# Final Year Project Report

**Full Unit – Interim Report**

---

# Fuel Delivery Mobile Application

Fahad Riaz (101027495)

---

A report submitted in part fulfilment of the degree of

**BSc (Hons) in Computer Science**

**Supervisor:** Vasudha Darbari


Department of Computer Science
Royal Holloway, University of London

December 10, 2024

# Declaration

This report has been prepared on the basis of my own work. Where other published and unpublished source materials have been used, these have been acknowledged.

Word Count: 5,231 words excluding title page, declaration page, table of contents, and bibliography.

Student Name: Fahad Riaz

Date of Submission: 10th December 2024

Signature:

# Table of Contents

# Introduction

There is an increasing demand for fuel because of many cars being introduced in the auto industry and this has led to a need for building fuel delivery applications that require the users to place an order for their vehicle when their fuel runs out or when it is low without the hassle of going to the petrol stations [1] . The app aims to provide the users with a convenient and efficient platform for ordering fuel to their vehicles at any given location, this can be done by using real-time GPS tracking of the delivery vehicle which will allow the users to monitor the status. Also, the app will implement secure payment integration for ordering fuel. Users will be presented with an intuitive platform that will ease the whole process of ordering and receiving fuel at their locations [2] . This will change how people approach refuelling in their day to day lives because it will be a time saving and seamless experience [2] . By using Flutter, the app will have support for both Android and iOS mobile phones making it cross platform with an attractive user interface because of Flutter. The app will also integrate with Google Maps because of its powerful navigation capabilities for real time GPS tracking. There are many benefits for this app, users will be able to order fuel from anywhere which will also provide accessibility to remote areas which will make sure fuel is available for everyone [2] . The process for the app will be simple: Users register on the app and login, share their current location, vehicle details and select the fuel type they want, order fuel, then, the fuel truck will reach the user's location at the given time. There was research conducted on this matter in which users were interviewed to find out if such an app would be successful or not, the response for this research showed that majority (68.8 percent) of the users were interested in having fuel delivered to them and they wanted such an app to be developed [3]. An in-built GPS is found in almost all mobile devices nowadays which enables such devices to accurately determine their location anywhere on Earth, this is done by receiving signals from GPS satellites [4] . Here is a simple image below to understand how the app will work [5].

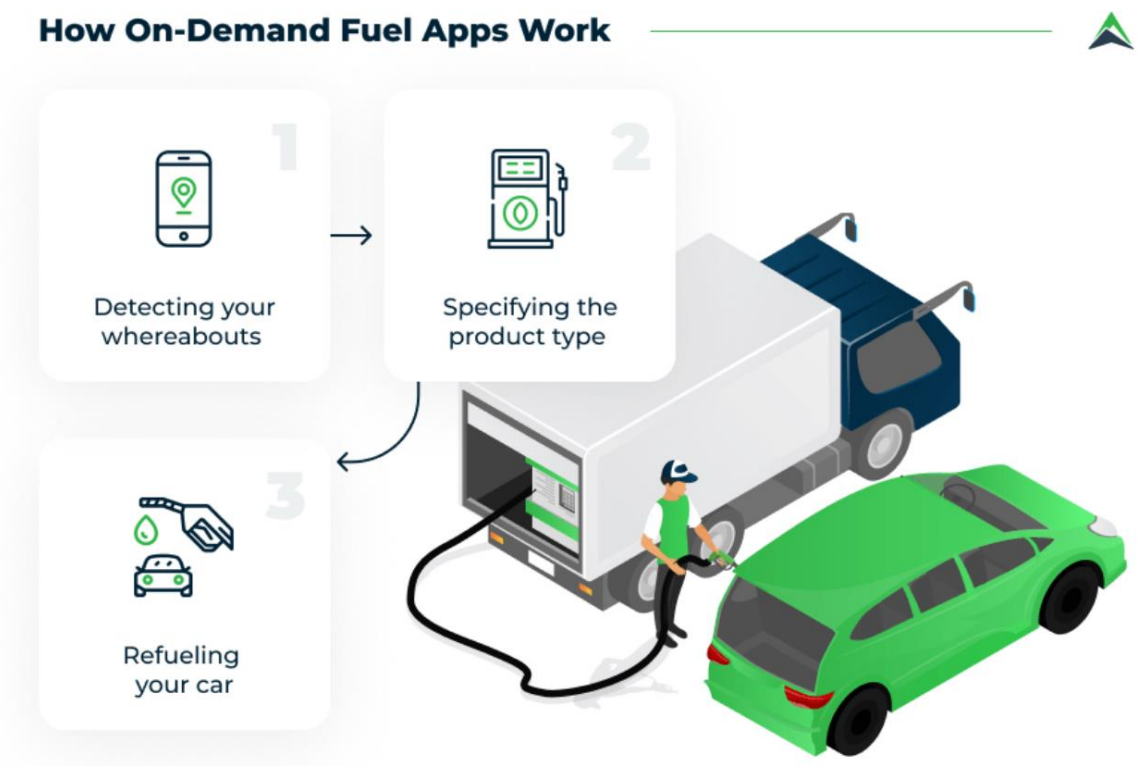



Figure 1: Workflow of the fuel delivery app.

The project's main objectives are:

1. Develop an easy to use and attractive interface/application where users can place fuel orders by selecting fuel type, and delivery location.

