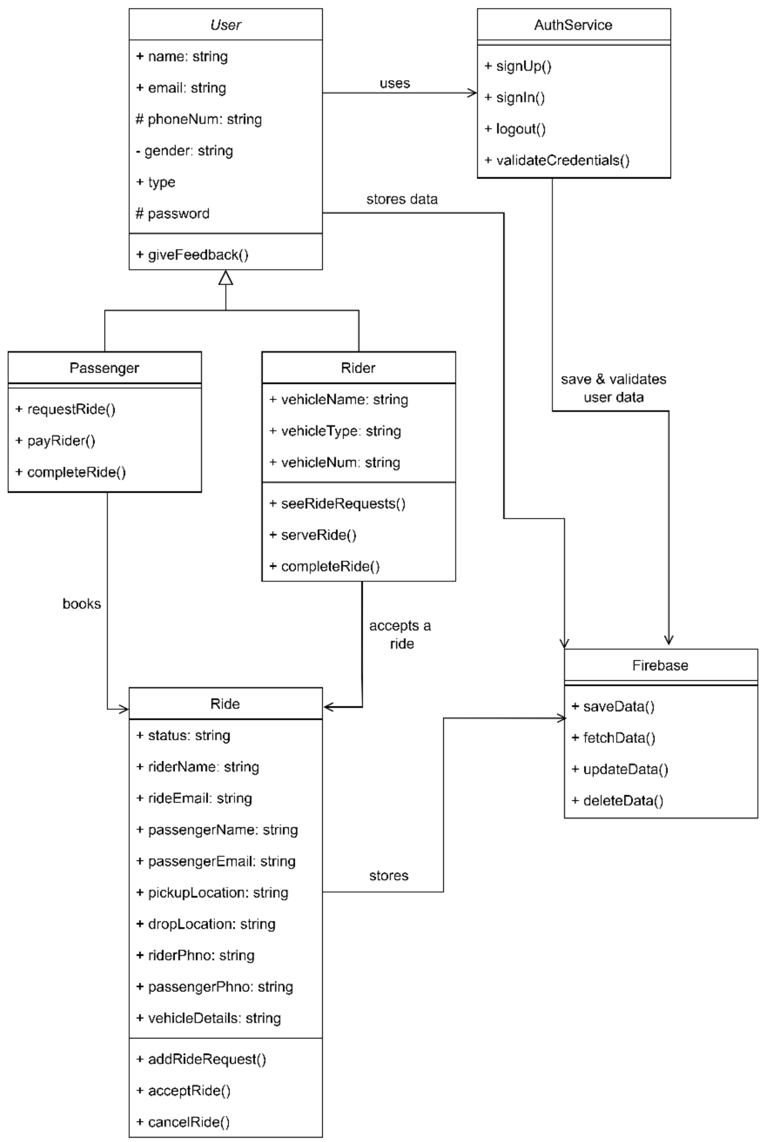

Figure 4.5 Class Diagram

The figure 4.5 depicts the class diagram of our project which consists of user class from which passenger and rider class are derived, ride class, firebase methods and authentication methods are presented.

4.6 Deployment diagram

A deployment diagram depicts the physical architecture of the system, showing how software components are distributed across servers and devices. In our project, it represents how the mobile application interacts with Firebase for authentication, database management, and real-time communication, ensuring a seamless ride-sharing experience.

