**Route4Me: A Real-Time GPS Tracking Application for PUV Passengers within Rizal Province**

An Undergraduate Thesis

Presented to the Faculty of the Department of Electronics Engineering

Polytechnic University of the Philippines

Sta. Mesa, Manila

In Partial Fulfillment of the Requirements for the Degree

Bachelor of Science in Electronics Engineering

By

**JACINTO, DAVE SIMON DS.**

**MERCADO, JYD REY A.**

**ORTEGA, JEAN IRAH M.**

**OSIAS, KIMBERLY JOY V.**

**SUAREZ, ARNEL D.**

**ENGR. GEOFFREY T. SALVADOR**

**ENGR. GERHARD P. TAN**Advisers

July 2024

**CERTIFICATION OF ORIGINALITY**

This is to certify that the work of research presented in this thesis/dissertation, *ROUTE4ME: A REAL-TIME GPS TRACKING APPLICATION FOR PUV PASSENGERS WITHIN RIZAL PROVINCE* for the degree of Bachelor of Science in Electronics Engineering at the Polytechnic University of the Philippines represents the outcome of original and scholarly research conducted by the undersigned. This dissertation does not include terms or concepts from published sources or other written works that have been used as the basis for earning a degree from any other institution, except where appropriate references and acknowledgments have been provided.

**DAVE SIMON DS. JACINTO**

Researcher

**JYD REY A. MERCADO**

Researcher

**JEAN IRAH M. ORTEGA**

Researcher

**KIMBERLY JOY V. OSIAS**

Researcher

**ARNEL D. SUAREZ**

Researcher

Signed September 6, 2024

**ACKNOWLEDGEMENTS**

The researchers would like to express their gratitude and gratefulness to all those who gave them support, encouragement, and assistance to turn this study into a reality. The completion and success of this study would not have been possible without the cooperation, untiring efforts, and support of the following persons who unselfishly imparted their time to accomplish this thesis.

First and foremost, this work is dedicated to Almighty God, for the gift of wisdom, blessings, and strength He provided in every way possible to complete this study.

The researchers could not have undertaken this journey without their thesis advisers, Engr. Geoffrey T. Salvador and Engr. Gerhard P. Tan, for their support, patience, guidance, valuable comments and criticisms that benefited the researchers much in the completion and success of this study.

They also extend their gratitude to the survey respondents, commuters, and drivers, for their efforts and willingness to participate and cooperate, which greatly helped the researchers to move forward in their study.

Appreciation is extended to the researcher’s parents, families, and loved ones, for the financial, moral, spiritual, and emotional support they have bestowed upon them and served as their source of inspiration that enabled them to withstand every challenge.

Their friends, classmates, and fellow researchers, for their thoughtfulness and encouragement, and those whom the researchers failed to mention, who have been in one way or another, have helped realize this study.

Thank you very much. To God be the highest glory!

* *The Researchers*

**ABSTRACT**

|  |  |  |
| --- | --- | --- |
| Title | : | Route4Me: A Real-Time GPS Tracking Application for PUV Passengers within Rizal Province |
| Researchers | : | Dave Simon DS. Jacinto |
|  |  | Jyd Rey A. Mercado |
|  |  | Jean Irah M. Ortega |
|  |  | Kimberly Joy V. Osias |
|  |  | Arnel D. Suarez |
| Degree | : | Bachelor of Science in Electronics Engineering |
| Institution | : | Polytechnic University of the Philippines |
| Year | : | 2024 |
| Adviser | : | Engr. Geoffrey T. Salvador & Engr. Gerhard P. Tan |

Abstract - This study presents Route4Me, a real-time GPS tracking application designed to enhance the commuting experience for Public Utility Vehicle (PUV) passengers in Rizal Province. The mobile application addresses common commuter challenges such as long waiting times, traffic congestion, and uncertainty in PUV arrival times by providing real-time tracking of PUV locations, route recommendations, and an estimated time of arrival (ETA) based on current traffic conditions. Additionally, it offers fare details for various routes to help commuters make informed and cost-effective travel decisions.

The development of Route4Me employs an agile software development methodology, utilizing Dart and the Flutter UI framework for compatibility with Android and iOS devices. Firebase is integrated for secure authentication and database management, while Google Maps and Geolocation APIs ensure accurate location services.

A commuter survey and numerous application tests were conducted to evaluate the app's accuracy, reliability, usability, portability, and maintainability. The theoretical framework for this study is based on the Intelligent Transport System (ITS) theory, which highlights the role of advanced technologies in enhancing transportation efficiency, safety, and sustainability.

The implementation of Route4Me significantly reduced waiting times and improved the overall commuting experience for PUV passengers. The application was well-received by users, who reported high levels of satisfaction with its real-time information and user-friendly navigation features. The study demonstrates the potential of Route4Me to modernize public transportation in Rizal Province and possibly other areas of the country, providing a reliable and efficient solution to common commuter challenges.

**Chapter 1**

**THE PROBLEM AND ITS BACKGROUND**

This chapter provides all the necessary background information and context for this study. The introduction, conceptual framework, theoretical framework, statement of the problem, significance of the study, scope and limitations, and definition of terms that are most frequently encountered will help readers better understand the scope of the research.

**Introduction**

Public utility vehicles (PUVs), consisting of diverse vehicles – from traditional tricycles to jeepneys, to trains, to taxis and buses – remain an integral part of everyday life in the Philippines. It is considered the primary mode of transportation in the Philippines. It provides much-needed mobility and conveyance to the general public and provides accessibility by covering several routes throughout the country (Chuenyindee et. al., 2022). Aside from this, for many Filipinos living in rural areas, it has become their economic backbone or source of income (Exploring the Main Modes of Transportation in the Philippines. Secret Philippines, 2023).

PUVs provide transportation services to commuters in the Philippines. Their primary goal is to provide accessible and affordable transportation options, allowing citizens to easily navigate the urban maze. Daily commuters in the Philippines use various modes of public transportation and face different challenges along their journey. Apart from enduring the notorious traffic in the metro, commuters face long queues, overcrowding in public transit, and limited walking spaces. In addition, the rising cost of transportation due to rising fuel prices raises concerns for both commuters and drivers of these public utility vehicles (PUVs) (Ignacio, 2023).

According to survey results from the transportation advocacy network The Passenger Forum, 79% of respondents agree that their waiting time for a ride is usually too long, while 96% disagree that the number of PUVs on the road, including both jeepneys and buses, is sufficient to service commuters (Luna, 2022).

Jeepneys are an essential mode of transportation for Filipino commuters, particularly students and low-income workers as they offer affordable fare and help thousands of independent operators make a living. However, the jeepney's continued use has raised environmental and safety concerns due to its high emissions output and outdated design (Asia Pacific Foundation of Canada, 2023).

As part of the Department of Transportations’ Public Utility Vehicle Modernization Program (PUVMP) which aims to modernize and unify the disparate PUV industry and seeks to update or replace PUVs that are over 15 years old to comply with safety and environmental regulations (‌Parrocha and Pateña, 2018). The Department of Science and Technology (DOST) proposed a locally made electric jeepney, or e-jeepney, as a solution to the significant challenge of executing the Department of Transportation (DOTr)'s public utility vehicle modernization program (PUVMP). E-jeepney has been described as a smart and practical technology built by local enterprises to address the nationwide issues with public transportation (Ronda, 2023). Considering that, the first batch of modern electricity-powered jeepneys, known as “E-Jeeps'' was rolled out last June 2018 (‌Parrocha and Pateña, 2018). It is currently used by commuters as a mode of their transportation.

Additionally, buses are among the most common forms of public transit in the country. Various bus types are out there, such as air-conditioned, provincial, shuttle, point-to-point (P2P), and traditional buses (rsvp-khalil, 2023). Other than that, UV Express is widely used primarily due to its affordable price as an alternative form of transportation for daily commuting, as compared with using a taxi. Although taxis tend to be cozy and private, UV Express is among the most frequently utilized vehicles, considering it costs less per kilometer (iFranchise Philippines, 2018).

Furthermore, with technological advances today, particularly in commuting, people can take control of their daily lives and use them for their benefit. Mobile phones are the most useful and easiest devices they can use. People can commute more easily, more efficiently, and more systematically than ever with the use of modern technology such as mobile phones (Cruz, 2016).

There are applications that people can utilize for their commuting, such as Moovit, which provides a real-time travel organizer mobile and web app to navigate public transit networks with the use of GPS across transit modes, such as trains, ferry trips, rapid transit (metro/subway/underground, etc.), buses, ride-hailing, shared bicycles, car sharing, and motorcycles (Moovit Inc., 2023), and Google Maps, in which commuters can access updates on ETA, traffic states, and incidents along their journey (Google, 2024).

Smart mobile applications have the potential benefit of improving traffic operations and safety. One of which is the route planning applications. Route planning applications are generally designed to help travelers find an efficient path from a point of origin to a destination and to help them navigate cities quickly and easily (Borole et al., 2013).

Moreover, in route planning, applications that deal with transit modes typically offer standard details like schedules and station locations. With the help of advanced applications, commuters can access real-time vehicle delays, stop locations, and arrival times for major transit agencies, as well as information about station services like parking rates, elevators, wheelchair accessibility, and restrooms. These applications use the GPS built into the mobile device to give users their current location, the closest station, and the time the next few trains are scheduled to depart from the station (Siuhi and Mwakalonge, 2016).

In this regard, the researchers seek to develop a mobile application that commuters can utilize. It aims to develop a real-time GPS tracking application for PUV passengers commuting within Rizal Province, in which they may use the app to track the present locations of PUVs, allowing them to plan their commutes more efficiently and reduce the waiting time for PUVs.

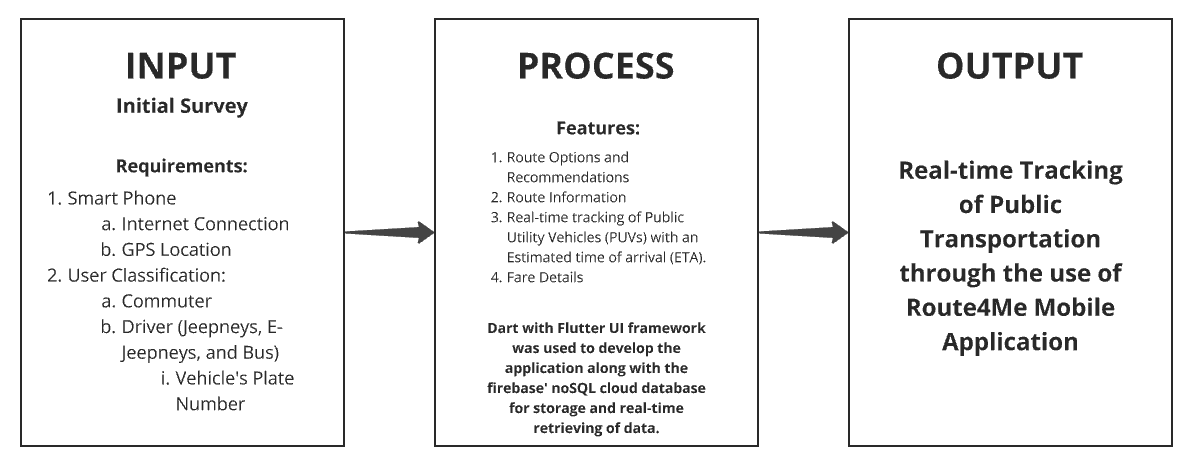
Also, it provides route recommendations within the application to guide commuters in choosing the most efficient path. This includes providing alternate routes depending on traffic conditions, road closures, or other commute-related factors. The application allows users to enter their location and suggests the most effective route based on relevant data and traffic conditions.

Lastly, it provides an estimated time of arrival (ETA) feature relative to the user's current location and displays fare details for different routes or distances. Based on their real-time GPS location, the application would determine and suggest an estimated time of arrival for PUVs at a specified location. When users enter their current location, the app estimates when the next PUV on their selected route will arrive. Moreover, the mobile application would also allow viewing fare details for various routes or distances. Commuters may review it before commencing their commute, allowing them to make cost-effective decisions.

In order to facilitate commuters' navigation and information access, the application would have a clear and user-friendly interface. The proposed solution of this study would work in the constantly changing environment of public transportation in Rizal Province. Integrating GPS tracking devices into PUVs is dependent on their assistance and availability. For the application to be functional, real-time data accuracy—such as GPS information and traffic conditions—is necessary. Also, a reliable internet connection must be available for real-time updates and data retrieval in the mobile app.

**Conceptual Framework**

This section outlines the conceptual framework of the research, employing an IPO model. The model comprises three fundamental components: inputs, which consist of variables and parameters of the study; process, which involves the creation of the Android application; and outputs, which involve the implementation and utilization of the application.



***Figure 1. Conceptual Framework***

The study focuses on meeting the demands of mobile application users by developing an application using agile software development methodology. The research considered user requirements and relevant literature, using them as inputs for the research variables. The mobile application development process involved strategic planning, user interface, and user experience design. Dart and the Flutter UI framework were chosen to develop the application, catering to Android and iOS devices. The application utilizes Firebase for both authentication and database needs. Firebase Authentication ensures secure user login and management, while Firebase's real-time NoSQL cloud database provides dynamic real-time data storage and retrieval. Additionally, Google Maps and Geolocation APIs from Google Cloud were integrated for location features.

Following development, the researchers will conduct a commuter survey to assess the application's viability. Application testing will be conducted for possible end users using proper test methods and technical standards. The collected data will then be summarized, tabulated, and analyzed.

# Theoretical Framework

The concepts and research methodology are supported by the theoretical framework's laws, theories, and principles.

**Intelligent Transport System**

Intelligent transportation systems are currently a vital aspect of each developing city, particularly relevant for transportation. This form of technology has been considered essential for obtaining effective transportation in cities, which explains why so many countries have already started or are considering using it (Chipilska, 2023).

Within the Internet of Things (IoT), an intelligent transportation system uses data and electronics technologies based on wireless and wireline connections. It consists of vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) technologies (Chipilska, 2023). Generally, this study, entitled *Route4Me: A Real-Time GPS Tracking Application for PUV Passengers within Rizal Province and Vice Versa* is supported by the theory of Intelligent Transport Systems which are employed in transportation and roadway management systems to improve the efficiency and long-term viability of several transportation systems and the user experience.

In connection with this study, the experts classify the Intelligent Transport System into three categories: mobility, safety, and environment. When it comes to transportation, Intelligent Transportation Systems (ITS) seek to find the fastest route between a starting point and a destination by considering variables such as time, distance, and energy consumption. Subsequently, it compares this information to an environment rich in data to determine the most efficient route. Numerous transportation networks and systems can benefit from this regarding administration and monitoring. Emergency and maintenance services can be quickly deployed, and traffic signals may be modified appropriately. Also, transit operators can be managed continuously (Chipilska, 2023).

Also, regarding reducing accidents, fatalities, and encounters on the roads, an intelligent transport system can be utilized to increase safety by providing speed warnings on slippery roads and accident situations. Commuters can get information and warnings from an ITS system, which can also be used for emergency management and vehicle safety applications. Moreover, real-time GPS tracking and monitoring can help make smart choices to improve efficiency across every transportation system. Improving daily traffic flow can also lead to significant fuel savings and allow commuters to reach their destinations more quickly and sustainably (Chipilska, 2023).

In addition to promoting the use of alternative modes, ITS can assist in optimizing journeys by providing route information and route options, real-time GPS tracking of Public Utility Vehicles (PUVs) with an Estimated Time of Arrival (ETA), fare information, and recommending a better route for commuters.

# Statement of the Problem

This study focuses on delivering a real-time GPS tracking application for PUV Passengers within Rizal Province. Specifically, this study aims to:

1. Develop a real-time GPS tracking application for PUV Passengers within Rizal Province.
2. Provide route recommendations within the application to guide commuters in choosing the most efficient path.
3. Provide an estimated time of arrival (ETA) feature relative to the user's current location and destination, and display fare details for different route options.
4. How is the real-time GPS tracking application for PUVs effective in terms of:
   1. Accuracy
   2. Reliability
   3. Usability
   4. Portability
   5. Maintainability

# Significance of the Study

The research aims to significantly improve the commuting experience by addressing challenges in transportation efficiency and environmental sustainability, aligning with the United Nations Sustainable Development Goals. This commitment seeks positive changes for drivers and commuters, fostering a more sustainable and efficient transportation system.

This study will significantly benefit the following:

**Commuters.** Enhancing the overall commuting experience, a real-time GPS tracking system designed for Public Utility Vehicles (PUVs) holds the potential to substantially diminish waiting times for commuters. This system ensures that passengers can access accurate information about the specific locations and expected time of arrival (ETA) of PUVs, empowering them to plan their journeys more efficiently.

**Drivers.** The project optimizes routes, potentially reducing congestion for drivers, resulting in more efficient travel times, lower fuel consumption, and a sustainable driving experience.

**Public Transportation Authorities.** Stand to benefit from streamlined operations, improved resource allocation, and a more effective transportation network, contributing to economic efficiency and environmental sustainability.

**Researchers.** Developing and implementing a real-time GPS tracking application introduces opportunities for technological innovators and software developers. Insights from this study may inspire further advancements in transportation-related technologies and applications.

**Scope and Limitations**

This study will primarily focus on the travel and waiting periods of commuters traveling within Rizal Province. The smartphone application should be able to provide real-time GPS tracking, route recommendations, estimated time of arrival (ETA), and fare details to help commuters manage their commute time effectively. Additionally, to assess the effectiveness of the smartphone application, participants in this study will be commuters within Rizal Province. They will be given a survey questionnaire to complete and answer through Google Forms. PUVs such as buses, traditional jeepneys, e-jeeps, and UV express are included in this study.

# Definition of Terms

**Artificial Intelligence (AI)** — Modern technology integrated into PUVs to develop a real-time mobile application to track their locations.

**Estimated Time of Arrival (ETA)** — The expected time for a PUV to reach a specific location, providing advance notice about its expected arrival at the destination.

**Google Maps** — A web service offering detailed global information, including road maps, aerial and satellite views, and street views with vehicle-captured photographs in select cities.

**Live Tracking** — The real-time tracking of PUV locations on a map.

**Moovit** — A trip planner with GPS to help commuters find a way around the public transportation system.

**Public Utility Vehicles (PUVs)** — Various transportation options provide travel options, such as pedicabs, tricycles, jeepneys, buses, taxis, and trains.

**Real-Time GPS Tracking Application** — The primary technology discussed in the study involves live tracking of PUV locations, displaying route information, providing fare details, and offering commuter recommendations.

**Route Information** — Information about the complete route within Rizal Province, including any deviations or changes.

**Urban Mobility Readiness Index (UMRI)** — A metric used to assess a city's performance in public transit density, efficiency, and utilization rate, indicating Rizal provinces’ position relative to other cities.

# Chapter 2

**REVIEW OF RELATED LITERATURE AND STUDIES**

This chapter analyzes and evaluates relevant studies and literature from national and international sources that can be used to gather and assess background data for this study. Additionally, it seeks to cover the most recent data regarding the topic of the study and any recent issues, developments, or innovations that may help the objectives of this study. Relevant information for the study under discussion has been researched through published publications, newspapers, featured articles, forums, and Internet resources.

*Philippine Public Utility Vehicles (PUVs)*

In 2021, the number of public utility vehicles (PUVs) in operation was classified by the various types of vehicles in the Philippines. Approximately 42.6 thousand public utility jeepneys were in service as of October 2021. Additionally, 21.7 thousand taxis and roughly 25.5 thousand transport network vehicle services (TNVS) were operating nationwide (Statista Research Department, 2023).

Sunio et al.'s (2019) research highlights the ongoing Public Utility Vehicle Modernization Program (PUVMP) in the Philippines as a focal point for socio-technical transition studies within the transport sector. Launched and executed by the national government, the PUVMP aims at transformative reconditioning in the country's public land transportation industry. This study provides insights into the PUVMP's advocacy as a crucial component in regulating the enhancement of public transit in the Philippines (Sunio et al., 2019).

Moreover, Salita (2024) stated that the three main modes of public transportation in the Metropolitan Manila area are taxis, jeepneys (little buses constructed on the chassis of jeeps), and buses.

*Challenges in the Public Transportation System in the Philippines*

In the Philippines, public transportation is considered toxic, and the experience has been called dehumanizing. Hours of waiting for a ride and getting trapped in congested streets are the main common challenges Filipino commuters encounter. These problems are caused by a severe scarcity of transportation supplies, which worsens due to an ineffective and disorganized public transportation system (Philippine Daily Inquirer, 2022).

The traffic conditions are extremely terrible for millions of commuters who travel through Metro Manila each day to work or school. The 52-year-old PNR Metro Commuter Line, the 38-year-old Manila Light Rail Transit (LRT), and the 22-year-old Manila Metro Rail Transit System (MRT) comprise the rail network that serves the metropolitan area. Commuting in this area has been impacted by long lines at train stations, passengers waiting by the side of the road, crowded public transportation, and hours lost in transit (Philippine Daily Inquirer, 2022).

Moreover, Salita (2024) argued that there is severe traffic congestion, particularly at the bridges during rush hour in the morning and evening. Also, numerous people commute to the city from neighboring towns, which increases the traffic issue.

*Commuter’s Time and Experiences*

Moreno (2023) emphasized the substantial influence of traffic congestion on public transportation commuters in the Philippines. Moreover, it identifies critical determinants contributing to traffic congestion, such as insufficient road infrastructure, unmonitored street parking, and ineffective traffic flow management. Consequently, the prevailing state of public commuting in the Philippines resulted in prolonged travel durations, increased stress levels, and decreased productivity.

Han et al. (2022) explore the impact of commuting time on the quality of life in China, revealing a negative correlation. Extended commutes are associated with lower satisfaction with work and life, health issues, and reduced physical activity. However, the study emphasizes a positive note: expanding public transportation, particularly the development of subways, can effectively alleviate commuting challenges, underscoring the crucial role of efficient transport infrastructure in enhancing overall well-being in urban environments.

The study by Fallaria et al. (2019) deliberated upon the psychological repercussions experienced by commuters due to extended travel durations during peak hours of congestion. This bears relevance to the current study as it offers insights that could aid in proposing potential remedies for issues encountered by commuters, including stress and anxiety stemming from the crisis. The research aims to address these challenges by presenting informed alternatives to alleviate the strain on individuals during their journeys home.

Fallaria et al. (2019) focus on Metropolitan Manila's status as a global leader in traffic congestion. The study identifies essential themes in commuting culture and preferences for public transit modes using covert participant observation and personal interviews. The findings emphasize integrating Filipino values, such as patience and resilience, into commuting practices. Recommendations include implementing an Internet-based platform for commuters and developing summary reports for stakeholders and officials to address the complex challenges of traffic congestion (Fallaria et al., 2019).

Furthermore, according to a Boston Consulting Group investigation, Filipinos lost around P100,000 in income each year due to the 16 days on average that they spent delayed in traffic. The Japan International Cooperation Agency estimates that the country's inefficient transportation system, poor connectivity, and terrible traffic cost it a daily P3.5 billion in lost revenue (Philippine Daily Inquirer, 2022).

*Application Development*

In the study conducted by Wasilewski and Zabierowski (2021), a comparison was made among three mobile application development tools: Java, Flutter, and Kotlin/Native, specifically in processing and analyzing data from sensors. The relevance of this investigation lies in the persistent preference among developers for Java, owing to its perceived advantages such as reduced file size and enhanced processing speed. However, it is noteworthy that the research pertains to developing an application across two distinct operating systems.

According to Payne (2024), the historical challenges and complexities associated with app development on multiple platforms have been a notable concern. Nevertheless, the Flutter UI framework presents a viable solution to this challenge, which enables the creation of iOS and Android applications using a singular codebase. Consequently, the proposed approach involves utilizing Flutter for the front-end development, while Java will be employed for the back-end programming aspect.

Jimoh et al. (2020) focus on developing a real-time vehicle tracking system using wireless technologies and GPS-based algorithms to enhance public transportation in urban areas. The study emphasizes the integration of GPS and GPRS technologies for efficient fleet management. Additionally, the authors introduce genetic algorithms for GPS satellite selection and cost calculation, ensuring effective end-to-end communication. The study details the integration of hardware and software components, including a Raspberry Pi processor and GSM modules. Rigorous testing demonstrates the system's accuracy, with minor dilution errors, fair navigation, and low latency. The conclusion underscores the system's potential for city-wide application at minimal cost per terminal, reducing waiting anxiety, improving fleet management, and contributing to environmental pollution reduction in urban transportation (Jimoh et al., 2020).

*Mobile Applications on Transportation*

* *Mobile Applications for Commuters*

Pelonia (2022) stated that Moovit, Sakay.ph, Angkas, Google Maps, and Reddit are the 5 must-have apps for commuters to navigate Metro Manila. Moovit provides users with a detailed route map showing how to get to their location. Depending on how they want to go, users can select from the app's recommended routes to walk, ride in a jeep, or take the bus. According to the current pricing matrix, Sakay.ph provides an estimate of the fare cost for jeepneys, buses, and other modes of transportation, along with information on commuting options and directions. Angkas can provide a ride, particularly in situations when there isn't any mass transit accessible. Google Maps can be used not only for driving and navigation, but also for commuting options since it gives users an estimated arrival time depending on how they want to travel. Lastly, Reddit gives users a first-hand experience guide from the other Redditors.

According to Wu et al. (2022), the widespread adoption of smartphones has facilitated the utilization of electronic ride-hailing applications. These apps have proven efficient in mitigating information disparities, offering convenience and simplicity in travel arrangements. Consequently, it can be inferred that contemporary technology holds significant potential to enhance the overall commuting experience for the public.

AltexSoft (2018) argued that the perfect transit app must consist of the following: a map, a trip assistant, a ticket vending machine, and a mini encyclopedia. When it comes to functionality, every developer of apps for public transport should aim for versatility. All apps need basic features like maps, schedules, location tracking, and real-time notifications, regardless of whether they are meant to assist hundreds of destinations, be customized for a particular transportation company, or educate users about the landmarks in a city.

* *GPS Tracking System*

Moreover, emphasizing the "Global Positioning System (GPS) Based Public Transit Tracking System (2015)" as a cost-effective solution, the system's real-time tracking capabilities provide current bus locations and timely messages to registered passengers when the bus reaches specific stops. The GPS-based tracking system efficiently collects real-time GPS data from the driver's Android mobile device. This reinforces the central themes of optimizing public transit, informed decision-making, and encouraging the preference for public transport over private vehicles.

* *Challenges and Opportunities of Mobile Applications*

The advantages and difficulties of transportation-related applications were discussed in the study conducted by Siuhi and Mwakalonge (2016). The study aims to identify and summarize key features of existing mobile applications that may have applications in the transportation industry. In this paper, planning routes, ridesharing and carpooling, traffic safety, parking information, gathering transportation data, fuel emissions, and consumption, and travel information are among the opportunities that are made clear. Transportation agencies, engineering students collecting field data (e.g., travel speed, travel time, vehicle count), and the general traveling public (e.g., route planning, ridesharing/carpooling, parking, traffic safety, and travel information) are among the potential users of the applications in the field of transportation. It is stated that the significant use of smart mobile applications has the potential to be very beneficial, particularly in reducing travel time, cost, and vehicle emissions. However, users' interactions with these applications can physically, visually, and cognitively distract them from their primary task of walking or driving.

Moreover, Ramizo's (2019) research significantly contributes to understanding urban transit dynamics, particularly in Metro Manila. The study compares conventional public transport with ride-hailing apps, addressing the limited empirical analysis available from the user's perspective. Through in-depth semi-structured interviews, Ramizo (2019) unveils challenges conventional transport users face, emphasizing infrastructure inadequacies and law enforcement challenges. Additionally, the study underscores ride-hailing technologies' perceived convenience and reliability as alternative solutions (Ramizo, 2019).

An article published by the United States Department of Transportation - Federal Highway Administration (2020) concludes that there are already several advantages to the widespread deployment of behavioral processes from psychology and economics in transportation apps. A few of these advantages are reducing cognitive load with robust search tools, enhancing traveler control over trips both literally and figuratively, boosting confidence in carpooling services, altering norms surrounding transportation, like the simplicity of mobile ticketing, directly influencing prices by enabling more affordable services and shifting perceptions of value across multiple modes, enhancing information availability and thereby influencing service usage, leveraging current social pressures that already exist and creating new ones to influence.

Abeygunawardana (2014) investigated ways to utilize mobile devices for data collection, with the ultimate goal of reducing travel time, and ultimately enhancing transportation systems; specifically, he reported on the development of data collection in the field of transportation research, such as using GPS sensor on the smartphone along with network communication capabilities of the phone. Also, the paper discussed a few novel case studies that used cell phones to identify road conditions and give end users useful information. Moreover, an in-depth analysis of one implementation of a smart data collection application was provided, specifically to collect accident scenes using iPhone applications which would extract data from various sources and submit them to a web service. In this part, they experimented by collecting data through mobile apps and paper forms and comparing them afterward. This revealed that the mobile app was much faster than the paper form.

The study of Li et.al. (2012) designed and developed a mobile application that allows users to report and visualize traffic data and a unique server architecture that takes advantage of Cloud storage while ensuring high query performance. In addition, they created a carpooling recommendation system based on traces collected from TrafficPulse users.

**Synthesis of Reviewed Literature and Studies**

In the Philippines, public transportation is considered toxic, and the experience has been called dehumanizing. Hours of waiting for a ride and getting trapped in congested streets are the main common challenges Filipino commuters encounter. These problems are caused by a severe scarcity of transportation supplies, which worsens due to an ineffective and disorganized public transportation system (Philippine Daily Inquirer, 2022).

The traffic conditions are extremely terrible for millions of commuters who travel through Metro Manila each day to work or school (Philippine Daily Inquirer, 2022). In 2021, the number of public utility vehicles (PUVs) in operation was classified by the various types of vehicles in the Philippines. Approximately 42.6 thousand public utility jeepneys were in service as of October 2021. Additionally, 21.7 thousand taxis and roughly 25.5 thousand transport network vehicle services (TNVS) were operating nationwide (Statista Research Department, 2023).

Moreover, Salita (2024) stated that the three main modes of public transportation in the Metropolitan Manila area are taxis, jeepneys (little buses constructed on the chassis of jeeps), and buses. He argued that there is severe traffic congestion, particularly at the bridges during rush hour in the morning and evening. Also, numerous people commute to the city from neighboring towns, which increases the traffic issue.

Additionally, Moreno (2023) emphasized the substantial influence of traffic congestion on public transportation commuters in the Philippines. It identifies critical determinants contributing to traffic congestion, such as insufficient road infrastructure, unmonitored street parking, and ineffective traffic flow management. Also, Han et al. (2022) explore the impact of commuting time on the quality of life in China, revealing a negative correlation. Extended commutes are associated with lower satisfaction with work and life, health issues, and reduced physical activity. Fallaria et al. (2019) deliberated upon the psychological repercussions experienced by commuters due to extended travel durations during peak hours of congestion.

Sunio et al.'s (2019) research highlights the ongoing Public Utility Vehicle Modernization Program (PUVMP) in the Philippines as a focal point for socio-technical transition studies within the transport sector. Launched and executed by the national government, the PUVMP aims at transformative reconditioning in the country's public land transportation industry.

According to Wu et al. (2022), the widespread adoption of smartphones has facilitated the utilization of electronic ride-hailing applications. These apps have proven efficient in mitigating information disparities, offering convenience and simplicity in travel arrangements. Consequently, it can be inferred that contemporary technology holds significant potential to enhance the overall commuting experience for the public.

Also, Pelonia (2022) stated that Moovit, Sakay.ph, Angkas, Google Maps, and Reddit are the 5 must-have apps for commuters to navigate Metro Manila. In connection with that, an article published by the United States Department of Transportation - Federal Highway Administration (2020), concludes that there are already several advantages to the widespread deployment of behavioral processes from psychology and economics in transportation apps.

The study of Ramizo (2019) underscores ride-hailing technologies' perceived convenience and reliability as alternative solutions. Additionally, Abeygunawardana (2014) investigated ways to utilize mobile devices for data collection, with the ultimate goal of reducing travel time, and ultimately enhancing transportation systems; specifically, he reported on the development of data collection in the field of transportation research, such as using GPS sensor on the smartphone along with network communication capabilities of the phone.

Furthermore, the study of Li et.al. (2012) designed and developed a mobile application that allows users to report and visualize traffic data and a unique server architecture that takes advantage of Cloud storage while ensuring high query performance. Also, AltexSoft (2018) argued that the perfect transit app must consist of the following: a map, a trip assistant, a ticket vending machine, and a mini encyclopedia. All apps need basic features like maps, schedules, location tracking, and real-time notifications, regardless of whether they are meant to assist hundreds of destinations, be customized for a particular transportation company, or educate users about the landmarks in a city.

Besides, emphasizing the "Global Positioning System (GPS) Based Public Transit Tracking System (2015)" as a cost-effective solution, the system's real-time tracking capabilities provide current bus locations and timely messages to registered passengers when the bus reaches specific stops. Jimoh et al. (2020) focus on developing a real-time vehicle tracking system using wireless technologies and GPS-based algorithms to enhance public transportation in urban areas. The conclusion underscores the system's potential for city-wide application at minimal cost per terminal, reducing waiting anxiety, improving fleet management, and contributing to environmental pollution reduction in urban transportation.

Transportation agencies, engineering students collecting field data (e.g., travel speed, travel time, vehicle count), and the general traveling public (e.g., route planning, ridesharing/carpooling, parking, traffic safety, and travel information) are among the potential users of the applications in the field of transportation. It is stated that the significant use of smart mobile applications can be very beneficial, particularly in reducing travel time, cost, and vehicle emissions (Siuhi and Mwakalonge, 2016).

# Chapter 3

**METHODOLOGY**

This chapter covers the research method that was employed, the research locale, the research participants, the sampling technique used for this study, a description of the research instrument that will be used, the data-gathering procedures, the statistical treatment of data, the flowchart of research design, the application process flowchart and flow diagram, and application system design.

**Research Method**

The primary goal of this research is to develop a mobile application that monitors public vehicles in real-time, which helps users determine the most convenient public transportation options based on various circumstances. The researchers initially identified transportation challenges in the Philippines and proposed a mobile application as a solution. To specify application details and features, surveys were conducted using Google Forms, divided into two phases: an initial survey and a final survey after application testing. Respondents' feedback provided data for analysis and employed a research approach.

Initial survey results guided the selection of the Android and iPhone operating systems, along with features such as estimated time of arrival, fare details, route information, and commuting options in the mobile application. From there, the researchers developed the application. The researchers utilized Figma in designing the mobile application's user interface and then used Visual Studio Code as the development environment. For the database of the mobile application, Firebase's real-time database was used, and its administration was managed through Firebase's built-in tools. After the development of the application, the researchers conducted application testing and evaluation. For this, the researchers conducted several trials to evaluate the system. The results of the testing and evaluation were then analyzed. The analyzed results were then used to conclude the study.

# Research Locale

This research focuses on the transportation route within Rizal Province —a vital thoroughfare for daily commuters involving various Public Utility Vehicles (PUVs). The study aims to develop a user-friendly GPS tracking and monitoring application. The focus on this commuter route facilitates a comprehensive examination of challenges typical of urban commuting, such as traffic congestion. Given the route's significance, marked by a substantial commuter volume, it serves as an optimal location for implementing and evaluating the proposed technology. This approach offers valuable insights into commuter challenges and potential enhancements to the overall transportation experience, providing perspectives on the efficacy of existing transportation methods and the potential impact of the GPS tracking and monitoring system on commuting within the broader transportation system.

**Research Participants**

In this study, the researchers focused on commuters and public utility vehicle (PUV) drivers in Rizal Province, particularly in municipalities such as Angono, Antipolo City, Cainta, Rodriguez, San Mateo, Tanay, Taytay, and Teresa. The study consisted of two main phases: a preliminary survey and field experiments (or post-survey).

During the preliminary survey phase, commuters from Rizal Province were asked to complete a questionnaire via Google Forms. From the respondents who agreed to participate in the second phase, which involved evaluating the Route4Me application, a sample of participants was selected to assess the accuracy, reliability, usability, portability, and maintainability of the mobile application.

**Sampling Technique**

The researchers used a combination of snowball and purposive sampling techniques in this study. Purposive sampling, also called judgmental sampling, relies on the researcher's assessment to choose which individuals will provide the most data to achieve the study's objectives (Nikolopoulou, 2022a).

Moreover, the snowball sampling method, commonly referred to as chain sampling or network sampling, begins with one or more respondents to the research and then proceeds based on recommendations from the individuals who participated. This process is repeated until you achieve the required sample size (Nikolopoulou, 2022b).

# Research Instrument

Researchers administered pre- and post-surveys via Google Forms. The surveys introduced the study, obtained consent, gathered demographics, addressed key objectives, and concluded with appreciation to respondents and a feedback request. Findings informed the development of a mobile application.

This research employed a software evaluation approach built upon the established criteria outlined in the first part of the ISO 9126 standard (ISO/IEC, 2011). This international standard provides a comprehensive framework for assessing software quality. The ISO 9126-1 model categorizes software quality into six key aspects: features that work as intended (functionality), dependable performance (reliability), how easy it is to use (usability), efficiency in resource usage, ease of future modifications (maintainability), and adaptability to different environments (portability). During the application testing phase, participants were presented with a questionnaire directly aligned with these ISO 9126-1 aspects (GeeksforGeeks, 2020).

The assessment of the application employed various methods to gauge its overall quality. To ensure the application functioned as intended and delivered consistent results, researchers conducted tests that examined its features and reliability. Additionally, a formula was used to evaluate how easy it would be to modify the application in the future, a key factor for long-term use.

To understand user experience, a user-friendly four-point rating scale was implemented. This scale offered clear response options such as "Very Satisfied," "Satisfied," "Dissatisfied," and "Very Dissatisfied." By eliminating a neutral middle ground, the scale encouraged participants to express a clear opinion on their experience with the application. The scores from this scale provided valuable insights into how users perceived the application's usability and overall functionality.

**Table 1. 4-Point Likert Scale for Satisfaction**

|  |  |  |  |
| --- | --- | --- | --- |
| Very Dissatisfied (1) | Dissatisfied (2) | Satisfied (3) | Very Satisfied (4) |

# Data Gathering Procedures

Survey questionnaires were given to them to assess the needs of the commuters within Rizal Province as they are the participants in this study. They should be 18 years of age and/or older, have experience commuting, and understand English or Filipino to comprehend the research survey questionnaire.

Before the respondents answer the survey form, they will be asked for consent to participate in the study, which the researchers gave along with a letter of approval. Google Forms was used to collect the survey data for the initial survey of the study. The sociodemographic information from the respondents was utilized exclusively for this study. Only the respondents had access to it, which would be deleted once the study was completed.

The researchers developed an application where all the required features were designed and implemented using the agile software development strategy with regard to mobile application user needs. To assess the effectiveness of the mobile application, field experiments were conducted to assess the accuracy, reliability, usability, portability, and maintainability of the Route4Me mobile application.

# Statistical Treatment of Data

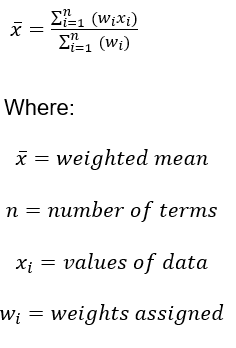
A structured questionnaire was developed for the initial survey, consisting of close-ended questions to formulate quantitative data. The researchers interpreted the results based on the numeric value or percentages of the gathered responses. The insights and perspectives of the respondents on geolocation technology for public commuting on mobile applications were obtained through analysis. Thereafter, a proposed solution was formulated from the gathered data.

Moreover, after the application's development stage, the developed system's quality will be determined by testing it against measured values. The study will perform a User Acceptance Test (UAT), a method for assessing the system's quality. User Acceptance Testing is the final stage in the system testing process after the system has completed the development stage. It aims to ensure that the solution created is following the needs of system users. In addition, this test will use ISO 9126 dimensions and the scale used will be the Likert scale (Wulandari et. al., 2023). The International Organization for Standardization (ISO) 9126 proposes several characteristics of software quality testing. ISO 9126 testing is a test to determine the quality of the software in terms of accuracy, reliability, usability, portability, and maintainability (Wulandari et.al., 2023). The ISO 9126 measurement formula is:

(1)

The results will be obtained from the survey questionnaires answered by the respondents for the testing.

Furthermore, to determine the feasibility of the mobile-based application, the researchers will use statistical treatment in which the weighted mean will be used to determine the average in each criterion and the overall weighted mean for the application. This measure will determine how much the respondents are personally satisfied or dissatisfied with the statements as shown in Equation 2.