2. Implement real time GPS tracking to provide users with live updates on the fuel delivery status.

3. Ensure secure payment system and processing to protect user data.

4. Design and build a scalable system capable of handling multiple user orders and managing delivery status updates.

5. Develop a secure system for user registration, login, and profile management, allowing users to easily create accounts, track their order history and manage their profiles.

6. Use route optimization algorithms to help delivery agents choose the most efficient routes for fuel delivery.

# Project Specification

1. Project Overview

   Mobile App Name: SwiftFuel – Fuel Delivery App

   Summary: SwiftFuel is a mobile application for Android and iOS designed to provide on-demand fuel delivery service, which will allow customers to refuel their vehicles conveniently at their provided location, without the need for going to the gas station. The app aims to deliver a seamless experience for users by offering contactless and efficient fuel delivery services, anytime, anywhere.

2. Functional Requirements

   - User Registration and Authentication: Users must be able to register, log in and log out safely using Firebase Authentication.
   - User Dashboard/Home: The app should provide a user-friendly home screen where users can view and easily navigate to different parts of the app.
   - Fuel Ordering System: Users should be able to select fuel type, enter their vehicle details, and select their location for delivery using Google Maps integration.
   - Payment: Integration of secure payment options for user transactions.
   - Order Confirmation: Users should see their order details for reviewing before placing their orders.
   - Notification System: Users should receive notifications about their placed orders.

3. Non-Functional Requirements

   - Usability: The app should be an intuitive, user-friendly interface that is easy to use and navigate.
   - Performance: The app must be responsive, with minimum delays during different tasks.
   - Scalability: The backend system should be able to handle multiple users and requests.
   - Security: User data must be well protected to ensure security and privacy compliance.

4. Technical Specifications

   - Platform: App is built for Android and iOS using Flutter.
   - Backend: Firebase services used, which includes Firestore database and Firebase Authentication.
   - APIs: Google Maps API for location-based features.
   - Programming Languages: Dart.

- Development Tools: Android Studio, Git

5. User Requirements

    - Target Users: Vehicle owners who want convenient and contactless refuelling for their vehicles without going to the petrol stations.

    - User Stories:

    - "*As a user, I want to register and login easily to access my account securely."*

    - *"As a user, I want to choose my current location so that I can have fuel delivered to me after ordering it from the app."*

    - *"As a user, I want to see my past orders to keep track of them."*

6. Constraints

    - Time Constraints: The project needs to be completed within the academic year (2024/25).

    - Budget Constraints: The app will rely on Firebase's free tier and Google Maps API within limited use.

    - Technical Constraints: Initial development focuses on Android due to hardware limitations, as the project development environment is Windows-based.

7. Assumptions

    - Users have a stable internet connection while using the app.

    - Users provide accurate location details to receive appropriate service.

    - The initial service area is restricted geographically to make logistics feasible.

8. Success Criteria

    - Successful user registration, login, and logout features.

    - Users can place fuel orders with selected location and fuel type etc.

    - Proper integration of Google Maps.

    - Proper integration of Payment system.

    - Successful saving of different data to the Firestore database.

    - Users can track their orders in real-time using Google Maps integration.

9. System Requirements

    - Hardware: Android device.

    - Software: Android OS with internet access.

# Chapter 1: **Literature Review**

The fuel delivery industry has gone through major phases with the emergence of new digital technologies, this literature review will explore existing research, solutions, market analysis, and mobile app development. On-demand fuel delivery services have emerged as a convenient alternative to the traditional fuelling method at gas stations, the main concept in this involves delivering fuel to customers at their locations which ensures convenience and time saving.

## 1.1 Market Analysis and Existing Solutions

There already exist some notable competitors in the market for on-demand fuel delivery services such as Booster in the USA and CAFU in the Middle East so the market has seen some steady growth, driven by consumer demand for more convenient services. Companies like Booster and CAFU have already successfully tapped into the market by providing on-demand fuel delivery services with easy to use mobile applications.

CAFU operates in the United Arab Emirates, specifically in Dubai, and they deliver Super 98 and Super 99 fuel to the people living there in Dubai when they order it using their mobile app [6]. When a user orders fuel from the official CAFU mobile app, they send CAFU trucks to the user's provided location so that the user's vehicle can be refuelled [6]. They have multiple stakeholders in their business and even the oil and fuel companies are an important part of their supply chain [6]. CAFU's existing solution changed the landscape of oil and gas transportation and logistics with their services and reports state that the UAE market has around 3.4 million cars on the road and the annual revenue per customer for CAFU is 264 AED which makes the market's annual estimate at around 897 million AED [6]. This growth in the UAE market has caught the attention of other countries as well and the growth of this market is connected to the general trend toward service-based mobile applications. An analysis of CAFU's existing application highlights features such as real-time GPS tracking, integrated payment systems, and fuel quality guarantees. Just like other on-demand services like food delivery, delivering a product like fuel to the user is not a revolutionary concept but it does become innovative because of the use of technology to enhance customer experience and to make it efficient and convenient [7]. CAFU also does not charge their customers for deliveries, instead, they create customer loyalty by rewarding the customers with points that can be redeemed against other services [7].