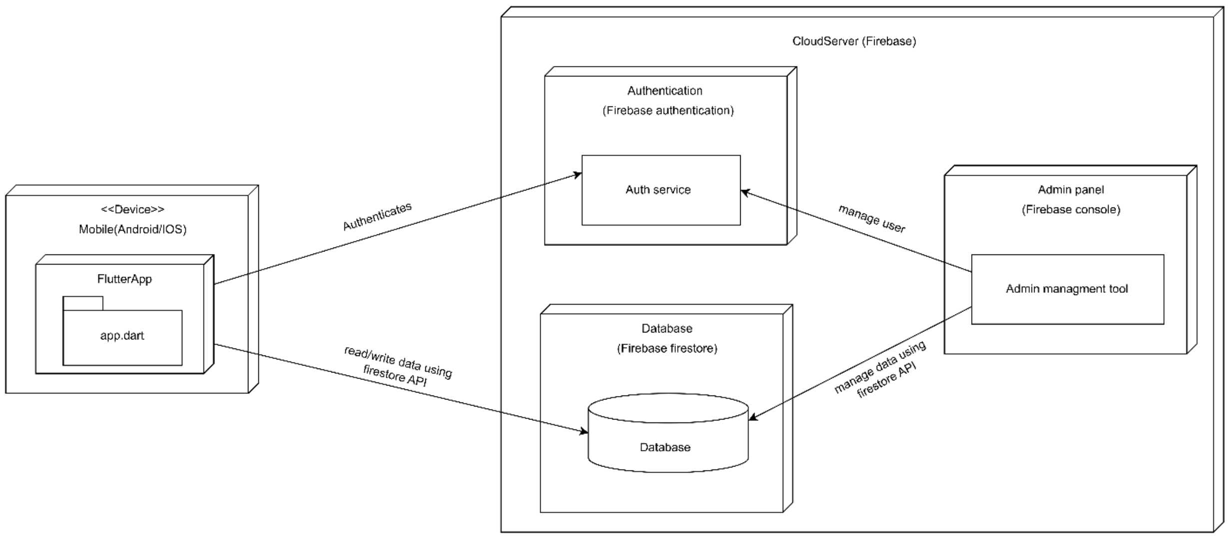

Figure 4.6 Deployment Diagram

The figure 4.6 depicts the deployment diagram which consists of how software components are deployed across servers and devices and their interaction.

4.7 Module description

This section consists of various modules descriptions that are implement in our project. We identified the necessary modules that must be implemented to achieve the objectives of our projects. The following below are the modules

1. **User Authentication**

   This module handles the secure registration and login processes for users. It verifies the identity of new users during sign-up by requiring a college email and secure password, and facilitates password recovery and multi-factor authentication as needed.

2. **Profile Management**

The profile management module allows users to update and manage their personal information. It includes features such as changing passwords, updating contact details, and modifying other account settings. Riders can also update vehicle details, ensuring their profiles remain current and accurate.

3. **Ride Creation & Search**

   This module enables passengers to create ride requests by selecting pickup and drop-off locations, setting preferred times, and providing additional ride details. Simultaneously, it allows riders to search for available ride requests that match their criteria, ensuring they can efficiently identify and serve requests within their vicinity.

4. **Gender-Based Matching**

   The gender-based matching module ensures that ride pairings are made based on the user's gender. This enhances security and comfort by matching riders and passengers of the same gender, addressing safety concerns that are particularly important in a campus environment.

5. **Payment Management**

   This module handles all aspects of payment processing within the platform. It integrates secure payment gateways to facilitate fare splitting or in-app transactions, ensuring that all financial exchanges are recorded, processed, and managed securely.

6. **Review & Feedback**

   The review and feedback module allows users to rate their ride experiences and provide comments regarding their interactions. This feedback system is critical for maintaining service quality, helping both the users and administrators identify areas for improvement.

7. **Admin Dashboard**

   The admin dashboard provides administrators with tools to manage the platform effectively. It includes features for verifying user profiles, monitoring ride requests and user interactions, handling reported issues, and accessing analytical data to ensure the system operates smoothly and securely.

ENVIRONMENTAL SETUP

5. ENVIRONMENTAL SETUP

This section outlines the necessary installations and configurations required to develop and test the ride-sharing application. The development environment is set up using Flutter, Dart, and Firebase, with essential tools such as Android Studio and VS Code.

5.1 Andriod Studio

Android Studio is the official Integrated Development Environment (IDE) for Android app development. It is used to build and test the ride-sharing application on Android devices.

Steps to Install Android Studio:

1. Download Android Studio
   - Visit the official website: https://developer.android.com/studio
   - Download the installer for Windows/macOS/Linux based on your operating system.
2. Install Android Studio
   - Run the downloaded installer and follow the setup instructions.
   - Choose the recommended settings during installation.
   - Allow Android Studio to install the required SDK components.
3. Verify Installation
   - Open Android Studio and create a new project to confirm that it is installed correctly.
   - Ensure the Android Virtual Device (AVD) Manager is working for running emulators.