(2)

# 

# 

# 

# 

# 

# Scoring and Interpretation

To interpret the level of satisfaction towards the application by the calculated weighted mean, the following scales and interpretation will be used:

|  |  |  |  |
| --- | --- | --- | --- |
| **Very Dissatisfied** | **Dissatisfied** | **Satisfied** | **Very Satisfied** |
| 1.00 ≤ x ≤ 1.75 | 1.75 ≤ x ≤ 2.50 | 2.50 ≤ x ≤ 3.25 | 3.25 ≤ x ≤ 4.00 |

**Table 2. Interpretation for the Computed Weighted Means**

Table 2 will also be used in interpreting the overall weighted mean obtained per criterion in software evaluation as the level of satisfaction equates to the acceptability of the application.

# Flowchart of Research Design

**Analysis of problems related to public transportation**

Formulation of research questions and the identification of purpose

**Development of solutions for the found problems**

Conceptualization and initial creation of Route4Me mobile application

**Conduction of preliminary survey**

Data collection related to public utility vehicle commuters and their mobile phone specifications, commuting schedule, and fare matrix

**Processing and integration of data to Route4Me mobile application**

Integration of data from the preliminary survey into application development

**Deployment and evaluation of the Route4Me mobile application**

Testing the mobile application to the respondents from the preliminary survey and through experiments

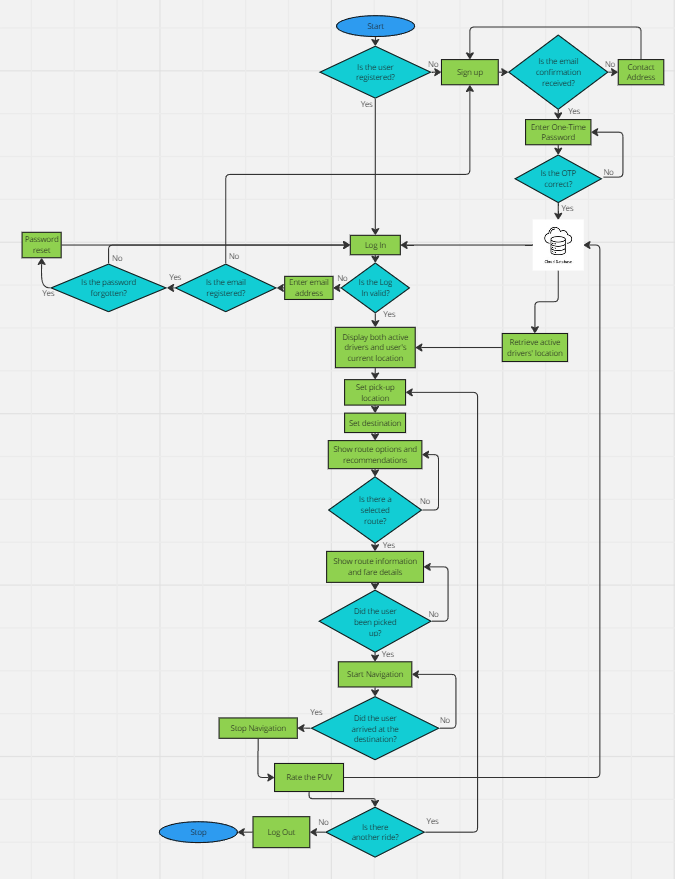
**Documentation and Analysis of results**

Design principles are generated and documented

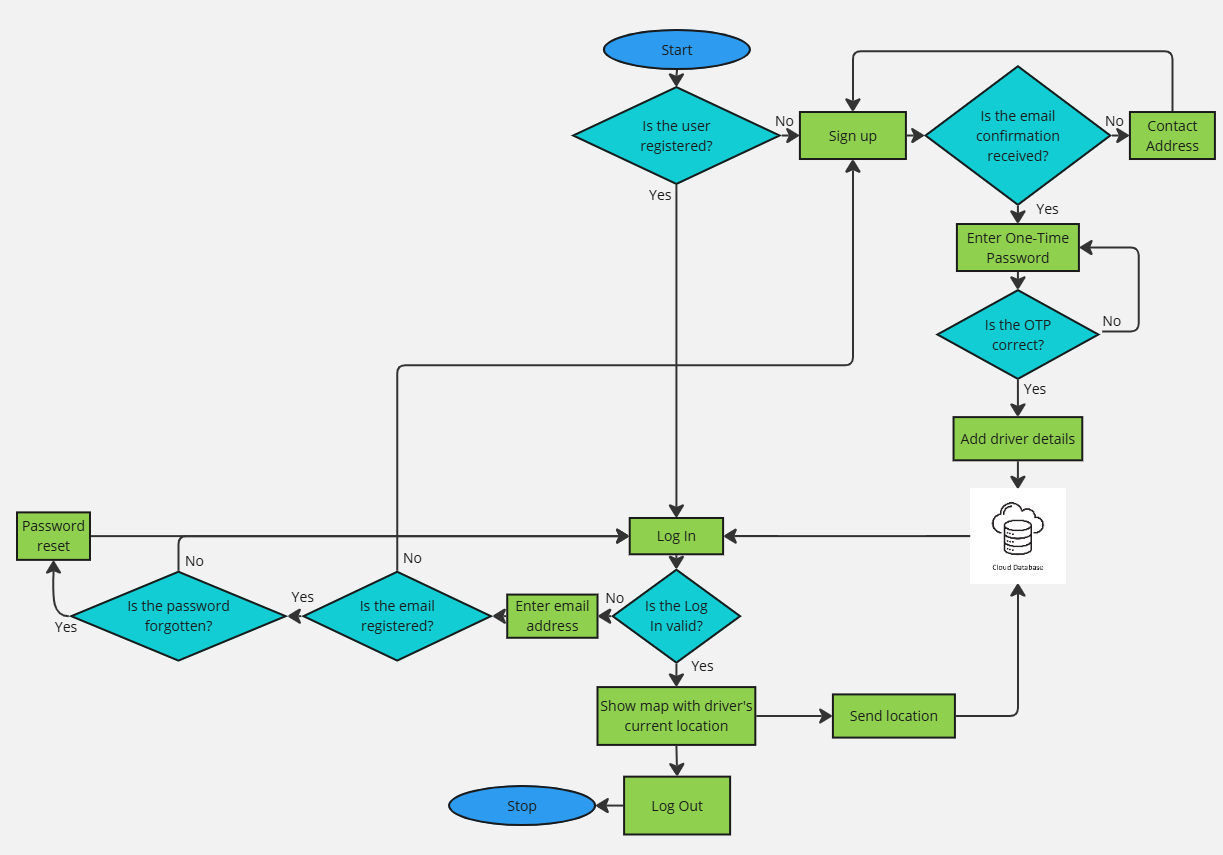
***Figure 2. Flowchart of Research Design***

The researchers devised a research design to outline their study, as shown in Figure 2. The research analyzes public utility vehicles and commuters’ issues and problems. It consists of the formulation of research questions and the identification of the purpose of the study. The researchers then develop a solution based on the problems found. The initial survey was conducted to gather data for the solution; it includes data related to public utility vehicle commuters and their mobile phone specifications, commuting schedules, and fare matrices. The application was then developed based on the identified problems and the gathered data. After the data integration and application development, the researchers will conduct experiments and survey respondents to test the system. Subsequently, they will document and analyze the results to generate design principles.

**Application Process Flowchart**

****

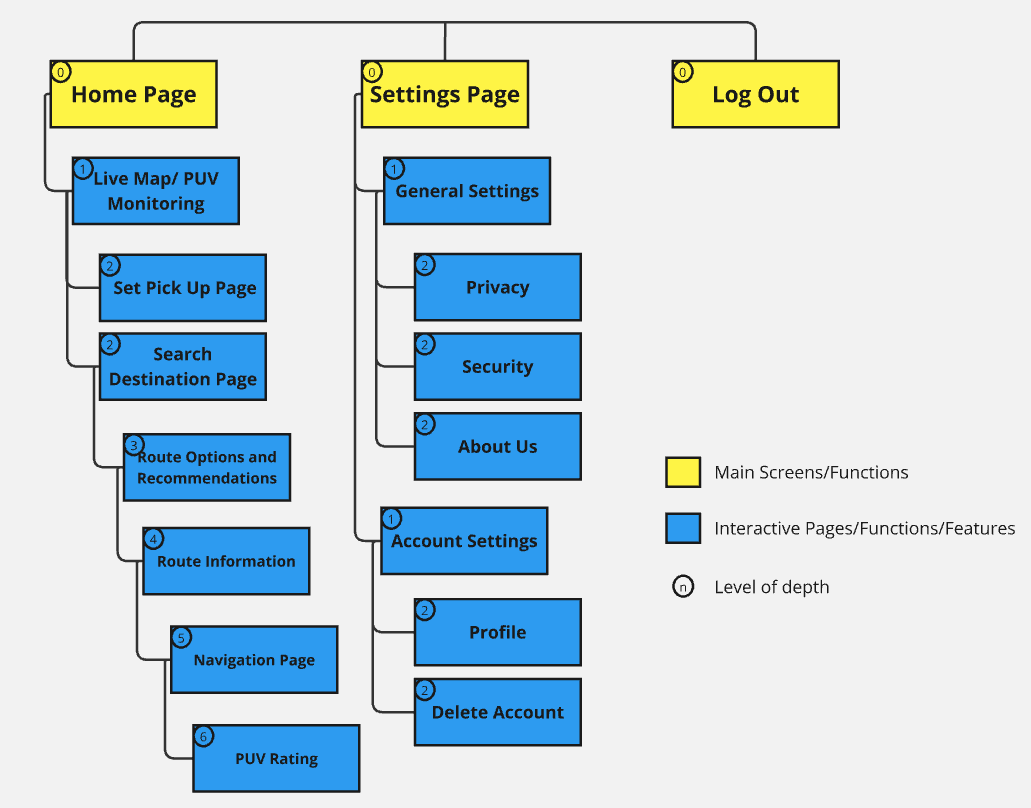
***Figure 3. Process Flow Chart for Commuter’s Route4Me Mobile Application***

****

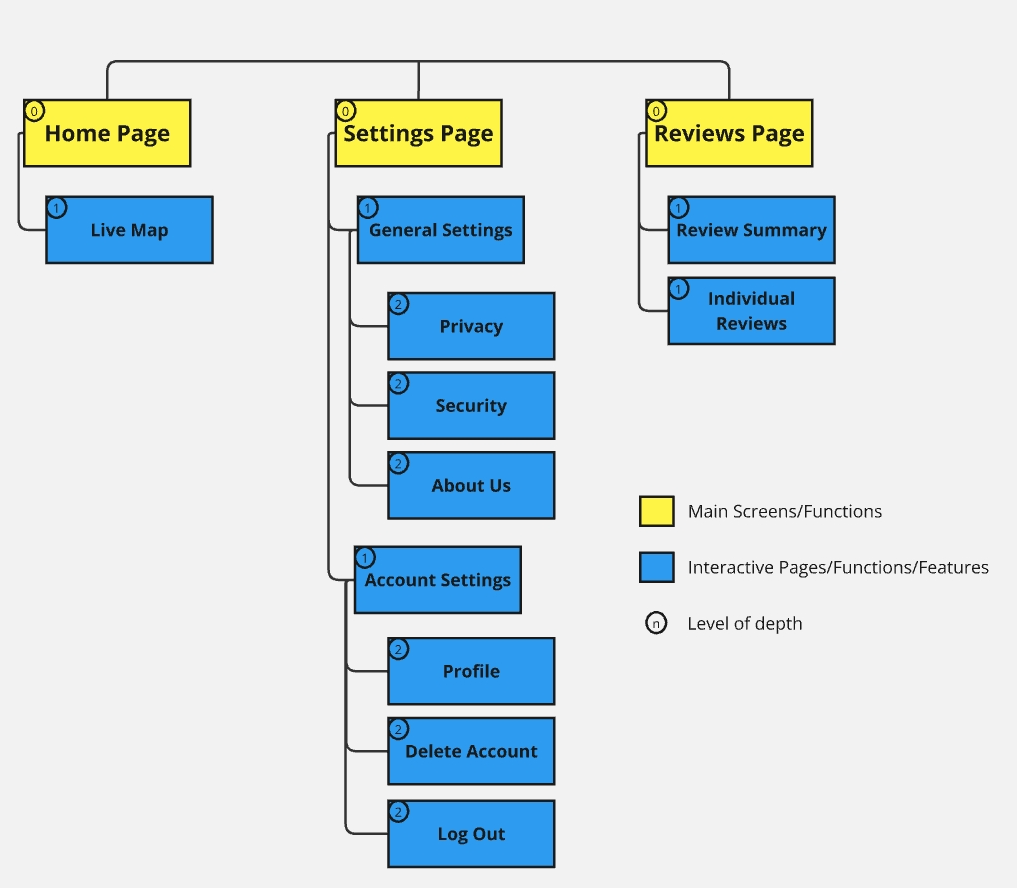
***Figure 4. Process Flow Chart for Driver’s Route4Me Mobile Application***

The flow charts depicted in Figures 4 and 5 illustrate the comprehensive workflow of the Route4Me mobile application. The application will be created in two categories catering to commuters and drivers. Both categories encompass processes such as registration, authentication, and the input of user details. For commuters, the Route4Me mobile application entails the monitoring and tracking of Public Utility Vehicles (PUVs), as well as the completion of it. In contrast, the Route4Me application for drivers does not involve monitoring of PUVs; instead, it operates in the background with an additional layer of user registry verification to enhance security.

**Application Flow Diagram**

****

***Figure 5. Flow Diagram for Commuter’s Route4Me Mobile Application***

****

***Figure 6. Flow Diagram for Driver’s Route4Me Mobile Application***

The above figures present a hierarchical diagram that describes the various components of the Route4Me mobile application, addressing both commuters and drivers. These visual representations offer insights into the overall organizational structure of the application. Moreover, they facilitate categorizing distinct elements within the application, enhancing navigability.

**Field Experiments**

The researchers conducted field experiments to assess the accuracy, reliability, usability, portability, and maintainability of the Route4Me mobile application.

**Experiment No. 1**

**Accuracy of the Features of the Route4Me Mobile Application**

**Objective:**

This experiment seeks to assess the accuracy of the features of the Route4Me mobile applications.

**Materials:**

1. Android Mobile Device and/or iOS Mobile Device
2. Network Connectivity

**Procedures:**

1. Open the application on each user’s smartphone.
2. Upon user registration, verify if the user information accurately appears within the mobile application.
3. If it does, indicate it in the checkbox column.

**Table. 3 Feature Testing**

|  |  |  |
| --- | --- | --- |
| **QUESTIONS** | **TRIALS** | |
| **✔** | **✘** |
| 1. Has the verification link been sent to the user's email? |  |  |
| 1. Is the email address for registration the same as the one used for logging in? |  |  |
| 1. Is the password created during registration the same as the one used for logging in? |  |  |
| 1. Is the registered name correctly displayed on the account profile? |  |  |
| 1. Is the user's current location accurately displayed on the account profile? |  |  |

**Experiment No. 2**

**Reliability of the Route4Me Mobile Application**

**Objective:**

This experiment assesses the application’s reliability, specifically focusing on its ability to maintain service provision under predefined conditions, including testing its resilience to failures and recovery capabilities thereafter.

**Materials:**

1. Android Mobile Device and/or iOS Mobile Device
2. Network Connectivity
3. Test Cases

**Procedures:**

1. Allow users to install the application on their mobile devices
2. Start the application and log in using the designated test passenger and test driver user profiles.
3. Utilize the application to track public utility vehicles (PUVs) and observe the performance in handling such tasks.
4. Force the mobile application to crash by inputting invalid user data, simulating network connection failures, or executing unexpected actions.
5. Confirm the application's capacity to resume functioning post-failure.
6. Upon completion of testing, uninstall the mobile application.

**Table. 4 Reliability Testing**

|  |  |  |
| --- | --- | --- |
| **QUESTIONS** | **TRIALS** | |
| **✔** | **✘** |
| 1. Is the software equipped to manage and handle errors effectively? |  |  |
| 1. Have the majority of software faults been mitigated over time? |  |  |
| 1. Is the software able to recover and retrieve lost data after the event of a failure? |  |  |

**Experiment No. 3**

**Usability of the Route4Me Application**

**Objective:**

This experiment seeks to assess how users perceive the usability of the Route4Me mobile application through a Likert scale. Each trial corresponds to a different user who has agreed to test the Route4Me mobile application. Additionally, it calculates the weighted mean for each question/statement, which is necessary for evaluating each criterion.

**Materials:**

1. Android Mobile Device and/or iOS Mobile Device
2. Network Connectivity

**Procedures:**

1. Allow the user to stay idle in the mobile application.
2. Using a Likert scale, let the user complete a prepared questionnaire based on their experience in using the mobile application. The Likert scale ratings are as follows: 1 for very dissatisfied, 2 for dissatisfied, 3 for satisfied, and 4 for very satisfied.
3. The responses from the PUV drivers and commuters, based on their level of satisfaction with every statement, will be recorded and tabulated to provide a structured format of data to visually compare the frequency of each scale.

**Table. 5 Usability Testing**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **QUESTIONS/STATEMENTS** | **Likert Scale** | | | |
| **1** | **2** | **3** | **4** |
| 1. The user can determine whether the Route4Me App meets their needs. (Appropriateness Recognizability) |  |  |  |  |
| 1. The users can utilize the Route4Me App to achieve specific goals, learning to use the system effectively, efficiently, safely, and with satisfaction in a given context. (Learnability) |  |  |  |  |
| 1. The Route4Me App has features that make it easy to operate and control. (Operability) |  |  |  |  |
| 1. The Route4Me App protects users from making errors. (User Error Protection) |  |  |  |  |
| 1. The Route4Me user interface offers pleasing and satisfying interaction for the user. (User Interface Aesthetics) |  |  |  |  |

**Experiment No. 4**

**Portability Test**

**Portability of the Route4Me Application**

**Objective:**

This experiment seeks to evaluate the application’s portability, or the system’s capability to be installed, run, and used under specified specifications.

**Materials:**

1. Android Mobile Device and/or iOS Mobile Device
2. Network Connectivity

**Procedures:**

1. Install the application on the user’s mobile devices.
2. Start the application.
3. Use the application in portrait and landscape mode.
4. Run the application using Wi-Fi and mobile data.
5. Check the system’s performance by running other applications in the background.
6. Explore the application in other environments and observe its performance.
7. Upon completion of testing, uninstall the mobile application.

**Table. 6 Portability Testing**

|  |  |  |
| --- | --- | --- |
| **QUESTIONS** | **Trials** | |
| **✔** | **✘** |
| 1. Is the application compatible with the operating system of the user’s device? |  |  |
| 1. Does the application display correctly on the screen of the user’s device? |  |  |
| 1. Does the application perform and respond well when switching between portrait and landscape orientations? |  |  |
| 1. Is the application suitable for the device’s memory capacities, such as RAM and ROM? |  |  |
| 1. Does the application continue to work when other applications are running in the background? |  |  |
| 1. Does the application operate with various network connectivity, such as Wi-Fi and mobile data? |  |  |
| 1. Is the application’s uninstallation process successful? |  |  |

**Experiment No. 5**

**Maintainability Index of the Route4Me Application**

**Objective:**

This experiment seeks to evaluate the maintainability of the application, which refers to the software product's ability to be modified. The maintainability index calculates a number between 0 and 100, indicating how easily the code can be maintained. A higher maintainability index signifies that the code is easier to maintain.

**Materials:**

1. Administration Tool (phpMyAdmin)
2. Android Studio
3. Calculator

**Procedures:**

1. Allow the application developer access the administration tool.
2. Use the calculator to compute the maintainability index of the application using the formula below.

**Maintainability Index**

**where:**

MI - Maintainability Index

HV - Halstead’s Volume

CC - Cyclomatic Complexity

LOC - Lines of Code

# Chapter 4

**RESULT AND DISCUSSION**

This chapter presents the result and discussion of the testing method used to assess the application's accuracy, reliability, usability, portability and maintainability and surveys carried out by the researchers. Initially, to aid in the development of the application, a preliminary survey was conducted to gather information on the participants' mobile devices and data usage. The survey included 65 participants: 59 commuters and 6 drivers, maintaining an approximately 1:6 ratio.

**Preliminary Survey Results**

1. **Mobile Device Specification**

**A. 1. Operating System of Mobile Device**

The first survey question addressed the operating system (OS) of the mobile devices used by the drivers and commuters. The results are displayed in Figure 7 and Figure 8.

**What is the operating system (OS) of your mobile device?**



Figure 7. Operating System (OS) of Commuters’ Mobile Devices

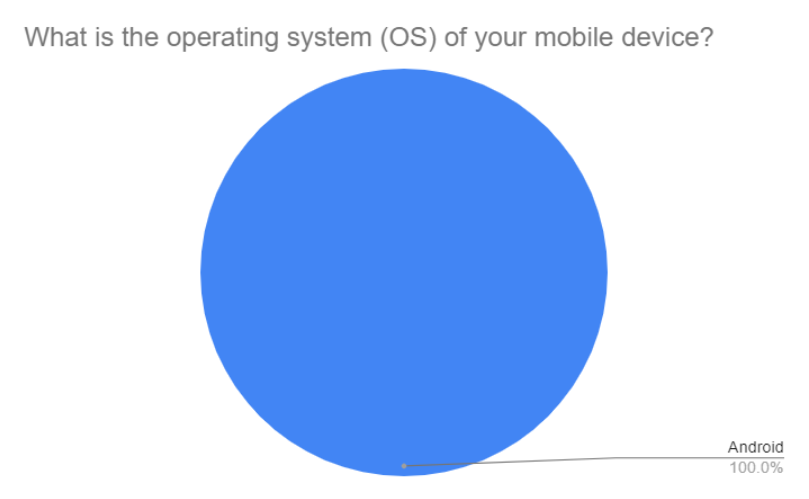


Figure 8. Operating System (OS) of Drivers' Mobile Devices

The study examines the types of phone operating systems (OS) used by commuters and drivers to better understand the target audience. The survey showed that many participants, both commuters (64.4%) and drivers (100%), used Android phones. Only 35.6% of commuters were identified as iOS users. This information about the prominence of different phone OS types was important for developing the application.

**A. 2. Android Version of Mobile Device**

**For Android users, what is the version of the operating system (OS) of your mobile device you commonly use while commuting?**

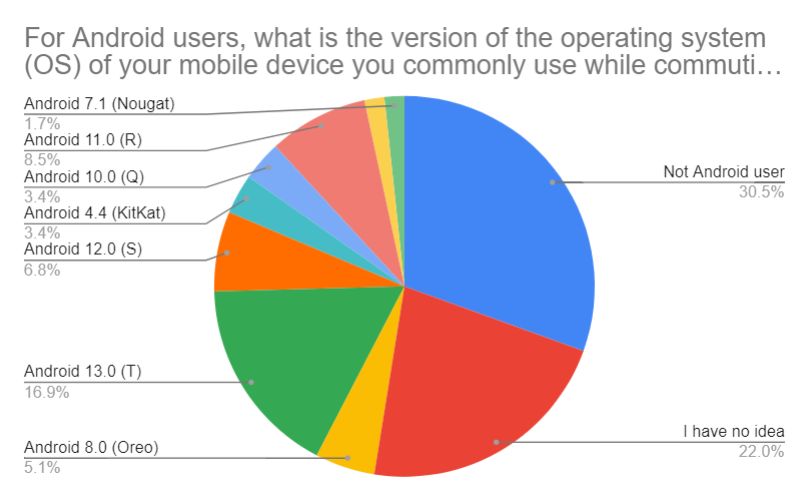


Figure 9. OS Android Version on Commuters' Devices

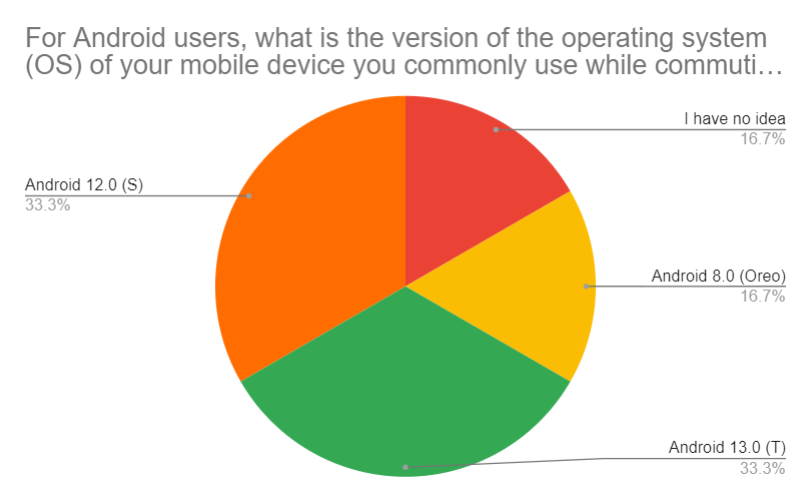


Figure 10. OS Android Version on Drivers' Devices

To ensure application compatibility across the target user base, the study investigated the distribution of Android versions among commuters and drivers (Figures 9 and 10). Compatibility with various Android versions is critical for application development, particularly for complex applications like Route4Me which likely utilizes multiple features and Application Programming Interfaces (APIs). Incompatibility between the application and a user's Android version could potentially hinder the application's functionality.

**A. 3. iOS Version of Mobile Device**

**For iOS users, what is the version of the operating system (OS) of your mobile device you commonly use while commuting?**

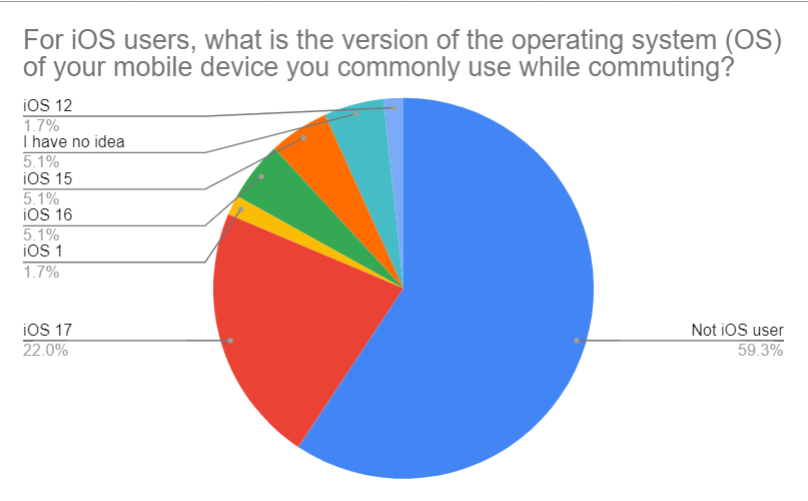


Figure 11. OS iOS Version on Commuters' Devices

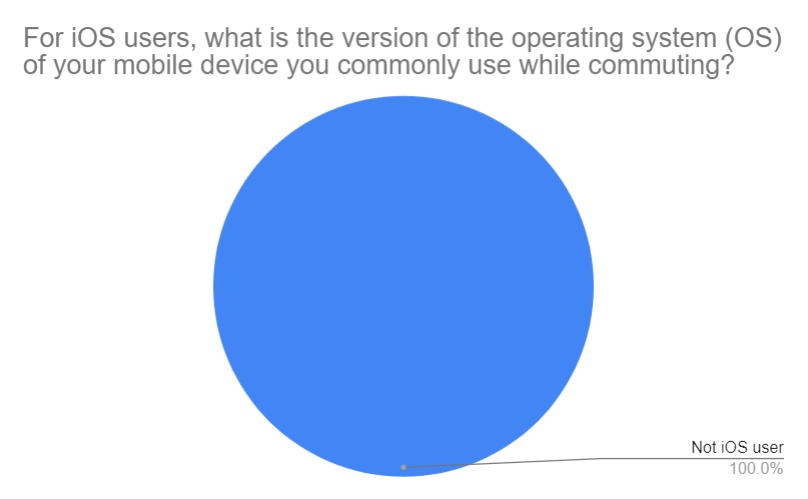


Figure 12. OS iOS Version on Drivers' Devices

Similar to the investigation of Android versions, researchers also examined the distribution of iOS versions among commuters' devices (Figure 11). Ensuring compatibility across different iOS versions is equally important for mobile application development. Like its Android version, the Route4Me application likely relies on various features and functionalities that might be hindered if a user's iOS version is incompatible. Understanding the spread of iOS versions among commuters is crucial for guaranteeing the application's smooth operation for a diverse user base.

1. **Mobile Data Usage**

**B. 1. Mobile Data Utilization Response**

This survey question addresses commuters’ and drivers' mobile data usage, providing valuable insights into their internet connectivity. The findings, illustrated in Figures 13 and 14, reveal the following results.

**How often do you have access to the internet data during your daily commute?**

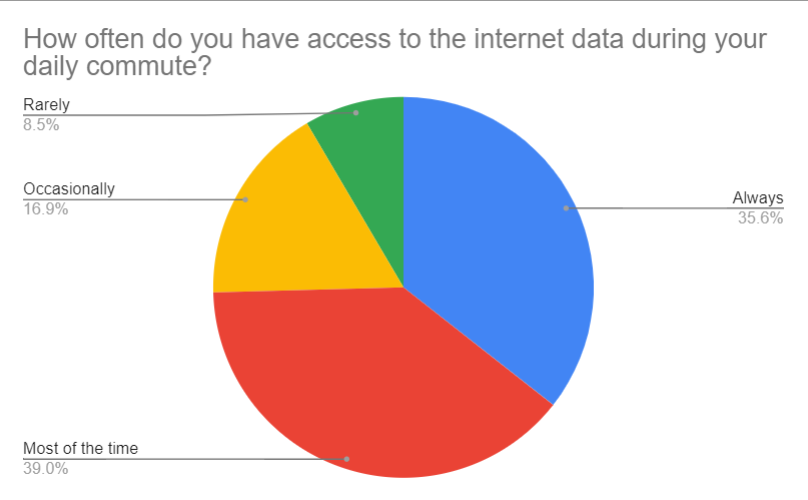


Figure 13. Commuters’ Response to Mobile Data Usage

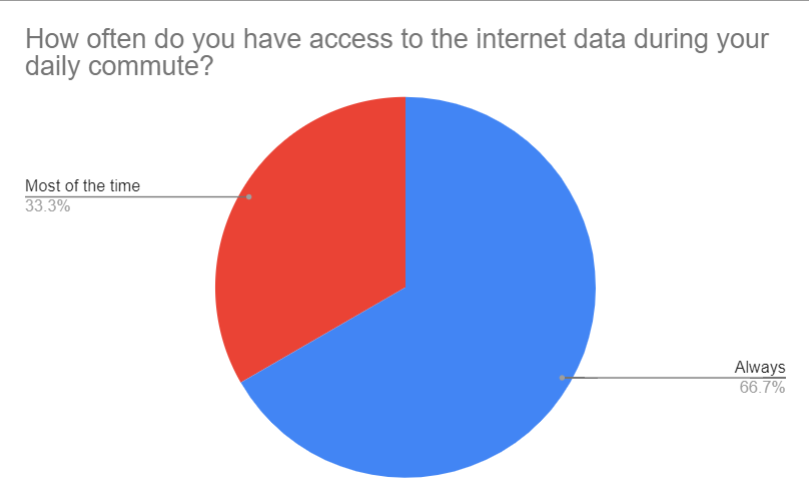


Figure 14. Drivers' Response to Mobile Data Usage

Figure 13 shows that all commuters participating in the study use mobile data services. This suggests that mobile data has become a widely used and essential way for commuters to connect to the internet. Based on Figure 13, the researchers can see that 100% of commuters who rely on mobile data emphasize the importance of ensuring the Route4Me application is optimized for mobile data usage. Since commuters depend heavily on mobile data, it is crucial for the application to function smoothly when accessed through this connection. This will allow them to take advantage of the various features and functionalities offered by the app.

On the other hand, Figure 14 reveals most drivers (66.7%) always use mobile data, and 33.3% use data most of the time. This finding suggests that while mobile data is the preferred way for most drivers to connect, a relevant percentage relies on less consistent access. Considering these results, future application development might involve exploring ways to accommodate users who may not have consistent mobile data access.

**B. 2. Average Daily Internet Usage Duration**

This survey question concerns the quantity of internet data consumed daily by commuters and drivers. The outcomes in Figures 15 and 16 reveal the following observations:

**On average, how many hours a day do you access the internet data?**

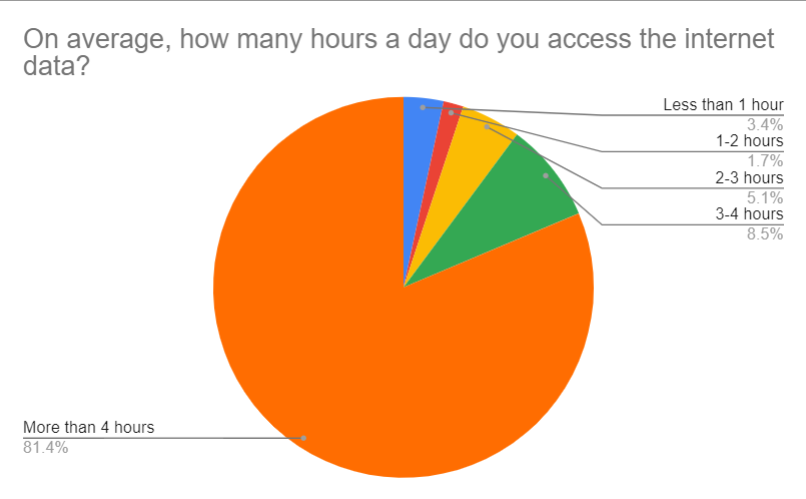


Figure 15. Commuters’ Daily Internet Data Usage

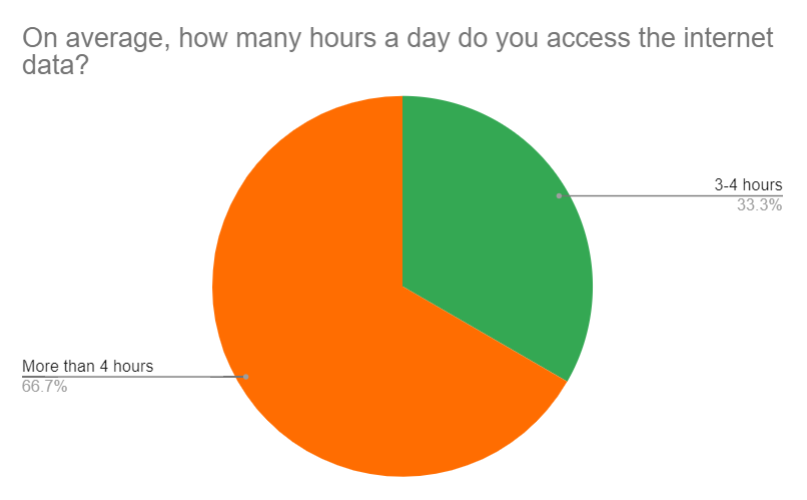


Figure 16. Drivers' Daily Internet Data Usage

Figure 15 shows that a large portion (81.4%) of commuters reported using the Internet or their mobile data for more than four hours daily. This suggests that many commuter participants spend a significant amount of time online.

Similarly, Figure 16 reveals a comparable pattern among drivers. Over half (66.7%) of drivers also reported using the internet for more than four hours daily. This data suggests that commuters and drivers use the internet for similar lengths of time each day.

The finding that both commuter and driver groups demonstrate similar internet usage habits strengthens the feasibility of the study. This overlap implies that both categories of participants have the technical skills necessary to use the application effectively.

**B. 3. Average Weekly Cellular Usage Cost**

**How much is your average mobile data/cellular data per week?**

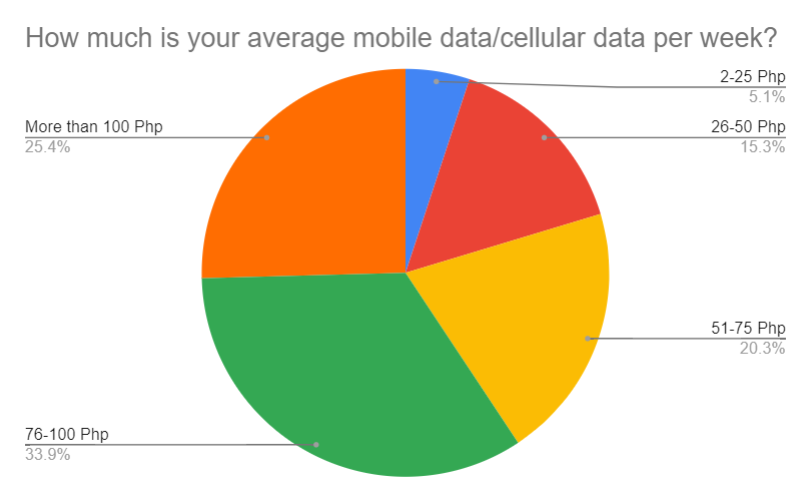


Figure 17. Commuters' Average Weekly Cellular Usage Cost

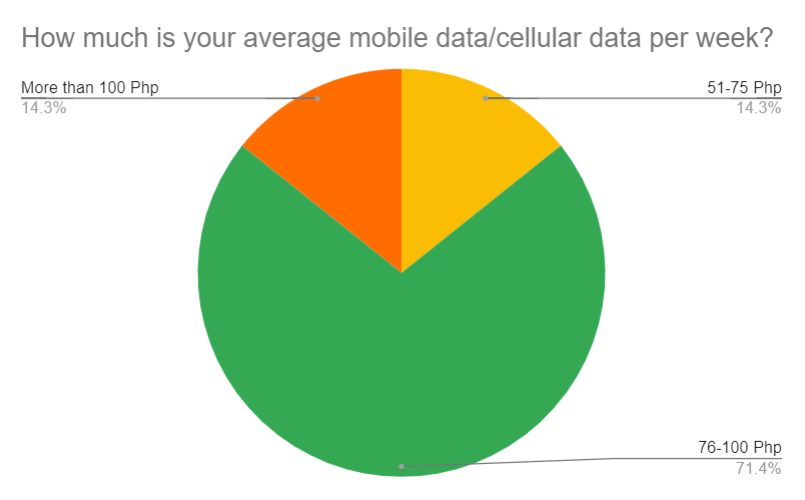


Figure 18. Drivers' Average Weekly Cellular Usage Cost

An analysis of Figures 17 and 18 explores the average weekly cellular usage cost for commuters and drivers. The findings suggest notable trends in mobile data spending habits correlated with daily internet usage.

Figures 15 and 16 reveal that a significant portion of commuters (81.4%) and drivers (66.7%) use the internet for more than four hours daily. This substantial usage indicates a likely need for higher data plans to support their consumption needs.

Figures 17 and 18 show both groups' average weekly cellular usage cost. The high daily data usage suggests commuters and drivers likely invest in phone plans that accommodate extensive internet needs through pre-paid or monthly plans with larger data allowances.

Understanding these spending patterns and data usage habits is valuable for optimizing applications like Route4Me for mobile data usage. Since commuters and drivers demonstrate high internet usage, ensuring the application functions efficiently with mobile data is important for enhancing user experience and satisfaction during commutes.

1. **Public Transport Commuting Schedule**

**C. 1. Usual Commuting Time**

In this question, participants were asked about their usual commuting schedules to work or school, providing insights into their commuting habits and busiest travel times.

**For passengers, what time do you usually commute to work or school?**

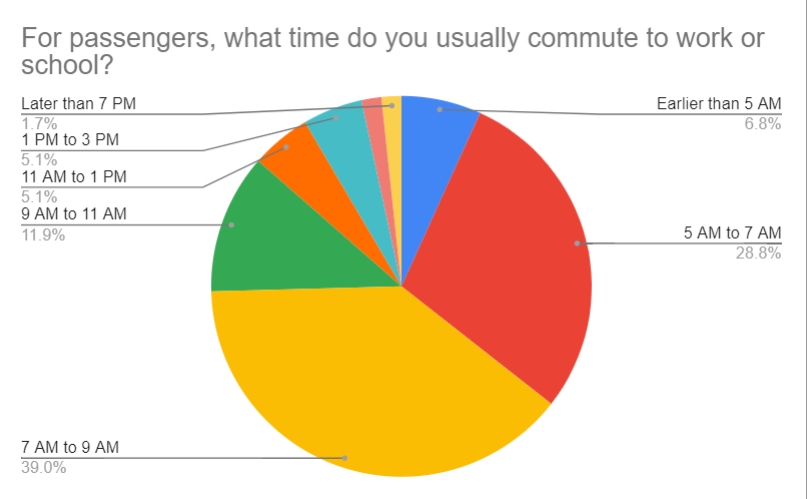


Figure 19. Passengers' Usual Commuting Hours

Figure 19 shows the distribution of times passengers typically leave for work or school. The data reveals a clear peak between 7 AM and 9 AM. This suggests a significant portion of passengers travel during these morning hours. In contrast, a smaller group (1.7%) of passengers commute from 5 PM to 7 PM and later than 7 PM, indicating a limited number of evening and late-night travelers.