The global on-demand fuel delivery market size was 0.33 billion USD in 2024 and it is projected to touch 2.04 billion USD by 2032 which shows a compound annual growth rate (CAGR) of 15.97 percent from 2024 to 2032 [8] . This shows how important this market really is and the increasing use of fuel vehicles in different countries is expected to support the expansion of worldwide on-demand fuel delivery market. (Shown in Figure 2 below)

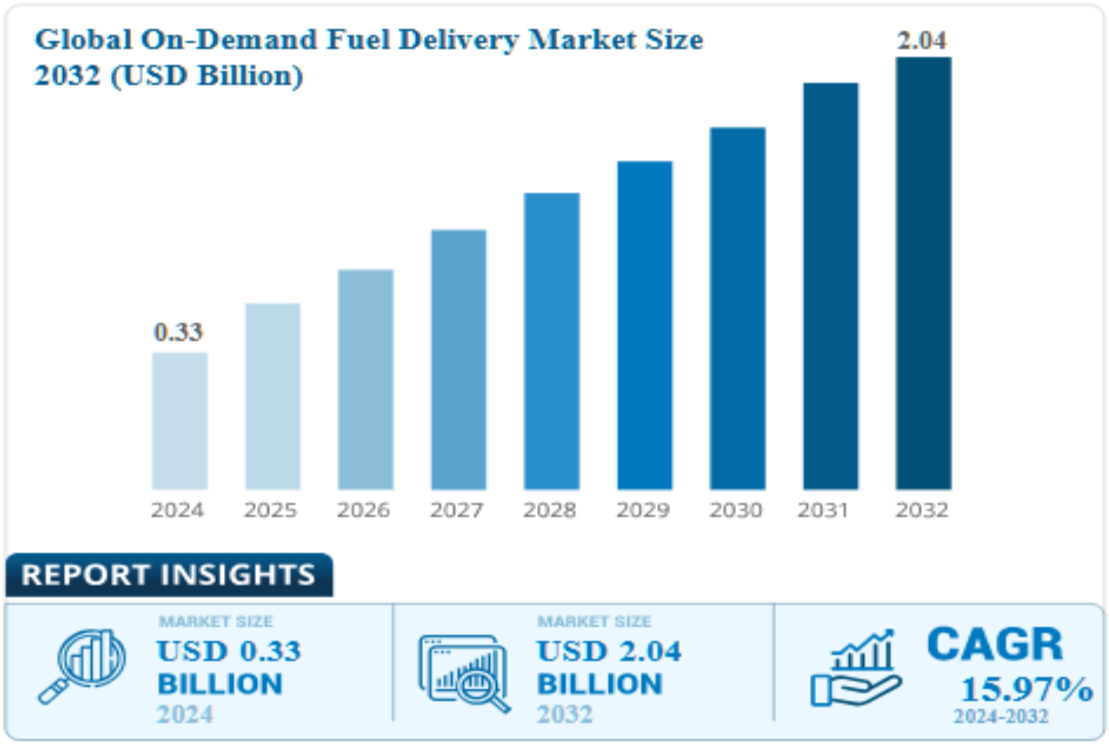



Figure 2

## 1.2 Technologies in Mobile App Development

The development of mobile applications for apps like fuel delivery requires robust and flexible technological solutions. Cross-platform development tools like Flutter are very popular because of their ability to reduce development costs and time. Flutter is an excellent framework for developing such applications with a single codebase for both iOS and Android.

Google has defined Flutter as a portable UI toolkit for developing beautiful looking natively compiled mobile apps from a single codebase [9] . A flutter project is written in Dart programming language, at the top level, Flutter provides widgets to work with which are used to create interface objects that we see usually on iOS and Android. Dart language is also faster as compared to other languages like JavaScript and it supports AOT and JIT compilation which is why it is perfect for the hot reload feature in Flutter [9] . Flutter comes with pre-made packages like Cupertino for iOS and Material for Android and these packages are used for creating the UI as they contain the interface elements which are specific to the themes of iOS and Android [9] . A survey was conducted in which 100 developers from different IT companies participated, these developers had 6 months to 8 years of experience and according to them in this survey, they thought that Flutter was good in terms of performance as compared to other technologies, 51.21 percent of the developers were in favour of Flutter [10] . These developers also claimed that Flutter has less learning curve than Java and Kotlin. The best feature in Flutter is the hot reload, this provides the ability to keep the app running and quickly reflect any changes made in the code at runtime without losing state on emulators or any hardware, in addition to this, Flutter also comes with a rendering engine and some development tools that help accelerate application development [11] .

Firebase provides a reliable Backend-as-a-Service (BaaS) solution that eases tasks like user authentication, database management, and real-time data synchronization. Firebase is a web application platform which is designed to aid developers in building mobile applications, it stores data in JSON format which doesn't use querying for any action and it is used as the backend system with many services available. One of these services is Firebase Authentication, this is a service that can authenticate users to create accounts and log-in and this comes with a user management system as well. Another service that Firebase provides is the real-time Firestore database [12] .

## 1.3 Real-Time GPS Tracking (Google Maps API)

Real-Time GPS Tracking is an important part of on-demand fuel delivery services. By integrating Google Maps API, mobile apps can offer accurate location tracking which will allow delivery agents to locate customers easily to deliver the fuel and the API can also be used to provide real-time updates on the delivery status.

There was a study conducted which demonstrated the use of GPS services to monitor the live locations of garbage trucks, displaying their real time locations on smartphones using Google Maps API. The app for this was developed in Android Studio and the approach was similar to what should be in a fuel delivery application. In the study, the mobile app gathers the coordinates of the garbage trucks and then displays it to the users using Google Maps API. This allows for the users to view the vehicle's location in real time. The location information is presented as latitude and longitude and it is collected through GPS services embedded within the garbage trucks. The Google Maps API plays an important role in this by displaying the vehicle's current live location and route, making it easier for the users to see where the truck is [13] .