5.2 VS code

VS Code is a lightweight IDE that supports Flutter and Dart development. It is an alternative to Android Studio for writing and debugging code.

Steps to Install VS Code:

1. Download VS Code
   - Visit the official website: https://code.visualstudio.com/
   - Download the appropriate installer for Windows/macOS/Linux.

2. Install VS Code
   - Run the installer and follow the setup process.
   - Choose default settings and install extensions as needed.
3. Install Flutter and Dart Extensions
   - Open VS Code.
   - Go to Extensions Marketplace (Ctrl + Shift + X).
   - Search for Flutter and Dart and install both extensions.

5.3 XCode

Xcode is the official Integrated Development Environment (IDE) for iOS app development. It is essential for compiling, testing, and running the ride-sharing application on Apple devices. Since the project is built using Flutter, Xcode is required to develop and deploy the iOS version of the application.

Steps to Install Xcode:

1. Download Xcode
   - Open the Mac App Store and search for "Xcode."
   - Click Install and wait for the installation to complete.
2. Install Xcode Command Line Tools
   - Open Terminal and run the following command:
     `xcode-select –install`
3. Accept Xcode License and Configure Flutter
   - Open Terminal and enter: `sudo xcodebuild -license`
   - Read and accept the terms to proceed.
4. Verify Installation
   - To check if Xcode is installed correctly, run: `xcodebuild -version`
   - Ensure flutter recognizes Xcode by running `flutter doctor`

5.4 Java

Java is a crucial requirement for running Flutter and Android development tools. Below are the steps to install Java on your system.

Steps to Install Java:

1. Download Java Development Kit (JDK)
   - Visit the official Oracle website: https://www.oracle.com/java/technologies/javase-downloads.html.
   - Download the latest stable version of the Java Development Kit (JDK) suitable for your operating system.
2. Install JDK
   - Run the downloaded installer and follow the on-screen instructions.
   - Choose the installation directory and complete the setup.
3. Set Up Environment Variables
   - Add the Java bin directory to the system's PATH variable to allow command-line execution.
   - Set the JAVA_HOME environment variable to point to the JDK installation folder.
4. Verify Installation
   - Open a terminal or command prompt and run: `java -version` and `javac -version`
   - If Java is installed correctly, it will display the installed version.

5.5 Flutter

Flutter is a cross-platform framework used to develop mobile, web, and desktop applications. Below are the steps to install Flutter.

Steps to Install Flutter:

1. Download Flutter SDK
   - Visit the official Flutter website: https://flutter.dev/docs/get-started/install.
   - Download the appropriate Flutter SDK for your operating system (Windows, macOS, or Linux).
2. Extract Flutter SDK
   - Unzip the downloaded file to a preferred directory (e.g., C:\flutter for Windows or ~/flutter for macOS/Linux).
3. Set Up Environment Variables
   - Add Flutter's bin directory to the system's PATH variable for global access.
4. Verify Installation
   - Open a terminal or command prompt and run: `flutter doctor`
   - This command checks for missing dependencies and provides installation

suggestions.

5. Install Required Dependencies
   - Install Android Studio and its Flutter/Dart plugins for a full development environment.
   - macOS users should install Xcode for iOS development.
6. Set Up an Emulator or Device
   - Configure an Android Emulator via Android Studio.
   - Alternatively, connect a real device with USB debugging enabled.

5.6 Firebase

Firebase provides authentication, a real-time database, and cloud storage for the app.

Steps to Set Up Firebase:

1. Create a Firebase Project
   - Go to https://console.firebase.google.com/
   - Click "Create a Project" and enter the project name.
2. Add Firebase to Flutter App
   - Follow the official guide: https://firebase.flutter.dev/docs/overview
   - Download the google-services.json (for Android) and GoogleService-Info.plist (for iOS) and place them in the respective directories.
3. Enable Required Services
   - Enable Authentication (for user login and registration).
   - Enable Realtime Database or Cloud Firestore (for storing ride data).
   - Enable Firebase Cloud Messaging (for push notifications).

APPLICATION SCREENS

6. APPLICATION SCREEN

This section consists of screenshots of the entire application and explanation of each screen in the application.

