

# **SYMBIOSIS UNIVERSITY OF APPLIED SCIENCES**

## **INDORE**



### **PROJECT REPORT**

#### **ON**

**“On-Demand Fuel Delivery Application Built using Flutter”**

Submitted to “Symbiosis University of Applied Sciences, Indore  
As a Project report for the partial fulfillment of the award of degree of

### **BACHELOR OF TECHNOLOGY**

#### **IN**

### **SCHOOL OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY**

Submitted To:  
Dr. Indrajeet Kumar  
Assistant Professor

Submitted By:  
Navneet Mishra 2019BTCS050  
Ritika Raghuwanshi 2019BTCS066

# **SYMBIOSIS UNIVERSITY OF APPLIED SCIENCES**

## **INDORE**

### **CERTIFICATE**

This is to certify that the project report entitled “On-Demand Fuel Delivery Application Built Using Flutter”, submitted by Navneet Mishra and Ritika Raghuwansi, students of final year towards partial fulfillment of the degree of Bachelor of Technology in School of Computer Science and Information Technology in year 2022-2023 Symbiosis University of Applied Sciences, Indore (M.P.)

Place: Indore

Date:

INTERNAL EXAMINER

EXTERNAL EXAMINER

# **SYMBIOSIS UNIVERSITY OF APPLIED SCIENCES**

## **INDORE**

### **RECOMMENDATION**

The work entitled “ On-Demand Fuel Delivery Application”, submitted by Navneet Mishra and Ritika Raghuwansi, student of final year Computer Science and Information Technology, towards the partial fulfillment for the award of degree of Bachelor of Technology in Computer Science and Information Technology of Symbiosis University of Applied Sciences Indore(M.P.) is a satisfactory account of their Final Project and is recommended for the award of the degree.

Endorsed By:

Dr. Neha Gupta

Director I/C, SCSIT

## **Student Undertaking**

I hereby undertake that the project work entitled “On-Demand Fuel Delivery Application Built Using Flutter” has been carried out by me from the period Jan – June 2023 and the report so prepared is a record of work done by me during my internship. I further declare that I have completed the internship in accordance with the Internship policy of the University. This Project report is submitted towards fulfillment of my academic requirement and not for any other purpose.

I hereby undertake that the material of this Project is my original work and I have not copied anything from anywhere else. The material obtained from other sources has been duly acknowledged. I understand that if at any stage, it is found that I have indulged in any malpractice or the project and the project report has been copied or not completed by me, the university shall cancel my degree/withhold my result and appropriate disciplinary action shall be initiated against me.

Navneet Mishra:

Ritika Raghuwansi:  
**Student Name and Signature**

Dr. Indrajeet Kumar  
**Mentor Name & Signature**

**Enrollment Number:**

2019BTCS050

2019BTCS066

Dr. Neha Gupta

**Date:**

**Head of School Name & Signature**

# **SYMBIOSIS UNIVERSITY OF APPLIED SCIENCES**

## **INDORE**

### **ACKNOWLEDGEMENT**

The successful completion of any work is generally not an individual effort. It is an outcome of dedicated and cumulative efforts of a number of people, each having its own importance to the objective. This section is a value of thanks and gratitude towards all those persons who have implicitly or explicitly contributed in their own unique way towards the completion of the project. For their invaluable comments and suggestions, I wish to thank them all.

Positive inspiration and right guidance are must in every aspect of life. Especially, when we arrive at academic stage for instance. For the success of our project a number of obligations have been taken. We have performed solemn duty of expressing a heartfelt thanks to all who have endowed us with their precious perpetual guidance, suggestions, and information. Any kind of help directly or indirectly has proved importance to us.

# CONTENTS

<b>Chapter 1: INTRODUCTION.....</b>	<b>Page 1</b>
1.1 Introduction	
1.2 Literature Review	
1.3 Problem Definition	
<b>Chapter 2: THE PROJECT.....</b>	<b>Page 11</b>
2.1 Project definition	
2.1.1 Objective	
2.1.2 Project scope	
<b>Chapter 3: REQUIREMENTS ANALYSIS.....</b>	<b>Page 15</b>
3.1 Functional Requirements	
3.2 Non-Functional Requirements	
3.3 Use-Case specification	
3.3.1 Find actors	
3.3.2 Documents actors	
3.3.3 Find Use Case	
<b>Chapter 4: DESIGN.....</b>	<b>Page 18</b>
4.1 Database Design	
4.1.1 ER Diagram	
4.2.2 Design Tables and Normalization	
4.2 Activity diagram	
4.3 Architecture and Flow Diagram	
4.3 Class diagram	
<b>Chapter 5: EXPERIMENT AND TESTING.....</b>	<b>Page 26</b>
5.1 Test cases developed.	
5.2 Testing used in our project.	
<b>Chapter 6: CONCLUSION .....</b>	<b>Page 29</b>
6.1 Problems and Issues in currents system	
6.2 Future extension	
<b>APPENDIX.....</b>	<b>Page 32</b>
<b>BIBLIOGRAPHY.....</b>	<b>Page 52</b>