Similar to the study mentioned above, GPS services embedded in smartphones can be used to track the live location of fuel delivery trucks or their delivery agents using their smartphone's GPS capabilities and this implementation will allow the users to view the live location of the delivery vehicle using an integrated Google Maps interface.

## 1.4 Conclusion

The literature reveals that on-demand fuel delivery services are poised to transform the way consumers access fuel. Advancements in mobile application development, combined with technologies such as Flutter, Firebase and Google Maps API can be leveraged to develop a robust and scalable Fuel Delivery App. The analysis and comparison of existing solutions and technologies highlight that innovation in this sector relies heavily on providing the best experience through use of modern technologies.

# Chapter 2: **Project Planning and Methodology**

This section outlines the strategic approach and timeline for developing the fuel delivery mobile application. It includes a detailed project plan, covering each key milestone, as well as the methods and practices used throughout the development process so far. The chosen methodology is designed to accommodate changing requirements while keeping a focus on delivering value to users and meeting project objectives within the specified time-period.

## 2.1 Development Methodology

The development of my Fuel Delivery Mobile Application follows an Agile methodology, which provides a flexible approach to software development. This methodology is well suited for this purpose as it allows for iterative improvements, incremental feature additions, and continuous adaptation based on user/supervisor feedback. Agile methodology was chosen for its ability to manage the changing requirements effectively, in a project like this where new features are tested and integrated regularly, Agile provides the framework to breakdown the development process into smaller sub-tasks which are manageable. Agile software development shows that developers should start with simple and predictable approximations to the final requirements and then continue to implement and increment the details of these requirements overtime [14] .

The project so far has been developed using iterative and incremental approach as the initial iterations focused on building the core functionalities of the mobile app, such as user authentication, user dashboard and the basic user interface. With time, more advanced features were implemented such as Google Maps API integration and fuel ordering process and system. All of this ensures that every feature is properly functional before moving onto the next one which helped in minimizing bugs and maintaining stability.

Frequent testing is also an important part of the development stages, every feature was properly tested on a virtual android device (emulator) to ensure a smooth user experience. The development process has also been very heavily user-centric to make sure that the user experience is intuitive and with enhanced convenience. All the features so far were implemented with the end-user in mind so that the best product is delivered.

## 2.2 Project Timeline

Term 1:

- Week 1: Setup the Flutter development environment for iOS and Android and research similar apps.
- Weeks 2–3: Working on the app design, finalizing the main features and interactions of the app, implementing firebase authentication, and building the UI for placing fuel orders.

- Week 4: Design database schema, some backend planning like using APIs etc.
- Weeks 5–6: Integrating Google maps for user location.
- Weeks 7–8: Testing for core features and remaining development from previous weeks.
- Weeks 9–10-11: Preparing and working on the interim report and presentation, any bug fixes or quality of life improvements.

Term 2:

- Weeks 1–2: Implementing push notifications and real time GPS updates, Integrating secure payment system.
- Weeks 3–4: Developing interface for delivery drivers/admins and developing user profiles (account management) + Implementing fuel order delivery tracking screen for users.
- Weeks 5–6: Any remaining development and testing the performance of the app.
- Weeks 7–8: User Acceptance Testing for feedback.
- Week 9: Deployment of the app if required.
- Weeks 10–11: Final testing, bug fixes, preparing and working on the final report and final submission.

## 2.3 Features Prioritization: MoSCoW Method

The MoSCoW method is an effective prioritization technique used in project management to determine the importance of several features and requirements. For the Fuel Delivery app, this method was used to categorize the features into two levels of importance: Must Have and Should Have. This ensures that the most important features of the application are developed first and this also contributes to the Agile methodology being used.

Must Have Features:

These are the core features that are required for the app/project to function properly as intended:

- User Registration and Login: Users must be able to create accounts and log in to place fuel orders
- Fuel Ordering System: The core feature where users select the fuel type, add all their details, select their delivery location, and place the order
- Real Time GPS Tracking: Users should be able to track the fuel delivery vehicle/driver in real time.
- Secure Payment Integration: A payment gateway must be implemented to deal with user transactions for placing orders.
- Google Maps Integration: This should be implemented to allow location selection and navigation for delivery.

- Notifications: Users should receive real time notifications about their order status and ETA.

- Driver Interface: A seperate interface for the fuel provider drivers to update and manage the fuel deliveries.

Should Have Features:

These are also important but not critical features, these will enhance the user experience or add value to the project:

- Push Notifications: Implementing push notifications to notify users about their order updates.

- Order History: Users should be able to view their past orders and details related to these orders.

- User Profile Management: Allow users to manage and update their personal details, payment methods and other details.

## 2.4 Risk Assessment and Mitigation Strategies

- Hardware Failure: While working on my mobile app, there is a possibility that the hardware on which I am doing the project might fail for some reasons leading to data loss and the progress made being wasted. To counter this, I will make sure that I am working on the project and keeping everything related to it under version control (Gitlab) and I will make sure that I am committing and pushing regularly to the repository.

- Uneven Balance Between Report and Code: Focusing too much on either the technical part (coding) or the written reports could lead in one of them being incomplete in time or insufficient in detail. To counter this, I will regularly review progress for both to make sure that both are receiving balanced attention, and I will make sure to follow my timeline and milestones to prevent this from happening.

- Technical Issues with GPS Tracking: Real time GPS Tracking may be inaccurate or may fail sometimes, to mitigate this I will properly integrate Google Maps with error handling.