6.1 Sign In and Sign Up

In this section the sign In page and sign Up page are depicted. The sign-in and sign-up screens serve as the primary authentication mechanisms for users. The sign-up screen allows new users to create an account by providing necessary details such as name, college email, password, and other relevant information. The sign-in screen enables registered users to log into the application using their credentials. These authentication processes ensure secure access and help maintain a verified user base within the platform.

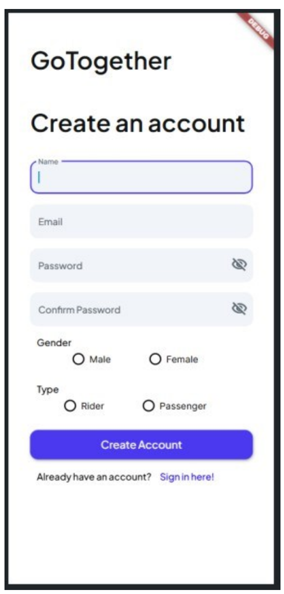

Figure 6.1.1 Sign Up

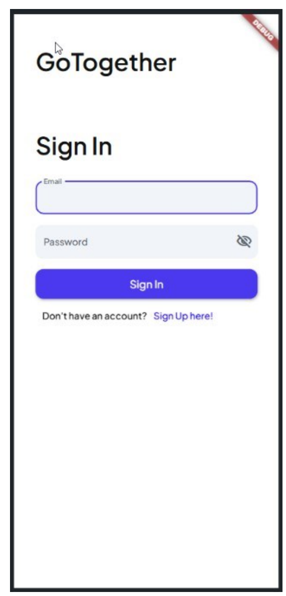

Figure 6.1.2 Sign In

Figure 6.1.1 illustrates the sign-up page, allowing new users to register by providing details such as name, college email, password, gender, user type, and vehicle details (if registering as a rider). It also includes an option to navigate to the sign-in page for existing users. Figure 6.1.2 depicts the sign-in page, where users can log in using their college-registered email ID and password. Additionally, this page provides a navigation option to the sign-up page for users who have not yet registered.

6.2 Home & Profile

In this section the screens of home screen and profile screen of our application are depicted. The home screen serves as the central hub for users, allowing them to interact with the core functionalities of the application. It typically includes options to input pickup and drop-off locations, view available rides, and request a ride. This screen provides a seamless and intuitive interface for users to navigate the ride-sharing process efficiently.

The profile screen enables users to manage their account settings and preferences. It may include options to update personal details, switch user roles (e.g., from passenger to rider), log out, and provide feedback. This section ensures a personalized and secure user experience within the application.

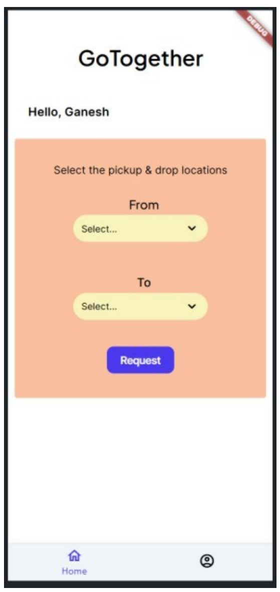

Figure 6.2.1 Home

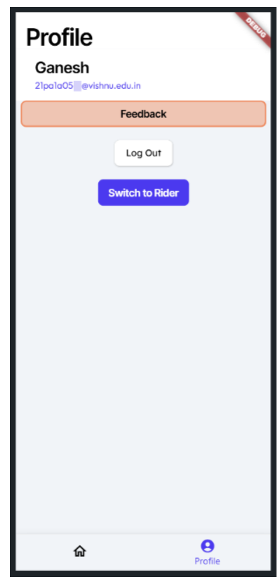

Figure 6.2.2 Profile

Figure 6.2.1 illustrates the home screen, where passengers can select their pickup and drop-off locations using dropdown boxes and submit a ride request. Figure 6.2.2 represents the profile screen, which provides options such as logging out, submitting feedback, and switching to a rider mode if the user has registered as a rider.