## **I. INTRODUCTION**

### **1.1 Introduction**

Our fuel delivery application, developed using Flutter, offers a convenient on-demand service that sets us apart from other delivery apps. Our primary objective is to provide customers with the ease of having fuel delivered directly to their location. With the rapid increase in demands in today's fast-paced world, we strive to modernize the fuel delivery industry through our user-friendly and secure On-Demand Fuel Delivery Application.

In this modern era where people rely heavily on mobile apps for various services, we recognized the need to extend the convenience of on-demand delivery to the fuel sector. Our application is designed to cater to the busy lifestyles of individuals, ensuring they no longer must take time out of their schedules to visit gas stations. By bringing fuel delivery right to their doorstep, we aim to save our customers valuable time and effort.

Through our app, customers can easily request fuel delivery at their preferred time and location. They can select the type and quantity of fuel they require, making the process quick and personalized. Our platform utilizes GPS technology to precisely track the customer's location, ensuring accurate delivery.

To enhance safety and reliability, we have implemented rigorous quality control measures in our fuel sourcing and delivery processes. We partner with trusted fuel suppliers who adhere to strict safety standards. Our delivery personnel are trained to handle fuel safely and efficiently, minimizing any potential risks.

In addition to the convenience of on-demand fuel delivery, our application provides a comprehensive online ordering experience. Customers can browse fuel prices in their nearby area, allowing them to make informed decisions and compare prices. The transparency we offer ensures that our customers are getting competitive rates and are well-informed about the fuel market.

Looking ahead, the scalability potential of on-demand fuel services is enormous. We can target various end-users, including individuals, businesses, and industries that heavily rely on fuel for their operations. For example, transportation companies, construction sites, and power generation facilities can greatly benefit from our on-demand fuel delivery solution.

Furthermore, our application contributes to environmental sustainability by optimizing fuel consumption and reducing carbon emissions. By eliminating the need for customers to drive to gas stations, we minimize traffic congestion and promote a greener approach to fuel procurement.

Through our innovative app, we aim to provide timely fuel delivery and meet the increasing expectations of customers. By leveraging the power of Flutter, we have created a robust platform that ensures a seamless user experience. With our On-Demand Fuel Delivery Application, customers can easily fulfill their fuel requirements with just a few taps on their smartphones, revolutionizing the way fuel is accessed and consumed. In addition to our commitment to providing timely fuel delivery and a seamless user experience, we understand the importance of building strong partnerships within the fuel industry. By collaborating with fuel suppliers, gas stations, and fleet management companies, we aim to create a comprehensive network that optimizes fuel sourcing, distribution, and logistics.

Through these partnerships, we can ensure a reliable and uninterrupted fuel supply chain. By working closely with fuel suppliers, we can establish strategic agreements that prioritize our customers' needs and always guarantee the availability of high-quality fuel. This not only enhances the convenience of our service but also instills confidence in our customers that they can rely on us to meet their fuel requirements.

Moreover, our application serves as a platform for gas stations to expand their customer base and reach a wider audience. By partnering with local gas stations, we provide them with an opportunity to connect with customers who prefer the convenience of on-demand fuel delivery. This mutually beneficial collaboration strengthens the fuel ecosystem and contributes to the growth of local businesses.

For fleet management companies, our On-Demand Fuel Delivery Application offers a significant advantage in optimizing their operations. Instead of dedicating resources and time to refueling their vehicles at gas stations, they can rely on our platform to handle their fuel needs efficiently. This frees up their workforce to focus on core business activities, leading to improved productivity and cost savings.

Furthermore, our app opens possibilities for innovative fuel management solutions. By leveraging data analytics and machine learning, we can provide insights and recommendations to customers on optimizing their fuel usage. This empowers individuals and businesses to make informed decisions about their fuel consumption, leading to greater efficiency and reduced environmental impact.

As we continue to grow and expand our services, we are committed to staying at the forefront of technological advancements. We are continuously exploring new features and functionalities to enhance the user experience. This includes incorporating emerging technologies such as Internet of Things (IoT) for smart fuel monitoring, integration with digital payment systems for seamless transactions, and further advancements in location tracking for precise and efficient deliveries.