- Cross-Platform Compatibility Issues: There are differences between apps for iOS and Android which could lead to inconsistent behaviour of the app or performance issues. For this, I will test frequently on both platforms and use Flutter as best as I can.

- Scope Creep: A project’s goal or requirements can change beyond what was originally planned upon, this can happen by adding too many features which could lead to missing deadlines. For this, I will stick to the original plan and use agile principles to manage priorities.

- Insufficient Testing: Poor testing during or after production might result in unfound bugs or poor user interface. For this, I will regularly test my product during development and afterwards as well.

- Time Management Issues: Delays in successfully achieving the milestones mentioned above in the timeline section could push the project beyond the given deadlines, so, to reduce this risk I will strictly follow the timeline, regularly review the progress, and break tasks into smaller sub-tasks which are manageable and easier to implement.

- Lack of Required Skills: It is possible that while working on this project, I might need to deal with new programming languages or technologies which I have not used before. For this, I will make sure that I also schedule time for learning these new skills by researching and doing.

# Chapter 3: **System Design**

The system design for the fuel delivery mobile app aims to balance usability, scalability, and functionality. This section outlines the design choices and UI elements, supported by a low fidelity storyboard that showcases the core features and user journey for the UX/UI of the project.

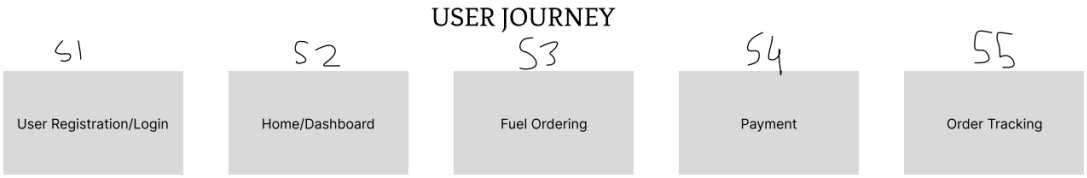



Figure 3

The user journey for the user experience is broken down into five main steps as shown in Figure 3 above:

1. User Registration/Login: The starting point for the users to register or log in with their email and password.

2. Home/Dashboard: After logging in, users access the dashboard to initiate fuel orders.

3. Fuel Ordering: Users provide details like delivery location, fuel type, and vehicle information to order fuel.

4. Payment: Users pay for their fuel order.

5. Order Tracking: After the order is successfully placed, users track their delivery using real-time GPS tracking.

The storyboard for this User Journey (Figure 4 below) showcases the low fidelity user interface design of the mobile app, highlighting the flow and functionality of each screen:

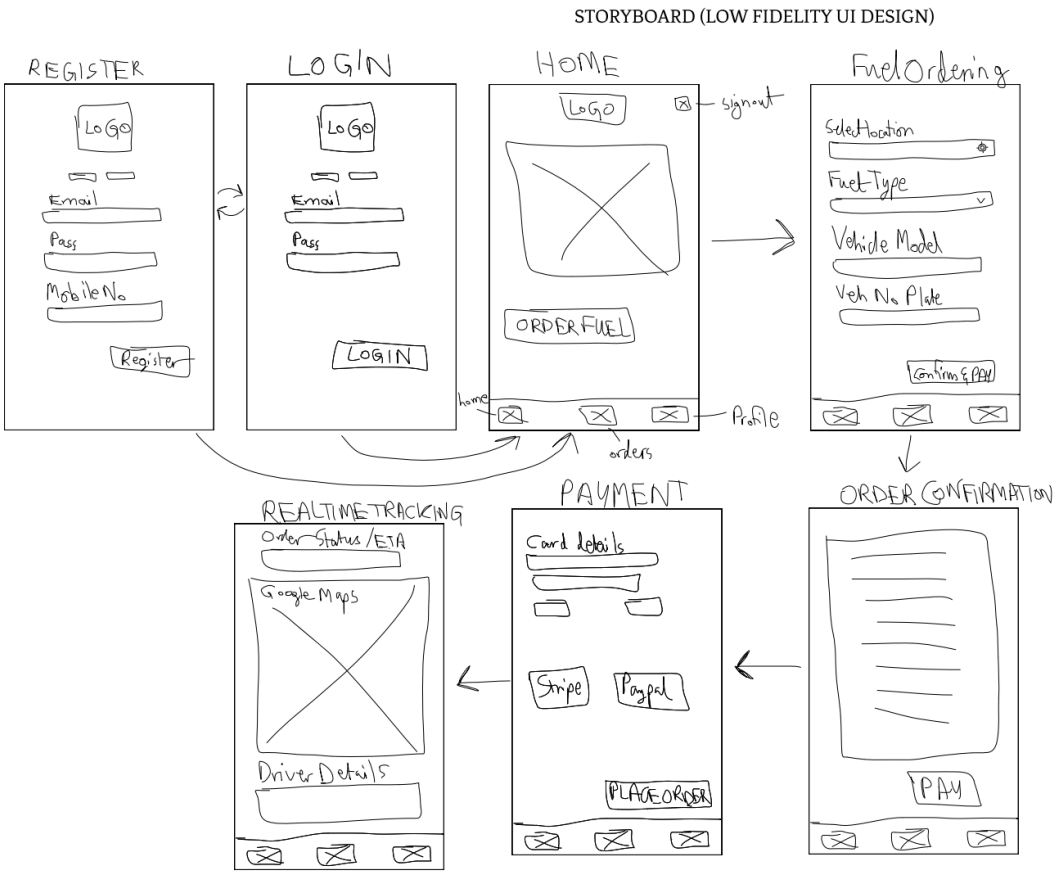



Figure 4

1. Register Screen: This allows the users to create an account by providing their email address, password, and mobile number. The screen is minimalistic, focusing on easy usage with clearly labelled input fields and prominent buttons.