6.3 Ride request & Rider's home

In this section the screens of ride request and rider's home page are presented. The ride request feature allows passengers to request a ride by selecting their desired pickup and drop-off locations. This ensures a seamless booking process, enabling efficient ride allocation based on user preferences.

The rider's home page serves as the interface where riders can view and manage incoming ride requests. It typically includes a list of ride requests filtered based on specific criteria, such as gender-based matching for enhanced safety and comfort. This ensures that riders receive only relevant requests, improving the overall ride-sharing experience.

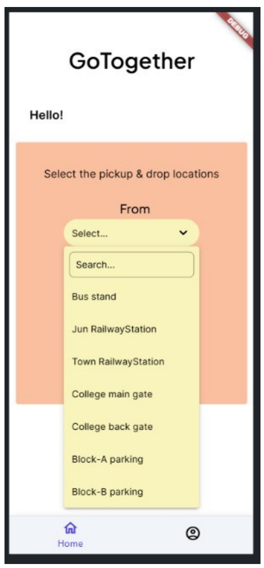

Figure 6.3.1 Ride request

Figure 6.3.2 Rider's home

Figure 6.3.1 illustrates the process of making a ride request, where passengers can select their pickup and drop-off locations. Figure 6.3.2 represents the rider's home page, which displays a list of ride requests from passengers of the same gender, ensuring that the rider receives only relevant and secure ride requests.

6.4 Rider’s Travel Screen & Passenger’s Travel Screen

This section presents the screens displayed during a ride for both the rider and the passenger. The rider's screen provides details about the passenger, including their name, email, pickup location, and destination. Similarly, the passenger's screen displays information about the rider, such as their name, email, vehicle type, vehicle name, and vehicle number.

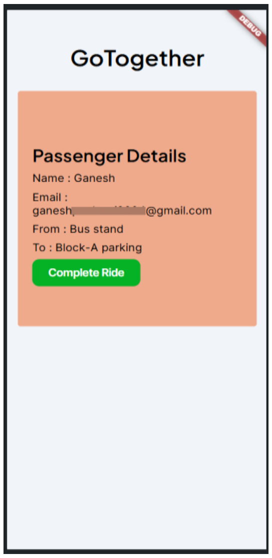

Figure 6.4.1 Rider’s Travel Screen

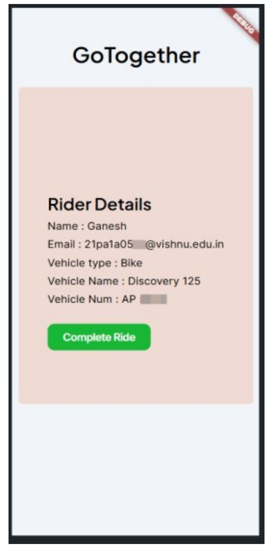

Figure 6.4.2 Passenger’s Travel Screen

Figure 6.4.1 illustrates the rider’s screen during the ride, showing the passenger’s details. Figure 6.4.2 represents the passenger’s screen during the ride, displaying the rider’s information for better transparency and communication

6.5 Feedback screen

This section showcases the feedback screen, where users can share their opinions and experiences regarding the application.

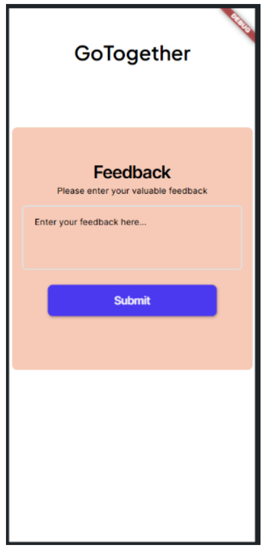

Figure 6.5 Rider's Travel Screen

Figure 6.5 illustrates the feedback page, allowing both passengers and riders to submit their feedback to help improve the platform.

CONCLUSION AND FUTURE SCOPE

7. CONCLUSION AND FUTURE SCOPE

7.1 Conclusion

This project addresses the critical commuting challenges faced by college students, particularly inefficiencies in public transport and safety concerns related to last-mile connectivity. The proposed mobile-based ride-sharing platform offers a tailored solution by incorporating essential features such as gender-sensitive ride matching, college-specific access control, and direct campus drop-offs. By leveraging cross-platform technologies like Flutter and Firebase, the system is designed to be secure, user-friendly, and cost-effective, effectively catering to the unique needs of student commuters.