**For passengers, what time do you usually go home from work or school?**

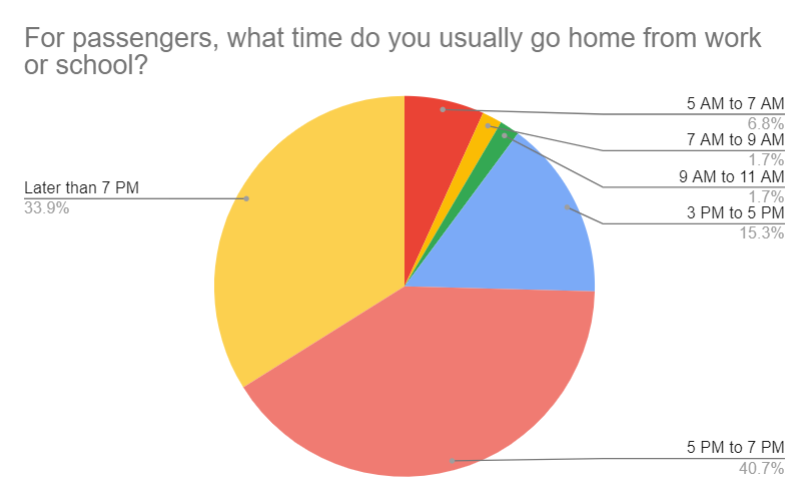


Figure 20. Passengers' Usual Time To Go Home

Figure 20 illustrates how passengers typically leave work or school to head home. The data shows a clear peak between 5 PM and 7 PM, indicating that many passengers travel during the evening rush hour, unlike the morning rush hour associated with commuting to work or school. A significant portion of passengers also commute later than 7 PM, suggesting a presence of late-evening travelers.

This information about passengers' commuting times is valuable for developing and improving the Route4Me application. Understanding peak travel times and passenger behavior allows the app to be tailored to efficiently provide real-time tracking of PUVs. Additionally, the application could incorporate features that gather user information (like location data) during peak commuting periods. This approach can optimize PUVs allocation and service during these busy times, leading to a smoother commuting experience for passengers.

1. **Mobility Applications Usage**

**D. 1. Mobility Applications Experience**

In this section, participants were asked about their experiences with various mobility applications used during commuting. The responses provide insights into the most-used applications, their helpfulness, and their safety features, highlighting user preferences and the importance of certain functionalities.

**Which mobility applications have you tried using when commuting? Check all that apply.**

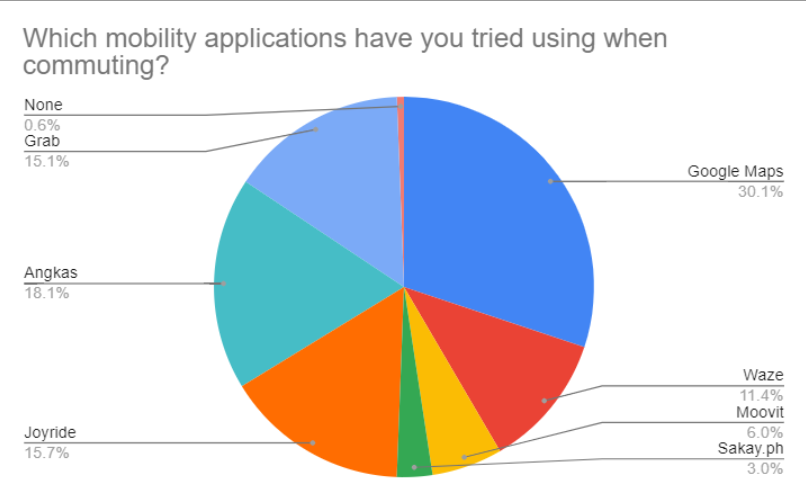
****

Figure 21. Mobility Applications Tried by Commuters

Figure 21 Google Maps is the most well-known among commuting apps, followed by Angkas and Joyride. Apps like Moovit and Sakay.ph show more limited usage, possibly due to their specialized functions or regional focus. The data highlights a preference for navigation and ride-hailing apps that offer convenience and real-time updates.

**Which mobility applications help you in your commuting experience? Check all that apply.**

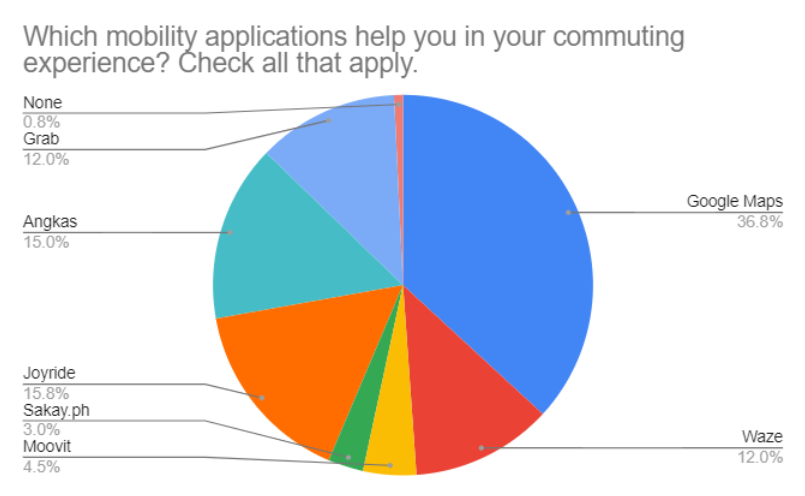
****

Figure 22. Mobility Applications That Help Commuters

Figure 22 shows that Google Maps stands out as the most helpful application for commuting, followed by Joyride and Angkas. The moderate helpfulness of Waze and Grab highlights their role in easing the commuting process for many users. Moovit and Sakay.ph, while less widely regarded as helpful, still serve important roles for their specific user groups.

**Which mobility applications make your commuting experience better, safer, and more secure? Check all that apply.**

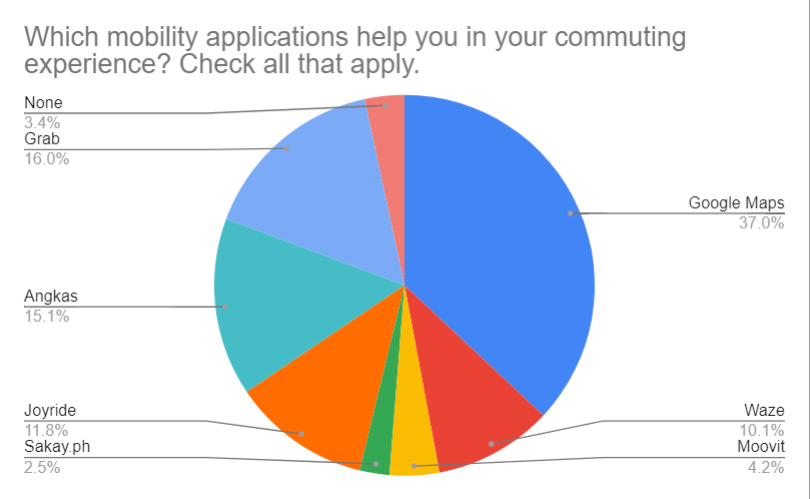
****

Figure 23. Mobility Applications That Help Commuters

Figure 23 emphasizes that Google Maps is the leading application for improving the safety and security of the commuting experience, followed by Grab and Angkas. Joyride also ranks relatively high, indicating its importance for a secure commute. Waze, Moovit, and Sakay. ph, while less frequently cited, still provides valuable safety and security features for their users. A few respondents do not find these applications helpful in enhancing their commute.

The data highlights a clear preference for comprehensive navigation tools like Google Maps, which dominate usage and perceived benefits. Ride-hailing services like Joyride, Angkas, and Grab also enhance the commuting experience by offering reliable and secure transportation options. Specialized apps like Waze, Moovit, and Sakay.ph serve important but more targeted functions. The results emphasize the importance of accurate navigation, real-time updates, and reliable ride-hailing services in improving the commuting experience, making it safer, more efficient, and more secure.

**Field Experiment Results**

In this section, each trial corresponds to a user who has consented to test the real-time GPS tracking application Route4Me. The feedback from passengers and PUV drivers regarding the application's usability, based on their satisfaction levels in each experiment, is represented by checkmarks. The data was presented in tables to compare the frequency of each scale, allowing for a structured format. This also included the weighted mean for each statement, which is necessary for evaluating the application's effectiveness. Several trials were conducted to assess and evaluate the criteria: accuracy of the features, reliability, usability, and portability. The maintainability of the Route4Me application was evaluated using the formula for the maintainability index.

1. **Accuracy of the Features**

**Experiment No. 1**

**Accuracy of the Features of the Route4Me Mobile Application**

**Table 7. Responses to Feature Testing for Android Mobile Devices**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Trials** | **Question 1** | **Question 2** | **Question 3** | **Question 4** | **Question 5** |
| **1** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **2** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **3** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **4** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **5** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **6** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **7** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **8** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **9** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **10** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **11** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **12** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **13** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **14** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **15** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **16** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **17** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **18** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **19** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **20** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **21** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **22** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **23** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **24** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **25** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **26** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **27** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **28** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **29** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **30** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **31** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **32** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **33** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **34** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **35** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **36** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **37** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **38** | **✔** | **✔** | **✔** | **✔** | **✔** |

Table 7 illustrates the results from the first experiment, which focused on evaluating the accuracy of the Route4Me mobile application's features. The researchers conducted a series of tests on Android devices to measure how well the app performed in various scenarios.

Throughout the testing phase, the application consistently demonstrated accurate feature performance. The 38 participants registered using Android devices, and the verification link was promptly sent to their email addresses. This ensured a smooth registration process. Additionally, the same email address used for registration could successfully log into the app, providing a seamless experience for users.

Moreover, the passwords created during registration were consistently accepted during logins, showcasing the app's reliability in handling user credentials. The registered names were correctly displayed on the users' account profiles, which is essential for a personalized user experience. Most importantly, the app accurately displayed the users' current locations on their profiles, validating its GPS tracking capability.

The data and results from each trial in Experiment 1, as detailed in Table 7, emphasize the Route4Me application's accuracy on Android platforms. This high level of performance is crucial for delivering a dependable and efficient tool for real-time GPS tracking and route management mobile application to its users.

**Table 8. Responses to Feature Testing for iOS Mobile Devices**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Trials** | **Question 1** | **Question 2** | **Question 3** | **Question 4** | **Question 5** |
| **1** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **2** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **3** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **4** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **5** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **6** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **7** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **8** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **9** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **10** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **11** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **12** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **13** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **14** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **15** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **16** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **17** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **18** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **19** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **20** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **21** | **✔** | **✔** | **✔** | **✔** | **✔** |

Table 8 presents the outcomes of feature testing on the iOS version of the Route4Me mobile application. Across 21 trials, every trial yielded positive responses, demonstrating that the application’s features performed consistently and reliably on iOS devices. These results confirm that the Route4Me application effectively supports all tested features on iOS platforms, maintaining dependable performance throughout multiple trials.

1. **Reliability**

**Experiment No. 2**

**Reliability of the Route4Me Application**

**Table 9. Responses to Reliability Testing for Android Mobile Devices**

|  |  |  |  |
| --- | --- | --- | --- |
| **Trials** | **Question 1** | **Question 2** | **Question 3** |
| **1** | **✔** | **✔** | **✔** |
| **2** | **✔** | **✔** | **✔** |
| **3** | **✔** | **✔** | **✔** |
| **4** | **✔** | **✔** | **✔** |
| **5** | **✔** | **✔** | **✔** |
| **6** | **✔** | **✔** | **✔** |
| **7** | **✔** | **✔** | **✔** |
| **8** | **✔** | **✔** | **✔** |
| **9** | **✔** | **✔** | **✔** |
| **10** | **✔** | **✔** | **✔** |
| **11** | **✔** | **✔** | **✔** |
| **12** | **✔** | **✔** | **✔** |
| **13** | **✔** | **✔** | **✔** |
| **14** | **✔** | **✔** | **✔** |
| **15** | **✔** | **✔** | **✔** |
| **16** | **✔** | **✔** | **✔** |
| **17** | **✔** | **✔** | **✔** |
| **18** | **✔** | **✔** | **✔** |
| **19** | **✔** | **✔** | **✔** |
| **20** | **✔** | **✔** | **✔** |
| **21** | **✔** | **✔** | **✔** |
| **22** | **✔** | **✔** | **✔** |
| **23** | **✔** | **✔** | **✔** |
| **24** | **✔** | **✔** | **✔** |
| **25** | **✔** | **✔** | **✔** |
| **26** | **✔** | **✔** | **✔** |
| **27** | **✔** | **✔** | **✔** |
| **28** | **✔** | **✔** | **✔** |
| **29** | **✔** | **✔** | **✔** |
| **30** | **✔** | **✔** | **✔** |
| **31** | **✔** | **✔** | **✔** |
| **32** | **✔** | **✔** | **✔** |
| **33** | **✔** | **✔** | **✔** |
| **34** | **✔** | **✔** | **✔** |
| **35** | **✔** | **✔** | **✔** |
| **36** | **✔** | **✔** | **✔** |
| **37** | **✔** | **✔** | **✔** |
| **38** | **✔** | **✔** | **✔** |

Table 9 shows that the reliability testing of the Route4Me mobile app conducted on Android devices consistently showed dependable performance throughout numerous trials. Across all 38 trials, the Route4Me application consistently executed its functions without any failures or deviations from expected behavior. This series of tests underscored the application's robustness and ability to reliably deliver its intended functionalities under varying conditions.

In each trial, the application's performance was assessed against a set of predefined criteria, and in every instance, it met the expected outcomes. The software is equipped to manage and handle errors effectively. Also, the majority of software faults have been mitigated over time. Lastly, the software can recover and retrieve lost data after the event of a failure. These findings affirm that Route4Me operates reliably on Android devices across diverse scenarios, highlighting its stability and effectiveness in real-world usage. The uniformity of positive outcomes across all trials indicates high reliability and consistency in the application's performance on Android platforms.

Overall, the results from the reliability testing underscore Route4Me's capability to maintain consistent functionality across a range of conditions, providing users with assurance of reliable performance on their Android mobile devices.

**Reliability of the Route4Me Application**

**Table 10. Responses to Reliability Testing for iOS Mobile Devices**

|  |  |  |  |
| --- | --- | --- | --- |
| **Trials** | **Question 1** | **Question 2** | **Question 3** |
| **1** | **✔** | **✔** | **✔** |
| **2** | **✔** | **✔** | **✔** |
| **3** | **✔** | **✔** | **✔** |
| **4** | **✔** | **✔** | **✔** |
| **5** | **✔** | **✔** | **✔** |
| **6** | **✔** | **✔** | **✔** |
| **7** | **✔** | **✔** | **✔** |
| **8** | **✔** | **✔** | **✔** |
| **9** | **✔** | **✔** | **✔** |
| **10** | **✔** | **✔** | **✔** |
| **11** | **✔** | **✔** | **✔** |
| **12** | **✔** | **✔** | **✔** |
| **13** | **✔** | **✔** | **✔** |
| **14** | **✔** | **✔** | **✔** |
| **15** | **✔** | **✔** | **✔** |
| **16** | **✔** | **✔** | **✔** |
| **17** | **✔** | **✔** | **✔** |
| **18** | **✔** | **✔** | **✔** |
| **19** | **✔** | **✔** | **✔** |
| **20** | **✔** | **✔** | **✔** |
| **21** | **✔** | **✔** | **✔** |

Table 10 presents the findings from the reliability testing of the iOS version of the Route4Me mobile application. In all 21 trials, the application consistently received positive responses, indicating that it operates without errors or failures. These results highlight the Route4Me application's dependable and consistent performance on iOS devices.

1. **Usability**

**Experiment No. 3**

**Usability of the Route4Me Application**

**Table 11. Responses to Usability Testing on the Commuter Side**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Trials** | **Statement 1**  *The user can determine whether the Route4Me App meets their needs. (Appropriateness Recognizability)* | | | | **Statement 2**  *The users can utilize the Route4Me App to achieve specific goals, learning to use the system effectively, efficiently, safely, and with satisfaction in a given context. (Learnability)* | | | | **Statement 3**  *The Route4Me App has features that make it easy to operate and control. (Operability)* | | | |
| **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** |
| **1** |  |  |  | **✔** |  |  |  | **✔** |  |  | **✔** |  |
| **2** |  |  |  | **✔** |  |  | **✔** |  |  |  |  | **✔** |
| **3** |  |  |  | **✔** |  |  | **✔** |  |  |  | **✔** |  |
| **4** |  |  | **✔** |  |  |  | **✔** |  |  |  |  | **✔** |
| **5** |  |  | **✔** |  |  |  |  | **✔** |  |  | **✔** |  |
| **6** |  |  | **✔** |  |  |  |  | **✔** |  |  | **✔** |  |
| **7** |  |  |  | **✔** |  |  | **✔** |  |  |  |  | **✔** |
| **8** |  |  |  | **✔** |  |  | **✔** |  |  |  |  | **✔** |
| **9** |  |  |  | **✔** |  |  | **✔** |  |  |  |  | **✔** |
| **10** |  |  | **✔** |  |  |  |  | **✔** |  |  | **✔** |  |
| **11** |  |  |  | **✔** |  |  | **✔** |  |  |  | **✔** |  |
| **12** |  |  |  | **✔** |  |  |  | **✔** |  |  | **✔** |  |
| **13** |  |  | **✔** |  |  |  |  | **✔** |  |  |  | **✔** |
| **14** |  |  | **✔** |  |  |  |  | **✔** |  |  |  | **✔** |
| **15** |  |  | **✔** |  |  |  | **✔** |  |  |  | **✔** |  |
| **16** |  |  |  | **✔** |  |  | **✔** |  |  |  | **✔** |  |
| **17** |  |  |  | **✔** |  |  | **✔** |  |  |  | **✔** |  |
| **18** |  |  | **✔** |  |  |  |  | **✔** |  |  | **✔** |  |
| **19** |  |  |  | **✔** |  |  | **✔** |  |  |  |  | **✔** |
| **20** |  |  | **✔** |  |  |  | **✔** |  |  |  | **✔** |  |
| **21** |  |  | **✔** |  |  |  | **✔** |  |  |  | **✔** |  |
| **22** |  |  | **✔** |  |  |  |  | **✔** |  |  |  | **✔** |
| **23** |  |  |  | **✔** |  |  |  | **✔** |  |  |  | **✔** |
| **24** |  |  |  | **✔** |  |  | **✔** |  |  |  |  | **✔** |
| **25** |  |  |  | **✔** |  |  | **✔** |  |  |  |  | **✔** |
| **26** |  |  | **✔** |  |  |  | **✔** |  |  |  | **✔** |  |
| **27** |  |  |  | **✔** |  |  |  | **✔** |  |  | **✔** |  |
| **28** |  |  |  | **✔** |  |  | **✔** |  |  |  |  | **✔** |
| **29** |  |  | **✔** |  |  |  |  | **✔** |  |  | **✔** |  |
| **30** |  |  | **✔** |  |  |  | **✔** |  |  |  | **✔** |  |
| **31** |  |  | **✔** |  |  |  |  | **✔** |  |  |  | **✔** |
| **32** |  |  |  | **✔** |  |  | **✔** |  |  |  |  | **✔** |
| **33** |  |  |  | **✔** |  |  |  | **✔** |  |  |  | **✔** |
| **34** |  |  | **✔** |  |  |  |  | **✔** |  |  | **✔** |  |
| **35** |  |  |  | **✔** |  |  |  | **✔** |  |  |  | **✔** |
| **36** |  |  |  | **✔** |  |  | **✔** |  |  |  | **✔** |  |
| **37** |  |  | **✔** |  |  |  | **✔** |  |  |  |  | **✔** |
| **38** |  |  |  | **✔** |  |  |  | **✔** |  |  |  | **✔** |
| **39** |  |  |  | **✔** |  |  |  | **✔** |  |  |  | **✔** |
| **40** |  |  | **✔** |  |  |  |  | **✔** |  |  |  | **✔** |
| **41** |  |  |  | **✔** |  |  |  | **✔** |  |  | **✔** |  |
| **42** |  |  | **✔** |  |  |  | **✔** |  |  |  |  | **✔** |
| **43** |  |  |  | **✔** |  |  |  | **✔** |  |  |  | **✔** |
| **44** |  |  |  | **✔** |  |  | **✔** |  |  |  | **✔** |  |
| **45** |  |  | **✔** |  |  |  |  | **✔** |  |  |  | **✔** |
| **46** |  |  | **✔** |  |  |  |  | **✔** |  |  | **✔** |  |
| **47** |  |  |  | **✔** |  |  |  | **✔** |  |  |  | **✔** |
| **48** |  |  | **✔** |  |  |  | **✔** |  |  |  |  | **✔** |
| **49** |  |  |  | **✔** |  |  | **✔** |  |  |  | **✔** |  |
| **50** |  |  |  | **✔** |  |  | **✔** |  |  |  | **✔** |  |
| **51** |  |  | **✔** |  |  |  |  | **✔** |  |  | **✔** |  |
| **52** |  |  | **✔** |  |  |  | **✔** |  |  |  |  | **✔** |
| **53** |  |  | **✔** |  |  |  |  | **✔** |  |  | **✔** |  |
| **54** |  |  |  | **✔** |  |  | **✔** |  |  |  |  | **✔** |
| **55** |  |  |  | **✔** |  |  |  | **✔** |  |  |  | **✔** |
| **56** |  |  | **✔** |  |  |  | **✔** |  |  |  | **✔** |  |
| **57** |  |  |  | **✔** |  |  |  | **✔** |  |  |  | **✔** |
| **58** |  |  |  | **✔** |  |  |  | **✔** |  |  | **✔** |  |
| **59** |  |  |  | **✔** |  |  |  | **✔** |  |  | **✔** |  |
| **x̄** | **3.56** | | | | **3.51** | | | | **3.51** | | | |

Table 11 shows the results from the third experiment, which focused on determining the usability of the Route4Me mobile application. The Route4Me App exhibits high user satisfaction in terms of recognizability, learnability, and operability. With a mean score of 3.56 for appropriateness recognizability, users find it clear and straightforward to assess if the app meets their needs. Both learnability and operability received mean scores of 3.51, indicating that users find the app easy to learn, use effectively, and control. These consistent and high ratings reflect a well-balanced and user-friendly experience.

**Table 11. Responses to Usability Testing on the Commuter Side**

***(Continuation)***

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Trials** | **Statement 4**  *The Route4Me protects users against making errors. (User Error Protection)* | | | | **Statement 5**  *The user interface of Route4Me enables pleasing and satisfying interaction for the user. (User Interface Aesthetics)* | | | |
| **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** |
| **1** |  |  |  | **✔** |  |  | **✔** |  |
| **2** |  |  | **✔** |  |  |  |  | **✔** |
| **3** |  |  |  | **✔** |  |  |  | **✔** |
| **4** |  |  | **✔** |  |  |  |  | **✔** |
| **5** |  |  | **✔** |  |  |  | **✔** |  |
| **6** |  |  |  | **✔** |  |  | **✔** |  |
| **7** |  |  |  | **✔** |  |  | **✔** |  |
| **8** |  |  | **✔** |  |  |  | **✔** |  |
| **9** |  |  | **✔** |  |  |  |  | **✔** |
| **10** |  |  | **✔** |  |  |  |  | **✔** |
| **11** |  |  | **✔** |  |  |  | **✔** |  |
| **12** |  |  |  | **✔** |  |  | **✔** |  |
| **13** |  |  |  | **✔** |  |  |  | **✔** |
| **14** |  |  |  | **✔** |  |  | **✔** |  |
| **15** |  |  | **✔** |  |  |  |  | **✔** |
| **16** |  |  |  | **✔** |  |  |  | **✔** |
| **17** |  |  |  | **✔** |  |  |  | **✔** |
| **18** |  |  | **✔** |  |  |  | **✔** |  |
| **19** |  |  | **✔** |  |  |  |  | **✔** |
| **20** |  |  | **✔** |  |  |  | **✔** |  |
| **21** |  |  |  | **✔** |  |  | **✔** |  |
| **22** |  |  |  | **✔** |  |  | **✔** |  |
| **23** |  |  | **✔** |  |  |  | **✔** |  |
| **24** |  |  |  | **✔** |  |  |  | **✔** |
| **25** |  |  | **✔** |  |  |  | **✔** |  |
| **26** |  |  | **✔** |  |  |  |  | **✔** |
| **27** |  |  | **✔** |  |  |  | **✔** |  |
| **28** |  |  |  | **✔** |  |  |  | **✔** |
| **29** |  |  |  | **✔** |  |  |  | **✔** |
| **30** |  |  |  | **✔** |  |  |  | **✔** |
| **31** |  |  |  | **✔** |  |  | **✔** |  |
| **32** |  |  | **✔** |  |  |  | **✔** |  |
| **33** |  |  | **✔** |  |  |  |  | **✔** |
| **34** |  |  | **✔** |  |  |  | **✔** |  |
| **35** |  |  | **✔** |  |  |  | **✔** |  |
| **36** |  |  |  | **✔** |  |  | **✔** |  |
| **37** |  |  |  | **✔** |  |  |  | **✔** |
| **38** |  |  | **✔** |  |  |  | **✔** |  |
| **39** |  |  |  | **✔** |  |  |  | **✔** |
| **40** |  |  |  | **✔** |  |  | **✔** |  |
| **41** |  |  | **✔** |  |  |  | **✔** |  |
| **42** |  |  |  | **✔** |  |  |  | **✔** |
| **43** |  |  |  | **✔** |  |  |  | **✔** |
| **44** |  |  |  | **✔** |  |  |  | **✔** |
| **45** |  |  | **✔** |  |  |  | **✔** |  |
| **46** |  |  | **✔** |  |  |  |  | **✔** |
| **47** |  |  | **✔** |  |  |  |  | **✔** |
| **48** |  |  | **✔** |  |  |  | **✔** |  |
| **49** |  |  |  | **✔** |  |  | **✔** |  |
| **50** |  |  |  | **✔** |  |  | **✔** |  |
| **51** |  |  | **✔** |  |  |  |  | **✔** |
| **52** |  |  | **✔** |  |  |  | **✔** |  |
| **53** |  |  |  | **✔** |  |  | **✔** |  |
| **54** |  |  |  | **✔** |  |  | **✔** |  |
| **55** |  |  | **✔** |  |  |  | **✔** |  |
| **56** |  |  |  | **✔** |  |  |  | **✔** |
| **57** |  |  |  | **✔** |  |  | **✔** |  |
| **58** |  |  |  | **✔** |  |  |  | **✔** |
| **59** |  |  |  | **✔** |  |  |  | **✔** |
| **x̄** | **3.53** | | | | **3.46** | | | |
| **OVERALL** | **3.51** | | | | | | | |

The Route4Me App performs well in user error protection and user-interface aesthetics, though with slightly varied results. The mean score for user error protection is 3.53, indicating that users are confident in the app's ability to protect them from making mistakes. On the other hand, the user interface aesthetics had a slightly lower mean score of 3.46, suggesting that although consumers found the interface to be aesthetically pleasant and fulfilling, there is still potential for development to improve the user experience. Overall, the computed weighted mean for this experiment was 3.51, which means that commuters are very satisfied with this mobile application's usability.

**Table 12. Responses to Usability Testing on the Driver Side**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Trials** | **Statement 1**  *The user can determine whether the Route4Me App meets their needs. (Appropriateness Recognizability)* | | | | **Statement 2**  *The users can utilize the Route4Me App to achieve specific goals, learning to use the system effectively, efficiently, safely, and with satisfaction in a given context. (Learnability)* | | | | **Statement 3**  *The Route4Me App has features that make it easy to operate and control. (Operability)* | | | |
| **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** |
| **1** |  |  | **✔** |  |  |  |  | **✔** |  |  |  | **✔** |
| **2** |  |  |  | **✔** |  |  |  | **✔** |  |  |  | **✔** |
| **3** |  |  |  | **✔** |  |  |  | **✔** |  |  | **✔** |  |
| **4** |  |  | **✔** |  |  |  | **✔** |  |  |  | **✔** |  |
| **5** |  |  | **✔** |  |  |  | **✔** |  |  |  |  | **✔** |
| **6** |  |  |  | **✔** |  |  | **✔** |  |  |  | **✔** |  |
| **x̄** | **3.5** | | | | **3.5** | | | | **3.5** | | | |

Table 12 shows that the usability testing on the driver side had a positive experience using the Route4Me mobile application. For the first statement (Appropriateness Recognizability), most drivers agreed that the app effectively met their needs, with a mean of 3.5. Regarding learnability, feedback was generally positive, indicating a 3.5 mean that drivers found the app straightforward to learn and use. Operability consistently received high marks with a 3.5 mean, showing that drivers found the app easy to control and operate.

**Table 12. Responses to Usability Testing on the Driver Side**

***(Continuation)***

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Trials** | **Statement 4**  *The Route4Me protects users against making errors. (User Error Protection)* | | | | **Statement 5**  *The user interface of Route4Me enables pleasing and satisfying interaction for the user. (User Interface Aesthetics)* | | | |
| **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** |
| **1** |  |  |  | **✔** |  |  |  | **✔** |
| **2** |  |  |  | **✔** |  |  | **✔** |  |
| **3** |  |  |  | **✔** |  |  | **✔** |  |
| **4** |  |  |  | **✔** |  |  |  | **✔** |
| **5** |  |  | **✔** |  |  |  |  | **✔** |
| **6** |  |  |  | **✔** |  |  |  | **✔** |
| **x̄** | **3.83** | | | | **3.67** | | | |
| **OVERALL** | **3.6** | | | | | | | |

Additionally, Table 12 shows that in terms of user error protection, the Route4Me mobile app performed well, as drivers felt it effectively prevented mistakes with a mean of 3.83. Lastly, drivers appreciated the app's user interface, which was aesthetically pleasing and satisfying, with a mean of 3.67.

In summary of the result of experiment 3 on the driver side, the Route4Me application was well-received, excelling in meeting their needs, being easy to learn and use, providing effective control, preventing errors, and offering a pleasant user interface. This positive feedback across multiple areas suggests that the app is functional and user-friendly for its intended audience. The overall calculated weighted mean for this experiment was 3.6, indicating that drivers are very satisfied with the usefulness of this mobile application.

1. **Portability**

**Experiment No. 4**

**Portability of the Route4Me Application**

**Table 13. Responses to Portability Testing for Android Mobile Devices**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Trials** | **Question 1** | **Question 2** | **Question 3** | **Question 4** | **Question 5** | **Question 6** | **Question 7** |
| **1** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **2** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **3** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **4** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **5** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **6** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **7** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **8** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **9** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **10** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **11** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **12** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **13** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **14** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **15** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **16** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **17** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **18** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **19** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **20** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **21** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **22** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **23** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **24** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **25** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **26** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **27** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **28** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **29** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **30** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **31** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **32** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **33** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **34** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **35** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **36** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **37** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **38** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** |

Table 13 shows the portability testing of the Route4Me mobile app conducted on Android devices throughout numerous trials. Across 38 trials, the Route4Me application consistently executed its functions without significant issues on various Android devices. This suggests that the Route4Me application is highly portable across different Android devices.  
 These various tests and positive outcomes ensure that users with various Android devices can reliably use the Route4Me application without facing compatibility problems. Overall, the results from the portability testing underscore Route4Me’s capability to operate seamlessly across a diverse set of Android mobile devices.

**Table 14. Responses to Portability Testing for iOS Mobile Devices**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Trials** | **Question 1** | **Question 2** | **Question 3** | **Question 4** | **Question 5** | **Question 6** | **Question 7** |
| **1** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **2** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **3** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **4** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **5** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **6** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **7** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **8** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **9** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **10** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **11** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **12** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **13** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **14** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **15** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **16** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **17** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **18** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **19** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **20** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** |
| **21** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** | **✔** |

Table 14 shows the results from the portability testing on the iOS version of the Route4Me mobile application. Across 21 trials, every trial showed positive responses. This indicates the Route4Me application functions correctly and consistently across various iOS devices. The Route4Me app is highly portable, maintaining consistent functionality across various iOS devices, ensuring users have a reliable experience regardless of the specific device being used​.

**Experiment No. 5**

**Maintainability Index of the Route4Me Application**

Based on Sharma (2024), the maintainability index of the application was calculated using the experiment given in Chapter 3 and the formula below.

**where:**

MI - Maintainability Index

HV - Halstead’s Volume

CC - Cyclomatic Complexity

LOC - Lines of Code

Indicated in an older guideline, 65 or below was thought to be low index, tough to sustain; 65-85 as moderate, sustainably index; and anything above 85 as high index, easily sustainably. Considering that software systems are evolving nowadays, making them more complicated, it has also become difficult to develop codes that are uncomplex in order to be understood and manipulated easily. Consequently, a set of code metrics is now used to calculate the maintainability index, and the range guidelines have been adjusted: an index number <10 is considered low to maintain, while 10-19 is a near cut-off point, and 20 and above is highly maintainable.

Using the necessary metrics and the maintainability index formula, the application scored an index of 54.9536, indicating that the code is highly maintainable.

# Chapter 5

**SUMMARY OF FINDINGS, CONCLUSIONS AND RECOMMENDATIONS**

This chapter summarizes the data gathered following a series of tests and provides conclusions about the study’s objectives. Finally, a list of recommendations is included for interested readers and future researchers.

**Summary of Findings**

This part of the paper presents the overview of the results and their corresponding interpretations based on the survey and experimentations conducted by the researchers. This encompasses the preliminary survey and the software evaluation of the Route4Me mobile application. The results are elaborated below.

The findings of the Preliminary Survey are as follows:

* The survey analyzed the distribution of phone operating systems (OS) among commuters and drivers to ensure compatibility with the Route4Me application. The survey revealed that most commuters and drivers use Android phones, while a smaller but significant portion of commuters use iOS, emphasizing that Android is the predominant OS among the target audience. This highlights the importance of cross-version platform compatibility in developing the Route4Me mobile application.
* The survey also examined mobile data usage among commuters and drivers, providing insights into their internet connectivity. The findings indicate that commuters and drivers rely heavily on mobile data for internet access. 60.7% of the drivers always access the internet data during their daily commute, while 39% of the commuters use it most of the time. This indicates that the application must function smoothly on mobile data connections to meet user needs. Additionally, both commuters and drivers spend substantial time online, with 81.4% of commuters and 66.7% of drivers using the internet for more than four hours daily. This high usage suggests that the respondents likely have higher data plans to support their internet consumption. Furthermore, both groups' average weekly cellular usage costs indicate investment in plans with larger data allowances, highlighting the application's need for efficient data usage. A significant portion of commuters (33.9%) reported that their average weekly load ranged from Php 76-100. Likewise, most drivers (71.4%) also spent over Php 76-100 weekly.
* Passengers' commuting times from work or school to home were also analyzed in the survey. The data reveals that commuters exhibit peak commuting hours during morning and evening rush periods. A substantial number traveled to work or school between 7 AM and 9 AM (39%), while between 5 PM and 7 PM (40.7%) was the observed peak hours as the commuters returned to their home.
* The participants' experiences with various mobility applications used during commuting, providing insights into their usage, helpfulness, and safety features were also examined in the survey. The results show a strong preference among commuters for comprehensive navigation tools such as Google Maps, which are not only dominant in usage (30.1%), but also in perceived benefits or helpfulness (36.8%) and safety enhancements (37%) as shown in Figure 22 and Figure 23 respectively. It was also found that ride-hailing services like Joyride, Angkas, and Grab still got a significant percentage as they also provide convenience and real-time updates, offering reliable and secure transportation options, further enhancing the commuting experience among respondents.

The software evaluation findings show that:

* The application was evaluated for its accuracy. The results yield a strong performance on Android and iOS platforms, rated 100% in all trials. This indicates that the application could consistently deliver accurate features across various tests, from registration and login processes to user profile management and accurate GPS tracking. The results affirm that the application meets the user expectations for the application’s accuracy among different platforms.
* The application was evaluated for its reliability. The responses were rated 100% in all trials, indicating that the application is highly reliable. The application demonstrated robust performance under varying conditions, effectively handling errors and maintaining data integrity. The results in all trials for Android and iOS versions of the application affirm its capability to maintain consistent functionality and performance across different platforms.
* The usability of the application was also assessed. The results from the testing were rated 3.51 by the commuters and 3.6 by the drivers, highlighting user satisfaction across various aspects. The high ratings across multiple usability aspects for commuter's and driver’s applications, such as appropriateness recognizability, learnability, operability, user error protection, and user interface, indicate that the application successfully delivered a user-friendly navigation experience, enhancing its usability and appeal among users.
* Testing on the portability of the application was conducted. The results rated 100% in all trials, indicating the application is highly portable. This suggests that the application operates effectively on various Android and iOS devices without encountering significant compatibility issues, enhancing the application’s accessibility and usability on different devices and platforms.
* The maintainability of the application had an index of \_\_\_\_ which means that the code is highly maintainable…

**Conclusions**

This study aims to develop and evaluate Route4Me, a real-time GPS tracking application designed for PUV passengers within Rizal Province. Through intensive testing across various dimensions—accuracy of features, reliability, usability, and portability—the application has demonstrated remarkable effectiveness in meeting identified challenges and fulfilling the outlined objectives.

Route4Me proved exceptionally accurate and reliable on Android and iOS platforms. Thorough testing confirmed its ability to consistently provide precise GPS tracking, accurately display user locations, and maintain seamless functionality across diverse scenarios. These findings highlight Route4Me's crucial role in delivering dependable service critical for real-time navigation and efficient route management.

Feedback from usability testing highlighted Route4Me's user-friendly design and functionality. Users expressed satisfaction with its ease of use, noting its clear interface and effective error-prevention mechanisms. This positive user experience enhances accessibility for passengers and PUV drivers, reinforcing Route4Me's utility and user satisfaction.

Furthermore, portability testing across various Android and iOS devices confirmed Route4Me's consistent performance without compatibility issues. This versatility ensures reliable access and utilization of the application across different mobile platforms, enhancing its accessibility and practicality for a broad user base.

Route4Me effectively addresses critical needs in public transport navigation and management by providing a reliable, user-friendly solution that enhances commuter convenience and efficiency in Rizal Province. This study contributes valuable insights into developing and evaluating transportation technology solutions tailored to local commuting challenges. Therefore, Route4Me's success paves the way for future enhancements in user interface refinement, expanded platform compatibility, and integration of advanced features based on continuous user feedback.

**Recommendations**

The following recommendations are provided for future researchers interested in further improving this study.

* Explore other software and algorithms to develop a mobile application related to this study.
* Develop a notification system within the mobile application, notifying commuters when a tracked PUV arrives at their pick-up location and when it is approaching their destination.
* Include a payment processing feature allowing users to pay for rides using an e-wallet such as GCash.
* Expand the application’s scope to include other types of PUVs aside from buses and traditional and modern jeepneys, and possibly other country areas.
* Consider implementing a subscription to generate revenue, ensuring the financial sustainability and continued maintenance of the mobile application.