2. Login Screen: Has input fields for users to log in with their registered details.

3. Home Screen: This is the central hub of the app, shows an “Order Fuel” button which allows the users to initiate the fuel ordering process. The home screen also includes navigation to other sections of the app like orders and profile.

4. Fuel Ordering Screen: This screen collects key details for the user’s order.

5. Order Confirmation Popup: This popup displays the summary of the user’s order, users can confirm then go to the next payment screen.

6. Payment Screen: This screen provides the users for using payment gateways like PayPal or Stripe to pay for their fuel orders.

7. Real Time Tracking Screen: This shows the user the status of their delivery and live location of the delivery agent/truck on a map. Users can also see ETA and driver details etc.

The storyboard provided has significantly informed the design considerations for the mobile app, key considerations include:

1. The apps UI design focuses on simplicity and clarity, which ensures that the users can navigate through the app easily. Each screen has distinct buttons and content which reduces cognitive load on the users.

2. A consistent visual and functional design across all screens.

3. Navigation is streamlined with a bottom navigation bar on the home screen which ensures the best user experience.

4. The storyboard also accounts for user feedback mechanisms such as confirmation popups and notifications.

This storyboard has played a very important role in developing and building the actual user interface of the app and by visually representing this, it has provided a clear framework for development while ensuring alignment with the project's objectives.

The system design for the mobile app implements main features of user-centred design (UCD), this emphasizes the importance of understanding user's needs, tasks, and goals to create the UI that aligns with their expectations [15] . To achieve this, the UI of the app has been designed with focus on usability and simplicity, ensuring that the users can successfully complete core tasks with minimal effort and cognitive load. By staying true to the design principles, the app aims to provide the best possible user experience, for example, the UI elements for navigation and interaction are visually consistent throughout the app to reduce confusion and build familiarity. In addition to this, the addition of real-time feedback mechanisms and order confirmation dialog enhances user trust and engagement. Hence, the app's design has been made in such way that it addresses the user's pain points and ensuring it meets both the functional requirements and aesthetic expectations by drawing on the best UCD practices in prototyping.

While this section focuses on the conceptual and low fidelity design for the app, the actual implementation of the UI with screenshots of the developed UI/app, is covered in the Preliminary Results section later. This ensures a clear distinction between the design process and the actual outcome achieved during development.

# Chapter 4: **Preliminary Results**

The SwiftFuel mobile application has made some major progress so far and the preliminary results highlight the milestones achieved so far, focusing on core features, user experience, and the integration of backend system.

The project started with extensive research to determine the app's purpose, platform, and functionality. After evaluating potential markets, it was decided to build a fuel delivery app targeting Android devices initially. This decision was later expanded to include iOS as well using Flutter. The development environment was setup which included the Flutter project folders and structure, Firebase, and Android studio IDE. All of this ensured that the app had a solid foundation before the development actually started.

In the early stages of development, a simple "Hello World" screen was developed using Flutter to familiarize with how Flutter works and its functionalities. After this, a low-fidelity storyboard of the app's UI was created using Figma software (as talked about in the previous chapter), this storyboard outlined the app's overall design and interactions.

With the foundational setup complete, Firebase authentication services were implemented to handle user registration and login processes. Then, login and registration screens were built and coded with proper UI and functionality. The app was officially named "SwiftFuel", reflecting its purpose and mission.

Efforts were also made to improve the app's UI by researching Flutter Flow templates for inspiration. This led to the development of the home/dashboard screen and the integration of the register, login, and home screens was achieved after this. In addition to this, the Firebase Firestore database was setup to store the user data, including mobile numbers entered during the registration process. Snackbar notifications were also implemented across the app to enhance user feedback during the interactions.

Some graphical content was added to the home screen to enhance its visual appeal. A major milestone was the successful integration of Google Maps API into the app, enabling the fuel ordering feature. The fuel ordering screen was also developed, which allowed the users to select their current location or pin a location on the map, choose the fuel type they want, and enter their vehicle number plate to place a fuel order.

Key functionality enhancements followed, including the implementation of a back button on the fuel ordering screen for easy navigation to the home screen. The fuel ordering system was also integrated with Firebase authentication to ensure secure user data handling and proper usage of the fuel ordering system. An order confirmation popup was added which allowed users to review their fuel order details before placing the order.

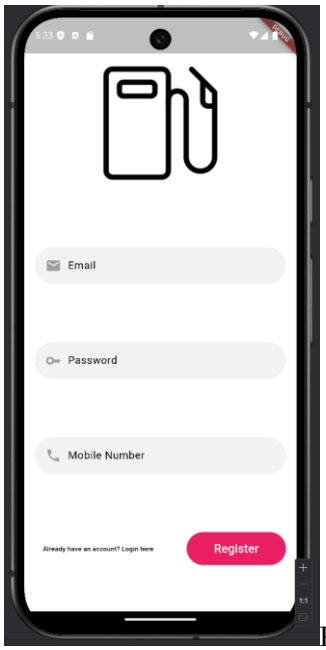
To validate the app's core functionalities, rigorous testing was also conducted during the development phase so far for all the implemented features. The registration and login screens were tested to ensure perfect user authentication with Firebase. The Google Maps integration for location selection was also tested to verify the accuracy of the location tracking and map rendering. Functional tests were regularly conducted on the fuel ordering screen to ensure that users could pin locations, select their current location and submit their orders after confirmation without any errors. Navigation between the screens was also tested for user-friendly transitions.