In summary, our On-Demand Fuel Delivery Application, powered by Flutter, not only revolutionizes the way fuel is accessed and consumed but also drives collaboration within the fuel industry. By establishing strong partnerships, optimizing the fuel supply chain, and embracing technological advancements, we are poised to reshape the fuel delivery landscape and meet the evolving needs of our customers and partners.

**Fuel Delivery App Development Statistics – Facts & Figures-**

- During the forecast period of 2022-2032, the mobile fuel delivery market is likely to increase at a CAGR of 6.8%.
- In 2022, this market is expected to reach around \$4.8 billion.
- The mobile fuel delivery market value will likely be \$6.2 billion by 2026.
- On-demand app revenue is likely to generate \$935 billion in 2023.

## 1.2 Literature Review

- **"All India Survey on Diesel and Gasoline Demand by Sector" from the Ministry of Oil and Gas of India. (2019-2020)**

The Ministry of Oil and Gas of India published a report on the "All India Survey on Diesel and Gasoline Demand by Sector" which outlines the consumption of diesel and gasoline in various sectors. The transportation sector is the largest consumer at 60%, followed by agriculture and industry. The report emphasizes the need to reduce dependency on fossil fuels and promote sustainable alternatives for a greener economy.

- **"Demand for Road Fuel in Small Developing Countries" by Sunil Chandrasiri 2016**

The research paper "Demand for Road Fuel in Small Developing Countries" [2] by Sunil Chandrasiri examines the challenges of meeting the growing demand for road fuel in these countries driven by economic development and population growth. The study analyzes factors affecting demand and the effectiveness of policy interventions like fuel taxes and subsidies. The paper suggests a more comprehensive approach is needed to promote sustainable transportation that considers technological innovations and behavioral change. It offers recommendations for policymakers to address these challenges.

- **"Fuel Management System" by Areeg Abubakr, Siddig Ali published in the Institute of Electrical and Electronics Engineers (IEEE) Journal on January 2017**

The "Fuel Management System" paper [1] by Areeg Abubakr and Siddig Ali published in the IEEE Journal in January 2017 provides a review of various fuel management systems used in transportation, highlighting their importance in reducing fuel consumption and costs. The study covers the technical aspects of FMS, challenges in implementing FMS, and their potential benefits in the transportation industry.

- **"Long-term Forecast of Energy and Fuel Demand for Ecuador's Sustainable Road Transport Sector (2016- 2035): Applying the LEAP Model" by Luis Rivera Gonzalez, David Bolognio published in the MDPI Journal on Energy and Fuel Requirements for 2019.**

The research paper "Long-term Forecast of Energy and Fuel Demand for Ecuador's Sustainable Road Transport Sector (2016- 2035): Applying the LEAP Model" [4] by Luis Rivera Gonzalez and David Bolognio analyzes the use of the LEAP model to forecast energy and fuel demand in Ecuador's road transport sector between 2016 and 2035. The study evaluates the impact of different policy scenarios on energy consumption and greenhouse gas emissions and recommends sustainable transport policies to reduce emissions and improve energy efficiency. The research offers valuable insights and recommendations for policymakers to promote sustainable development.

- **"India's Oil Demand: Empirical Estimates and Future Forecasts by Pradeep Agarwal published at IEG University in Delhi in 2012.**

"India's Oil Demand: Empirical Estimates and Future Forecasts" [5] by Pradeep Agarwal is a research paper that examines the factors driving oil demand in India, the impact of policy interventions on oil demand, and recommendations for promoting energy conservation and renewable energy sources. The study suggests that India's oil demand will continue to grow due to economic development and urbanization, and offers insights and recommendations for policymakers.

- **"All India Study on Sectoral Demand of Diesel & Petrol" by Nielsen India Private Limited submitted in the Ministry of Petroleum and Natural gas in 2013.**

The "All India Study on Sectoral Demand of Diesel & Petrol" [3] by Nielsen India Private Limited is a report that examines the sector-wise demand for diesel and petrol in India, with a focus on key drivers of demand. The study provides insights for policymakers and industry stakeholders and contributes to a better understanding of the energy sector in India.