Additionally, the research highlights the importance of safety in mobility within academic environments. With verified user participation and gender-sensitive ride pairing, the platform not only enhances commuter convenience and reduces travel costs but also promotes a safer and more sustainable transportation ecosystem. While the system provides a solid foundation for student-centric ride-sharing, continuous improvements based on user feedback will be crucial to further refining its efficiency and usability.

7.2 Future Scope

To further enhance the platform, several improvements are planned:

- **Timeout Mechanism:** Implementing a robust timeout mechanism to optimize system performance during peak hours, ensuring smooth operation for a growing user base.
- **Scalability:** Expanding the platform's capabilities to accommodate a larger number of students across multiple institutions.
- **Admin Interface Enhancements:** Developing an advanced admin dashboard with additional functionalities, including a report module to efficiently handle user complaints, particularly concerning gender misrepresentation and other security concerns.

By implementing these future enhancements, the platform will continue to evolve, ensuring greater efficiency, security, and accessibility for student commuters.

BIBLIOGRAPHY

8. BIBLIOGRAPHY

1. Mugoro, J. (2014). *Transport problems for students and their effects on attendance in community secondary schools in Dar es Salaam city, Tanzania*. PhD dissertation, The Open University of Tanzania.
2. Joshi, K.M. (2015). *Higher education, social demand, and social equity in India*. In *Higher Education in the BRICS Countries: Investigating the pact between higher education and society* (pp. 125–147). Springer Netherlands, Dordrecht.
3. Chowdhury, A., Jamal, A., Alam, R., & Palit, R. (2016). *Campus ride: An environment-friendly ride-sharing platform for academic institutions*. In *2016 IEEE International Conference on Computer and Information Technology (CIT)* (pp. 120–124). IEEE.
4. Chatterjee, A., & Skr, P. (2023). *Last mile connectivity in the Indian scenario: A literary review*. In *Proceedings of the World Conference on Social Science* (Vol. 1, No. 1, pp. 40–52).
5. Mugoro, J. (2014). *Transport problems for students and their effects on attendance in community secondary schools in Dar es Salaam city, Tanzania*.
6. Meshram, A., Choudhary, P., & Velaga, N.R. (2020). *Assessing and modelling perceived safety and comfort of women during ridesharing*. *Transportation Research Procedia*, 48, 2852–2869.
7. Natarajan, M., Schmuhl, M., Sudula, S., & Mandala, M. (2017). *Sexual victimization of college students in public transport environments: A whole journey approach*. *Crime Prevention and Community Safety*, 19, 168–182.
8. Mehnaj, R., Raida, A., & Murshed, M.N. (2022). *Mode choice preferences of female university students and impacts of ride-sharing thereon*. *Journal of Engineering Science*, 13(2), 43–49.
9. Kamaruddin, K.A., & Mohd Rozlis, N.R. (2019). *UITM share ride: Requirements validation, design and development of a campus ride-sharing mobile application*. In *2019 6th International Conference on Research and Innovation in Information Systems (ICRIIS)* (pp. 1–6). IEEE.
10. Gallo, M., & Buonocore, C. (2017). *The inclination of university students towards carpooling: Critical aspects and opportunities*. *International Journal of Education and Learning Systems*, 2.
11. Divyesh, P. (2016). *A Smart Real-Time Ridesharing Android Application*. *International*

Journal on Recent and Innovation Trends in Computing and Communication, 4, 188–192.

12. Ashokkumar, M., Balaji, J., & Maheswaran, K. *Bike Pooling System Using Flutter*.
13. Kusuma, G.R., Handana, A.R., & Warnars, H.L.H.S. (2022). *Web Application-based Ridesharing: Your Transportation in Jakarta*. In *2022 Second International Conference on Artificial Intelligence and Smart Energy (ICAIS)* (pp. 827–832). IEEE. doi:10.1109/ICAIS53314.2022.9743102.
14. Flutter Homepage, https://flutter.dev, last accessed 2025/01/07.
15. Firebase Homepage, https://firebase.google.com, last accessed 2025/02/21.