Several bug fixes were done to improve app stability and performance. Additionally, a detailed README file was created to guide users on how to deploy and run the main code of the app. These results demonstrate that significant progress has been made so far in both technical and design aspects of the project. The project is on track with the foundation laid for remaining development.

**Challenges faced during development so far:**

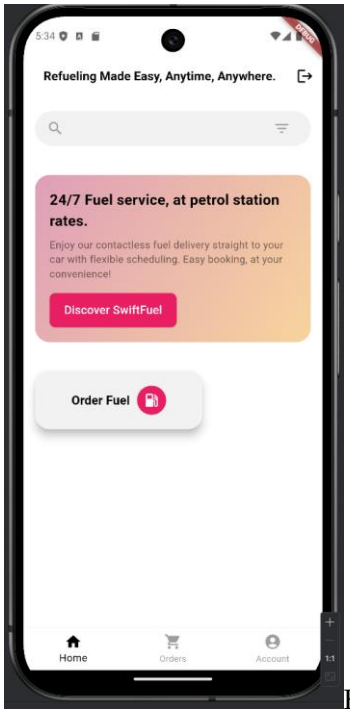
Several challenges were also faced and resolved during the development of the app, such as Firebase rule misconfigurations which in the start prevented the app from properly saving data to the Firestore database. These problems were debugged by reviewing the Firestore security rules, making sure that only authenticated users could interact with the database. In addition to this, integration the Google Maps API presented some difficulties at first, which included map rendering failures and incorrect location initialisation. These issues were also fixed by properly setting up the API key and verifying project configurations.

To showcase the progress made so far for the development of the app, the following screenshots showcase the app's core features and user interface design so far, each screen reflects the key milestones achieved so far:

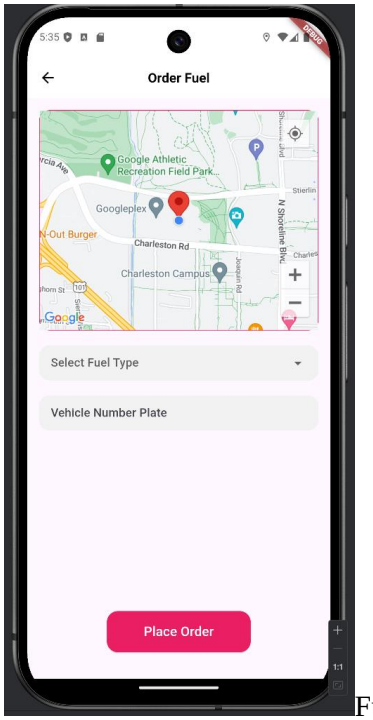


Register

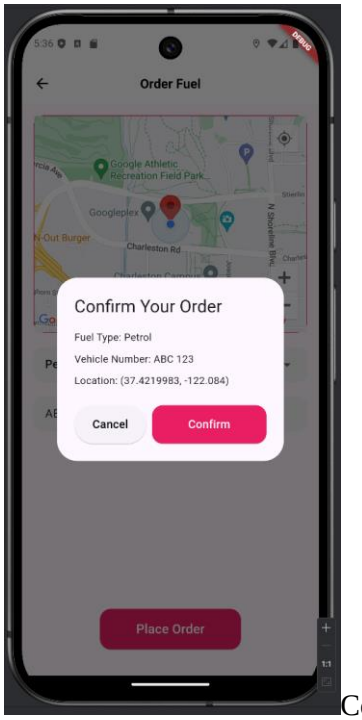
Login



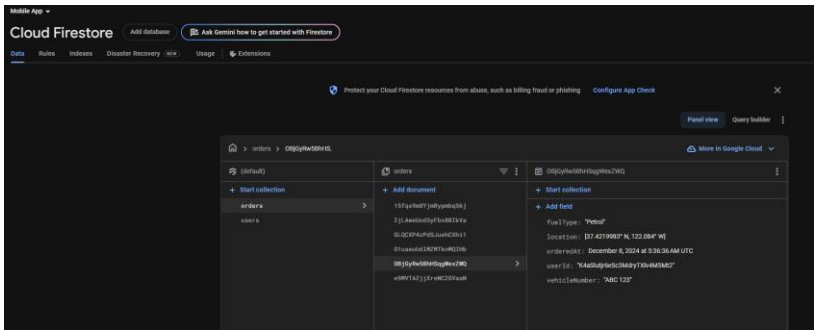

Home




Fuel Ordering




Confirmation




Backend Database

The screenshots demonstrate the app's design aesthetics and provide evidence of technical implementation such as Firebase integration, Google Maps feature, and secure user authentication. The designs from the storyboard were instrumental in shaping the current UI, demonstrating how they were translated into actual and functional screens. This highlights the connection between the design planning and practical execution of the project.

# Chapter 5: **Conclusion**

The development of the SwiftFuel mobile application represents a major step towards modernizing refuelling services, offering customer's convenience and efficiency. This report has outlined the progress made so far, which includes planning, system design, literature review, and the successful implementation of the Term 1 milestones. By using modern technologies like Flutter, Firebase, and Google Maps API, the project has set a strong foundation for future work.