**References**

Abeygunawardana, T. (2014). Smart Data Collection Using Mobile Devices To Improve Transportation Systems. UNLV Theses, Dissertations, Professional Papers, and Capstones. https://doi.org/10.34917/5836071

AltexSoft. (2018, October 19). Best Public Transportation Apps: Bus, Rail & Public Transit. https://www.altexsoft.com/blog/public-transportation-apps-apis-and-platforms-maps-scheduling-trip-planning-and-mobile-ticketing/?fbclid=IwAR13Ceq9KmCroWbReTvYm-7lczcy30XgARXOrTX-5VU5hNc6DaO\_bTwC3Ak

Asia Pacific Foundation of Canada. (2023, March 8). Jeepneys, Backbone of Public Transport in Philippines, Face Modernization by Government. Asia Pacific Foundation of Canada. https://www.asiapacific.ca/publication/jeepneys-backbone-public-transport-philippines-face

Borole, N., Rout, D., Goel, N., Vedagiri, P., & Mathew, T. V. (2013). Multimodal Public Transit Trip Planner with Real-time Transit Data. Procedia - Social and Behavioral Sciences, 104, 775–784. https://doi.org/10.1016/j.sbspro.2013.11.172

Chipilska, I. (2023, September 14). What Is An Intelligent Transport System And How Does It Work? Modeshift. https://www.modeshift.com/what-is-an-intelligent-transport-system-and-how-does-it-work/#:~:text=The%20technology%20itself%20relies%20on

Chuenyindee, T., Ong, A. K. S., Ramos, J. P., Prasetyo, Y. T., Nadlifatin, R., Kurata, Y. B., & Sittiwatethanasiri, T. (2022). Public utility vehicle service quality and customer satisfaction in the Philippines during the COVID-19 pandemic. Utilities Policy, 75(101336), 101336. https://doi.org/10.1016/j.jup.2022.101336

Cruz, C. A. (2016, November 17). Top 10 Mobile Apps for Drivers and Commuters. Carmudi. https://www.carmudi.com.ph/journal/top-10-mobile-apps-drivers-commuters/

Exploring the Main Modes of Transportation in the Philippines. Secret Philippines (2023, March 25). https://secret-ph.com/exploring-the-main-modes-of-transportation-in-the-philippines/

Fallaria, A. J., Jesus, R. de, Carpio, M., Jacinto, F. L., Leon, L. D., Agapito, J., & Ramos, J. G. (2019). Emerging from the “worst”: An ethnography of the modern Filipino commuting culture behind the Metro Manila traffic crisis. MATEC Web of Conferences, 272, 01032. https://doi.org/10.1051/matecconf/201927201032

GeeksforGeeks. (2020, December 25). ISO/IEC 9126 in Software Engineering. GeeksforGeeks. https://www.geeksforgeeks.org/iso-iec-9126-in-software-engineering/

Global Positioning System (GPS) Based Transit Tracking System. (2015). International Journal of Technology Enhancements and Emerging Engineering Research, 3(6), 22–25. https://typeset.io/papers/global-positioning-system-gps-based-transit-tracking-system-1x63lx3289

Google. (2024). Plan your commute or trip - Android - Google Maps Help. Support.google.com. https://support.google.com/maps/answer/7565193?hl=en&co=GENIE.Platform%3DAndroid#:~:text=Find%20traffic%20%26%20transit%20info

Han, L., Peng, C., & Xu, Z. (2022). The Effect of Commuting Time on Quality of Life: Evidence from China. International Journal of Environmental Research and Public Health, 20(1), 573. https://doi.org/10.3390/ijerph20010573

iFranchise Philippines. (2018, November 20). UV Express Franchise - Information, Fees and Franchise Investment. IFranchise.ph. https://ifranchise.ph/uv-express-franchise/

Ignacio, C. P. (2023, August 23). Philippine public transport’s route to sustainability. BusinessWorld Online. https://www.bworldonline.com/special-features/2023/08/23/541412/philippine-public-transports-route-to-sustainability/

ISO. (2011). ISO/IEC 25010:2011. ISO. https://www.iso.org/standard/35733.html

Jimoh, O. D., Ajao, L. A., Adeleke, O. O., & Kolo, S. S. (2020). A Vehicle Tracking System Using Greedy Forwarding Algorithms for Public Transportation in Urban Arterial. IEEE Access, 8, 191706–191725. https://doi.org/10.1109/access.2020.3031488

Li, R.-Y., Liang, S., Lee, D.-W., & Byon, Y.-J. (2012). TrafficPulse : A Mobile GISystem for Transportation. Retrieved January 5, 2024, from http://wcl.cs.rpi.edu/pilots/library/papers/ACM-SIGSPATIAL-GIS02/MobiGIS2012-USB/MobiGIS2012\_files/MobiGIS2012-02.pdf

Luna, F. (2022, July 19). Commuter group: Most Filipino commuters say waiting time too long, PUV supply lacking. Philstar.com. https://www.philstar.com/headlines/2022/07/19/2196498/commuter-group-most-filipino-commuters-say-waiting-time-too-long-puv-supply-lacking

Moovit Inc. (2023). About Moovit: The leading MaaS solutions provider. Moovit. https://moovit.com/about-us/

Moreno, F. (2023). Traffic congestion and management in Zamboanga City, Philippines: The public transport commuters' point of view. 1, 44561. http://eprints.rclis.org/44561/1/Traffic%20Management%20-%20MORENO%20RESEARCH%20PAPER%20-%20July%2014%2C%202023%20-%20FINAL.pdf

Nikolopoulou, K. (2022a). What is purposive sampling? | definition & examples. Scribbr. https://www.scribbr.com/methodology/purposive-sampling/#:~:text=Purposive%20sampling%20refers%20to%20a

Nikolopoulou, K. (2022b, August 17). What Is Snowball Sampling? | Definition & Examples. Scribbr. https://www.scribbr.com/methodology/snowball-sampling/#:~:text=Also%20known%20as%20chain%20sampling

Parrocha, A. and Pateña, A. (2018, May 17). 300 e-jeepneys to ply Metro Manila in June. Www.pna.gov.ph. https://www.pna.gov.ph/articles/1035634

Payne, R. (2019). Beginning App Development with Flutter. In Apress eBooks. https://doi.org/10.1007/978-1-4842-5181-2

Pelonia, A. (2022, July 28). #TripTips: 5 must-have apps for commuters to navigate Metro Manila [Review of #TripTips: 5 must-have apps for commuters to navigate Metro Manila]. PhilSTAR Life. https://philstarlife.com/living/247681-apps-commuters-navigation-metro?page=3

Philippine Daily Inquirer. (2022, November 13). Commuters deserve better. INQUIRER.net. https://opinion.inquirer.net/158672/commuters-deserve-better

Ramizo, G. (2019, November 1). Comparing conventional public transport to ride-hailing apps: A snapshot of user experiences from Metro Manila. Www.atlantis-Press.com; Atlantis Press. https://doi.org/10.2991/icsgs-18.2019.21

Ronda, R. A. (2023, March 10). DOST pitches 23-seater e-jeepney. Philstar.com. https://www.philstar.com/headlines/2023/03/10/2250549/dost-pitches-23-seater-e-jeepney

rsvp-khalil. (2023, April 25). Transportation in the Philippines: The Ultimate Guide > Abraham Home. Abraham Home. https://www.abraham.travel/transportation-philippines/#:~:text=Buses%20are%20one%20of%20the

Salita, D. C. (2024, January 4). Manila. Encyclopedia Britannica. https://www.britannica.com/place/Manila

Sharma, V. (2024, February 5). Analyzing Software Code — Maintainability Index - Vineet Sharma - Medium. Medium; Medium. https://mvineetsharma.medium.com/analyzing-software-code-maintainability-index-9765896c80f9

Siuhi, S., & Mwakalonge, J. (2016). Opportunities and challenges of smart mobile applications in transportation. Journal of Traffic and Transportation Engineering (English Edition), 3(6), 582–592. https://doi.org/10.1016/j.jtte.2016.11.001

Statista Research Department. (2023, September 7). Philippines: number of PUVs operating by vehicle type 2021. Statista. https://www.statista.com/statistics/1268623/philippines-number-of-puvs-operating-by-vehicle-type/#:~:text=Number%20of%20public%20utility%20vehicles%20in%20operation%20Philippines%202021%2C%20by%20vehicle%20type&text=As%20of%20October%202021%2C%20about

Streefkerk, R. (2022, November 4). Primary vs. Secondary Sources | Difference & Examples. Scribbr. https://www.scribbr.com/working-with-sources/primary-and-secondary-sources/

Sunio, V., Gaspay, S., Guillen, M. D., Mariano, P., & Mora, R. (2019). Analysis of the public transport modernization via system reconfiguration: The ongoing case in the Philippines. Transportation Research Part A: Policy and Practice, 130, 1–19. https://doi.org/10.1016/j.tra.2019.09.004

‌United States Department of Transportation - Federal Highway Administration. (2020, May 13). Smartphone Applications to Influence Travel Choices Practices and Policies: Chapter 4. Transportation Apps and Their Impacts on Traveler Behavior. Ops.fhwa.dot.gov. https://ops.fhwa.dot.gov/publications/fhwahop16023/ch4.htm?fbclid=IwAR0KB5r9hzfVsw3OTOcQIfmJGZ-e9vEVWRjYikgsnuvca-kkH63YuDFOozs#:~:text=In%20transportation%2C%20smartphone%20apps%20have

Wasilewski, K., & Zabierowski, W. (2021). A comparison of Java, Flutter and Kotlin/Native Technologies for Sensor Data-Driven applications. Sensors, 21(10), 3324. https://doi.org/10.3390/s21103324

Wu, S., Ma, E., Wang, J., & Li, D. (2022). Experience with Travel Mobile Apps and Travel Intentions—The Case of University Students in China. Sustainability, 14(19), 12603. https://doi.org/10.3390/su141912603

Wulandari, W., Nofiyani, N., & Hasugian, H. (2023). USER ACCEPTANCE TESTING (UAT) PADA ELECTRONIC DATA PREPROCESSING GUNA MENGETAHUI KUALITAS SISTEM. Jurnal Mahasiswa Ilmu Komputer, 4(1), 20–27. https://doi.org/10.24127/ilmukomputer.v4i1.3383

**APPENDIX A**

**Researcher’s Profile**









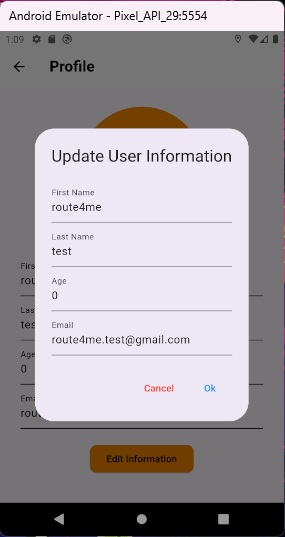
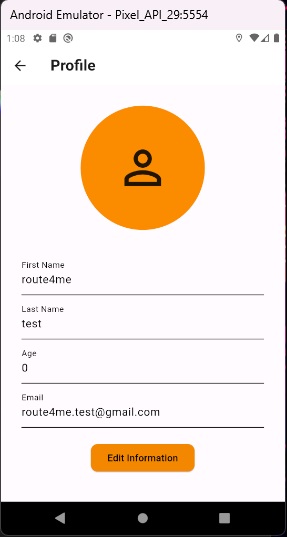
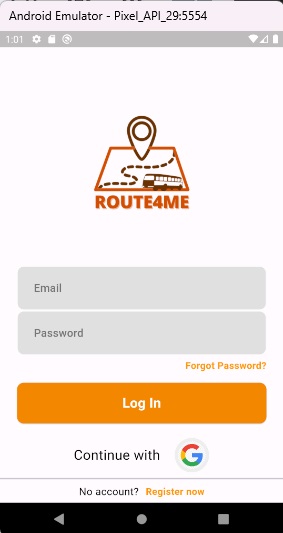


**APPENDIX B**

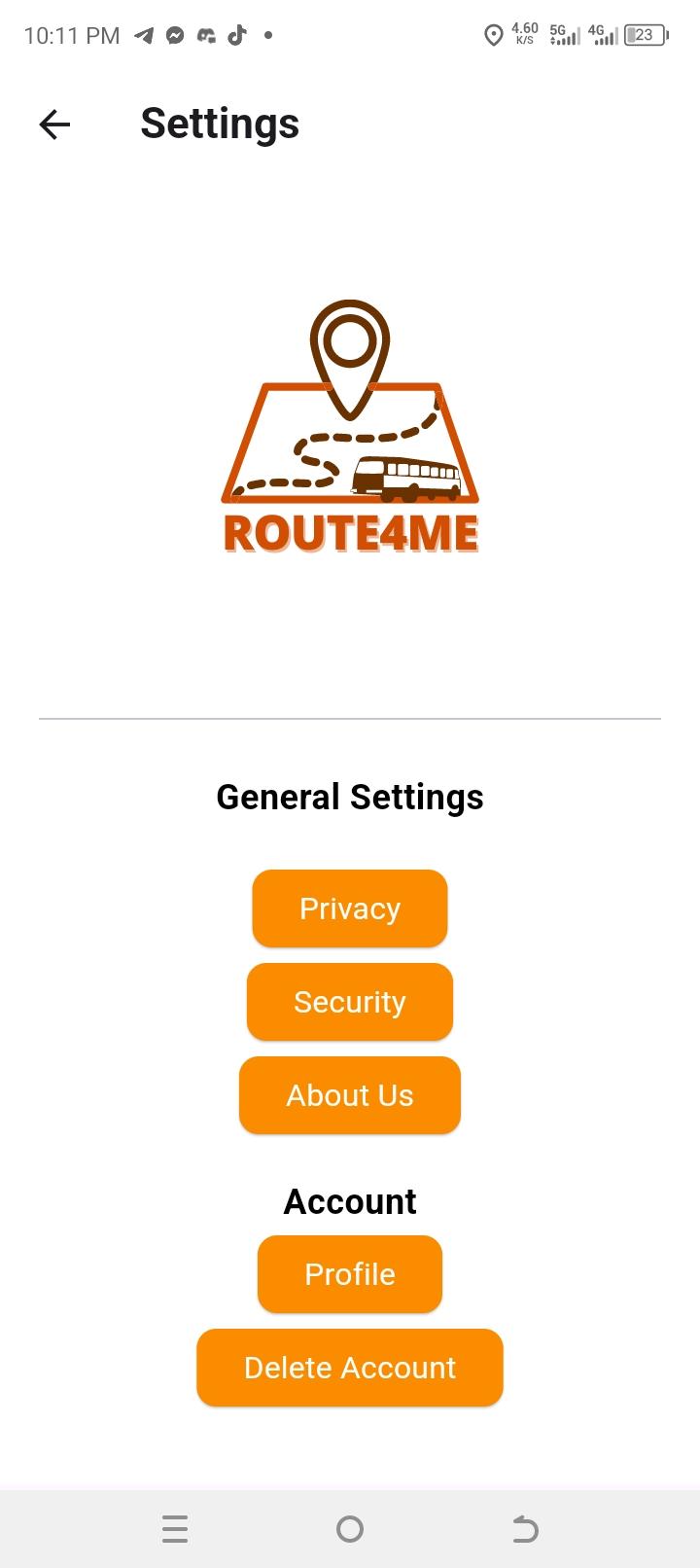
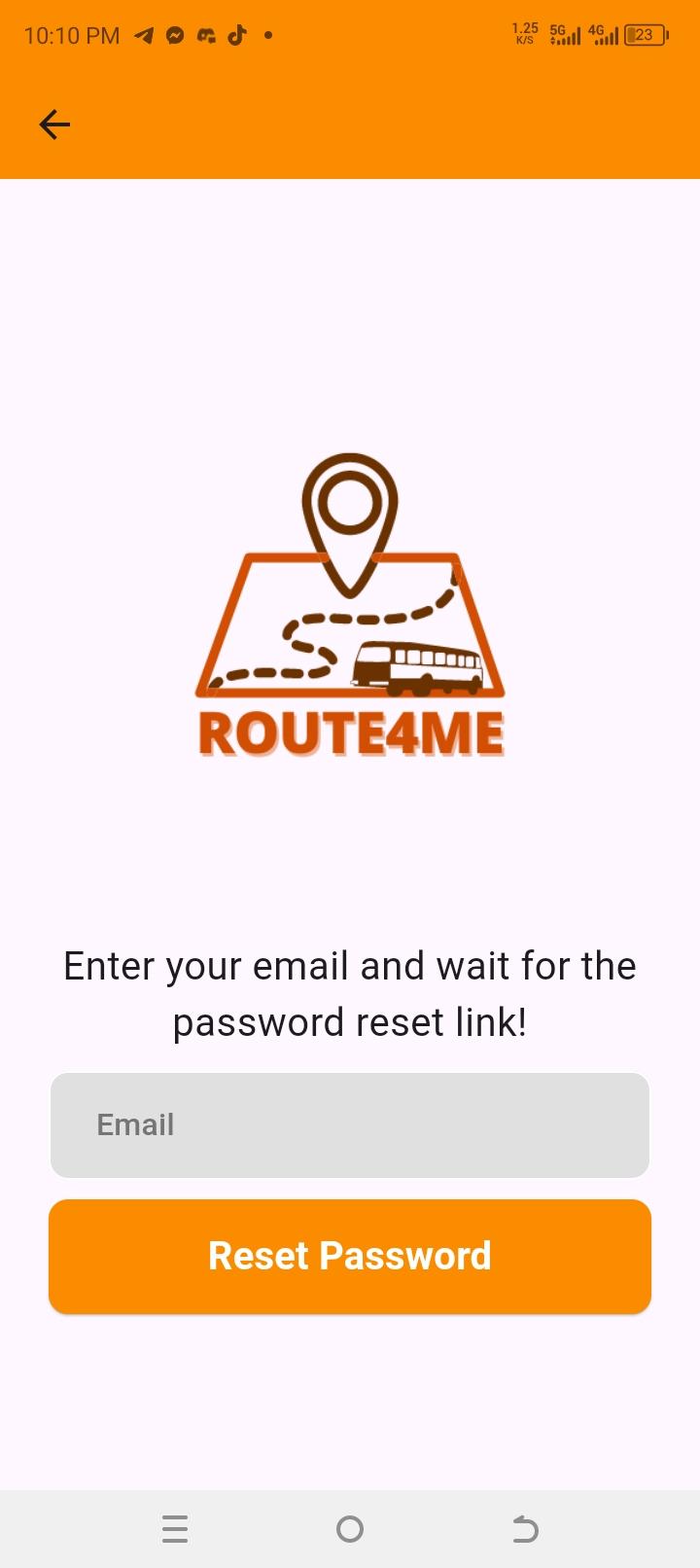
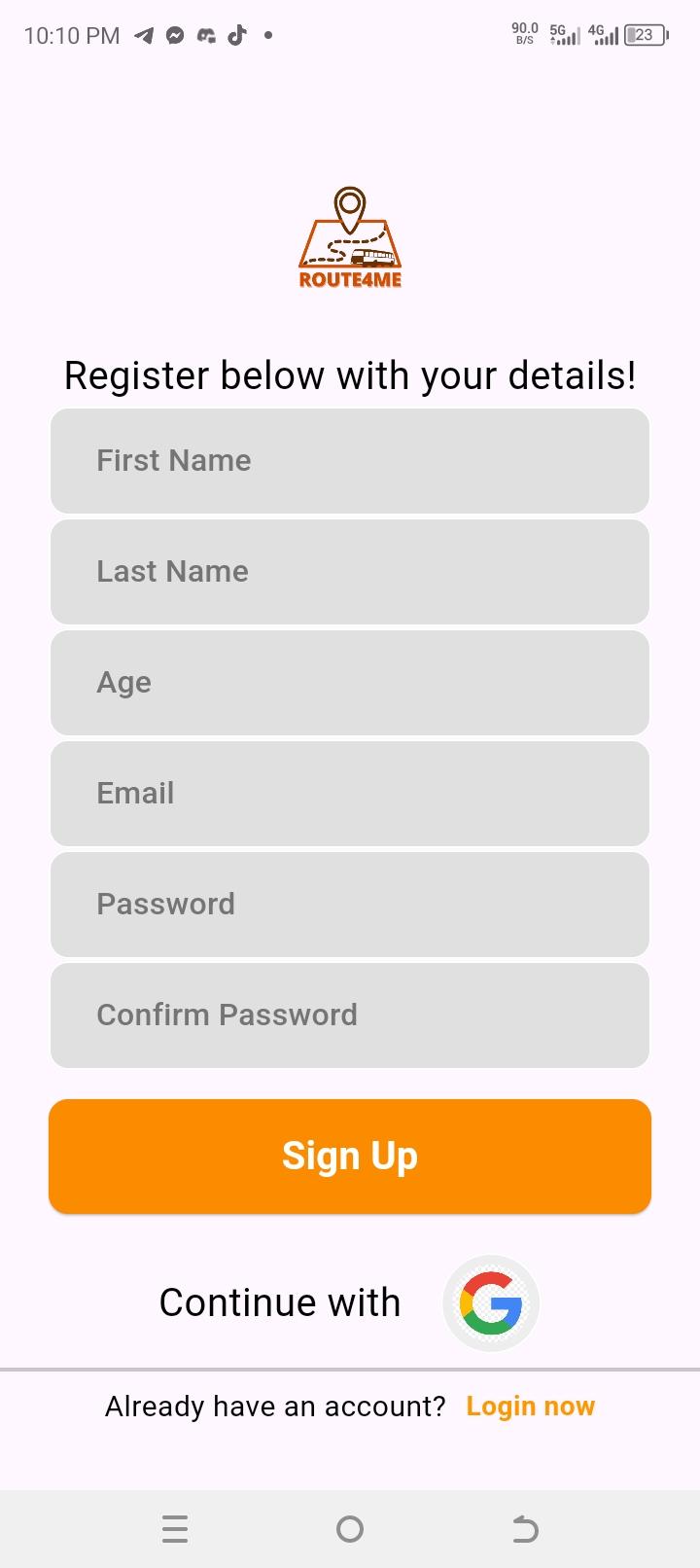
**Design of Route4Me Application**

**Application System Design**

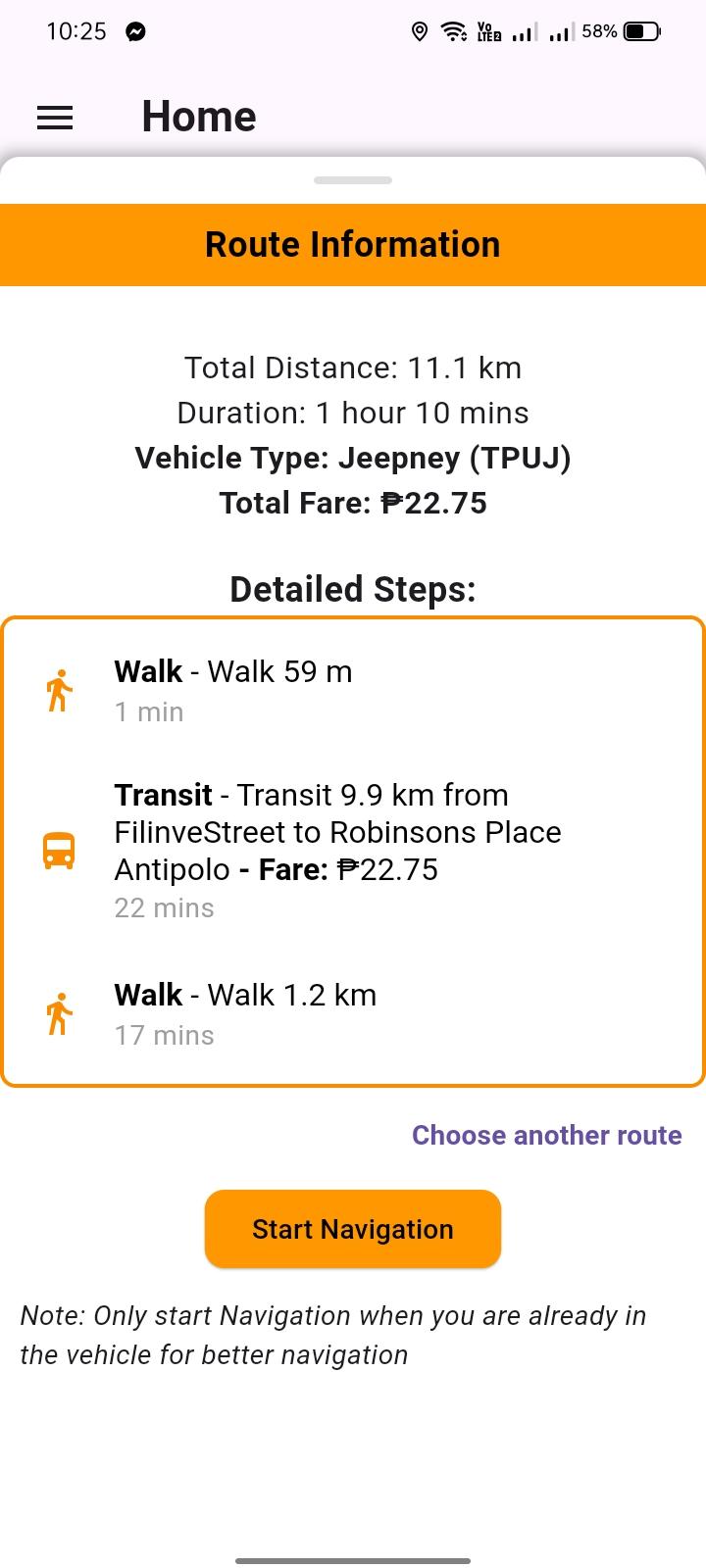
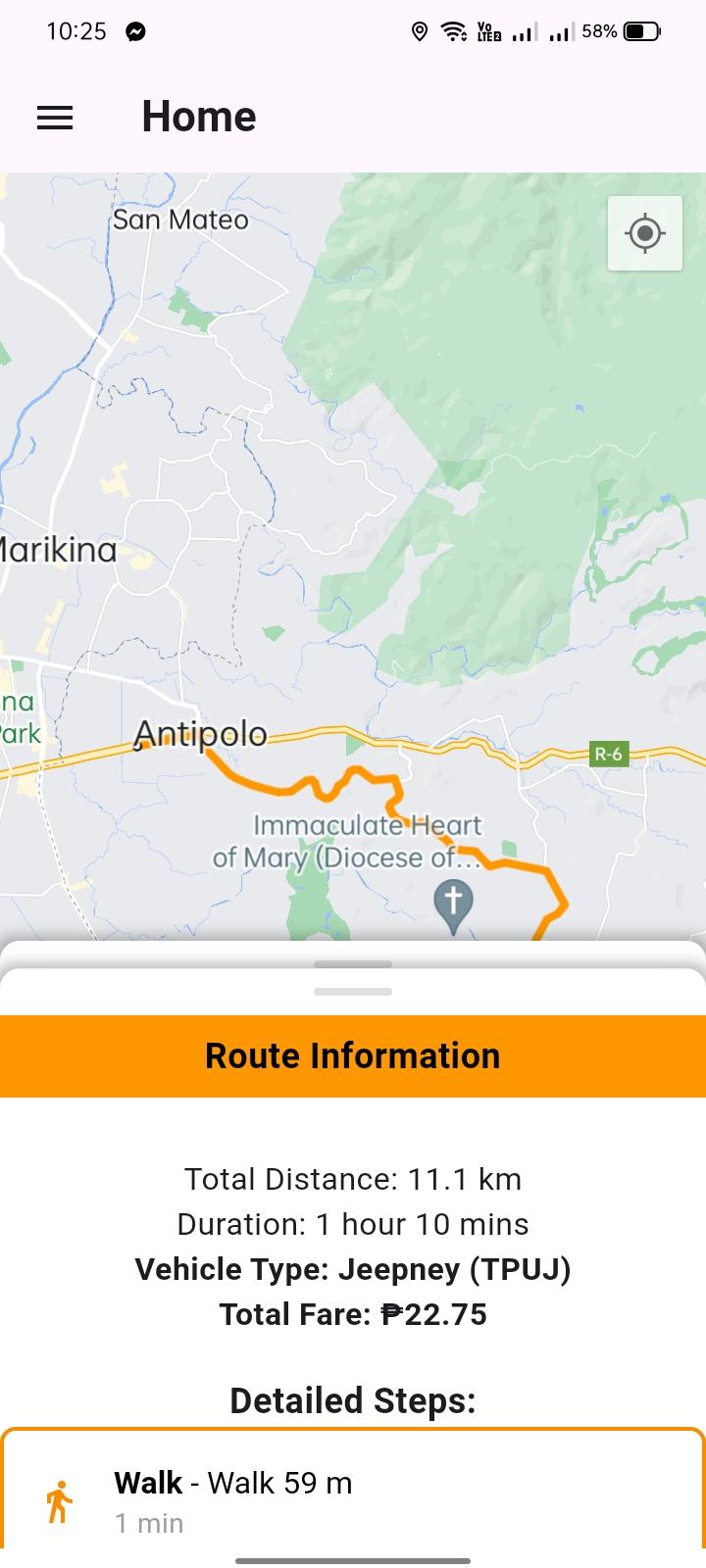
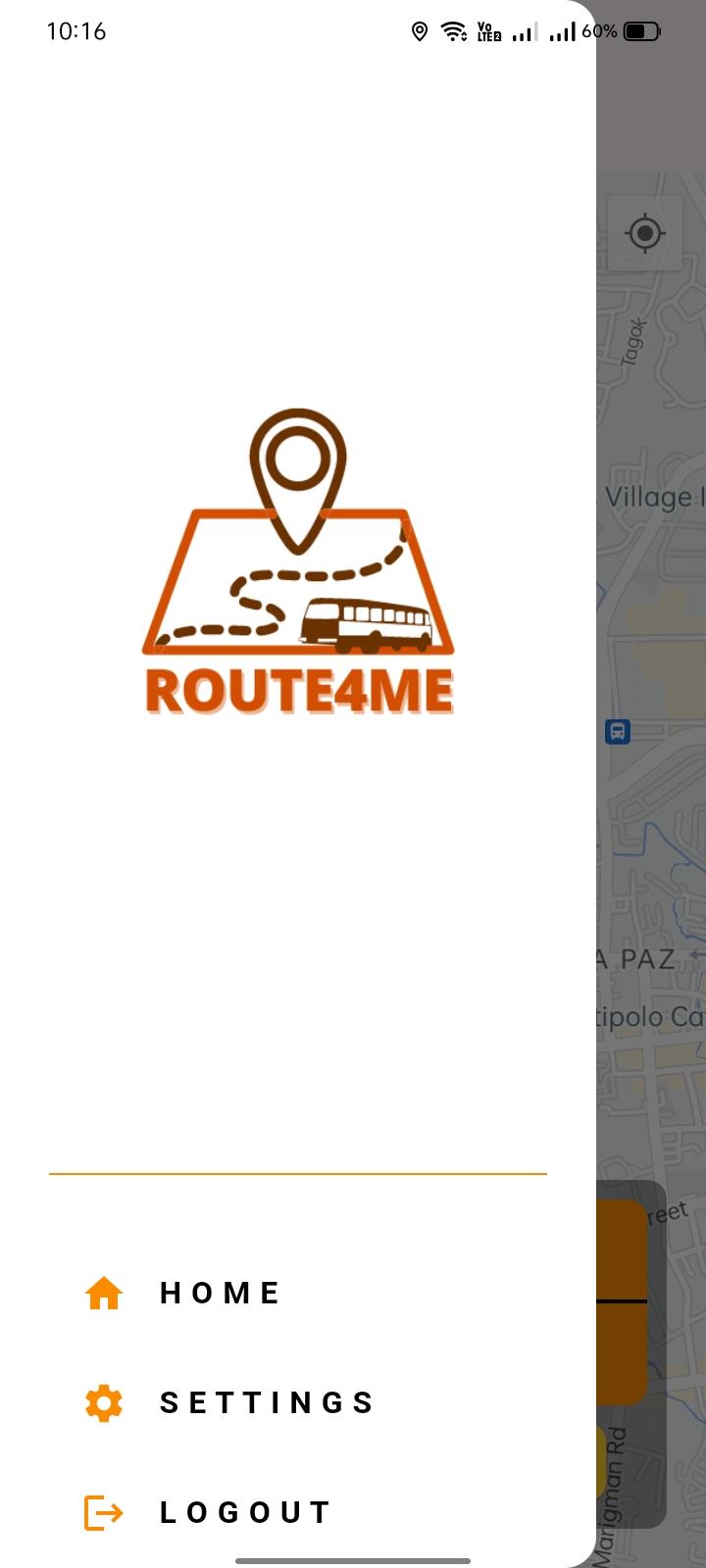
**Log-In and Profile Page**

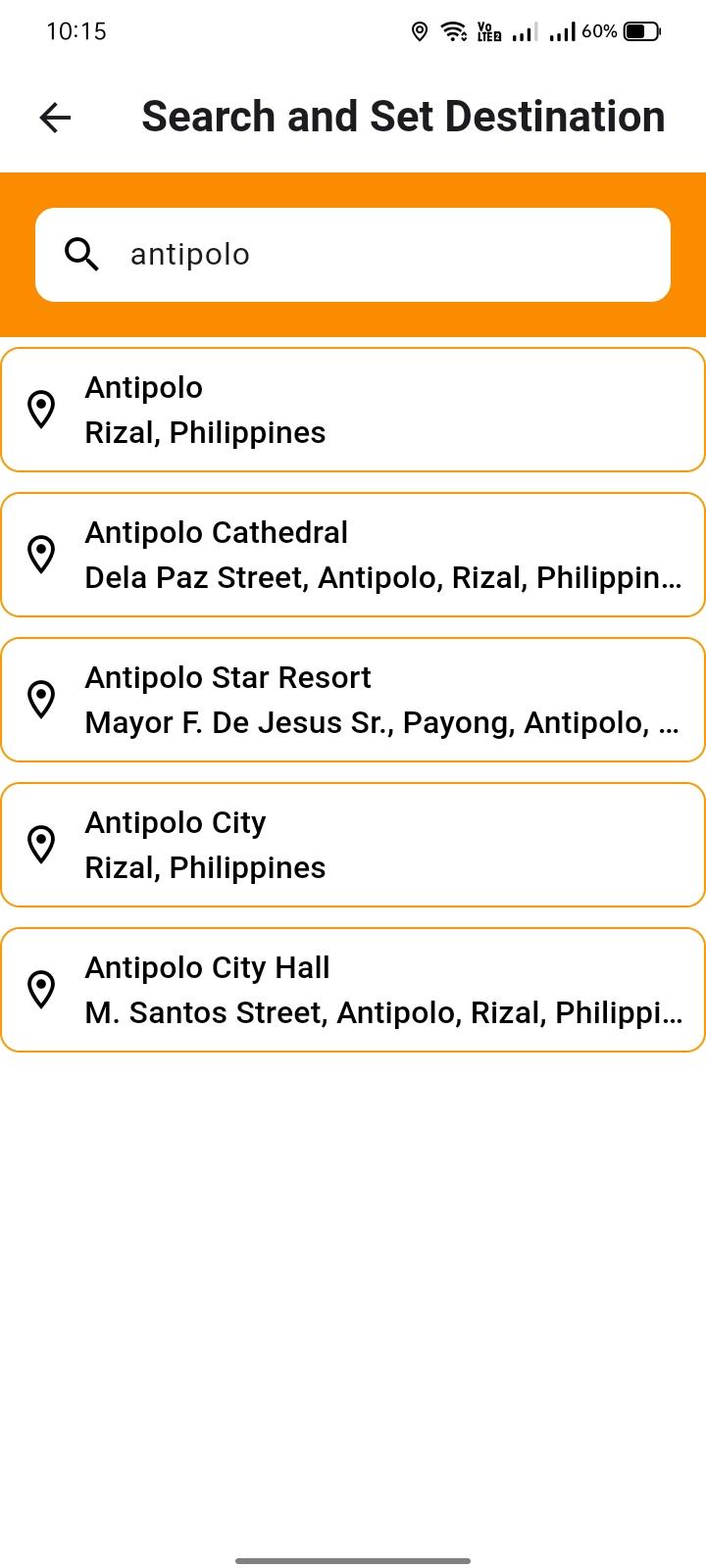
****

**Register, Forgot Password, and Settings Page**

****

**Home Page**

****

****

**APPENDIX C**

**Preliminary Survey**

**Name** (Optional)**:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **Sample Group** (Driver/Commuters)**:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **Technological availability, accessibility, and capability of drivers/commuters**

**Directions** (Direksyon)**:**

Please choose the best option for you by ticking the corresponding box.

1. What is the operating system (OS) of your mobile device?

▢ Android

▢ iOS

1. For Android users, what is the version of the operating system (OS) of your mobile device you commonly use while commuting?

|  |  |
| --- | --- |
| ▢ Android 4.4 (KitKat)  ▢ Android 5.0 (Lollipop)  ▢ Android 5.1 (Lollipop)  ▢ Android 6.0 (Marshmallow)  ▢ Android 7.0 (Nougat)  ▢ Android 7.1 (Nougat)  ▢ Android 8.0 (Oreo)  ▢ Android 8.1 (Oreo) | ▢ Android 9.0 (Pie)  ▢ Android 10.0 (Q)  ▢ Android 11.0 (R)  ▢ Android 12.0 (S)  ▢ Android 13.0 (T)  ▢ I have no idea  ▢ Not Android user |

1. For iOS users, what is the version of the operating system (OS) of your mobile device you commonly use while commuting?

|  |  |
| --- | --- |
| ▢ iOS 1  ▢ iOS 2  ▢ iOS 3  ▢ iOS 4  ▢ iOS 5  ▢ IOS 6  ▢ iOS 7  ▢ iOS 8  ▢ iOS 9  ▢ iOS 10 | ▢ iOS 11  ▢ iOS 12  ▢ IOS 13  ▢ IOS 14  ▢ iOS 15  ▢ IOS 16  ▢ iOS 17  ▢ I have no idea  ▢ Not iOS user |

1. How often do you have access to the internet data during your daily commute?

|  |  |
| --- | --- |
| ▢ Always  ▢ Most of the time  ▢ Occasionally | ▢ Rarely  ▢ Never |

1. On average, how many hours a day do you access the internet data?

|  |  |
| --- | --- |
| ▢ Less than 1 hour  ▢ 1-2 hours  ▢ 2-3 hours | ▢ 3-4 hours  ▢ More than 4 hours |

1. How much is your average mobile data/cellular data per week?

|  |  |
| --- | --- |
| ▢ 2-25 Php  ▢ 26-50 Php  ▢ 51-75 Php | ▢ 76-100 Php  ▢ More than 100 Php |

1. For passengers, what time do you usually commute to work or school?

|  |  |
| --- | --- |
| ▢ Earlier than 5 AM  ▢ 5 AM to 7 AM  ▢ 7 AM to 9 AM  ▢ 9 AM to 11 AM  ▢ 11 AM to 1 PM | ▢ 1 PM to 3 PM  ▢ 3 PM to 5 PM  ▢ 5 PM to 7 PM  ▢ Later than 7 PM |

1. For passengers, what time do you usually go home from work or school?

|  |  |
| --- | --- |
| ▢ Earlier than 5 AM  ▢ 5 AM to 7 AM  ▢ 7 AM to 9 AM  ▢ 9 AM to 11 AM  ▢ 11 AM to 1 PM | ▢ 1 PM to 3 PM  ▢ 3 PM to 5 PM  ▢ 5 PM to 7 PM  ▢ Later than 7 PM |

1. Which mobility applications have you tried using when commuting? Check all that apply.

|  |  |
| --- | --- |
| ▢ Google Maps  ▢ Waze  ▢ Moovit  ▢ Sakay.ph | ▢ Joyride  ▢ Angkas  ▢ Grab  ▢ None |

1. Which mobility applications help you in your commuting experience? Check all that apply.

|  |  |
| --- | --- |
| ▢ Google Maps  ▢ Waze  ▢ Moovit  ▢ Sakay.ph | ▢ Joyride  ▢ Angkas  ▢ Grab  ▢ None |

1. Which mobility applications make your commuting experience better, safer, and more secure? Check all that apply.

|  |  |
| --- | --- |
| ▢ Google Maps  ▢ Waze  ▢ Moovit  ▢ Sakay.ph | ▢ Joyride  ▢ Angkas  ▢ Grab  ▢ None |

The researchers will develop a real-time GPS tracking application for PUV passengers in Rizal Province, offering route recommendations and estimated arrival times, along with fare details for various routes. Will you be willing to participate in the implementation of the application?