References	Title	Authors	Year	Main Area of Focus	Summary
1	Fuel management system	A. A. I. Ahmed, S. A. E. Mohammed, M. A. M. H. Satte	2017	Fuel management system	This paper discusses the design and implementation of a fuel management system for efficient fuel consumption in various applications, addressing challenges and presenting solutions.
2	Demand for road-fuel in a small developing economy	S. Chandrasiri	2006	Road fuel demand in small developing economies	The study examines the factors influencing road fuel demand in small developing economies, using Sri Lanka as a case study, and discusses the implications for energy policy and sustainability.
3	All India Study on Sectoral Demand of Diesel & Petrol	Nielsen India Private Limited	2013	Sectoral demand for diesel and petrol in India	This study provides insights into the sector-wise demand for diesel and petrol in India, contributing to a better understanding of the energy sector and offering valuable information for policymakers and industry stakeholders.
4	Long-Term Forecast of Energy and Fuels Demand	L. Rivera-González, D. Bolonio, G. A. García-López, M. Alvarez	2019	Long-term energy and fuel demand forecast	The paper applies the Long-range Energy Alternatives Planning (LEAP) model to forecast energy and fuel demand in Ecuador's road transport sector, evaluating different policy scenarios and

					providing recommendations for a sustainable road transport sector.
5	India's Petroleum Demand: Empirical Estimations and Projections	P. Agarwal	2012	Petroleum demand in India	The research paper presents empirical estimations and future projections for petroleum demand in India, focusing on the factors driving demand, policy interventions, and recommendations for energy conservation and renewable energy sources.
6	Fuel Management System	A. Abubakr, S. Ali	2017	Fuel management system	This study discusses various fuel management systems (FMS) used in transportation, highlighting their importance in reducing fuel consumption and costs. It covers technical aspects, challenges, and potential benefits of FMS in the transportation industry.
7	Assessment of fuel delivery system of a high-performance UAV engine	A. Rabinovich, Y. Azuri, L. Shtilman	2018	Fuel delivery system for high-performance UAV engine	The paper focuses on the assessment of the fuel delivery system for a high-performance unmanned aerial vehicle (UAV) engine, evaluating its efficiency and performance to ensure reliable and optimal fuel delivery for UAV operations.

8	Fault diagnosis of fuel delivery system for diesel engine	H. Gao, J. Liu, Q. Huang	2019	Fault diagnosis of fuel delivery system for diesel engine	The study proposes a fault diagnosis method for the fuel delivery system in diesel engines based on a dynamic Bayesian network. It aims to improve the reliability and performance of fuel delivery systems by detecting and diagnosing faults.
9	Design and simulation of a fuel delivery system for a variable compression ratio engine	K. Huang, S. Xie, X. Wang, L. Sun	2020	Fuel delivery system for a variable compression ratio engine	The paper presents the design and simulation of a fuel delivery system for an engine with variable compression ratio, aiming to optimize engine performance, fuel efficiency, and emissions control.
10	Design of a fuel delivery system for high-speed diesel engine	J. Wang, J. Liu, Q. Huang	2021	Fuel delivery system for high-speed diesel engine	This study focuses on the design of a fuel delivery system for a high-speed diesel engine using digital simulation technology. The research aims to improve the performance, reliability, and fuel efficiency of the engine through an optimized fuel delivery system design.
11	Fuel delivery systems for gasoline direct injection engines	T. M. Williams, J. M. Pearson	2018	Fuel delivery systems for gasoline direct	The paper discusses fuel delivery systems for gasoline direct injection engines, covering topics such as system

				injection engines	architecture, components, and optimization techniques. It addresses the challenges and advancements in fuel delivery systems for efficient combustion and reduced emissions in gasoline direct injection engines.
12	Modeling and control of a fuel delivery system for gasoline engines	N. P. Manh, H. G. Jeong	2017	Fuel delivery system for gasoline engines	This study focuses on the modeling and control of a fuel delivery system for gasoline engines, aiming to improve fuel efficiency, performance, and emissions control. It presents a control strategy and simulation results for an optimized fuel delivery system in gasoline engines.
13	Design and optimization of a fuel delivery system for a diesel engine	Y. S. Kuo, W. L. Chen	2016	Fuel delivery system for a diesel engine	The research paper presents the design and optimization of a fuel delivery system for a diesel engine using computational fluid dynamics (CFD) simulation and response surface methodology (RSM). It aims to improve the fuel injection process and optimize the performance of the diesel engine.

### **1.3 Problem Definition**

With the rapid increase in vehicles and transport facilities, the traditional model of relying solely on gas stations for refueling is becoming increasingly inconvenient, especially for individuals with busy schedules. To address this challenge, we propose an innovative solution: the On-Demand Fuel Delivery Application. This application aims to provide users with a convenient and hassle-free way to check fuel prices and order fuel at any location of their choice.