The literature review section highlighted the increasing relevance of on-demand fuel delivery services. Market analysis and existing solutions like CAFU underscored the potential for innovation in this sector, while the exploration of technologies like Flutter showed their ability to develop robust mobile apps. The system design and project planning sections provided careful insights into the planning and user centred approach taken for the project. The preliminary results section showed the transformation of early concepts into proper results.

Key milestones achieved during Term 1 include the implementation of user authentication, implementation of Google Maps API for location selection, and the main fuel ordering system, all of which align with the project's main objectives. Moving forward, the focus will be on completing the remaining work which is planned for Term 2 and also further refining and polishing the existing app with quality-of-life updates. The progress made so far reflects the projects alignment with its objectives, and the methodology used has ensured a structured and effective approach to building the app.

SwiftFuel is perfectly positioned to achieve its final goal of delivering a reliable and user-friendly solution for delivering fuel to anyone, anytime, anywhere. The remaining development will build upon what has already been developed to deliver a final product that meets and exceeds the expectations.

# Bibliography

[1] Mishra, N., Raghuwanshi, R., Maurya, N.K. and Kumar, I., 2023, August. Efficient Fuel Delivery at Your Fingertips: Developing a Seamless On-Demand Fuel Delivery App with Flutter. In International Conference on Cognitive Computing and Cyber Physical Systems (pp. 134-147). Cham: Springer Nature Switzerland.

[2] Kumar, A., Surya, S., Keerthiga, A. and Shalini, S., 2024. FUEL PULSE: Digital Fuel Logistics Revolutionizing Delivery Management. Available at SSRN 4776337.

[3] Prabhu, M. and Sarker, Y. . (2023) "On-Demand Fuel Delivery Mobile App for OmanOil", Journal of Student Research. Houston, U.S. Available at: https://www.jsr.org/index.php/path/article/view/2317 (Accessed: 9October2024).

[4] Gunthe, S., Sangale, A., Brahmankar, Y., Kulkarni, P. and Baddi, P., 2023. Fuel Delivery Application. International Journal for Research in Applied Science and Engineering Technology (IJRASET), 11(5), pp.1-7. Available at: https://doi.org/10.22214/ijraset.2023.52655.

[5] On-Demand Fuel Delivery Apps Development Explained Blog, Available at: https://easternpeak. com/blog/on-demand-fuel-delivery-apps-development/.

[6] Esferasoft Solutions. *How to build An On-Demand Fuel Delivery App Like CAFU in 2022*. Retrieved from https://www.esferasoft.com/blog/cafu-on-demand-fuel-delivery-app-business-model-development-cost/

[7] Wamda. (2023). *CAFU on the road to world domination*. Retrieved from https://www.wamda.com/2023/07/cafu-road-world-domination

[8] Business Research Insights (2024) 'On-Demand Fuel Delivery Market REPORT OVERVIEW', *Business Research Insights*, 18 November. Available at: https://www.businessresearchinsights.com/market-reports/on-demand-fuel-delivery-market-102906

[9] Dagne, L. (2019) 'Flutter for cross-platform App and SDK development', *Bachelor's Thesis*, Metropolia University of Applied Sciences, 01 May 2019. Available at: https://www.theseus.fi/bitstream/handle/10024/172866/Lukas%20Dagne%20Thesis.pdf

[10] Hussain, H., Khan, K., Farooqui, F., Arain, Q.A. and Siddiqui, I.F. (2021) 'Comparative Study of Android Native and Flutter App Development', *Proceedings of the 13th International Conference on Internet (ICONI) 2021*, December. Available at: https://www.researchgate.net/profile/Hina-Hussain-2/publication/361208165_Comparative_Study_of_Android_Native_and_Flutter_App_Development/links/635a857c12cbac6a3efff6b6/Comparative-Study-of-Android-Native-and-Flutter-App-Development.pdf

[11] Fentaw, A.E. (2020) 'Cross platform mobile application development: a comparison study of React Native vs Flutter', *Master's Thesis*, University of Jyväskylä. Available at: https://jyx.jyu.fi/handle/123456789/70969

[12] Khawas, C. and Shah, P. (2018) 'Application of Firebase in Android App Development—A Study', *International Journal of Computer Applications*, 179(46), pp. 49–53. Available at: https://www.researchgate.net/profile/Chunnu-Khawas/publication/325791990_Application_of_Firebase_in_Android_App_Development-A_Study/links/5bab55ed45851574f7e6801e/Application-of-Firebase-in-Android-App-Development-A-Study.pdf

[13] Kassim M et al. (2019) 'Mobile Application on Garbage Collector Tracker Using Google Maps', *Proceedings of the 2019 IEEE 9th International Conference on System Engineering and Technology (ICSET)*, Shah Alam, Malaysia, 7 October. Available at: https://ieeexplore.ieee.org/document/8906374

[14] Kumar, G. and Bhatia, P.K. (2012) 'Impact of Agile Methodology on Software Development Process', *International Journal of Computer Technology and Electronics Engineering*, 2(4), pp. 46–50. Available at: https://www.researchgate.net/profile/Gaurav-Kumar-175/publication/255707851_Impact_of_Agile_Methodology_on_Software_Development_Process/links/00b49520489442e12d000000/Impact-of-Agile-Methodology-on-Software-Development-Process.pdf

[15] Stone, D., Jarrett, C., Woodroffe, M., and Minocha, S. (2005) *User Interface Design and Evaluation*. San Francisco: Morgan Kaufmann. Available at: https://books.google.co.uk/books?id=VvSoyqPBPbMC

THE END