▢ Yes

▢ No

**APPENDIX D**

**Route4Me Mobile Application Documentation (PUVs Drivers)**

**Air-Conditioned Bus**





**E-Jeep**





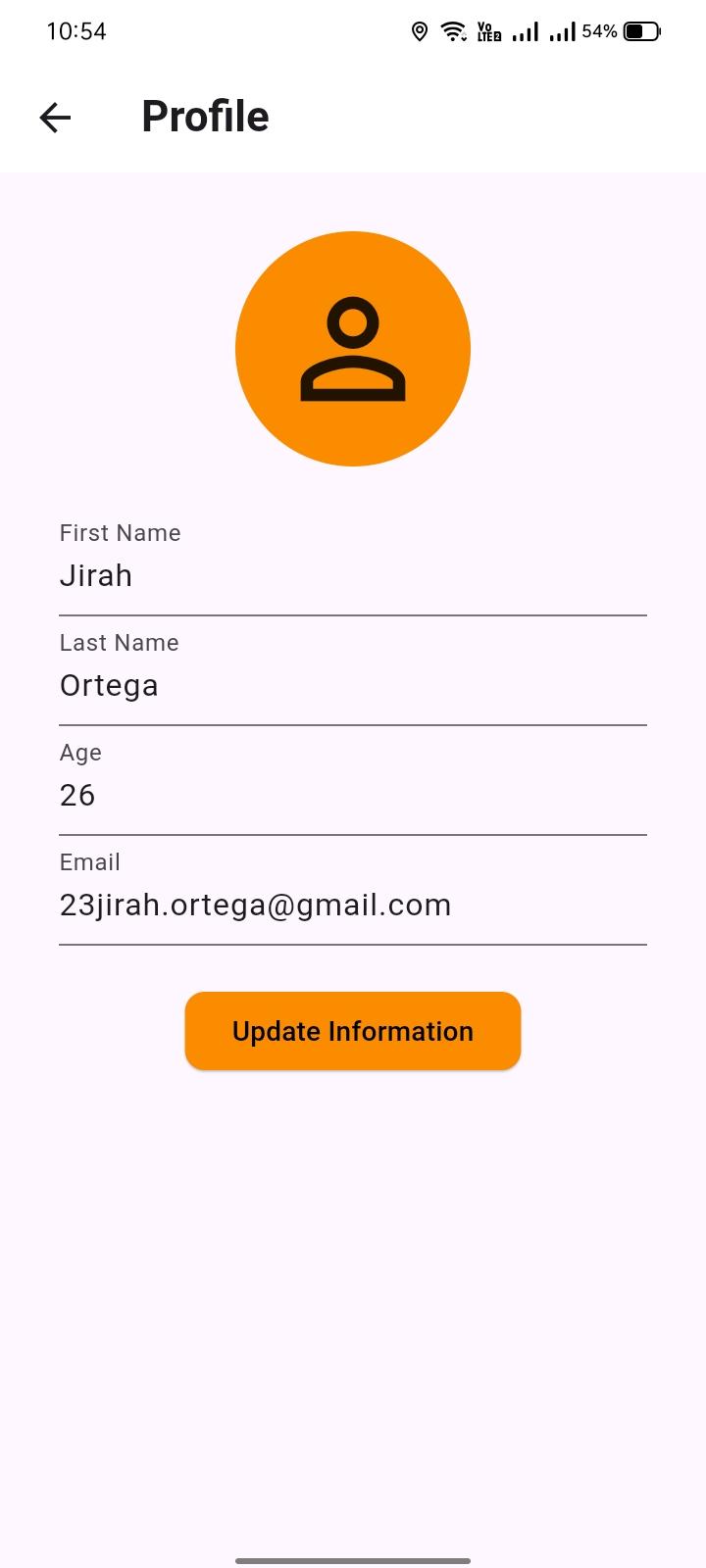
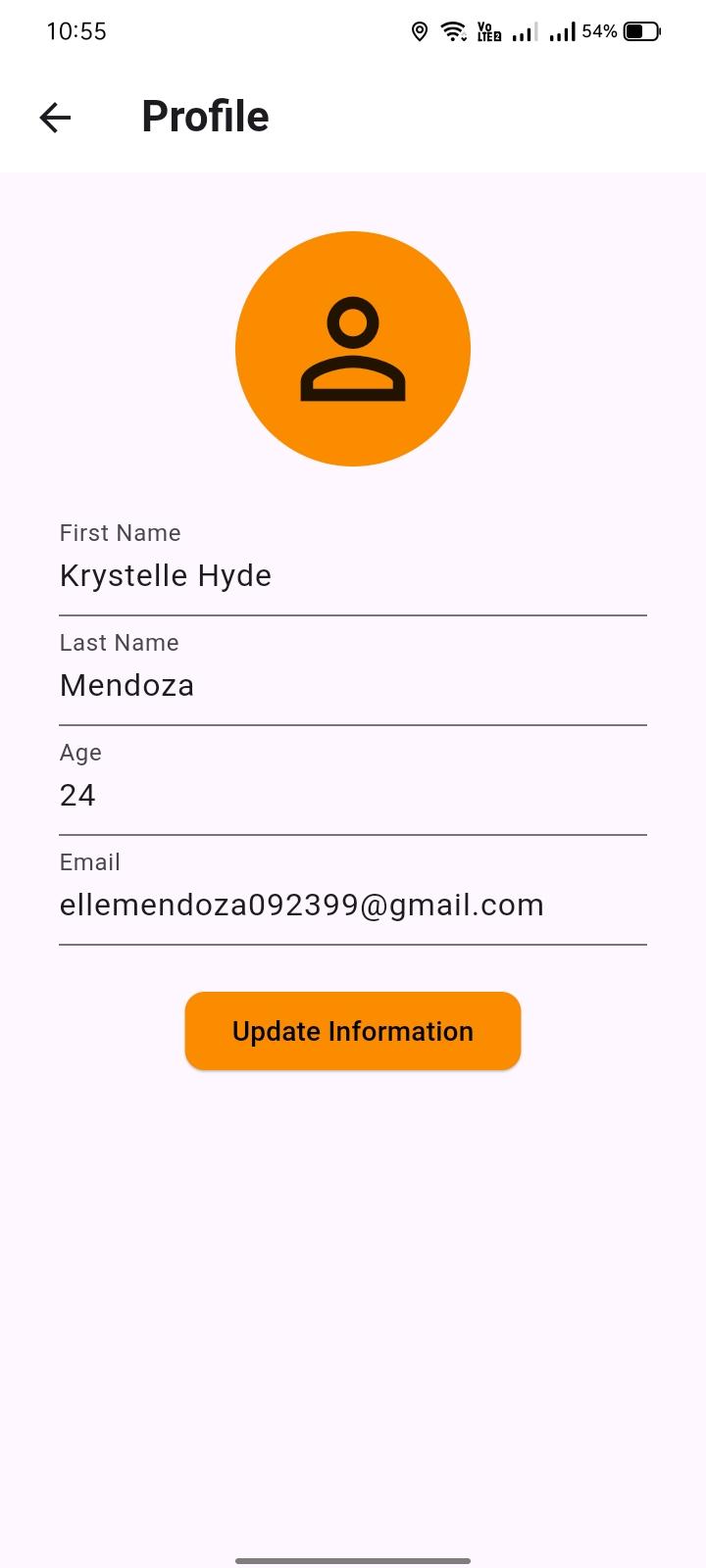
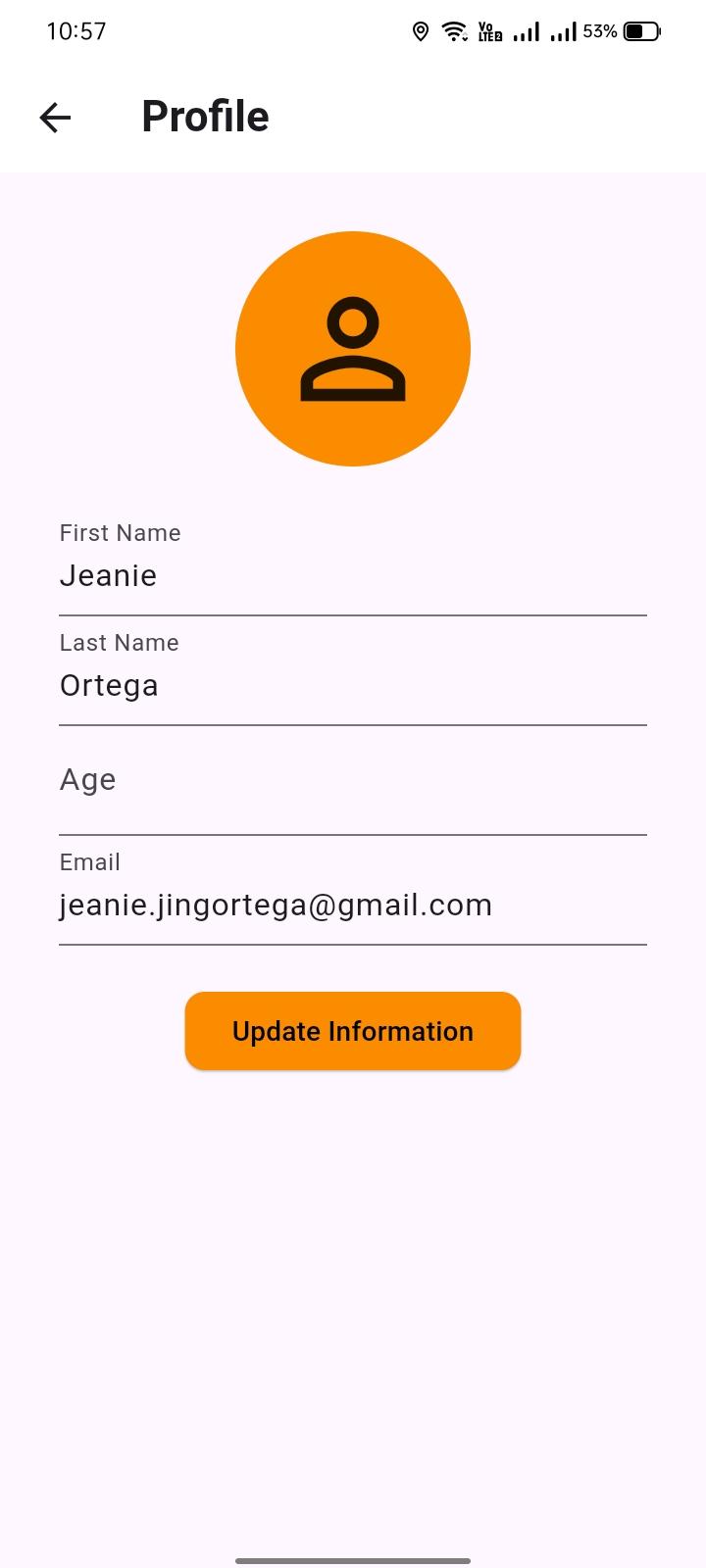
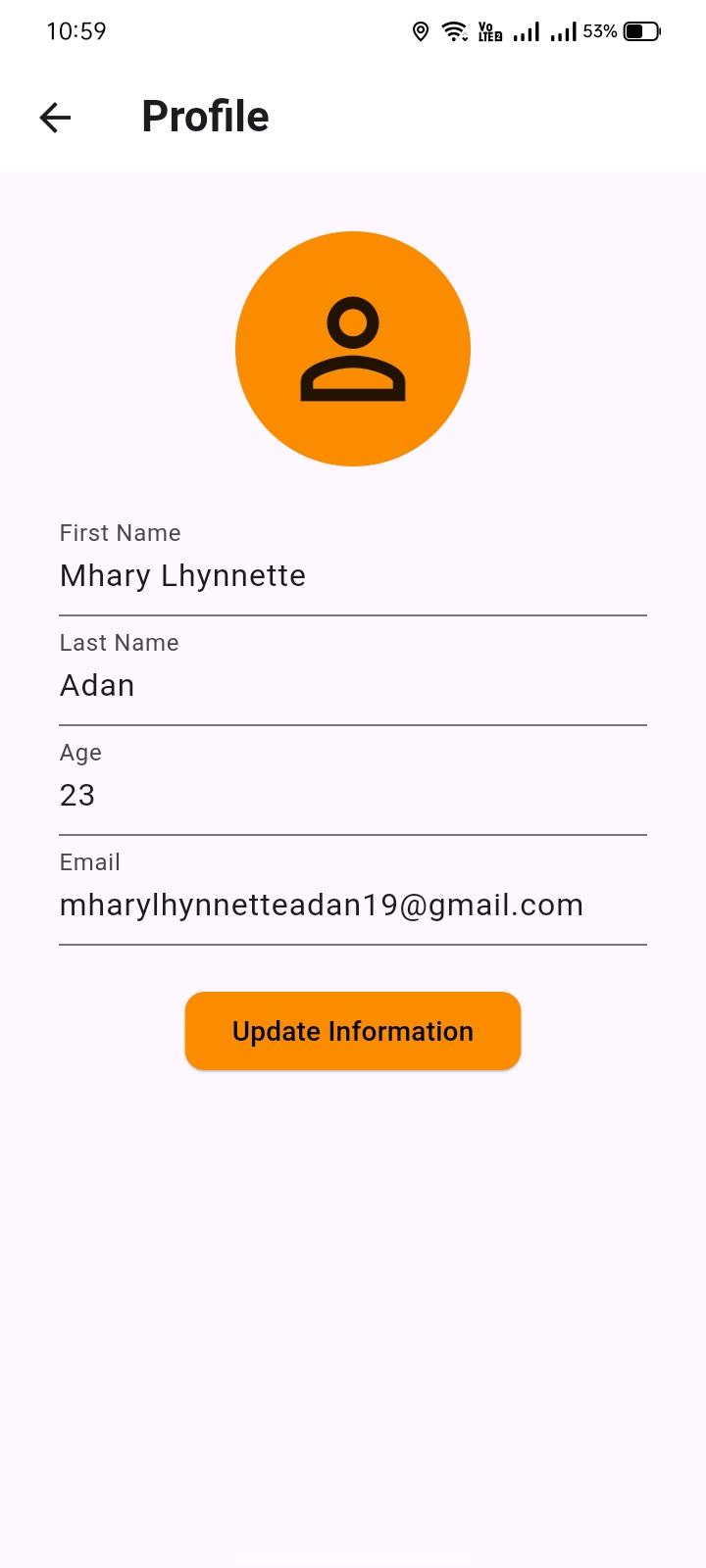
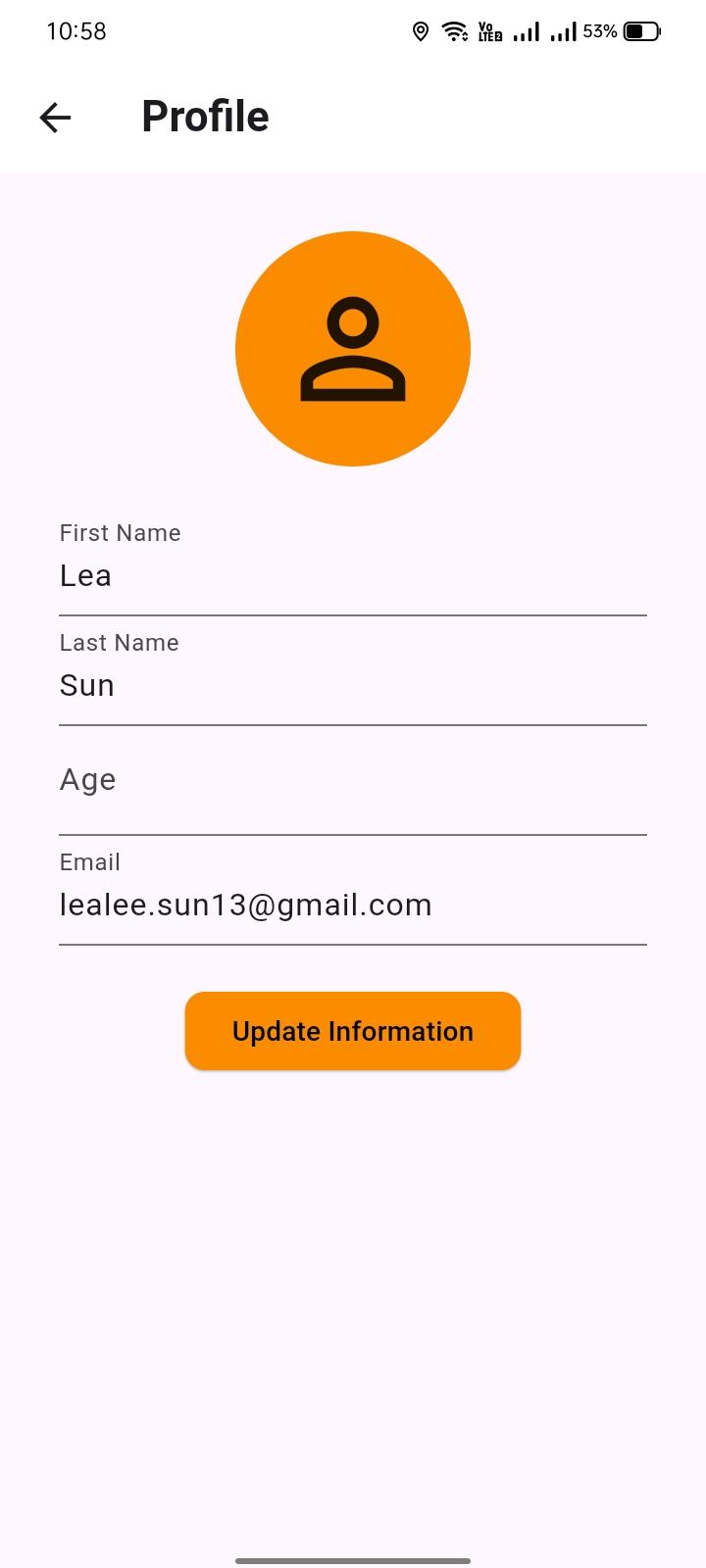
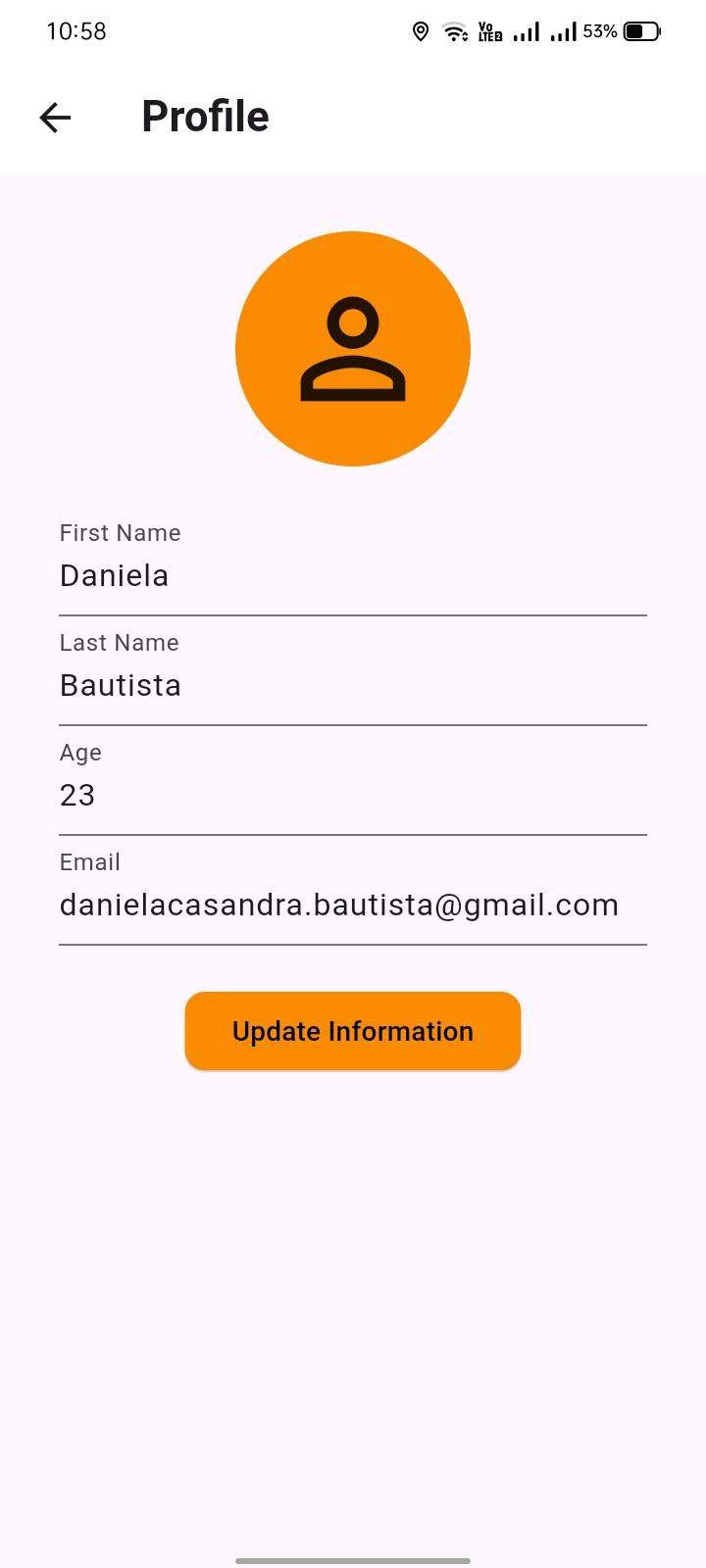
**Traditional Jeepney**

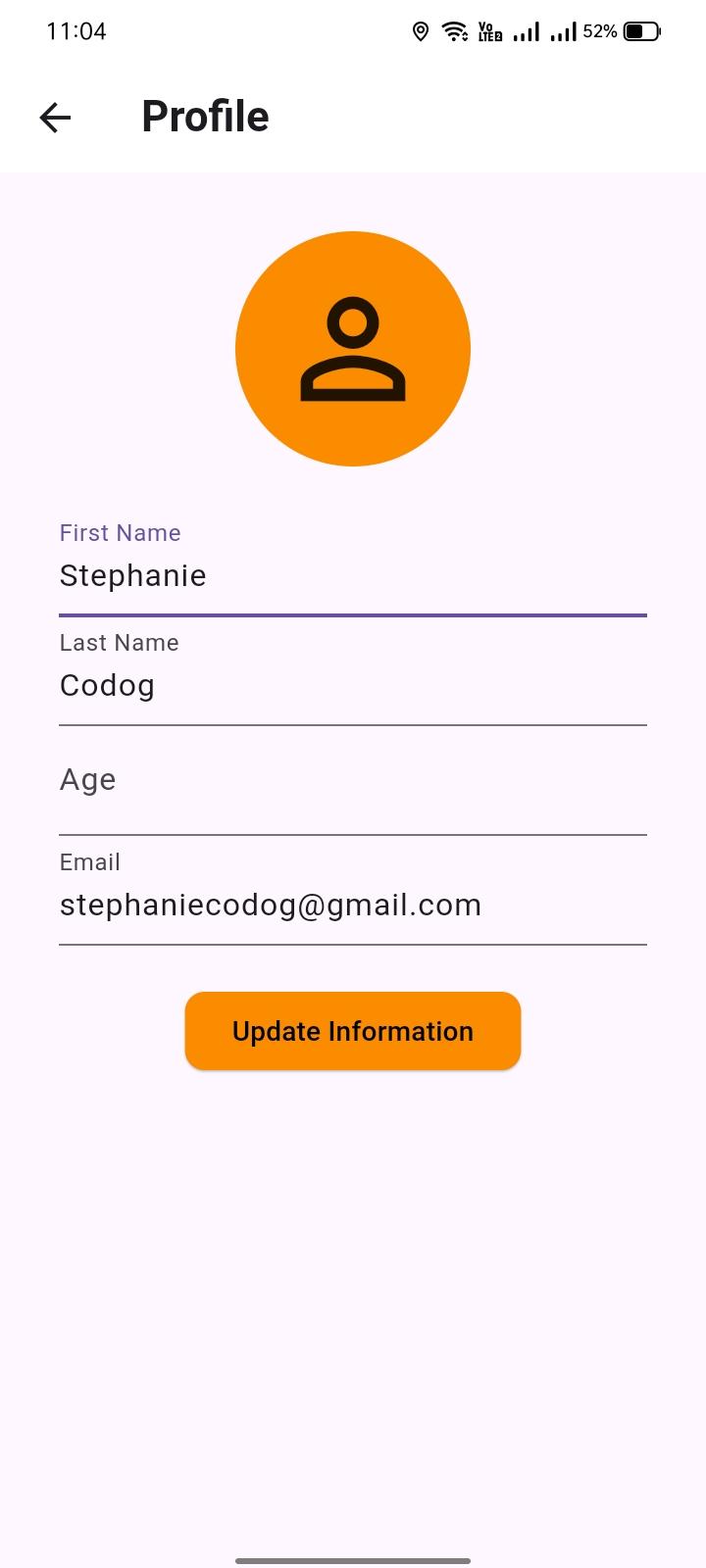
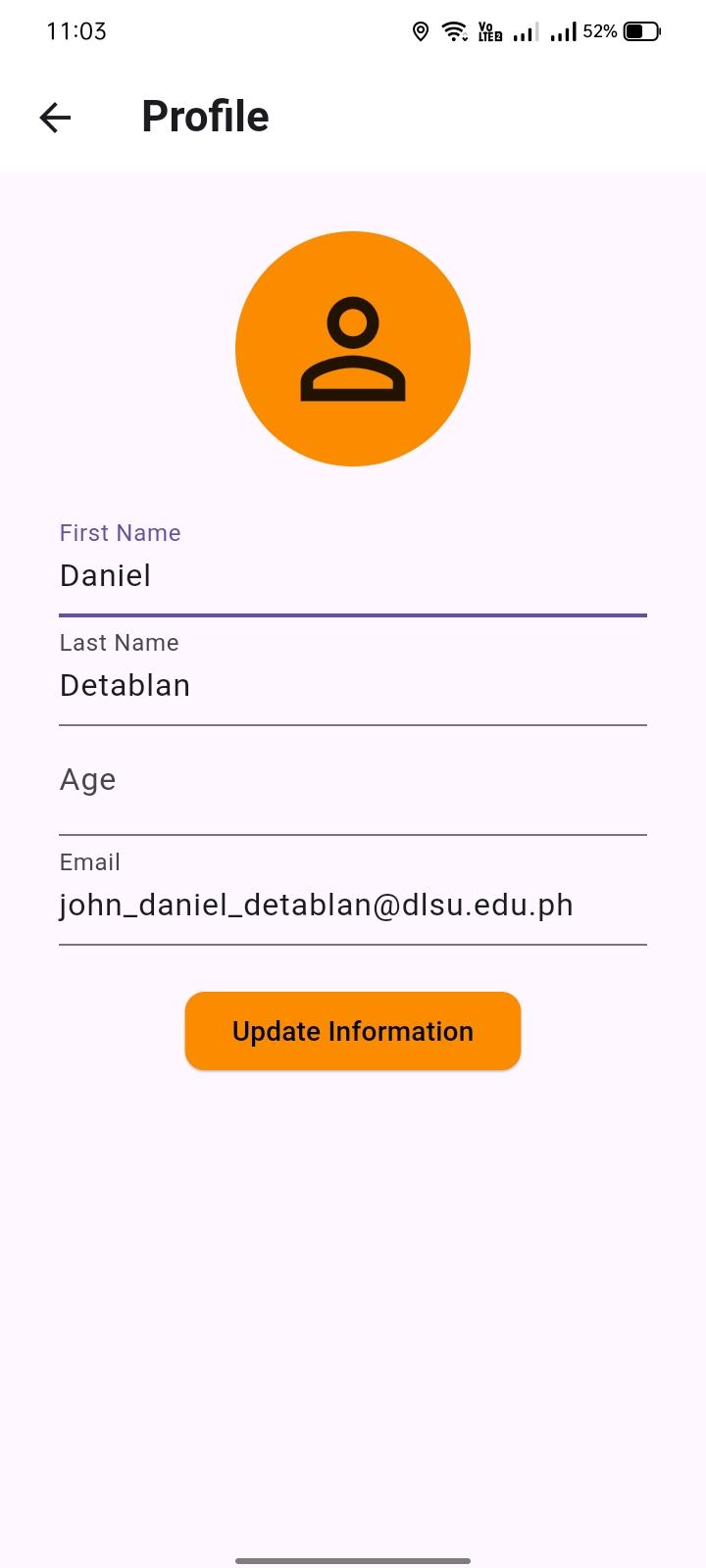
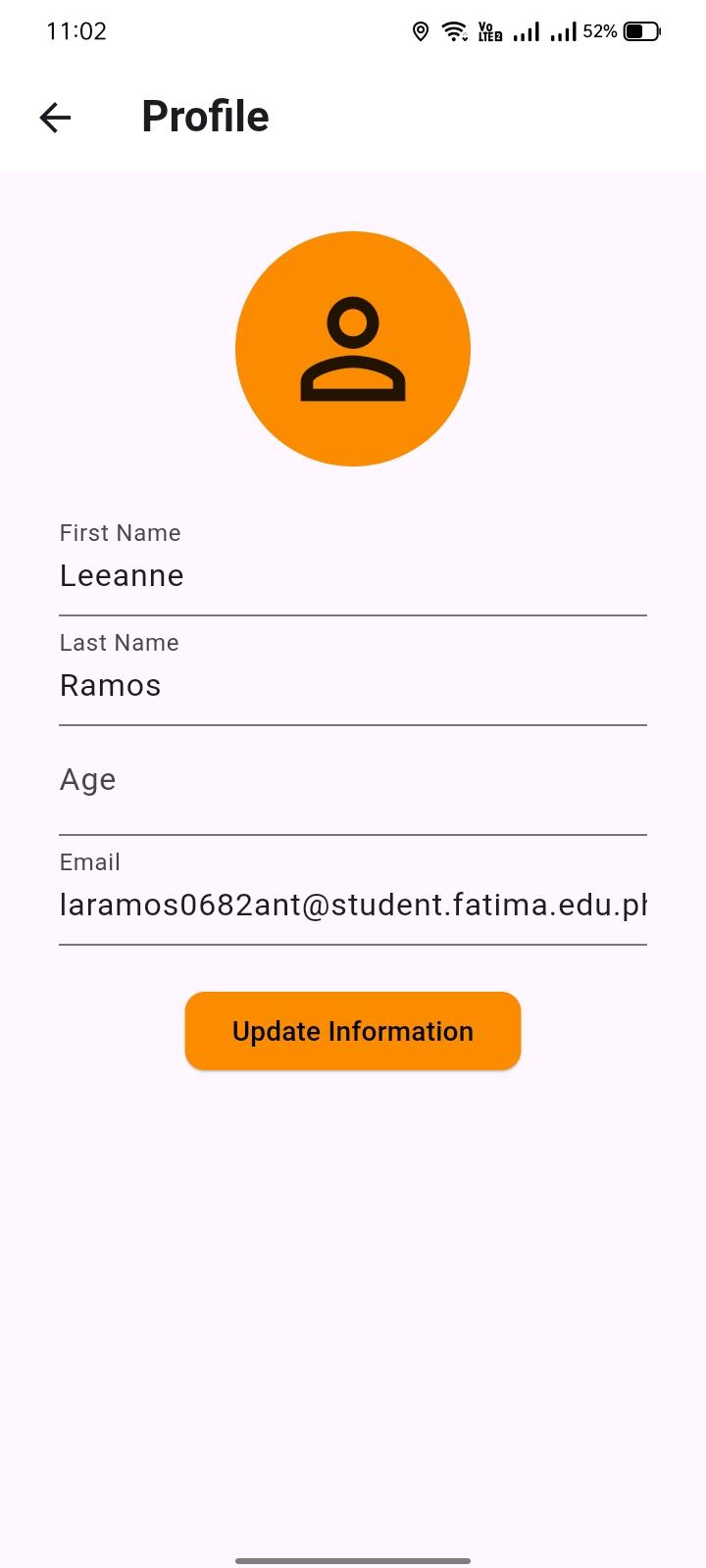
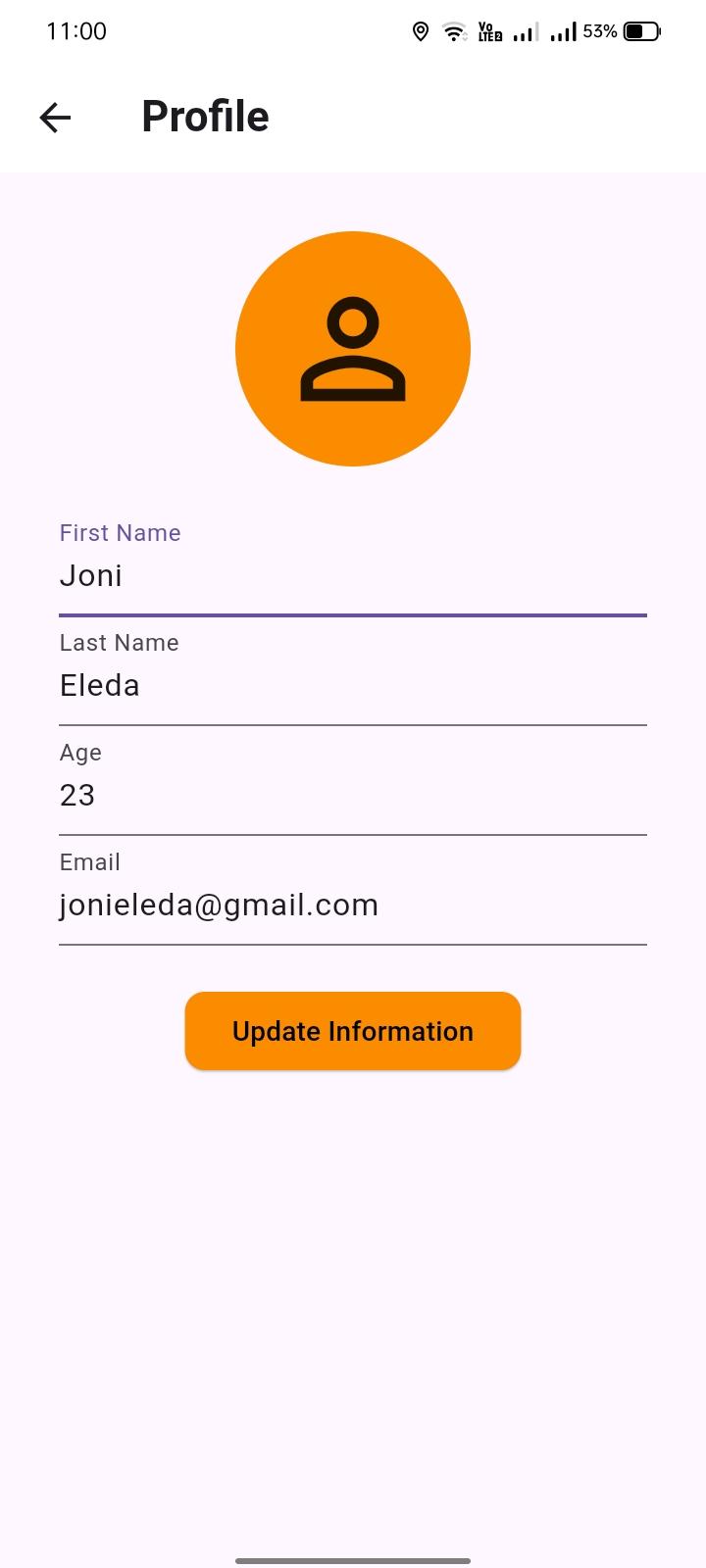
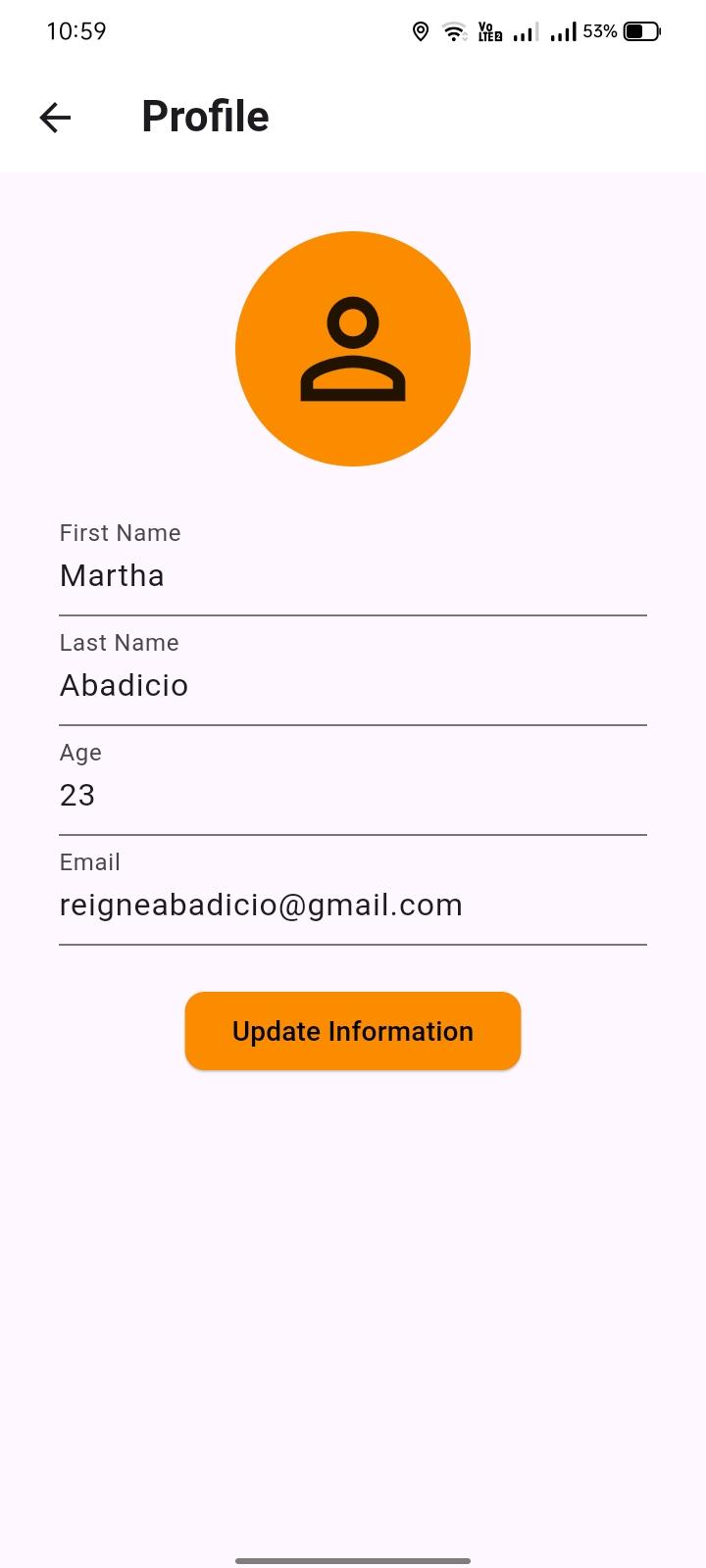


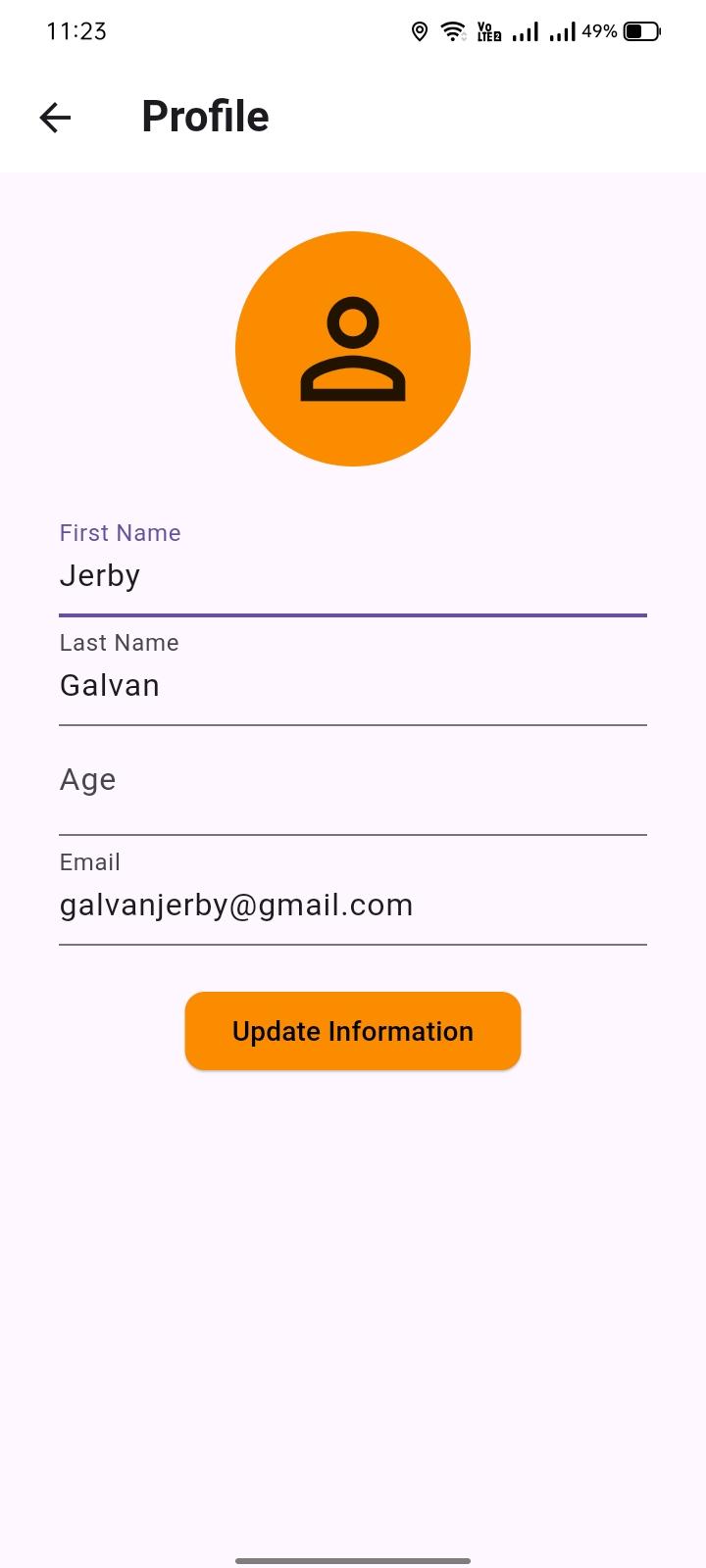
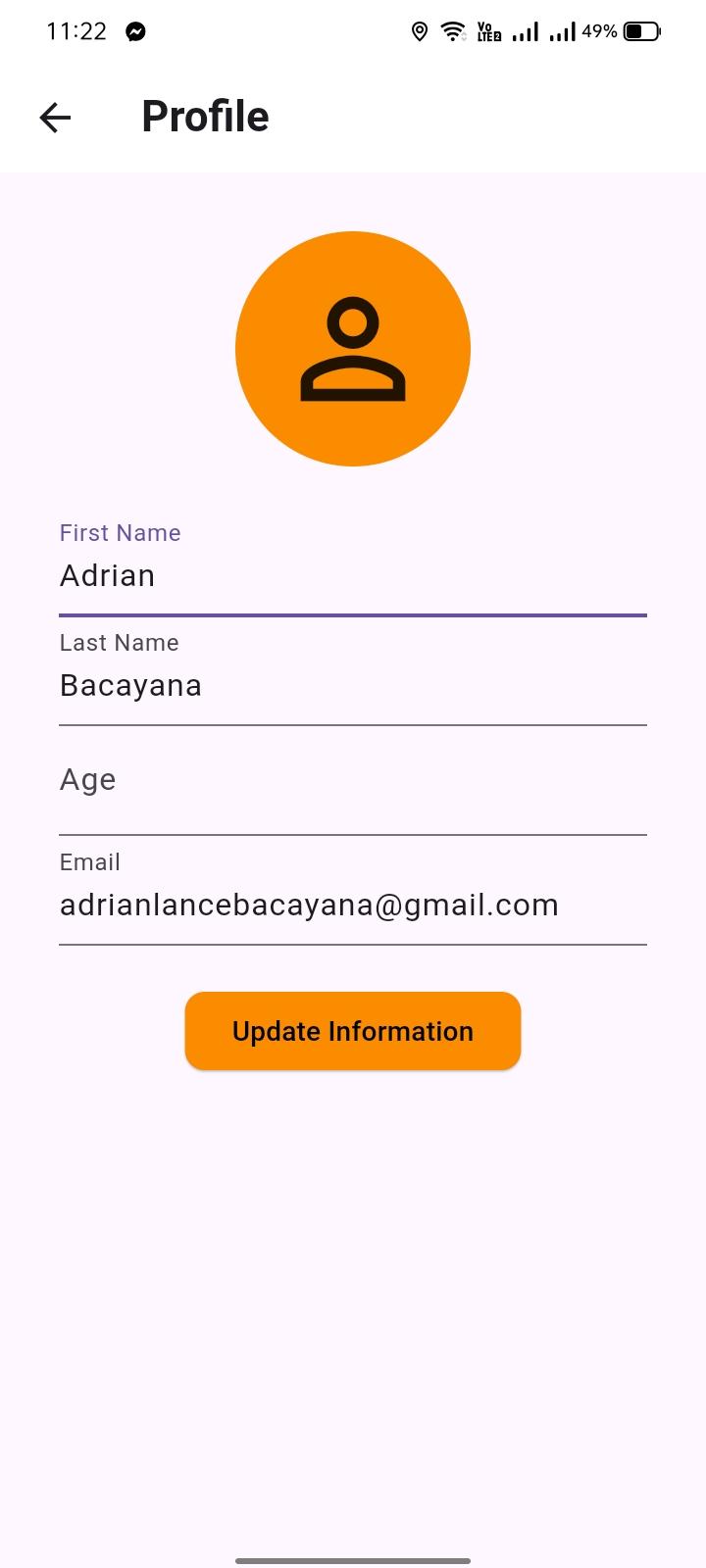
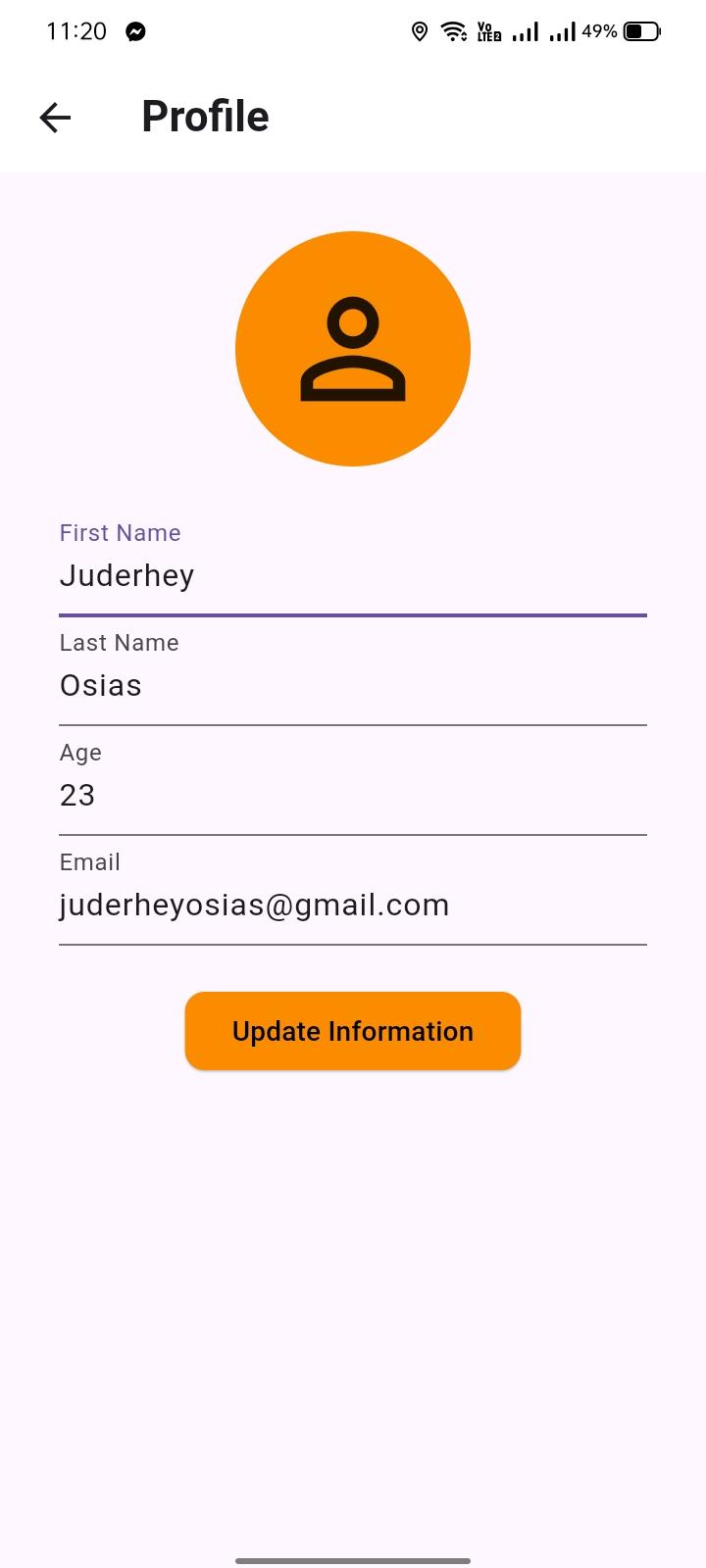
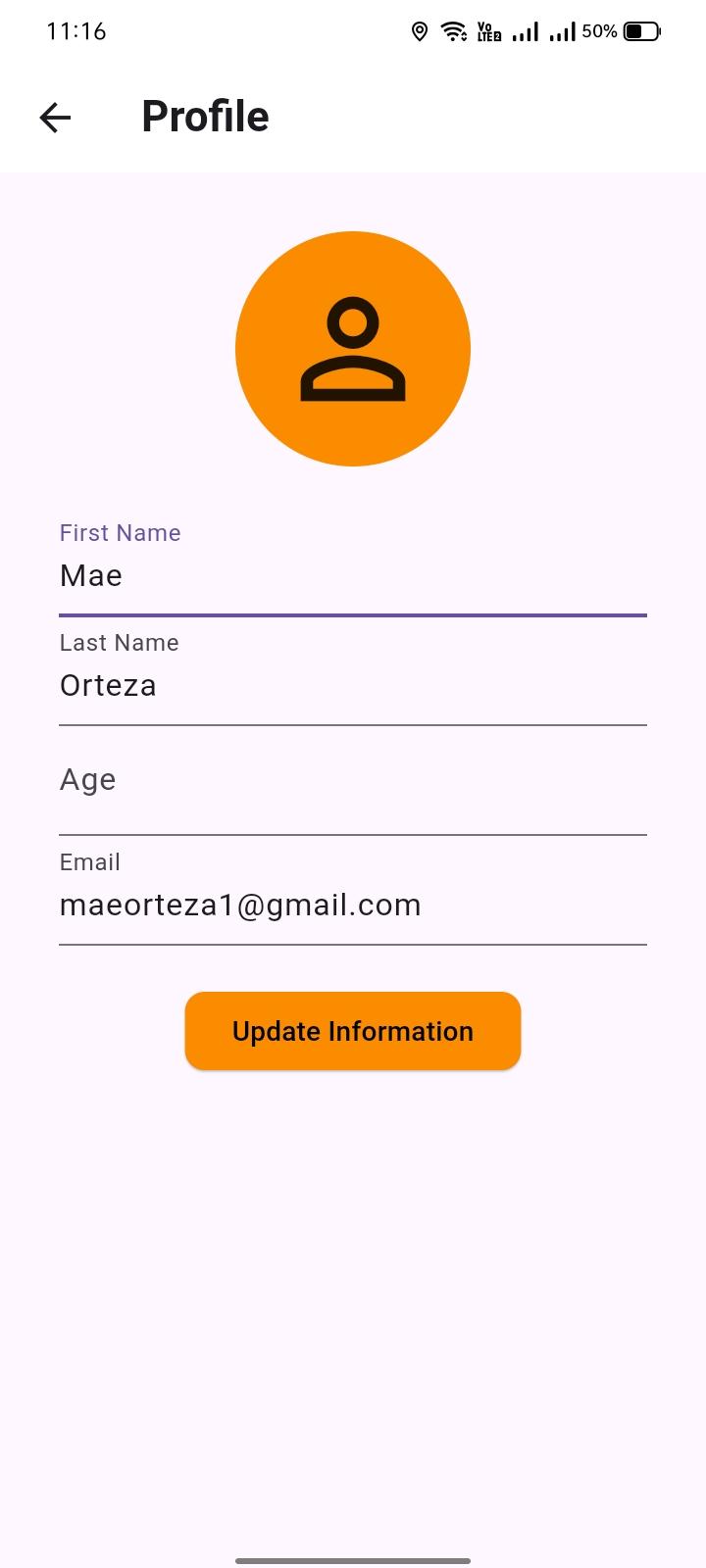
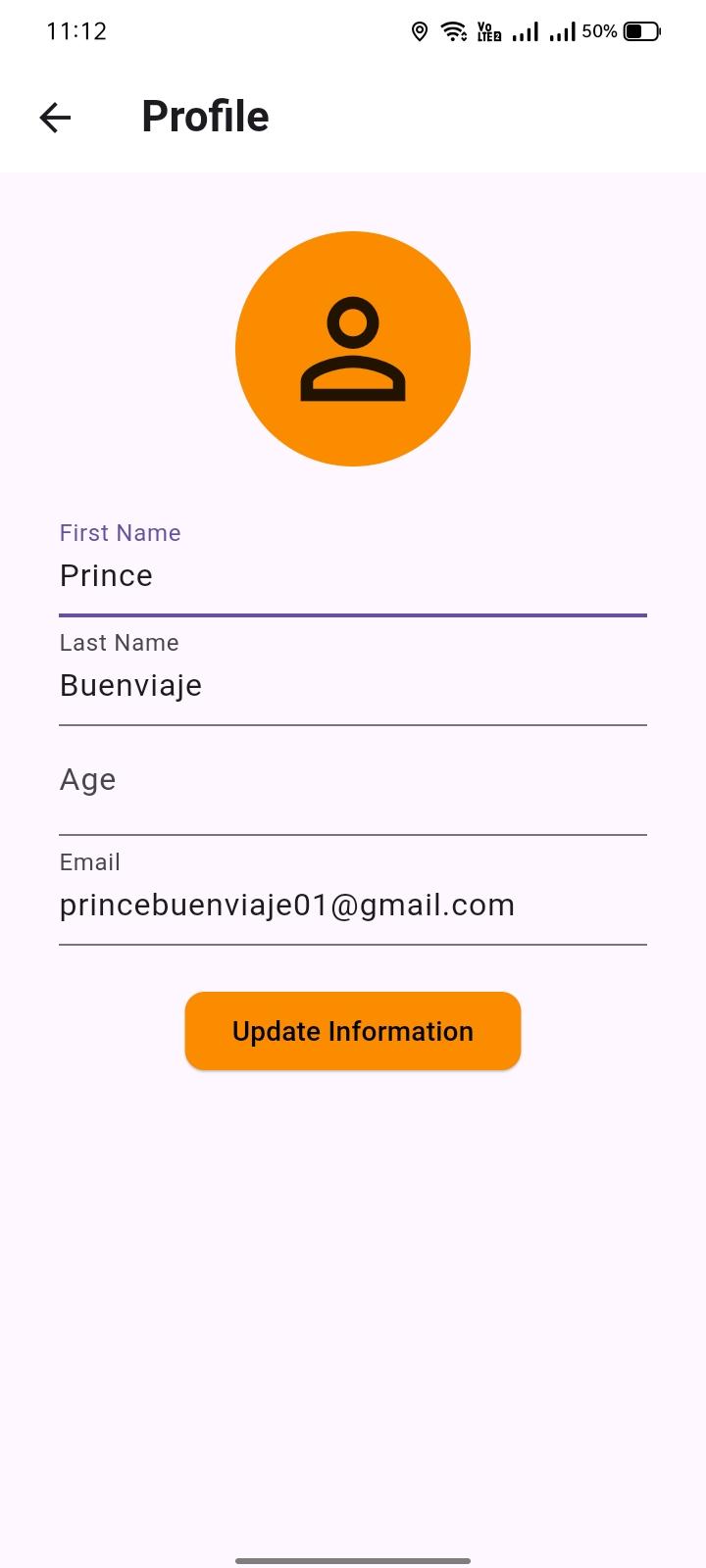
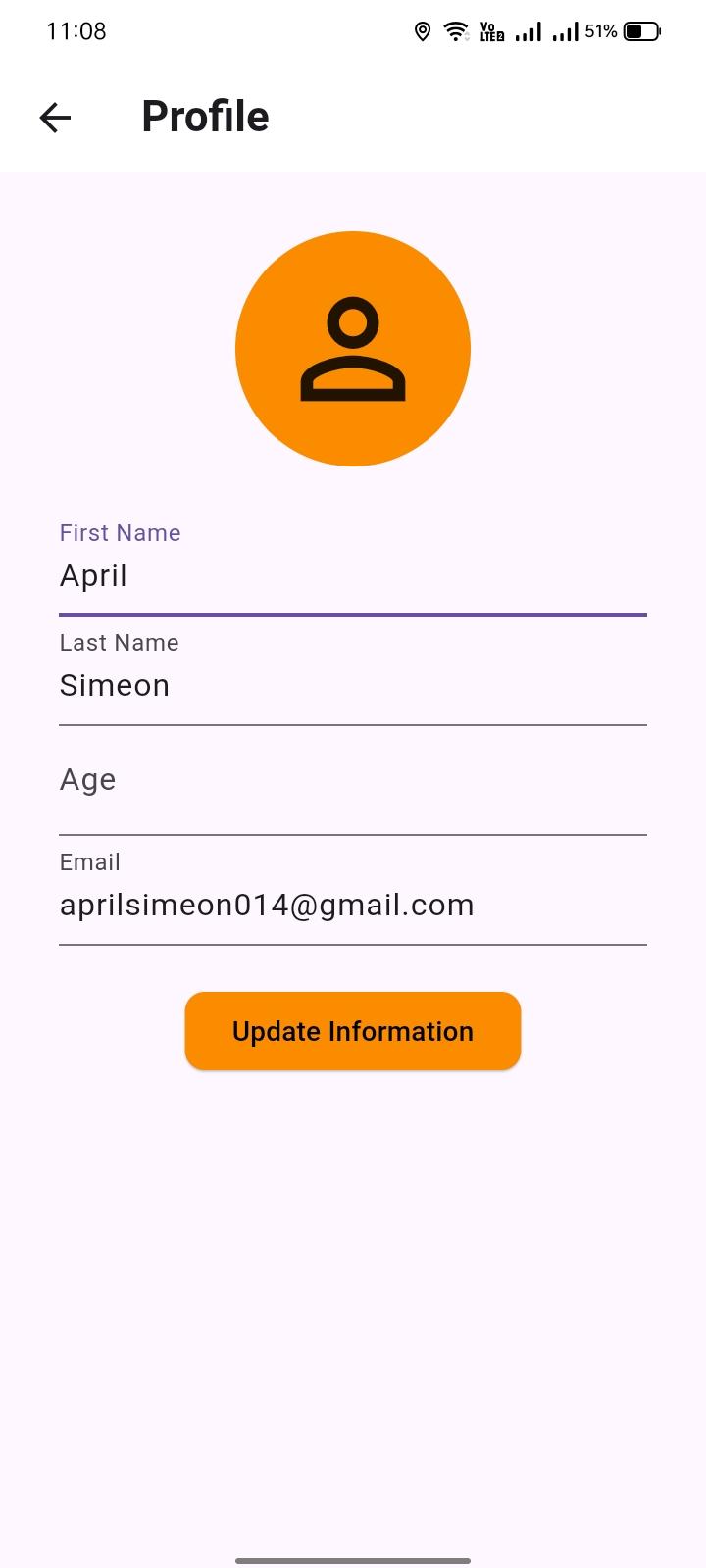
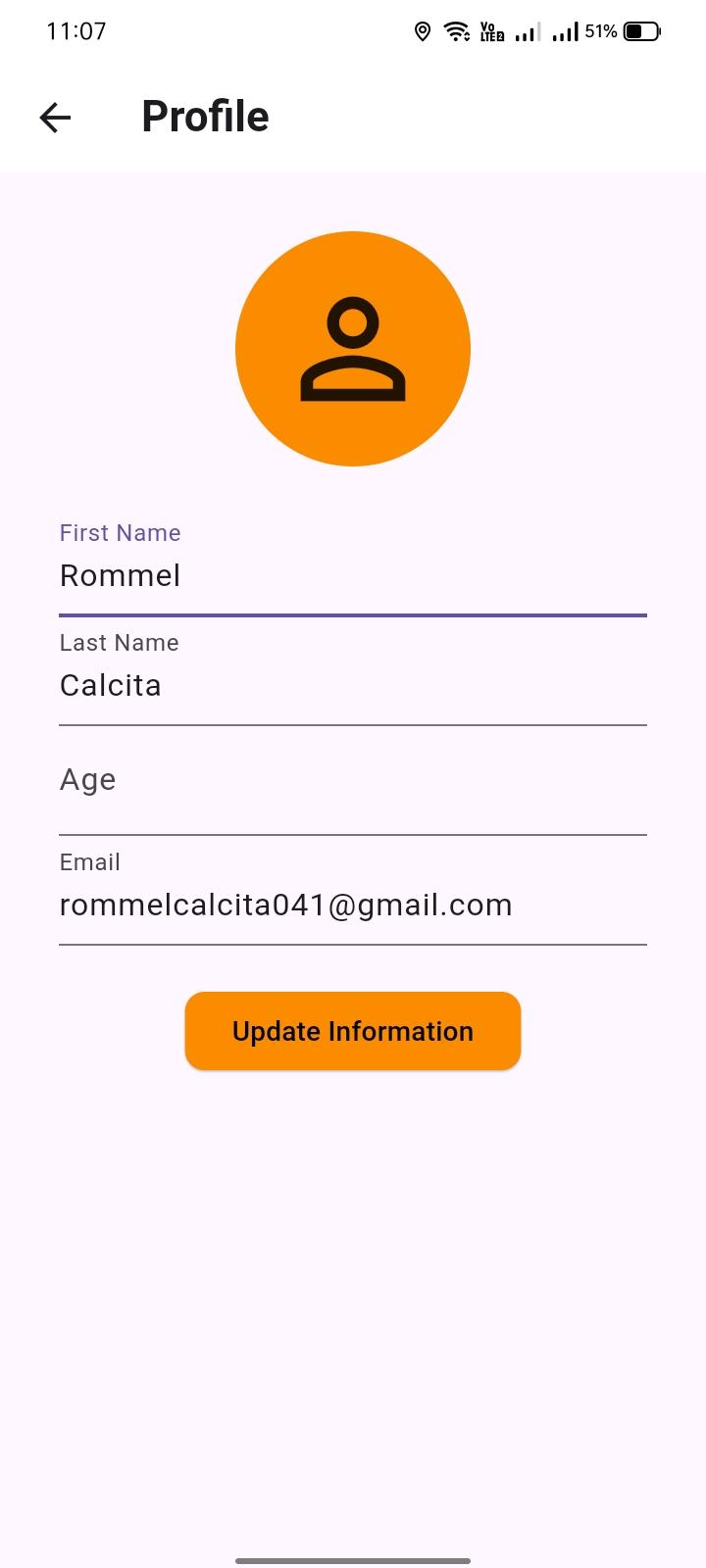
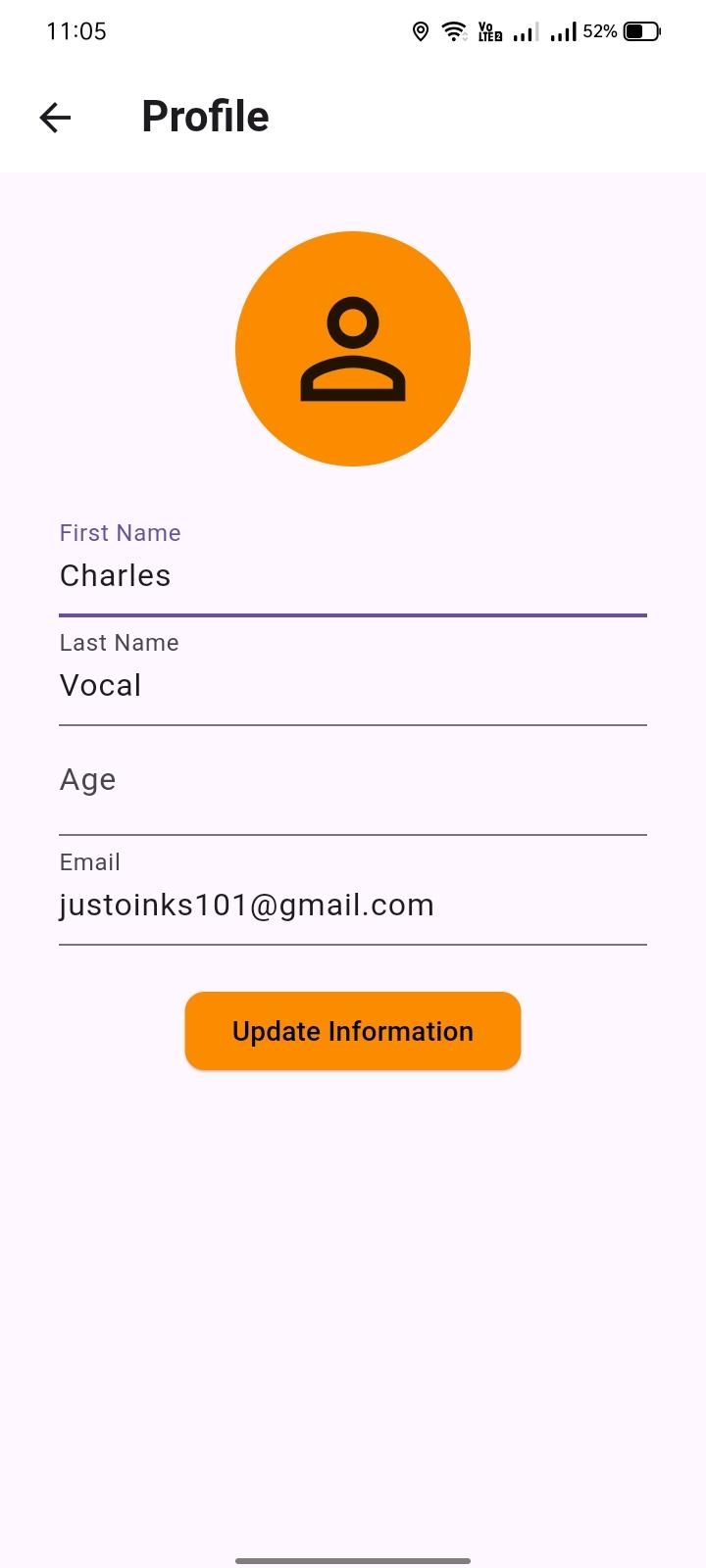


**APPENDIX E**

**Route4Me Mobile Application Documentation (Commuters)**

**  ’**

****

****

**APPENDIX F**

**Source Code of Route4Me Mobile Application**

DRIVERS’ APP

Assistant\_Methods

import 'package:firebase\_database/firebase\_database.dart';

import 'package:geolocator/geolocator.dart';

import 'package:google\_maps\_flutter/google\_maps\_flutter.dart';

import 'package:provider/provider.dart';

import 'package:route4me\_driver/assistants/request\_assistant.dart';

import 'package:route4me\_driver/global/directions.dart';

import 'package:route4me\_driver/global/global.dart';

import 'package:route4me\_driver/global/map\_key.dart';

import 'package:route4me\_driver/info%20handler/app\_info.dart';

import 'package:route4me\_driver/models/direction\_infos.dart';

import 'package:route4me\_driver/models/user\_model.dart';

class assistantMethods {

static Future<UserModel> readCurrentOnlineUserInfo() async {

try {

final currentUser = firebaseAuth.currentUser;

if (currentUser != null) {

final userRef = FirebaseDatabase.instance

.ref()

.child('Drivers')

.child(currentUser.uid);

DatabaseEvent event = await userRef.once();

DataSnapshot snapshot = event.snapshot;

if (snapshot.value != null) {

Map<String, dynamic> data =

Map<String, dynamic>.from(snapshot.value as Map);

UserModel userModel = UserModel(

firstName: data['First Name'] ?? '',

lastName: data['Last Name'] ?? '',

age: data['Age'] ?? 0,

email: data['Email'] ?? '',

uid: data['Uid'] ?? '',

profileImageUrl: data['carImage'],

);

return userModel;

} else {

throw Exception('User document does not exist');

}

} else {

throw Exception('Current user is null');

}

} catch (error) {

rethrow; // Propagate the error for handling by the caller

}

}

static Future<void> updateUserInfo(UserModel updatedUserModel) async {

try {

final currentUser = firebaseAuth.currentUser;

if (currentUser != null) {

final userRef = FirebaseDatabase.instance

.ref()

.child('Drivers')

.child(currentUser.uid);

await userRef.update({

'First Name': updatedUserModel.firstName,

'Last Name': updatedUserModel.lastName,

'Age': updatedUserModel.age,

'Email': updatedUserModel.email,

'carImage': updatedUserModel.profileImageUrl,

});

} else {

throw Exception('Current user is null');

}

} catch (error) {

rethrow; // Propagate the error for handling by the caller

}

}

static Future<String> searchAddressForGeographicCoordinates(

Position position, context) async {

String apiURL =

"https://maps.googleapis.com/maps/api/geocode/json?latlng=${position.latitude},${position.longitude}&key=$mapKey";

String humanReadableAddress = "";

var requestResponse = await RequestAssistant.receiveRequest(apiURL);

if (requestResponse != "Error Occurred. Failed. No Response.") {

humanReadableAddress = requestResponse["results"][0]["formatted\_address"];

Directions userPickUpAddress = Directions();

userPickUpAddress.locationLatitude = position.latitude;

userPickUpAddress.locationLongitude = position.longitude;

userPickUpAddress.locationName = humanReadableAddress;

Provider.of<appInfo>(context, listen: false)

.updatePickUpAddress(userPickUpAddress);

}

return humanReadableAddress;

}

static Future<DirectionDetailsInfo> obtainOriginToDestinationDirectionDetails(

LatLng originPosition, LatLng destinationPosition) async {

String urlOriginToDestinationDirectionDetails =

"https://maps.googleapis.com/maps/api/directions/json?origin=${originPosition.latitude},${originPosition.longitude}&destination=${destinationPosition.latitude},${destinationPosition.longitude}&key=$mapKey";

var responseDirectionApi = await RequestAssistant.receiveRequest(

urlOriginToDestinationDirectionDetails);

DirectionDetailsInfo directionDetailsInfo = DirectionDetailsInfo();

directionDetailsInfo.e\_points =

responseDirectionApi["routes"][0]["overview\_polyline"]["points"];

directionDetailsInfo.distance\_text =

responseDirectionApi["routes"][0]["legs"][0]["distance"]["text"];

directionDetailsInfo.distance\_value =

responseDirectionApi["routes"][0]["legs"][0]["distance"]["value"];

directionDetailsInfo.duration\_text =

responseDirectionApi["routes"][0]["legs"][0]["duration"]["text"];

directionDetailsInfo.duration\_value =

responseDirectionApi["routes"][0]["legs"][0]["duration"]["value"];

return directionDetailsInfo;

}

}

Request\_Assistant

import 'dart:convert';

import 'package:http/http.dart' as http;

class RequestAssistant{

static Future<dynamic> receiveRequest(String url) async{

http.Response httpResponse = await http.get(Uri.parse(url));

try {

if (httpResponse.statusCode == 200) {

String responseData = httpResponse.body;

var decodeResponseData = jsonDecode(responseData);

return decodeResponseData;

} else {

return "Error Occured. Failed. No Response.";

}

} catch(e) {

return "Error Occured. Failed. No Response.";

}

}

}

Button

import 'package:flutter/material.dart';

class button extends StatelessWidget {

final Function()? onTap;

final String text;

const button({

super.key,

required this.onTap,

required this.text,

});

@override

Widget build(BuildContext context) {

return Padding(

padding: const EdgeInsets.symmetric(horizontal: 25),

child: Row(

children: [

Expanded(

child: ElevatedButton(

onPressed: onTap,

style: ElevatedButton.styleFrom(

backgroundColor: Colors.orange[600],

shape: RoundedRectangleBorder(

borderRadius: BorderRadius.circular(10),

),

padding: const EdgeInsets.all(15),

),

child: Text(

text,

style: const TextStyle(

color: Colors.white,

fontWeight: FontWeight.bold,

fontSize: 20,

),

),

),

),

],

),

);

}

}

Circle\_Tile

import 'package:flutter/material.dart';

class CircleTile extends StatelessWidget {

final String imagePath;

final Function()? onTap;

const CircleTile({

super.key,

required this.imagePath,

required this.onTap,

});

@override

Widget build(BuildContext context) {

return GestureDetector(

onTap: onTap,

child: Container(

padding: const EdgeInsets.all(5),

decoration: BoxDecoration(

border: Border.all(color: Colors.white),

shape: BoxShape.circle, // Use BoxShape.circle to make it circular

color: Colors.grey[200],

),

child: CircleAvatar(

backgroundImage: AssetImage(imagePath),

radius: 20, // Set the radius as needed

),

),

);

}

}

Drawer\_Tile

import 'package:flutter/material.dart';

class DrawerTile extends StatelessWidget {

final String text;

final IconData? icon;

final void Function()? onTap;

const DrawerTile({

super.key,

required this.text,

required this.icon,

required this.onTap,

});

@override

Widget build(BuildContext context) {

return Padding(

padding: const EdgeInsets.only(left: 25.0),

child: ListTile(

title: Text(

text,

style: const TextStyle(

color: Colors.black,

fontWeight: FontWeight.bold,

),

),

leading: Icon(

icon,

color: Colors.orange[600],

),

onTap: onTap,

),

);

}

}

Text\_Field

import 'package:flutter/material.dart';

class textfield extends StatelessWidget {

final TextEditingController controller;

final String hintText;

final bool obscureText;

const textfield({

super.key,

required this.controller,

required this.hintText,

required this.obscureText,

});

@override

Widget build(BuildContext context) {

return Padding(

padding: const EdgeInsets.symmetric(horizontal: 25.0),

child: TextField(

controller: controller,

obscureText: obscureText,

decoration: InputDecoration(

enabledBorder: OutlineInputBorder(

borderRadius:

BorderRadius.circular(10), // Adjust border radius as needed

borderSide: const BorderSide(color: Colors.white),

),

focusedBorder: OutlineInputBorder(

borderRadius:

BorderRadius.circular(10), // Adjust border radius as needed

borderSide: BorderSide(color: Colors.orange.shade600),

),

fillColor: Colors.grey[300],

filled: true,

hintText: hintText,

hintStyle: TextStyle(color: Colors.grey[600]),

),

),

);

}

}

Directions

class Directions{

String? humanReadableAddress;

String? locationName;

String? locationId;

double? locationLatitude;

double? locationLongitude;

Directions(

{this.humanReadableAddress,

this.locationName,

this.locationId,

this.locationLatitude,

this.locationLongitude,}

);

}

Global

import 'package:firebase\_auth/firebase\_auth.dart';

import 'package:geolocator/geolocator.dart';

import 'package:route4me\_driver/models/driver\_data.dart';

import 'package:route4me\_driver/models/user\_model.dart';

final FirebaseAuth firebaseAuth = FirebaseAuth.instance;

User? currentUser;

UserModel? userModelCurrentInfo;

Position? driverCurrentPosition;

DriverData onlineDriverData = DriverData();

String? driverVehicleType = '';

Map\_Key

String mapKey = "AIzaSyCsKzXQHefWMPcBUibiEx-WAuODIjMzhDw";

App\_Info

import 'package:flutter/cupertino.dart';

import 'package:route4me\_driver/global/directions.dart';

class appInfo extends ChangeNotifier {

Directions? userPickUpLocation, userDestinationLocation;

int countTotalTrips = 0;

//List<String> historyTripsKeysList = [];

//List<TripsHistoryModel> allTripHistoryInformationList = [];

void updatePickUpAddress(Directions userPickUpAddress) {

userPickUpLocation = userPickUpAddress;

notifyListeners();

}

void updateDestinationAddress(Directions userDestinationAddress) {

userDestinationLocation = userDestinationAddress;

notifyListeners();

}

}

Direction\_Infos

class DirectionDetailsInfo {

int? distance\_value;

int? duration\_value;

String? e\_points;

String? distance\_text;

String? duration\_text;

DirectionDetailsInfo({

this.distance\_text,

this.duration\_text,

this.e\_points,

this.distance\_value,

this.duration\_value,

});

}

Driver\_Data

class DriverData {

int? age;

String? firstName;

String? lastName;

String? email;

String? carPlate;

String? carType;

DriverData({

this.age,

this.firstName,

this.lastName,

this.email,

this.carPlate,

this.carType,

});

}

User\_Model

import 'package:firebase\_database/firebase\_database.dart';

class UserModel {

String firstName;

String lastName;

int age;

String email;

String uid;

String? profileImageUrl;

UserModel({

required this.firstName,

required this.lastName,

required this.age,

required this.email,

required this.uid,

this.profileImageUrl,

});

factory UserModel.fromSnapshot(DataSnapshot snapshot) {

Map<String, dynamic> data = snapshot.value as Map<String, dynamic>;

return UserModel(

firstName: data['First Name'] ?? '',

lastName: data['Last Name'] ?? '',

age: data['Age'] ?? 0,

email: data['Email'] ?? '',

uid: data['Uid'] ?? '',

profileImageUrl: data['Profile Image URL'],

);

}

Map<String, dynamic> toMap() {

return {

'First Name': firstName,

'Last Name': lastName,

'Age': age,

'Email': email,

'Uid': uid,

'Profile Image URL': profileImageUrl,

};

}

}

Auth\_Page

import 'package:flutter/material.dart';

import 'package:firebase\_auth/firebase\_auth.dart';

import 'package:route4me\_driver/global/global.dart';

import 'package:route4me\_driver/pages/home\_page.dart';

import 'package:route4me\_driver/pages/login\_register\_page.dart';

class AuthPage extends StatelessWidget {

const AuthPage({super.key});

@override

Widget build(BuildContext context) {

return Scaffold(

body: StreamBuilder<User?>(

stream: firebaseAuth.authStateChanges(),

builder: (context, snapshot) {

//user is logged in

if (snapshot.hasData) {

return const HomePage();

}

//user is not logged in

else {

return const LoginOrRegisterPage();

}

},

),

);

}

}

Car\_Info\_Page

import 'package:firebase\_database/firebase\_database.dart';

import 'package:firebase\_storage/firebase\_storage.dart';

import 'package:flutter/material.dart';

import 'dart:io';

import 'package:image\_picker/image\_picker.dart';

import 'package:route4me\_driver/components/button.dart';

import 'package:route4me\_driver/components/text\_field.dart';

import 'package:route4me\_driver/global/global.dart';

import 'package:route4me\_driver/pages/home\_page.dart';

class carInfoPage extends StatefulWidget {

const carInfoPage({super.key});

@override

State<carInfoPage> createState() => \_carInfoPageState();

}

class \_carInfoPageState extends State<carInfoPage> {

final carPlateController = TextEditingController();

List<String> carTypes = [

'Jeepney (TPUJ)',

'E-Jeepney Aircon (A-MPUJ)',

'E-Jeepney Non-Aircon (Na-MPUJ)',

'Bus Aircon (A-PUB)',

'Bus Ordinary (O-PUB)'

];

String? selectedCarType;

File? \_image;

@override

void dispose() {

carPlateController.dispose();

super.dispose();

}

Future<void> \_pickImage() async {

final picker = ImagePicker();

final pickedFile = await picker.pickImage(source: ImageSource.gallery);

if (pickedFile != null) {

setState(() {

\_image = File(pickedFile.path);

});

}

}

Future<void> \_captureImage() async {

final picker = ImagePicker();

final pickedFile = await picker.pickImage(source: ImageSource.camera);

if (pickedFile != null) {

setState(() {

\_image = File(pickedFile.path);

});

}

}

Future<String?> \_uploadImage(File image) async {

try {

final currentUser = firebaseAuth.currentUser;

if (currentUser == null) {

throw Exception('No authenticated user');

}

final userId = currentUser.uid;

final storageRef = FirebaseStorage.instance

.ref()

.child('car\_images')

.child(userId)

.child('${DateTime.now().millisecondsSinceEpoch}.jpg');

final uploadTask = storageRef.putFile(image);

final snapshot = await uploadTask.whenComplete(() => null);

final downloadUrl = await snapshot.ref.getDownloadURL();

return downloadUrl;

} catch (e) {

print("Failed to upload image: $e");

return null;

}

}

void savePUVDetails() async {

try {

final currentUser = firebaseAuth.currentUser;

if (currentUser != null) {

final userRef = FirebaseDatabase.instance

.ref()

.child('Drivers')

.child(currentUser.uid);

String? imageUrl;

if (\_image != null) {

imageUrl = await \_uploadImage(\_image!);

}

await userRef.update({

'carPlate': carPlateController.text,

'carType': selectedCarType,

'carImage': imageUrl,

});

// Show success dialog

showDialog(

context: context,

builder: (context) {

return AlertDialog(

shape: RoundedRectangleBorder(

borderRadius: BorderRadius.circular(10),

side: const BorderSide(color: Colors.orange, width: 2),

),

title: const Text(

'Success',

style: TextStyle(color: Colors.black),

),

content: const Text(

'PUV details added successfully!',

style: TextStyle(color: Colors.black),

),

actions: <Widget>[

TextButton(

onPressed: () {

Navigator.of(context).pop();

Navigator.pushReplacement(

context,

MaterialPageRoute(builder: (context) => const HomePage()),

);

},

child: const Text(

'OK',

style: TextStyle(color: Colors.black),

),

),

],

);

},

);

} else {

throw Exception('Current user is null');

}

} catch (error) {

print("Failed to save PUV details: $error");

// Show error dialog

showDialog(

context: context,

builder: (context) {

return AlertDialog(

shape: RoundedRectangleBorder(

borderRadius: BorderRadius.circular(10),

side: const BorderSide(color: Colors.orange, width: 2),

),

title: const Text(

'Error',

style: TextStyle(color: Colors.black),

),

content: const Text(

'Failed to save PUV details. Please try again later.',

style: TextStyle(color: Colors.black),

),

actions: <Widget>[

TextButton(

onPressed: () {

Navigator.of(context).pop();

},

child: const Text(

'OK',

style: TextStyle(color: Colors.black),

),

),

],

);

},

);

}

}

@override

Widget build(BuildContext context) {

return GestureDetector(

onTap: () {

FocusScope.of(context).unfocus();

},

child: Scaffold(

body: ListView(

padding: const EdgeInsets.all(10),

children: [

Column(

children: [

const SizedBox(

height: 60,

),

Icon(

Icons.bus\_alert\_outlined,

size: 100,

color: Colors.orange[600],

),

const SizedBox(

height: 30,

),

const Text(

'Add PUV Details',

style: TextStyle(

color: Colors.black,

fontSize: 20,

fontWeight: FontWeight.normal,

),

),

const SizedBox(

height: 20,

),

textfield(

controller: carPlateController,

hintText: ' Plate No.',

obscureText: false,

),

const SizedBox(

height: 20,

),

Padding(

padding: const EdgeInsets.symmetric(horizontal: 20.0),

child: DropdownButtonFormField(

decoration: InputDecoration(

hintText: ' Choose PUV Type',

filled: true,

fillColor: Colors.grey[300],

border: OutlineInputBorder(

borderRadius: BorderRadius.circular(10),

borderSide: const BorderSide(

width: 0,

style: BorderStyle.none,

),

),

),

items: carTypes.map((car) {

return DropdownMenuItem(

value: car,

child: Text(

car,

style: TextStyle(color: Colors.grey[800]),

),

);

}).toList(),

onChanged: (newValue) {

setState(() {

selectedCarType = newValue.toString();

});

}),

),

const SizedBox(

height: 20,

),

\_image != null

? Image.file(\_image!)

: const Text('No image selected.'),

Row(

mainAxisAlignment: MainAxisAlignment.center,

children: [

ElevatedButton(

onPressed: \_pickImage,

style: ElevatedButton.styleFrom(

foregroundColor: Colors.black,

backgroundColor: Colors.white, // Text color

side: const BorderSide(

color: Colors.orange), // Border color

),

child: const Text('Upload Image'),

),

const SizedBox(width: 20),

ElevatedButton(

onPressed: \_captureImage,

style: ElevatedButton.styleFrom(

foregroundColor: Colors.black,

backgroundColor: Colors.white, // Text color

side: const BorderSide(

color: Colors.orange), // Border color

),

child: const Text('Capture Image'),

),

],

),

const SizedBox(

height: 20,

),

const SizedBox(height: 15),

button(

text: "Add Details",

onTap: savePUVDetails,

),

const SizedBox(height: 20),

],

)

],

),

),

);

}

}

Forgot\_Page

import 'package:firebase\_auth/firebase\_auth.dart';

import 'package:flutter/material.dart';

import 'package:route4me\_driver/components/button.dart';

import 'package:route4me\_driver/components/text\_field.dart';

import 'package:route4me\_driver/global/global.dart';

class ForgotPasswordPage extends StatefulWidget {

const ForgotPasswordPage({super.key});

@override

State<ForgotPasswordPage> createState() => \_ForgotPasswordPageState();

}

class \_ForgotPasswordPageState extends State<ForgotPasswordPage> {

final emailController = TextEditingController();

@override

void dispose() {

emailController.dispose();

super.dispose();

}

Future passwordReset() async {

// show loading circle

showDialog(

context: context,

builder: (context) {

return const Center(

child: CircularProgressIndicator(),

);

},

);

try {

await firebaseAuth.sendPasswordResetEmail(

email: emailController.text.trim());

Navigator.pop(context);

// If no exception is thrown, show a success message

showDialog(

context: context,

builder: (context) {

return const AlertDialog(

content: Text(

textAlign: TextAlign.center,

'Password reset link was sent successfully, you can now set a new password and go back to the login page'),

);

});

} on FirebaseAuthException catch (e) {

Navigator.pop(context);

showDialog(

context: context,

builder: (context) {

return AlertDialog(

content: Text(e.message.toString()),

);

},

);

}

}

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

backgroundColor: Colors.orange[600],

elevation: 0,

),

body: SingleChildScrollView(

child: Column(

mainAxisAlignment: MainAxisAlignment.center,

children: [

Image.asset(

'lib/images/route4me logo.png',

height: 300,

width: 300,

),

const Padding(

padding: EdgeInsets.symmetric(horizontal: 25.0),

child: Text(

'Enter your email and wait for the password reset link!',

textAlign: TextAlign.center,

style: TextStyle(fontSize: 20),

),

),

//emailfield

const SizedBox(height: 10),

textfield(

controller: emailController,

hintText: ' Email',

obscureText: false,

),

//reset pass button

const SizedBox(height: 10),

button(

text: "Reset Password",

onTap: passwordReset,

),

],

),

),

);

}

}

Home\_Page

import 'package:flutter/material.dart';

import 'package:route4me\_driver/tab%20pages/ratings\_tab.dart';

import 'package:route4me\_driver/tab%20pages/settings\_tab.dart';

import '../tab pages/home\_tab.dart';

class HomePage extends StatefulWidget {

const HomePage({super.key});

@override

State<HomePage> createState() => \_HomePageState();

}

class \_HomePageState extends State<HomePage>

with SingleTickerProviderStateMixin {

TabController? tabController;

int selectedIndex = 0;

onItemClicked(int index) {

setState(() {

selectedIndex = index;

tabController!.index = selectedIndex;

});

}

@override

void initState() {

super.initState();

tabController = TabController(length: 3, vsync: this);

}

@override

Widget build(BuildContext context) {

return Scaffold(

body: TabBarView(

physics: const NeverScrollableScrollPhysics(),

controller: tabController,

children: const [

HomeTab(),

RatingsTabPage(),

SettingsPage(),

],

),

bottomNavigationBar: BottomNavigationBar(

items: const [

BottomNavigationBarItem(

icon: Icon(

Icons.home\_outlined,

),

label: 'Home'),

BottomNavigationBarItem(

icon: Icon(

Icons.star\_border\_outlined,

),

label: 'Reviews'),

BottomNavigationBarItem(

icon: Icon(

Icons.settings\_applications\_outlined,

),

label: 'Settings'),

],

unselectedItemColor: Colors.black,

selectedItemColor: Colors.white,

backgroundColor: Colors.orange[600],

type: BottomNavigationBarType.fixed,

selectedLabelStyle: const TextStyle(fontSize: 14),

showUnselectedLabels: true,

currentIndex: selectedIndex,

onTap: onItemClicked,

),

);

}

}

Log-in\_Page

import 'package:firebase\_auth/firebase\_auth.dart';

import 'package:flutter/material.dart';

import 'package:route4me\_driver/components/button.dart';

import 'package:route4me\_driver/components/text\_field.dart';

import 'package:route4me\_driver/components/circle\_tile.dart';

import 'package:route4me\_driver/pages/forgot\_page.dart';

import 'package:route4me\_driver/pages/home\_page.dart';

import 'package:route4me\_driver/services/auth\_service.dart';

class LoginPage extends StatefulWidget {

final Function()? onTap;

const LoginPage({super.key, required this.onTap});

@override

State<LoginPage> createState() => \_LoginPageState();

}

class \_LoginPageState extends State<LoginPage> {

//text editing controllers

final emailController = TextEditingController();

final passwordController = TextEditingController();

//sign user in method

void logIn() async {

showLoadingDialog();

try {

await FirebaseAuth.instance.signInWithEmailAndPassword(

email: emailController.text,

password: passwordController.text,

);

// Optionally fetch additional user details here if necessary

dismissLoadingDialog(); // Ensure dialog is dismissed before navigating

// Navigate to HomePage or similar

navigateToHome();

} on FirebaseAuthException catch (e) {

dismissLoadingDialog();

showErrorDialog(e.message ?? "An unknown error occurred.");

} catch (e) {

dismissLoadingDialog();

showErrorDialog('An unexpected error occurred: ${e.toString()}');

}

}

void showLoadingDialog() {

if (!mounted) return;

showDialog(

context: context,

barrierDismissible: false,

builder: (context) => const Center(child: CircularProgressIndicator()),

);

}

void dismissLoadingDialog() {

if (Navigator.canPop(context)) {

Navigator.of(context, rootNavigator: true).pop();

}

}

void navigateToHome() {

Navigator.of(context).pushReplacement(

MaterialPageRoute(builder: (context) => const HomePage()),

);

}

void showErrorDialog(String message) {

if (!mounted) return;

showDialog(

context: context,

builder: (context) => AlertDialog(

content: Text(message),

actions: <Widget>[

TextButton(

child: const Text('OK'),

onPressed: () {

if (Navigator.canPop(context)) {

Navigator.of(context).pop(); // Dismiss the error dialog

}

},

),

],

),

);

}

@override

void dispose() {

emailController.dispose();

passwordController.dispose();

super.dispose();

}

@override

Widget build(BuildContext context) {

return Scaffold(

body: SafeArea(

child: Center(

child: SingleChildScrollView(

child: Column(mainAxisAlignment: MainAxisAlignment.center, children: [

const SizedBox(height: 50),

Image.asset(

'lib/images/route4me logo.png',

height: 260,

width: 300,

),

//emailfield

const SizedBox(height: 1),

textfield(

controller: emailController,

hintText: ' Email',

obscureText: false,

),

//passwordfield

const SizedBox(height: 1),

textfield(

controller: passwordController,

hintText: ' Password',

obscureText: true,

),

//forgotpassword

const SizedBox(height: 5),

Padding(

padding: const EdgeInsets.symmetric(horizontal: 25.0),

child: Row(

mainAxisAlignment: MainAxisAlignment.end,

children: [

GestureDetector(

onTap: () {

Navigator.push(

context,

MaterialPageRoute(

builder: (context) {

return const ForgotPasswordPage();

},

),

);

},

child: const Text(

'Forgot Password?',

style: TextStyle(

color: Colors.orange,

fontWeight: FontWeight.bold,

),

),

),

],

),

),

//signIn button

const SizedBox(height: 15),

button(

text: "Log In",

onTap: logIn,

),

//divider

const SizedBox(height: 20),

Row(

mainAxisAlignment: MainAxisAlignment.center,

children: [

const Text(

'Continue with',

style: TextStyle(

fontSize: 20,

color: Colors.black,

),

),

const SizedBox(width: 20),

CircleTile(

onTap: () => AuthService().signInWithGoogle(),

imagePath: 'lib/images/Google.png'),

/\*SizedBox(width: 0),

CircleTile(onTap: () {}, imagePath: 'lib/images/Facebook.png'),\*/

],

),

const Divider(thickness: 2),

Row(

mainAxisAlignment: MainAxisAlignment.center,

children: [

const Text(

'No account?',

style: TextStyle(

fontSize: 15,

color: Colors.black,

),

),

const SizedBox(width: 10),

GestureDetector(

onTap: widget.onTap,

child: const Text(

'Register now',

style: TextStyle(

color: Colors.orange,

fontWeight: FontWeight.bold,

),

),

),

],

),

]),

),

)),

);

}

}

Log-in\_Register\_Page

import 'package:flutter/material.dart';

import 'package:route4me\_driver/pages/login\_page.dart';

import 'package:route4me\_driver/pages/register\_page.dart';

class LoginOrRegisterPage extends StatefulWidget {

const LoginOrRegisterPage({super.key});

@override

State<LoginOrRegisterPage> createState() => \_LoginOrRegisterPageState();

}

class \_LoginOrRegisterPageState extends State<LoginOrRegisterPage> {

//initially show login page

bool showLoginPage = true;

//toggle between login and register page

void togglePages() {

setState(() {

showLoginPage = !showLoginPage;

});

}

@override

Widget build(BuildContext context) {

if (showLoginPage) {

return LoginPage(

onTap: togglePages,

);

} else {

return RegisterPage(

onTap: togglePages,

);

}

}

}

Profile\_Page

import 'package:flutter/material.dart';

import 'package:image\_picker/image\_picker.dart';

import 'dart:io';

import 'package:firebase\_storage/firebase\_storage.dart';

import 'package:route4me\_driver/assistants/assistant\_methods.dart';

import 'package:route4me\_driver/global/global.dart';

import 'package:route4me\_driver/models/user\_model.dart';

class ProfilePage extends StatefulWidget {

const ProfilePage({super.key});

@override

\_ProfilePageState createState() => \_ProfilePageState();

}

class \_ProfilePageState extends State<ProfilePage> {

final firstNameController = TextEditingController();

final lastNameController = TextEditingController();

final ageController = TextEditingController();

final emailController = TextEditingController();

UserModel? currentUser;

String? profileImageUrl;

@override

void initState() {

super.initState();

fetchCurrentUser();

}

Future<void> fetchCurrentUser() async {

try {

UserModel user = await assistantMethods.readCurrentOnlineUserInfo();

setState(() {

currentUser = user;

firstNameController.text = user.firstName;

lastNameController.text = user.lastName;

ageController.text = user.age.toString();

emailController.text = user.email;

profileImageUrl = user.profileImageUrl;

});

} catch (error) {

print("Failed to fetch user info: $error");

}

}

Future<void> updateUserInfo() async {

try {

if (currentUser != null) {

final updatedUserModel = UserModel(

firstName: firstNameController.text,

lastName: lastNameController.text,

age: int.tryParse(ageController.text) ?? 0,

email: emailController.text,

profileImageUrl: profileImageUrl,

uid: currentUser!.uid,

);

await assistantMethods.updateUserInfo(updatedUserModel);

setState(() {

currentUser = updatedUserModel;

});

print('User information updated successfully');

}

} catch (error) {

print("Failed to update user information: $error");

}

}

Future<void> showUserNameDialogAlert(BuildContext context) {

return showDialog(

context: context,

builder: (context) {

return AlertDialog(

title: const Text('Update User Information'),

content: SingleChildScrollView(

child: Column(

children: [

TextFormField(

controller: firstNameController,

decoration: const InputDecoration(labelText: 'First Name'),

),

TextFormField(

controller: lastNameController,

decoration: const InputDecoration(labelText: 'Last Name'),

),

TextFormField(

controller: ageController,

decoration: const InputDecoration(labelText: 'Age'),

),

TextFormField(

controller: emailController,

decoration: const InputDecoration(labelText: 'Email'),

),

],

),

),

actions: [

TextButton(

onPressed: () {

Navigator.pop(context);

},

child: const Text(

'Cancel',

style: TextStyle(color: Colors.red),

),

),

TextButton(

onPressed: () {

updateUserInfo();

Navigator.pop(context);

},

child: const Text(

'Ok',

style: TextStyle(color: Colors.blue),

),

),

],

);

},

);

}

Future<void> \_pickImage() async {

final picker = ImagePicker();

final pickedFile = await picker.pickImage(source: ImageSource.gallery);

if (pickedFile != null) {

final imageUrl = await \_uploadImage(File(pickedFile.path));

setState(() {

profileImageUrl = imageUrl;

});

updateUserInfo();

}

}

Future<String?> \_uploadImage(File image) async {

try {

final currentUser = firebaseAuth.currentUser;

if (currentUser == null) {

throw Exception('No authenticated user');

}

final userId = currentUser.uid;

final storageRef = FirebaseStorage.instance

.ref()

.child('car\_images')

.child(userId)

.child('${DateTime.now().millisecondsSinceEpoch}.jpg');

final uploadTask = storageRef.putFile(image);

final snapshot = await uploadTask.whenComplete(() => null);

final downloadUrl = await snapshot.ref.getDownloadURL();

return downloadUrl;

} catch (e) {

print("Failed to upload image: $e");

return null;

}

}

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: const Text(

'Profile',

style: TextStyle(

fontWeight: FontWeight.bold,

),

),

backgroundColor: Colors.white,

),

body: SingleChildScrollView(

child: Center(

child: Padding(

padding: const EdgeInsets.all(30),

child: Column(

crossAxisAlignment: CrossAxisAlignment.center,

children: [

GestureDetector(

onTap: \_pickImage,

child: Stack(

alignment: Alignment.center,

children: [

Container(

width: 125,

height: 125,

decoration: BoxDecoration(

color: Colors.orange,

shape: BoxShape.circle,

),

),

ClipOval(

child: Container(

width: 120,

height: 120,

decoration: BoxDecoration(

shape: BoxShape.circle,

image: profileImageUrl != null

? DecorationImage(

image: NetworkImage(profileImageUrl!),

fit: BoxFit.cover,

)

: null,

),

child: profileImageUrl == null

? const Icon(Icons.person\_outline, size: 80)

: null,

),

),

],

),

),

const SizedBox(height: 20),

TextFormField(

controller: firstNameController,

decoration: const InputDecoration(labelText: 'First Name'),

readOnly: true,

),

TextFormField(

controller: lastNameController,

decoration: const InputDecoration(labelText: 'Last Name'),

readOnly: true,

),

TextFormField(

controller: ageController,

decoration: const InputDecoration(labelText: 'Age'),

readOnly: true,

),

TextFormField(

controller: emailController,

decoration: const InputDecoration(labelText: 'Email'),

readOnly: true,

),

const SizedBox(height: 20),

ElevatedButton(

onPressed: () {

showUserNameDialogAlert(context);

},

style: ElevatedButton.styleFrom(

shape: RoundedRectangleBorder(

borderRadius: BorderRadius.circular(10),

),

backgroundColor: Colors.orange[600],

),

child: const Text(

'Update Information',

style: TextStyle(color: Colors.black),

),

),

],

),

),

),

),

);

}

}

Register\_Page

import 'dart:async';

import 'package:firebase\_auth/firebase\_auth.dart';

import 'package:flutter/material.dart';

import 'package:route4me\_driver/components/button.dart';

import 'package:route4me\_driver/components/text\_field.dart';

import 'package:route4me\_driver/components/circle\_tile.dart';

import 'package:route4me\_driver/global/global.dart';

import 'package:route4me\_driver/pages/car\_info\_page.dart';

import 'package:route4me\_driver/services/auth\_service.dart';

import 'package:firebase\_database/firebase\_database.dart';

class RegisterPage extends StatefulWidget {

final Function()? onTap;

const RegisterPage({super.key, required this.onTap});

@override

State<RegisterPage> createState() => \_RegisterPageState();

}

class \_RegisterPageState extends State<RegisterPage> {

// text editing controllers

final emailController = TextEditingController();

final passwordController = TextEditingController();

final confirmPasswordController = TextEditingController();

final firstNameController = TextEditingController();

final lastNameController = TextEditingController();

final ageController = TextEditingController();

Timer? \_timer;

@override

void dispose() {

\_timer?.cancel();

emailController.dispose();

passwordController.dispose();

confirmPasswordController.dispose();

firstNameController.dispose();

lastNameController.dispose();

ageController.dispose();

super.dispose();

}

// sign user up method

void signUp() {

if (passwordController.text == confirmPasswordController.text) {

// Use AuthService to register the user

var authService = AuthService();

authService

.registerWithEmailPassword(

emailController.text.trim(),

passwordController.text.trim(),

)

.then((user) {

if (user != null) {

// User has been created and verification email sent

addUserDetails(

firstNameController.text.trim(),

lastNameController.text.trim(),

int.parse(ageController.text.trim()),

emailController.text.trim(),

user.uid,

).then((\_) {

// Show a dialog asking the user to verify their email

showDialog(

context: context,

barrierDismissible: false, // Prevent dialog from being dismissed

builder: (context) {

\_startEmailVerificationCheck();

return AlertDialog(

shape: RoundedRectangleBorder(

borderRadius: BorderRadius.circular(10),

side: const BorderSide(color: Colors.orange, width: 2),

),

content: Column(

mainAxisSize: MainAxisSize.min,

children: [

const Text(

'Registration successful! Please verify your email and wait a moment.',

textAlign: TextAlign.center,

style: TextStyle(color: Colors.black),

),

const SizedBox(height: 20),

ElevatedButton(

onPressed: () {

user.sendEmailVerification().then((\_) {

ScaffoldMessenger.of(context).showSnackBar(

const SnackBar(

content: Text("Verification email sent")),

);

}).catchError((error) {

print("Failed to send verification email: $error");

});

},

style: ElevatedButton.styleFrom(

foregroundColor: Colors.black,

backgroundColor: Colors.white,

side:

const BorderSide(color: Colors.orange, width: 2),

shape: RoundedRectangleBorder(

borderRadius: BorderRadius.circular(10),

),

),

child: const Text("Resend Verification Email"),

),

],

),

);

},

);

}).catchError((error) {

print("Failed to add user details: $error");

});

} else {

// Show an error message if registration failed

showDialog(

context: context,

builder: (context) {

return const AlertDialog(

content: Text(

textAlign: TextAlign.center,

'Failed to register. Please try again.',

),

);

},

);

}

}).catchError((error) {

print("Registration failed: $error");

// Handle registration error

});

} else {

// Show error if passwords do not match

showDialog(

context: context,

builder: (context) {

return const AlertDialog(

content: Text(

textAlign: TextAlign.center,

'Passwords do not match!',

),

);

},

);

}

}

Future<void> addUserDetails(String firstName, String lastName, int age,

String email, String uid) async {

DatabaseReference userRef =

FirebaseDatabase.instance.ref().child("Drivers").child(uid);

await userRef.set({

'First Name': firstName,

'Last Name': lastName,

'Age': age,

'Email': email,

'Uid': uid,

}).then((\_) {

print("User added with ID: $uid");

}).catchError((error) {

print("Failed to add user: $error");

});

}

void \_startEmailVerificationCheck() {

\_timer = Timer.periodic(const Duration(seconds: 3), (timer) {

FirebaseAuth.instance.currentUser?.reload().then((\_) {

User? user = FirebaseAuth.instance.currentUser;

if (user != null && user.emailVerified) {

\_timer?.cancel();

Navigator.of(context).pop(); // Close the dialog

Navigator.pushReplacement(

context,

MaterialPageRoute(builder: (context) => const carInfoPage()),

);

}

}).catchError((error) {

print("Error reloading user: $error");

});

});

}

@override

Widget build(BuildContext context) {

return Scaffold(

body: SafeArea(

child: Center(

child: SingleChildScrollView(

child: Column(

mainAxisAlignment: MainAxisAlignment.center,

children: [

const SizedBox(height: 10),

Image.asset(

'lib/images/route4me logo.png',

height: 100,

width: 100,

),

const Row(

mainAxisAlignment: MainAxisAlignment.center,

children: [

Text(

'Register below with your details!',

style: TextStyle(

fontSize: 20,

color: Colors.black,

fontStyle: FontStyle.normal,

),

),

],

),

// First name textfield

const SizedBox(height: 1),

textfield(

controller: firstNameController,

hintText: ' First Name',

obscureText: false,

),

// Last name textfield

const SizedBox(height: 1),

textfield(

controller: lastNameController,

hintText: ' Last Name',

obscureText: false,

),

// Age textfield

const SizedBox(height: 1),

textfield(

controller: ageController,

hintText: ' Age',

obscureText: false,

),

// Email textfield

const SizedBox(height: 1),

textfield(

controller: emailController,

hintText: ' Email',

obscureText: false,

),

// Password field

const SizedBox(height: 1),

textfield(

controller: passwordController,

hintText: ' Password',

obscureText: true,

),

// Confirm password field

const SizedBox(height: 1),

textfield(

controller: confirmPasswordController,

hintText: ' Confirm Password',

obscureText: true,

),

// Sign up button

const SizedBox(height: 15),

button(

text: "Sign Up",

onTap: signUp,

),

// Divider

const SizedBox(height: 20),

Row(

mainAxisAlignment: MainAxisAlignment.center,

children: [

const Text(

'Continue with',

style: TextStyle(

fontSize: 20,

color: Colors.black,

),

),

const SizedBox(width: 20),

CircleTile(

onTap: () => AuthService().signInWithGoogle(),

imagePath: 'lib/images/Google.png',

),

],

),

const Divider(thickness: 2),

Row(

mainAxisAlignment: MainAxisAlignment.center,

children: [

const Text(

'Already have an account?',

style: TextStyle(

fontSize: 15,

color: Colors.black,

),

),

const SizedBox(width: 10),

GestureDetector(

onTap: widget.onTap,

child: const Text(

'Login now',

style: TextStyle(

color: Colors.orange,

fontWeight: FontWeight.bold,

),

),

),

],

),

],

),

),

),

),

);

}

}

Splash\_Page

import 'package:flutter/material.dart';

import 'dart:async';

import 'package:route4me\_driver/pages/auth\_page.dart';

class SplashPage extends StatefulWidget {

const SplashPage({super.key});

@override

State<SplashPage> createState() => \_SplashPageState();

}

class \_SplashPageState extends State<SplashPage> {

startTimer() {

Timer(const Duration(seconds: 3), () async {

Navigator.push(

context, MaterialPageRoute(builder: (c) => const AuthPage()));

});

}

@override

void initState() {

super.initState();

startTimer();

}

@override

Widget build(BuildContext context) {

return Scaffold(

body: Center(

child: Column(

mainAxisAlignment: MainAxisAlignment.center,

children: [

Image.asset(

'lib/images/route4me logo.png',

height: 260,

width: 300,

),

],

),

),

);

}

}

Acc\_Del

import 'package:flutter/material.dart';

import 'package:firebase\_auth/firebase\_auth.dart';

import 'package:firebase\_database/firebase\_database.dart';

import 'package:google\_sign\_in/google\_sign\_in.dart';

import 'package:route4me\_driver/pages/login\_register\_page.dart';

class AccountManagement {

static void showDeleteConfirmation(BuildContext context) {

showDialog(

context: context,

builder: (BuildContext context) {

return AlertDialog(

title: const Text(

'Delete Account',

textAlign: TextAlign.center,

),

content: const Text(

'Are you sure you want to delete your account? This action cannot be undone.',

textAlign: TextAlign.center,

),

shape: RoundedRectangleBorder(

borderRadius: BorderRadius.circular(10),

side: const BorderSide(color: Colors.orange, width: 2),

),

actions: <Widget>[

TextButton(

child: const Text('No'),

onPressed: () {

Navigator.of(context).pop(); // Close the dialog

},

),

TextButton(

child: const Text('Yes'),

onPressed: () {

deleteAccount(context);

},

),

],

);

},

);

}

static Future<void> deleteAccount(BuildContext context) async {

final FirebaseAuth firebaseAuth = FirebaseAuth.instance;

final currentUser = firebaseAuth.currentUser;

if (currentUser != null) {

try {

final userRef = FirebaseDatabase.instance

.ref()

.child('Drivers')

.child(currentUser.uid);

await userRef.remove();

await currentUser.delete();

await firebaseAuth.signOut();

GoogleSignIn().signOut();

GoogleSignIn().disconnect();

Navigator.of(context).popUntil((route) => route.isFirst);

showDialog(

context: context,

builder: (BuildContext context) {

return AlertDialog(

title: const Text(

'Account Deleted',

textAlign: TextAlign.center,

),

content: const Text(

'Your account has been successfully deleted.',

textAlign: TextAlign.center,

),

shape: RoundedRectangleBorder(

borderRadius: BorderRadius.circular(10),

side: const BorderSide(color: Colors.orange, width: 2),

),

actions: <Widget>[

TextButton(

child: const Text('OK'),

onPressed: () {

Navigator.of(context).pop(); // Close the success dialog

Navigator.of(context).popUntil((route) => route.isFirst);

Navigator.pushReplacement(

context,

MaterialPageRoute(

builder: (context) => const LoginOrRegisterPage()));

},

),

],

);

},

);

} catch (error) {

showDialog(

context: context,

builder: (BuildContext context) {

return AlertDialog(

title: const Text(

'Error',

textAlign: TextAlign.center,

),

content: Text(

'Failed to delete account: $error',

textAlign: TextAlign.center,

),

shape: RoundedRectangleBorder(

borderRadius: BorderRadius.circular(10),

side: const BorderSide(color: Colors.orange, width: 2),

),

actions: <Widget>[

TextButton(

child: const Text('OK'),

onPressed: () {

Navigator.of(context).pop(); // Close the error dialog

},

),

],

);

},

);

}

} else {

showDialog(

context: context,

builder: (BuildContext context) {

return AlertDialog(

title: const Text(

'Error',

textAlign: TextAlign.center,

),

content: const Text(

'No active user session found.',

textAlign: TextAlign.center,

),

shape: RoundedRectangleBorder(

borderRadius: BorderRadius.circular(10),

side: const BorderSide(color: Colors.orange, width: 2),

),

actions: <Widget>[

TextButton(

child: const Text('OK'),

onPressed: () {

Navigator.of(context).pop(); // Close the dialog

},

),

],

);

},

);

}

}

}

Auth\_Service

import 'package:firebase\_auth/firebase\_auth.dart';

import 'package:google\_sign\_in/google\_sign\_in.dart';

class AuthService {

final FirebaseAuth \_firebaseAuth = FirebaseAuth.instance;

// Google Sign In

Future<UserCredential?> signInWithGoogle() async {

try {

final GoogleSignInAccount? gUser = await GoogleSignIn().signIn();

if (gUser == null) {

print("Google sign-in was aborted.");

return null;

}

final GoogleSignInAuthentication gAuth = await gUser.authentication;

final OAuthCredential credential = GoogleAuthProvider.credential(

accessToken: gAuth.accessToken,

idToken: gAuth.idToken,

);

return await \_firebaseAuth.signInWithCredential(credential);

} catch (error) {

print("Error during Google sign-in: $error");

throw error; // Or handle this error appropriately in your app

}

}

// Register with Email and Password

Future<User?> registerWithEmailPassword(String email, String password) async {

try {

UserCredential userCredential = await \_firebaseAuth

.createUserWithEmailAndPassword(email: email, password: password);

User? user = userCredential.user;

user?.sendEmailVerification();

return user;

} catch (e) {

print("Error in registering user: $e");

return null;

}

}

// Check if email is verified

Future<bool> isEmailVerified() async {

User? user = \_firebaseAuth.currentUser;

return user?.emailVerified ?? false;

}

// Sign in with Email and Password

Future<User?> signInWithEmailPassword(String email, String password) async {

try {

UserCredential userCredential = await \_firebaseAuth

.signInWithEmailAndPassword(email: email, password: password);

User? user = userCredential.user;

if (user != null && !user.emailVerified) {

throw FirebaseAuthException(

code: "email-not-verified",

message: "Please verify your email first.");

}

return user;

} catch (e) {

print("Error during sign-in: $e");

return null;

}

}

}

Home\_Tab

import 'dart:async';

import 'package:firebase\_database/firebase\_database.dart';

import 'package:flutter/material.dart';

import 'package:flutter/services.dart';

import 'package:flutter\_geofire/flutter\_geofire.dart';

import 'package:geolocator/geolocator.dart';

import 'package:google\_maps\_flutter/google\_maps\_flutter.dart';

import 'package:route4me\_driver/assistants/assistant\_methods.dart';

import 'package:route4me\_driver/global/global.dart';

class HomeTab extends StatefulWidget {

const HomeTab({super.key});

@override

State<HomeTab> createState() => \_HomeTabState();

}

class \_HomeTabState extends State<HomeTab> with AutomaticKeepAliveClientMixin {

GoogleMapController? newGoogleMapController;

bool get wantKeepAlive => true;

final Completer<GoogleMapController> \_controllerGoogleMap = Completer();

static const CameraPosition Manila = CameraPosition(

target: LatLng(14.599512, 120.984222),

zoom: 14.4746,

);

var geoLocator = Geolocator();

LocationPermission? locationPermission;

StreamSubscription<Position>? streamSubscriptionPosition;

String statusText = 'Now Offline';

Color buttonColor = Colors.orange.shade600;

bool isDriverActive = false;

bool isVehicleFull = false;

checkIfLocationPermissionAllowed() async {

locationPermission = await Geolocator.requestPermission();

if (locationPermission == LocationPermission.denied) {

locationPermission = await Geolocator.requestPermission();

}

}

locateDriverPosition() async {

Position cPosition = await Geolocator.getCurrentPosition(

desiredAccuracy: LocationAccuracy.high);

driverCurrentPosition = cPosition;

LatLng latLngPosition = LatLng(

driverCurrentPosition!.latitude, driverCurrentPosition!.longitude);

CameraPosition cameraPosition =

CameraPosition(target: latLngPosition, zoom: 15);

newGoogleMapController!

.animateCamera(CameraUpdate.newCameraPosition(cameraPosition));

String humanReadableAddress = await assistantMethods

.searchAddressForGeographicCoordinates(driverCurrentPosition!, context);

print('This is our address = $humanReadableAddress');

}

readCurrentDriverInformation() async {

currentUser = firebaseAuth.currentUser;

FirebaseDatabase.instance

.ref()

.child('Drivers')

.child(currentUser!.uid)

.once()

.then((snap) {

if (snap.snapshot.value != null) {

onlineDriverData.firstName = (snap.snapshot.value as Map)['First Name'];

onlineDriverData.lastName = (snap.snapshot.value as Map)['Last Name'];

onlineDriverData.age = (snap.snapshot.value as Map)['Age'];

onlineDriverData.email = (snap.snapshot.value as Map)['Email'];

onlineDriverData.carPlate = (snap.snapshot.value as Map)["carPlate"];

onlineDriverData.carType = (snap.snapshot.value as Map)["carType"];

driverVehicleType = (snap.snapshot.value as Map)['type'];

}

});

}

@override

void initState() {

super.initState();

checkIfLocationPermissionAllowed();

readCurrentDriverInformation();

}

@override

void dispose() {

streamSubscriptionPosition

?.cancel(); // Cancel the subscription when the state is disposed.

super.dispose();

}

@override

Widget build(BuildContext context) {

super.build(context);

return Scaffold(

body: Stack(

children: [

GoogleMap(

padding: const EdgeInsets.only(top: 40),

mapType: MapType.normal,

myLocationEnabled: true,

zoomControlsEnabled: false,

zoomGesturesEnabled: true,

initialCameraPosition: Manila,

onMapCreated: (GoogleMapController controller) {

\_controllerGoogleMap.complete(controller);

newGoogleMapController = controller;

if (isDriverActive) {

locateDriverPosition();

}

},

),

// Overlay for online/offline UI

statusText != 'Now Online'

? Container(

height: MediaQuery.of(context).size.height,

width: double.infinity,

color: Colors.black87,

)

: Container(),

Positioned(

top: statusText != 'Now Online'

? MediaQuery.of(context).size.height \* 0.45

: 40,

left: 0,

right: 0,

child: Row(

mainAxisAlignment: MainAxisAlignment.center,

children: [

ElevatedButton(

onPressed: toggleDriverStatus,

style: ElevatedButton.styleFrom(

backgroundColor: buttonColor,

padding: const EdgeInsets.symmetric(horizontal: 18),

shape: RoundedRectangleBorder(

borderRadius: BorderRadius.circular(10),

)),

child: Text(

statusText,

style: const TextStyle(

fontSize: 16,

fontWeight: FontWeight.bold,

color: Colors.white,

),

),

),

],

),

),

// Floating action buttons for vehicle status

if (isDriverActive) ...[

Positioned(

bottom: 50,

right: 10,

child: FloatingActionButton(

onPressed: () => setVehicleStatus(

!isVehicleFull), // Toggle the vehicle status

backgroundColor: isVehicleFull

? Colors.red

: Colors.green, // Red if full, green if available

tooltip: isVehicleFull

? 'Set Vehicle as Available'

: 'Set Vehicle as Full', // Tooltip updates based on the state

child: Icon(isVehicleFull

? Icons.bus\_alert\_outlined

: Icons.directions\_bus\_filled\_outlined),

),

)

],

],

),

);

}

void toggleDriverStatus() {

if (!isDriverActive) {

driverIsOnlineNow();

updateDriversLocationAtRealtime();

setState(() {

statusText = 'Now Online';

isDriverActive = true;

buttonColor = Colors.transparent;

});

} else {

driverIsOfflineNow();

setState(() {

statusText = 'Now Offline';

isDriverActive = false;

buttonColor = Colors.orange.shade600;

});

}

}

driverIsOnlineNow() async {

Position pos = await Geolocator.getCurrentPosition(

desiredAccuracy: LocationAccuracy.high,

);

driverCurrentPosition = pos;

Geofire.initialize('activeDrivers');

Geofire.setLocation(currentUser!.uid, driverCurrentPosition!.latitude,

driverCurrentPosition!.longitude);

DatabaseReference ref = FirebaseDatabase.instance

.ref()

.child('Drivers')

.child(currentUser!.uid)

.child('newRideStatus');

ref.set('idle');

ref.onValue.listen((event) {});

}

updateDriversLocationAtRealtime() {

streamSubscriptionPosition

?.cancel(); // First, cancel any existing subscription.

streamSubscriptionPosition =

Geolocator.getPositionStream().listen((Position position) {

if (isDriverActive && !isVehicleFull) {

// Check if the driver is active and the vehicle is not full.

Geofire.setLocation(

currentUser!.uid, position.latitude, position.longitude);

newGoogleMapController!.animateCamera(CameraUpdate.newLatLng(

LatLng(position.latitude, position.longitude)));

}

});

}

driverIsOfflineNow() {

Geofire.removeLocation(currentUser!.uid);

DatabaseReference? ref = FirebaseDatabase.instance

.ref()

.child('Drivers')

.child(currentUser!.uid)

.child('newRideStatus');

ref.onDisconnect();

ref.remove();

ref = null;

Future.delayed(const Duration(milliseconds: 2000), () {

SystemChannels.platform.invokeMethod('SystemNavigator.pop');

});

}

void setVehicleStatus(bool full) {

setState(() {

isVehicleFull = full;

});

if (isVehicleFull) {

Geofire.removeLocation(currentUser!.uid);

streamSubscriptionPosition

?.cancel(); // Cancel the location stream when the vehicle is full.

} else {

updateDriversLocationAtRealtime();

}

}

}

Ratings\_Tab

import 'package:firebase\_auth/firebase\_auth.dart';

import 'package:flutter/material.dart';

import 'package:firebase\_database/firebase\_database.dart';

class RatingsTabPage extends StatefulWidget {

const RatingsTabPage({Key? key}) : super(key: key);

@override

\_RatingsTabPageState createState() => \_RatingsTabPageState();

}

class \_RatingsTabPageState extends State<RatingsTabPage> {

List<Review> reviews = [];

double averageRating = 0.0;

@override

void initState() {

super.initState();

fetchReviews();

}

void fetchReviews() {

final FirebaseAuth auth = FirebaseAuth.instance;

final User? currentUser = auth.currentUser;

if (currentUser == null) {

print("No current user found.");

return;

}

final uid = currentUser.uid;

DatabaseReference reviewsRef =

FirebaseDatabase.instance.ref('Drivers/$uid/reviews');

reviewsRef.onValue.listen((event) {

final data = event.snapshot.value as Map<dynamic, dynamic>? ?? {};

final allReviews = <Review>[];

double totalRating = 0.0;

data.forEach((key, value) {

try {

final review = Review.fromMap(Map<String, dynamic>.from(value));

allReviews.add(review);

totalRating += review.rating;

} catch (e) {

print("Error parsing review data: $e");

}

});

if (allReviews.isNotEmpty) {

setState(() {

reviews = allReviews;

averageRating = totalRating / allReviews.length;

});

}

});

}

@override

Widget build(BuildContext context) {

return Scaffold(

backgroundColor: Colors.white,

body: SingleChildScrollView(

child: Padding(

padding: const EdgeInsets.only(top: 50, left: 20, right: 20),

child: Column(

crossAxisAlignment: CrossAxisAlignment.start,

children: [

Container(

padding: const EdgeInsets.all(10),

decoration: BoxDecoration(

border: Border.all(color: Colors.orange.shade600, width: 2),

borderRadius: BorderRadius.circular(10),

),

child: Column(

crossAxisAlignment: CrossAxisAlignment.start,

children: [

Text('Review Summary',

style: TextStyle(

fontSize: 18,

fontWeight: FontWeight.bold,

color: Colors.black)),

SizedBox(height: 8),

Row(

children: [

Text('${averageRating.toStringAsFixed(1)} ',

style: TextStyle(

fontSize: 24,

fontWeight: FontWeight.bold,

color: Colors.orange.shade800)),

Icon(Icons.star,

color: Colors.orange.shade800, size: 24),

],

),

Divider(color: Colors.orange.shade600),

...List.generate(5, (index) {

return Padding(

padding: const EdgeInsets.symmetric(vertical: 4),

child: Row(

children: [

Text('${5 - index} ',

style: TextStyle(color: Colors.black)),

Icon(Icons.star, color: Colors.orange, size: 16),

const SizedBox(width: 10),

Expanded(

child: LinearProgressIndicator(

value: (reviews

.where((review) =>

review.rating.round() == 5 - index)

.length) /

(reviews.length == 0 ? 1 : reviews.length),

backgroundColor: Colors.grey[200],

valueColor: AlwaysStoppedAnimation<Color>(

Colors.orange.shade600),

minHeight: 10,

),

),

const SizedBox(width: 10),

Text(

'${reviews.where((review) => review.rating.round() == 5 - index).length}'),

],

),

);

}),

],

),

),

const SizedBox(height: 20),

const Text('All Reviews',

style: TextStyle(

fontSize: 20,

fontWeight: FontWeight.bold,

color: Colors.black)),

...reviews.map((review) => ReviewCard(review: review)),

],

),

),

),

);

}

}

class ReviewCard extends StatelessWidget {

final Review review;

const ReviewCard({Key? key, required this.review}) : super(key: key);

@override

Widget build(BuildContext context) {

return Card(

margin: const EdgeInsets.only(top: 10, bottom: 10),

elevation: 2,

shape: RoundedRectangleBorder(

borderRadius: BorderRadius.circular(10),

side: BorderSide(color: Colors.orange.shade600, width: 2),

),

child: Padding(

padding: const EdgeInsets.all(10),

child: ListTile(

title: Text(review.comment, style: TextStyle(color: Colors.black87)),

subtitle: Row(

children: List.generate(

5,

(index) => Icon(

index < review.rating.floor()

? Icons.star

: Icons.star\_border,

color: Colors.orange,

)),

),

),

),

);

}

}

class Review {

final double rating;

final String comment;

Review({required this.rating, required this.comment});

static Review fromMap(Map<String, dynamic> data) {

return Review(

rating: data['rating'].toDouble(),

comment: data['comment'],

);

}

}

Settings\_Tab

import 'package:firebase\_auth/firebase\_auth.dart';

import 'package:flutter/material.dart';

import 'package:google\_sign\_in/google\_sign\_in.dart';

import 'package:route4me\_driver/pages/login\_register\_page.dart';

import 'package:route4me\_driver/pages/profile\_page.dart';

import 'package:route4me\_driver/services/acc\_del.dart';

class SettingsPage extends StatelessWidget {

const SettingsPage({super.key});

Future<void> \_confirmLogout(BuildContext context) async {

final bool confirm = await showDialog<bool>(

context: context,

builder: (BuildContext context) => AlertDialog(

title: const Text("Confirm Logout"),

content: const Text("Are you sure you want to log out?"),

shape: RoundedRectangleBorder(

borderRadius: BorderRadius.circular(10),

side: const BorderSide(color: Colors.orange, width: 2),

),

actionsAlignment: MainAxisAlignment.center,

actions: <Widget>[

TextButton(

onPressed: () => Navigator.of(context).pop(false),

child: const Text('Cancel'),

),

TextButton(

onPressed: () => Navigator.of(context).pop(true),

style: TextButton.styleFrom(

foregroundColor: Colors.white,

backgroundColor: Colors.orange,

),

child: const Text('Logout'),

),

],

),

) ??

false;

if (confirm) {

try {

await FirebaseAuth.instance.signOut();

GoogleSignIn().signOut();

GoogleSignIn().disconnect();

Navigator.of(context).popUntil((route) => route.isFirst);

Navigator.pushReplacement(

context,

MaterialPageRoute(

builder: (context) => const LoginOrRegisterPage()));

;

} catch (error) {

print("Failed to log out: $error");

}

}

}

@override

Widget build(BuildContext context) {

return Scaffold(

backgroundColor: Colors.white,

body: SingleChildScrollView(

child: Container(

padding: const EdgeInsets.only(top: 80, right: 50, left: 50),

child: Column(

crossAxisAlignment: CrossAxisAlignment.center,

children: [

Image.asset(

'lib/images/route4me logo.png',

width: 250,

height: 250,

),

const Divider(

color: Colors.black,

thickness: 1.0,

),

const SizedBox(height: 20),

const Text(

'General Settings',

style: TextStyle(

color: Colors.black,

fontSize: 18.0,

fontWeight: FontWeight.bold,

),

),

ElevatedButton(

onPressed: () {

\_showPrivacyPolicy(context);

},

style: ElevatedButton.styleFrom(

shape: RoundedRectangleBorder(

borderRadius: BorderRadius.circular(10),

),

backgroundColor: Colors.orange[600],

),

child: const Text(

"Privacy",

style: TextStyle(

fontWeight: FontWeight.normal,

fontSize: 16,

color: Colors.white,

),

),

),

ElevatedButton(

onPressed: () {

\_showSecurityInfo(context);

},

style: ElevatedButton.styleFrom(

shape: RoundedRectangleBorder(

borderRadius: BorderRadius.circular(10),

),

backgroundColor: Colors.orange[600],

),

child: const Text(

"Security",

style: TextStyle(

fontWeight: FontWeight.normal,

fontSize: 16,

color: Colors.white,

),

),

),

ElevatedButton(

onPressed: () {

\_showAboutUs(context);

},

style: ElevatedButton.styleFrom(

shape: RoundedRectangleBorder(

borderRadius: BorderRadius.circular(10),

),

backgroundColor: Colors.orange[600],

),

child: const Text(

"About Us",

style: TextStyle(

fontWeight: FontWeight.normal,

fontSize: 16,

color: Colors.white,

),

),

),

const SizedBox(height: 28.0),

const Text(

'Account',

style: TextStyle(

color: Colors.black,

fontSize: 18.0,

fontWeight: FontWeight.bold,

),

),

ElevatedButton(

onPressed: () {

Navigator.push(

context,

MaterialPageRoute(builder: (c) => const ProfilePage()),

);

},

style: ElevatedButton.styleFrom(

shape: RoundedRectangleBorder(

borderRadius: BorderRadius.circular(10),

),

backgroundColor: Colors.orange[600],

),

child: const Text(

"Profile",

style: TextStyle(

fontWeight: FontWeight.normal,

fontSize: 16,

color: Colors.white,

),

),

),

ElevatedButton(

onPressed: () {

AccountManagement.showDeleteConfirmation(context);

},

style: ElevatedButton.styleFrom(

shape: RoundedRectangleBorder(

borderRadius: BorderRadius.circular(10),

),

backgroundColor: Colors.orange[600],

),

child: const Text(

"Delete Account",

style: TextStyle(

fontWeight: FontWeight.normal,

fontSize: 16,

color: Colors.white,

),

),

),

ElevatedButton(

onPressed: () {

\_confirmLogout(context); // Use the confirm logout function

},

style: ElevatedButton.styleFrom(

shape: RoundedRectangleBorder(

borderRadius: BorderRadius.circular(10),

),

backgroundColor: Colors.orange[600],

),

child: const Text(

"Log Out",

style: TextStyle(

fontWeight: FontWeight.normal,

fontSize: 16,

color: Colors.white,

),

),

),

],

),

),

),

);

}

}

void \_showPrivacyPolicy(BuildContext context) {

showDialog(

context: context,

builder: (BuildContext context) {

return AlertDialog(

title: const Text(

'Privacy Policy',

textAlign: TextAlign.center,

),

content: const SingleChildScrollView(

child: ListBody(

children: <Widget>[

Text(

'We take your privacy seriously. This means we only collect personal information that is necessary to provide you with our service. You will always have choices about what information you share and how we use it.',

textAlign: TextAlign.center,

),

Text(

'We will never sell your personal information to third parties, and we will take appropriate security measures to protect your data.',

textAlign: TextAlign.center,

),

],

),

),

shape: RoundedRectangleBorder(

borderRadius: BorderRadius.circular(10),

side: const BorderSide(color: Colors.orange, width: 2),

),

actions: <Widget>[

TextButton(

child: const Text(

'Close',

style: TextStyle(color: Colors.orange),

),

onPressed: () {

Navigator.of(context).pop(); // Close the dialog

},

),

],

);

},

);

}

void \_showAboutUs(BuildContext context) {

showDialog(

context: context,

builder: (BuildContext context) {

// Determine the available screen height and width

double screenWidth = MediaQuery.of(context).size.width;

double screenHeight = MediaQuery.of(context).size.height;

return AlertDialog(

title: const Text(

'About Us',

textAlign: TextAlign.center,

),

content: Container(

// Set a maximum width to ensure the dialog looks good on all devices

width: screenWidth \* 0.9,

// Limit the height to prevent overflow on small devices

constraints: BoxConstraints(

maxHeight: screenHeight \* 0.7,

),

child: SingleChildScrollView(

child: Column(

mainAxisSize: MainAxisSize.min,

children: [

// Use AspectRatio to maintain the image's aspect ratio

AspectRatio(

aspectRatio: 16 / 9,

child: Image.asset(

'lib/images/route4me logo.png',

),

),

Padding(

padding: const EdgeInsets.all(8.0),

child: const Text(

'Our team of developers has poured their expertise into creating these options, allowing you to customize features and optimize your workflow. We\'re constantly working to improve Route4Me. If you have any suggestions, please don\'t hesitate to send us feedback through the app.',

textAlign: TextAlign.center,

),

),

],

),

),

),

shape: RoundedRectangleBorder(

borderRadius: BorderRadius.circular(10),

side: const BorderSide(color: Colors.orange, width: 2),

),

actions: <Widget>[

TextButton(

child: const Text(

'Close',

style: TextStyle(color: Colors.orange),

),

onPressed: () => Navigator.of(context).pop(),

),

],

);

},

);

}

void \_showSecurityInfo(BuildContext context) {

showDialog(

context: context,

builder: (BuildContext context) {

return AlertDialog(

title: const Text(

'Security Information',

textAlign: TextAlign.center,

),

content: SingleChildScrollView(

child: ListBody(

children: const <Widget>[

Text(

'Your security is our top priority. We use industry-standard encryption to protect your data at rest and in transit. Our application ensures secure connections to prevent unauthorized access.',

textAlign: TextAlign.center,

),

Text(

'For your safety, always keep your software updated and be cautious of unsolicited requests asking for your personal information.',

textAlign: TextAlign.center,

),

],

),

),

shape: RoundedRectangleBorder(

borderRadius: BorderRadius.circular(10),

side: const BorderSide(color: Colors.orange, width: 2),

),

actions: <Widget>[

TextButton(

child: const Text(

'Close',

style: TextStyle(color: Colors.orange),

),

onPressed: () {

Navigator.of(context).pop(); // Close the dialog

},

),

],

);

},

);

}

Firebase\_Options

// File generated by FlutterFire CLI.

// ignore\_for\_file: lines\_longer\_than\_80\_chars, avoid\_classes\_with\_only\_static\_members

import 'package:firebase\_core/firebase\_core.dart' show FirebaseOptions;

import 'package:flutter/foundation.dart'

show defaultTargetPlatform, kIsWeb, TargetPlatform;

/// Default [FirebaseOptions] for use with your Firebase apps.

///

/// Example:

/// ```dart

/// import 'firebase\_options.dart';

/// // ...

/// await Firebase.initializeApp(

/// options: DefaultFirebaseOptions.currentPlatform,

/// );

/// ```

class DefaultFirebaseOptions {

static FirebaseOptions get currentPlatform {

if (kIsWeb) {

throw UnsupportedError(

'DefaultFirebaseOptions have not been configured for web - '

'you can reconfigure this by running the FlutterFire CLI again.',

);

}

switch (defaultTargetPlatform) {

case TargetPlatform.android:

return android;

case TargetPlatform.iOS:

return ios;

case TargetPlatform.macOS:

throw UnsupportedError(

'DefaultFirebaseOptions have not been configured for macos - '

'you can reconfigure this by running the FlutterFire CLI again.',

);

case TargetPlatform.windows:

throw UnsupportedError(

'DefaultFirebaseOptions have not been configured for windows - '

'you can reconfigure this by running the FlutterFire CLI again.',

);

case TargetPlatform.linux:

throw UnsupportedError(

'DefaultFirebaseOptions have not been configured for linux - '

'you can reconfigure this by running the FlutterFire CLI again.',

);

default:

throw UnsupportedError(

'DefaultFirebaseOptions are not supported for this platform.',

);

}

}

static const FirebaseOptions android = FirebaseOptions(

apiKey: 'AIzaSyCsKzXQHefWMPcBUibiEx-WAuODIjMzhDw',

appId: '1:315854700999:android:a3172a97c2b128990f19e2',

messagingSenderId: '315854700999',

projectId: 'route4me-8aae4',

databaseURL: 'https://route4me-8aae4-default-rtdb.asia-southeast1.firebasedatabase.app',

storageBucket: 'route4me-8aae4.appspot.com',

);

static const FirebaseOptions ios = FirebaseOptions(

apiKey: 'AIzaSyCjJWZUacvU1vIz\_hYkPG8o2s-1oWdNyOE',

appId: '1:315854700999:ios:fb345ad69106a2770f19e2',

messagingSenderId: '315854700999',

projectId: 'route4me-8aae4',

databaseURL: 'https://route4me-8aae4-default-rtdb.asia-southeast1.firebasedatabase.app',

storageBucket: 'route4me-8aae4.appspot.com',

iosClientId: '315854700999-2lehoj0f916jqas8i9g490keg6mm5uu4.apps.googleusercontent.com',

iosBundleId: 'com.example.route4me',

);

}

Main

import 'package:firebase\_core/firebase\_core.dart';

import 'package:flutter/material.dart';

import 'package:provider/provider.dart';

import 'package:route4me\_driver/firebase\_options.dart';

import 'package:route4me\_driver/info%20handler/app\_info.dart';

import 'package:route4me\_driver/pages/splash\_page.dart';

void main() async {

WidgetsFlutterBinding.ensureInitialized();

await Firebase.initializeApp(

options: DefaultFirebaseOptions.currentPlatform,

);

runApp(const route4me\_driver());

}

class route4me\_driver extends StatelessWidget {

const route4me\_driver({super.key});

@override

Widget build(BuildContext context) {

return ChangeNotifierProvider(

create: (context) => appInfo(),

child: const MaterialApp(

debugShowCheckedModeBanner: false,

title: 'Route4Me',

home: SplashPage(),

),

);

}

}

USERS’ APP

Assistant\_Method

import 'package:geolocator/geolocator.dart';

import 'package:provider/provider.dart';

import 'package:route4me/assistants/request\_assistant.dart';

import 'package:route4me/global/directions.dart';

import 'package:route4me/global/map\_key.dart';

import 'package:route4me/info%20handler/app\_info.dart';

class assistantMethods{

static Future<String> searchAddressForGeographicCoordinates(Position position, context) async{

String apiURL = "https://maps.googleapis.com/maps/api/geocode/json/latlng=${position.latitude},${position.longitude}&key=$mapKey";

String humanReadableAddress = "";

var requestResponse = await RequestAssistant.receiveRequest(apiURL);

if(requestResponse != "Error Occured. Failed. No Response."){

humanReadableAddress = requestResponse["results"][0]["formatted\_address"];

Directions userPickUpAddress = Directions();

userPickUpAddress.locationLatitude = position.latitude;

userPickUpAddress.locationLongitude = position.longitude;

userPickUpAddress.locationName = humanReadableAddress;

Provider.of<appInfo>(context, listen: false).updatePickUpAddress(userPickUpAddress);

}

return humanReadableAddress;

}

}

Request\_Assistant

import 'dart:convert';

import 'package:http/http.dart' as http;

class RequestAssistant{

static Future<dynamic> receiveRequest(String url) async{

http.Response httpResponse = await http.get(Uri.parse(url));

try {

if (httpResponse.statusCode == 200) {

String responseData = httpResponse.body;

var decodeResponseData = jsonDecode(responseData);

return decodeResponseData;

} else {

return "Error Occured. Failed. No Response.";

}

} catch(e) {

return "Error Occured. Failed. No Response.";

}

}

}

Button\_Dart

import 'package:flutter/material.dart';

class button extends StatelessWidget {

final Function()? onTap;

final String text;

const button({

super.key,

required this.onTap,

required this.text,

});

@override

Widget build(BuildContext context) {

return GestureDetector(

onTap: onTap,

child: Container(

padding: const EdgeInsets.all(15),

margin: const EdgeInsets.symmetric(horizontal: 25),

decoration: BoxDecoration(

color: Colors.orange[600],

borderRadius: BorderRadius.circular(30),

),

child: Center(

child: Text(

text,

style: const TextStyle(

color: Colors.white,

fontWeight: FontWeight.bold,

fontSize: 20,

),

),

),

),

);

}

}

Circle\_Tile

import 'package:flutter/material.dart';

class CircleTile extends StatelessWidget {

final String imagePath;

final Function()? onTap;

const CircleTile({

super.key,

required this.imagePath,

required this.onTap,

});

@override

Widget build(BuildContext context) {

return GestureDetector(

onTap: onTap,

child: Container(

padding: const EdgeInsets.all(5),

decoration: BoxDecoration(

border: Border.all(color: Colors.white),

shape: BoxShape.circle, // Use BoxShape.circle to make it circular

color: Colors.grey[200],

),

child: CircleAvatar(

backgroundImage: AssetImage(imagePath),

radius: 20, // Set the radius as needed

),

),

);

}

}

Drawer

import 'package:firebase\_auth/firebase\_auth.dart';

import 'package:flutter/material.dart';

import 'package:route4me/components/drawer\_tile.dart';

import 'package:route4me/pages/settings\_page.dart';

class MyDrawer extends StatelessWidget {

const MyDrawer({super.key});

@override

Widget build(BuildContext context) {

return Drawer(

child: Column(

children: [

//app logo

Padding(

padding: const EdgeInsets.only(top: 50.0),

child: Image.asset(

'lib/images/route4me logo.jpg',

height: 260,

width: 300,

),

),

Padding(

padding: const EdgeInsets.all(25.0),

child: Divider(

color: Colors.orange[600],

),

),

//home list tile

DrawerTile(

text: 'H O M E',

icon: Icons.home,

onTap: () {

Navigator.pop(context);

},

),

//setting list tile

DrawerTile(

text: 'S E T T I N G S',

icon: Icons.settings,

onTap: () {

Navigator.pop(context);

Navigator.push(

context,

MaterialPageRoute(

builder: (context) => const SettingsPage(),

),

);

},

),

//logout list tile

const Spacer(),

DrawerTile(

text: 'L O G O U T',

icon: Icons.logout,

onTap: () {

FirebaseAuth.instance.signOut();

},

),

],

),

);

}

}

Text\_Field

import 'package:flutter/material.dart';

class textfield extends StatelessWidget {

final TextEditingController controller;

final String hintText;

final bool obscureText;

const textfield({

super.key,

required this.controller,

required this.hintText,

required this.obscureText,

});

@override

Widget build(BuildContext context) {

return Padding(

padding: const EdgeInsets.symmetric(horizontal: 25.0),

child: TextField(

controller: controller,

obscureText: obscureText,

decoration: InputDecoration(

enabledBorder: OutlineInputBorder(

borderRadius:

BorderRadius.circular(30), // Adjust border radius as needed

borderSide: const BorderSide(color: Colors.white),

),

focusedBorder: OutlineInputBorder(

borderRadius:

BorderRadius.circular(30), // Adjust border radius as needed

borderSide: BorderSide(color: Colors.orange.shade600),

),

fillColor: Colors.grey[300],

filled: true,

hintText: hintText,

hintStyle: TextStyle(color: Colors.grey[600]),

),

),

);

}

}

Directions

class Directions{

String? humanReadableAddress;

String? locationName;

String? locationId;

double? locationLatitude;

double? locationLongitude;

Directions(

{this.humanReadableAddress,

this.locationName,

this.locationId,

this.locationLatitude,

this.locationLongitude,}

);

}

Map\_Key

String mapKey = "AIzaSyCsKzXQHefWMPcBUibiEx-WAuODIjMzhDw";

App\_Info

import 'package:flutter/cupertino.dart';

import 'package:route4me/global/directions.dart';

class appInfo extends ChangeNotifier{

Directions? userPickUpLocation, userDestinationLocation;

int countTotalTrips = 0;

//List<String> historyTripsKeysList = [];

//List<TripsHistoryModel> allTripHistoryInformationList = [];

void updatePickUpAddress(Directions userPickUpAddress){

userPickUpLocation = userPickUpAddress;

notifyListeners();

}

void updateDestinationAddress(Directions userDestinationAddress){

userDestinationLocation = userDestinationAddress;

notifyListeners();

}

}

Auth\_Page

import 'package:flutter/material.dart';

import 'package:firebase\_auth/firebase\_auth.dart';

import 'package:route4me/pages/home\_page.dart';

import 'package:route4me/pages/login\_register\_page.dart';

class AuthPage extends StatelessWidget {

const AuthPage({super.key});

@override

Widget build(BuildContext context) {

return Scaffold(

body: StreamBuilder<User?>(

stream: FirebaseAuth.instance.authStateChanges(),

builder: (context, snapshot) {

//user is logged in

if (snapshot.hasData) {

return const HomePage();

}

//user is not logged in

else {

return const LoginOrRegisterPage();

}

},

),

);

}

}

Forgot\_Page

import 'package:firebase\_auth/firebase\_auth.dart';

import 'package:flutter/material.dart';

import 'package:route4me/components/button.dart';

import 'package:route4me/components/text\_field.dart';

class ForgotPasswordPage extends StatefulWidget {

const ForgotPasswordPage({super.key});

@override

State<ForgotPasswordPage> createState() => \_ForgotPasswordPageState();

}

class \_ForgotPasswordPageState extends State<ForgotPasswordPage> {

final emailController = TextEditingController();

@override

void dispose() {

emailController.dispose();

super.dispose();

}

Future passwordReset() async {

// show loading circle

showDialog(

context: context,

builder: (context) {

return const Center(

child: CircularProgressIndicator(),

);

},

);

try {

await FirebaseAuth.instance

.sendPasswordResetEmail(email: emailController.text.trim());

Navigator.pop(context);

// If no exception is thrown, show a success message

showDialog(

context: context,

builder: (context) {

return const AlertDialog(

content: Text(

textAlign: TextAlign.center,

'Password reset link was sent successfully, you can now set a new password and go back to the login page'),

);

});

} on FirebaseAuthException catch (e) {

Navigator.pop(context);

showDialog(

context: context,

builder: (context) {

return AlertDialog(

content: Text(e.message.toString()),

);

},

);

}

}

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

backgroundColor: Colors.orange[600],

elevation: 0,

),

body: Column(

mainAxisAlignment: MainAxisAlignment.center,

children: [

Image.asset(

'lib/images/route4me icon lang.jpg',

height: 100,

width: 100,

),

const Padding(

padding: EdgeInsets.symmetric(horizontal: 25.0),

child: Text(

'Enter your email and wait for the password reset link!',

textAlign: TextAlign.center,

style: TextStyle(fontSize: 20),

),

),

//emailfield

const SizedBox(height: 10),

textfield(

controller: emailController,

hintText: ' Email',

obscureText: false,

),

//reset pass button

const SizedBox(height: 10),

button(

text: "Reset Password",

onTap: passwordReset,

),

],

),

);

}

}

Home\_Page

import 'dart:async';

import 'package:firebase\_auth/firebase\_auth.dart';

import 'package:flutter/material.dart';

import 'package:geocoder2/geocoder2.dart';

import 'package:geolocator/geolocator.dart';

import 'package:google\_maps\_flutter/google\_maps\_flutter.dart';

import 'package:location/location.dart' hide LocationAccuracy;

import 'package:provider/provider.dart';

import 'package:route4me/assistants/assistant\_methods.dart';

import 'package:route4me/components/drawer.dart';

import 'package:route4me/global/directions.dart';

import 'package:route4me/global/map\_key.dart';

import '../info handler/app\_info.dart';

class HomePage extends StatefulWidget {

const HomePage({super.key});

@override

State<HomePage> createState() => \_HomePageState();

}

class \_HomePageState extends State<HomePage> {

final user = FirebaseAuth.instance.currentUser!;

final Completer<GoogleMapController> \_controllerGoogleMap = Completer();

GoogleMapController? newGoogleMapController;

LatLng? pickLocation;

Location location = Location();

String? address;

static const CameraPosition \_kGooglePlex = CameraPosition(

target: LatLng(37.42796133580664, -122.085749655962),

zoom: 14.4746,

);

/\* static const CameraPosition \_kLake = CameraPosition(

bearing: 192.8334901395799,

target: LatLng(37.43296265331129, -122.08832357078792),

tilt: 59.440717697143555,

zoom: 19.151926040649414);\*/

//final GlobalKey<ScaffoldState> \_scaffoldState = GlobalKey<ScaffoldState>();

double searchLocationContainerHeight = 220;

double waitingResponsefromDriverContainerHeight = 0;

double assignedDriverInfoContainerHeight = 0;

Position? userCurrentPosition;

var geoLocation = Geolocator();

LocationPermission? locationPermission;

double bottomPaddingofMap = 0;

List<LatLng> pLineCoordinatedList = [];

Set<Polyline> polylineSet = {};

Set<Marker> markerSet = {};

Set<Circle> circleSet = {};

String? userName = "";

String? userEmail = "";

bool openNavigationDrawer = true;

bool activeNearbyDriverKeysLoaded = false;

BitmapDescriptor? activeNearbyIcon;

locateUserPosition() async {

Position cPosition = await Geolocator.getCurrentPosition(

desiredAccuracy: LocationAccuracy.high);

userCurrentPosition = cPosition;

LatLng latLngPosition =

LatLng(userCurrentPosition!.latitude, userCurrentPosition!.longitude);

CameraPosition cameraPosition =

CameraPosition(target: latLngPosition, zoom: 15);

newGoogleMapController!

.animateCamera(CameraUpdate.newCameraPosition(cameraPosition));

String humanReadableAddress = await assistantMethods

.searchAddressForGeographicCoordinates(userCurrentPosition!, context);

print('This is our address = $humanReadableAddress');

// userName = userModelCurrentInfo!.name!;

// userEmail = userModelCurrentInfo!.email!;

// initializeGeofireListener();

//assistantMethods.readTripsKeysForOnlineUser(context);

}

getAddressFromLatLng() async {

try {

GeoData data = await Geocoder2.getDataFromCoordinates(

latitude: pickLocation!.latitude,

longitude: pickLocation!.longitude,

googleMapApiKey: mapKey,

);

setState(() {

Directions userPickUpAddress = Directions();

userPickUpAddress.locationLatitude = pickLocation!.latitude;

userPickUpAddress.locationLongitude = pickLocation!.longitude;

userPickUpAddress.locationName = data.address;

//address = data.address;

Provider.of<appInfo>(context, listen: false)

.updatePickUpAddress(userPickUpAddress);

});

} catch (e) {

print(e);

}

}

checkIfLocationPermissionAllowed() async {

locationPermission = await Geolocator.requestPermission();

if (locationPermission == LocationPermission.denied) {

locationPermission = await Geolocator.requestPermission();

}

}

@override

void initState() {

// TODO: implement initState

super.initState();

checkIfLocationPermissionAllowed();

}

@override

Widget build(BuildContext context) {

return GestureDetector(

onTap: () {

FocusScope.of(context).unfocus();

},

child: Scaffold(

appBar: AppBar(

title: const Text("Home"),

),

drawer: const MyDrawer(),

body: Stack(

children: [

GoogleMap(

mapType: MapType.normal,

myLocationEnabled: true,

zoomGesturesEnabled: true,

zoomControlsEnabled: true,

initialCameraPosition: \_kGooglePlex,

polylines: polylineSet,

markers: markerSet,

circles: circleSet,

onMapCreated: (GoogleMapController controller) {

\_controllerGoogleMap.complete(controller);

newGoogleMapController = controller;

setState(() {});

locateUserPosition();

},

onCameraMove: (CameraPosition? position) {

if (pickLocation != position!.target) {

setState(() {

pickLocation = position.target;

});

}

},

onCameraIdle: () {

getAddressFromLatLng();

},

),

Align(

alignment: Alignment.center,

child: Padding(

padding: const EdgeInsets.only(bottom: 35.0),

child: Icon(

Icons.location\_on,

color: Colors.orange[600],

size: 50,

)),

),

Positioned(

top: 40,

right: 20,

left: 20,

child: Container(

decoration: BoxDecoration(

border: Border.all(color: Colors.orange),

color: Colors.white,

),

padding: const EdgeInsets.all(20),

child: Text(

Provider.of<appInfo>(context).userPickUpLocation != null

? "${(Provider.of<appInfo>(context).userPickUpLocation!.locationName!).substring(0, 24)}..."

: "Not Getting Address",

overflow: TextOverflow.visible,

softWrap: true,

),

),

)

],

),

),

);

}

}

Log-in\_Page

import 'package:firebase\_auth/firebase\_auth.dart';

import 'package:flutter/material.dart';

import 'package:route4me/components/button.dart';

import 'package:route4me/components/text\_field.dart';

import 'package:route4me/components/circle\_tile.dart';

import 'package:route4me/pages/forgot\_page.dart';

import 'package:route4me/services/auth\_service.dart';

class LoginPage extends StatefulWidget {

final Function()? onTap;

const LoginPage({super.key, required this.onTap});

@override

State<LoginPage> createState() => \_LoginPageState();

}

class \_LoginPageState extends State<LoginPage> {

//text editing controllers

final emailController = TextEditingController();

final passwordController = TextEditingController();

//sign user in method

void logIn() async {

// show loading circle

showDialog(

context: context,

builder: (context) {

return const Center(

child: CircularProgressIndicator(),

);

},

);

//try signing in

try {

await FirebaseAuth.instance.signInWithEmailAndPassword(

email: emailController.text, password: passwordController.text);

//pop the loading circle

Navigator.pop(context);

} on FirebaseAuthException catch (e) {

//pop the loading circle

Navigator.pop(context);

showDialog(

context: context,

builder: (context) {

return AlertDialog(

content: Text(e.message.toString()),

);

},

);

}

@override

void dispose() {

emailController.dispose();

passwordController.dispose();

super.dispose();

}

}

@override

Widget build(BuildContext context) {

return Scaffold(

body: SafeArea(

child: Center(

child: SingleChildScrollView(

child: Column(mainAxisAlignment: MainAxisAlignment.center, children: [

const SizedBox(height: 50),

Image.asset(

'lib/images/route4me logo.jpg',

height: 260,

width: 300,

),

//emailfield

const SizedBox(height: 1),

textfield(

controller: emailController,

hintText: ' Email',

obscureText: false,

),

//passwordfield

const SizedBox(height: 1),

textfield(

controller: passwordController,

hintText: ' Password',

obscureText: true,

),

//forgotpassword

const SizedBox(height: 5),

Padding(

padding: const EdgeInsets.symmetric(horizontal: 25.0),

child: Row(

mainAxisAlignment: MainAxisAlignment.end,

children: [

GestureDetector(

onTap: () {

Navigator.push(

context,

MaterialPageRoute(

builder: (context) {

return const ForgotPasswordPage();

},

),

);

},

child: const Text(

'Forgot Password?',

style: TextStyle(

color: Colors.orange,

fontWeight: FontWeight.bold,

),

),

),

],

),

),

//signIn button

const SizedBox(height: 15),

button(

text: "Log In",

onTap: logIn,

),

//divider

const SizedBox(height: 20),

Row(

mainAxisAlignment: MainAxisAlignment.center,

children: [

const Text(

'Continue with',

style: TextStyle(

fontSize: 20,

color: Colors.black,

),

),

const SizedBox(width: 20),

CircleTile(

onTap: () => AuthService().signInWithGoogle(),

imagePath: 'lib/images/Google.png'),

/\*SizedBox(width: 0),

CircleTile(onTap: () {}, imagePath: 'lib/images/Facebook.png'),\*/

],

),

const Divider(thickness: 2),

Row(

mainAxisAlignment: MainAxisAlignment.center,

children: [

const Text(

'No account?',

style: TextStyle(

fontSize: 15,

color: Colors.black,

),

),

const SizedBox(width: 10),

GestureDetector(

onTap: widget.onTap,

child: const Text(

'Register now',

style: TextStyle(

color: Colors.orange,

fontWeight: FontWeight.bold,

),

),

),

],

),

]),

),

)),

);

}

}

Log-in\_Register\_Page

import 'package:flutter/material.dart';

import 'package:route4me/pages/login\_page.dart';

import 'package:route4me/pages/register\_page.dart';

class LoginOrRegisterPage extends StatefulWidget {

const LoginOrRegisterPage({super.key});

@override

State<LoginOrRegisterPage> createState() => \_LoginOrRegisterPageState();

}

class \_LoginOrRegisterPageState extends State<LoginOrRegisterPage> {

//initially show login page

bool showLoginPage = true;

//toggle between login and register page

void togglePages() {

setState(() {

showLoginPage = !showLoginPage;

});

}

@override

Widget build(BuildContext context) {

if (showLoginPage) {

return LoginPage(

onTap: togglePages,

);

} else {

return RegisterPage(

onTap: togglePages,

);

}

}

}

Register\_Page

import 'package:cloud\_firestore/cloud\_firestore.dart';

import 'package:firebase\_auth/firebase\_auth.dart';

import 'package:flutter/material.dart';

import 'package:route4me/components/button.dart';

import 'package:route4me/components/text\_field.dart';

import 'package:route4me/components/circle\_tile.dart';

import 'package:route4me/services/auth\_service.dart';

class RegisterPage extends StatefulWidget {

final Function()? onTap;

const RegisterPage({super.key, required this.onTap});

@override

State<RegisterPage> createState() => \_RegisterPageState();

}

class \_RegisterPageState extends State<RegisterPage> {

//text editing controllers

final emailController = TextEditingController();

final passwordController = TextEditingController();

final confirmPasswordController = TextEditingController();

final firstNameController = TextEditingController();

final lastNameController = TextEditingController();

final ageController = TextEditingController();

@override

void dispose() {

emailController.dispose();

passwordController.dispose();

confirmPasswordController.dispose();

firstNameController.dispose();

lastNameController.dispose();

ageController.dispose();

super.dispose();

}

//sign user up method

void signUp() async {

try {

Future addUserDetails(

String firstName, String lastName, int Age, String Email) async {

await FirebaseFirestore.instance.collection("Users").add({

'First Name': firstName,

'Last Name': lastName,

'Age': Age,

'Email': Email,

});

}

//check if password is confirmed

if (passwordController.text == confirmPasswordController.text) {

//create user

await FirebaseAuth.instance.createUserWithEmailAndPassword(

email: emailController.text, password: passwordController.text);

//add user details

addUserDetails(

firstNameController.text.trim(),

lastNameController.text.trim(),

int.parse(ageController.text.trim()),

emailController.text.trim(),

);

//if no exception were thrown

showDialog(

context: context,

builder: (context) {

return const AlertDialog(

content: Text(

textAlign: TextAlign.center, 'You have been registered!'),

);

});

} else {

//show error message that passwords don't match

showDialog(

context: context,

builder: (context) {

return const AlertDialog(

content:

Text(textAlign: TextAlign.center, 'Password don\'t match!'),

);

});

}

} on FirebaseAuthException catch (e) {

// show the error code

showDialog(

context: context,

builder: (context) {

return AlertDialog(

content: Text(

textAlign: TextAlign.center,

e.message.toString(),

),

);

});

}

}

@override

Widget build(BuildContext context) {

return Scaffold(

body: SafeArea(

child: Center(

child: SingleChildScrollView(

child: Column(mainAxisAlignment: MainAxisAlignment.center, children: [

const SizedBox(height: 10),

Image.asset(

'lib/images/route4me icon lang.jpg',

height: 100,

width: 100,

),

const Row(

mainAxisAlignment: MainAxisAlignment.center,

children: [

Text(

'Register below with your details!',

style: TextStyle(

fontSize: 20,

color: Colors.black,

fontStyle: FontStyle.normal,

),

),

],

),

//first name textfield

const SizedBox(height: 1),

textfield(

controller: firstNameController,

hintText: ' First Name',

obscureText: false,

),

//last name textfield

const SizedBox(height: 1),

textfield(

controller: lastNameController,

hintText: ' Last Name',

obscureText: false,

),

//age textfield

const SizedBox(height: 1),

textfield(

controller: ageController,

hintText: ' Age',

obscureText: false,

),

//email textfield

const SizedBox(height: 1),

textfield(

controller: emailController,

hintText: ' Email',

obscureText: false,

),

//passwordfield

const SizedBox(height: 1),

textfield(

controller: passwordController,

hintText: ' Password',

obscureText: true,

),

//confirmpasswordfield

const SizedBox(height: 1),

textfield(

controller: confirmPasswordController,

hintText: ' Confirm Password',

obscureText: true,

),

//signUp button

const SizedBox(height: 15),

button(

text: "Sign Up",

onTap: signUp,

),

//divider

const SizedBox(height: 20),

Row(

mainAxisAlignment: MainAxisAlignment.center,

children: [

const Text(

'Continue with',

style: TextStyle(

fontSize: 20,

color: Colors.black,

),

),

const SizedBox(width: 20),

CircleTile(

onTap: () => AuthService().signInWithGoogle(),

imagePath: 'lib/images/Google.png'),

],

),

const Divider(thickness: 2),

Row(

mainAxisAlignment: MainAxisAlignment.center,

children: [

const Text(

'Already have an account?',

style: TextStyle(

fontSize: 15,

color: Colors.black,

),

),

const SizedBox(width: 10),

GestureDetector(

onTap: widget.onTap,

child: const Text(

'Login now',

style: TextStyle(

color: Colors.orange,

fontWeight: FontWeight.bold,

),

),

),

],

),

]),

),

)),

);

}

}

Settings\_Page

import 'package:flutter/material.dart';

class SettingsPage extends StatelessWidget {

const SettingsPage({super.key});

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: const Text("Settings"),

),

);

}

}

Auth\_Service

import 'package:firebase\_auth/firebase\_auth.dart';

import 'package:google\_sign\_in/google\_sign\_in.dart';

class AuthService {

//Google Sign In

signInWithGoogle() async {

//begin interactive sign in process

final GoogleSignInAccount? gUser = await GoogleSignIn().signIn();

//obtain auth details from the requests

final GoogleSignInAuthentication gAuth = await gUser!.authentication;

//create a new credential for the user

final credential = GoogleAuthProvider.credential(

accessToken: gAuth.accessToken,

idToken: gAuth.idToken,

);

//finally let's sign in

return await FirebaseAuth.instance.signInWithCredential(credential);

}

}

Firebase\_Options

// File generated by FlutterFire CLI.

// ignore\_for\_file: lines\_longer\_than\_80\_chars, avoid\_classes\_with\_only\_static\_members

import 'package:firebase\_core/firebase\_core.dart' show FirebaseOptions;

import 'package:flutter/foundation.dart'

show defaultTargetPlatform, kIsWeb, TargetPlatform;

/// Default [FirebaseOptions] for use with your Firebase apps.

///

/// Example:

/// ```dart

/// import 'firebase\_options.dart';

/// // ...

/// await Firebase.initializeApp(

/// options: DefaultFirebaseOptions.currentPlatform,

/// );

/// ```

class DefaultFirebaseOptions {

static FirebaseOptions get currentPlatform {

if (kIsWeb) {

throw UnsupportedError(

'DefaultFirebaseOptions have not been configured for web - '

'you can reconfigure this by running the FlutterFire CLI again.',

);

}

switch (defaultTargetPlatform) {

case TargetPlatform.android:

return android;

case TargetPlatform.iOS:

return ios;

case TargetPlatform.macOS:

throw UnsupportedError(

'DefaultFirebaseOptions have not been configured for macos - '

'you can reconfigure this by running the FlutterFire CLI again.',

);

case TargetPlatform.windows:

throw UnsupportedError(

'DefaultFirebaseOptions have not been configured for windows - '

'you can reconfigure this by running the FlutterFire CLI again.',

);

case TargetPlatform.linux:

throw UnsupportedError(

'DefaultFirebaseOptions have not been configured for linux - '

'you can reconfigure this by running the FlutterFire CLI again.',

);

default:

throw UnsupportedError(

'DefaultFirebaseOptions are not supported for this platform.',

);

}

}

static const FirebaseOptions android = FirebaseOptions(

apiKey: 'AIzaSyCsKzXQHefWMPcBUibiEx-WAuODIjMzhDw',

appId: '1:315854700999:android:a3172a97c2b128990f19e2',

messagingSenderId: '315854700999',

projectId: 'route4me-8aae4',

databaseURL: 'https://route4me-8aae4-default-rtdb.asia-southeast1.firebasedatabase.app',

storageBucket: 'route4me-8aae4.appspot.com',

);

static const FirebaseOptions ios = FirebaseOptions(

apiKey: 'AIzaSyCjJWZUacvU1vIz\_hYkPG8o2s-1oWdNyOE',

appId: '1:315854700999:ios:fb345ad69106a2770f19e2',

messagingSenderId: '315854700999',

projectId: 'route4me-8aae4',

databaseURL: 'https://route4me-8aae4-default-rtdb.asia-southeast1.firebasedatabase.app',

storageBucket: 'route4me-8aae4.appspot.com',

iosClientId: '315854700999-2lehoj0f916jqas8i9g490keg6mm5uu4.apps.googleusercontent.com',

iosBundleId: 'com.example.route4me',

);

}

Main

import 'package:firebase\_core/firebase\_core.dart';

import 'package:flutter/material.dart';

import 'package:provider/provider.dart';

import 'package:route4me/firebase\_options.dart';

import 'package:route4me/info%20handler/app\_info.dart';

import 'package:route4me/pages/auth\_page.dart';

void main() async {

WidgetsFlutterBinding.ensureInitialized();

await Firebase.initializeApp(

options: DefaultFirebaseOptions.currentPlatform,

);

runApp(const route4me());

}

class route4me extends StatelessWidget {

const route4me({super.key});

@override

Widget build(BuildContext context) {

return ChangeNotifierProvider(

create: (context) => appInfo(),

child: const MaterialApp(

debugShowCheckedModeBanner: false,

title: 'Route4Me',

home: AuthPage(),

),

);

}

}