The On-Demand Fuel Delivery Application leverages the power of technology and mobile connectivity to revolutionize the fuel delivery experience. Users can easily access the application on their smartphones and with just a few taps, they can check the current fuel prices in their vicinity. This feature enables users to make informed decisions about their fuel purchases, ensuring they get the best value for their money.

Moreover, the application eliminates the need for users to visit a physical gas station. Instead, users can simply provide their current location, and the On-Demand Fuel Delivery Application will arrange for fuel delivery to their specified address. This means that users no longer must waste time and effort in finding a nearby gas station, queuing up for fuel, and dealing with the inconvenience of crowded stations.

The convenience of this app-based service is especially valuable for individuals with hectic lifestyles, such as professionals, parents, or anyone who wants to optimize their time and energy. By offering a seamless and efficient fuel delivery experience, the On-Demand Fuel Delivery Application aims to simplify the fueling process, allowing users to focus on their priorities without the added stress of refueling their vehicles.

In addition to its user-centric approach, the On-Demand Fuel Delivery Application also contributes to a more sustainable future. By minimizing unnecessary trips to gas stations, the application helps reduce traffic congestion and carbon emissions associated with vehicle movements. It aligns with the growing global emphasis on eco-friendly practices and supports the transition towards cleaner and greener transportation alternatives.

Overall, the On-Demand Fuel Delivery Application presents a modern and convenient solution to the challenges posed by traditional gas stations. By providing users with the ability to check fuel prices and order fuel from any location, this application redefines the fueling experience, saving users time, effort, and providing them with greater flexibility. It combines convenience, sustainability, and technological innovation to meet the evolving needs of today's busy individuals and contribute to a more efficient and environmentally conscious future.



## **II. THE PROJECT**

### **2.1 Project Definition**

The increasing demand for fuel in the market, driven by the growth of automobiles, has highlighted the need for innovative solutions to ensure a convenient and hassle-free fuel supply experience. Traditional methods of refueling, such as visiting gas stations, can be inconvenient, time-consuming, and challenging for certain individuals, including the elderly, those with medical conditions, or during emergencies.

To address these challenges and improve the fueling experience, we propose the development of an on-demand fuel delivery application. This application aims to provide a seamless and efficient solution for vehicle refueling and emergency power supplies.

By leveraging the power of technology and mobile connectivity, the on-demand fuel delivery application enables users to order fuel and have it delivered directly to their location. This eliminates the need for users to physically travel to gas stations, saving them time, effort, and providing a higher level of convenience. Whether it's refueling a vehicle on the go or supplying fuel for generators, the application caters to a wide range of fueling needs.

One of the key features of the application is its ability to offer a variety of fuel options. Users can choose the type of fuel they require, such as gasoline, diesel, or alternative fuels, based on their specific needs and vehicle requirements. This flexibility ensures that the application can cater to a diverse range of customers and fuel preferences.

In addition to the on-demand fuel delivery service, the application also provides users with information about available gas stations in their vicinity. Users can easily locate nearby gas stations, check their availability, and access real-time information about fuel prices and station amenities. This feature empowers users with the information they need to make informed decisions and optimize their fueling experience.

Furthermore, the on-demand fuel delivery application incorporates features that enhance user convenience and streamline the fueling process. Users can track their fuel orders in real-time, ensuring transparency and peace of mind. The application also offers secure payment options, allowing users to complete transactions seamlessly within the app.

The successful implementation of this project will result in an efficient and user-friendly on-demand fuel delivery application. Users will have the convenience of ordering fuel as per their requirements, eliminating the need to visit gas stations and providing door-to-door coverage. The application will also facilitate easy tracking of orders and access to information about available gas stations, empowering users with a comprehensive fueling solution.

### 2.1.1 Objective

The main objective of the On-Demand Fuel Delivery Application is to provide a convenient and efficient solution for users to get fuel for their vehicles delivered to their desired location or address. The application aims to simplify the fueling process by eliminating the need for users to physically visit gas stations and instead allowing them to make online orders for fuel delivery.

By incorporating user-friendly features and leveraging advanced technologies, the application strives to achieve the following objectives:

**Convenience:** The application aims to offer a seamless and hassle-free fueling experience for users. By allowing them to order fuel with just a few taps on their smartphones, the application eliminates the need for users to take time out of their busy schedules to visit a gas station. Users can simply register, provide their location, and place an order for fuel delivery, saving them time and effort.

**Accessibility:** The On-Demand Fuel Delivery Application aims to make fuel more accessible to a wider range of users. It addresses the challenges faced by individuals who may find it difficult to physically travel to a gas station, such as the elderly, people with medical conditions, or those with mobility constraints. By offering door-to-door fuel delivery, the application ensures that users can easily access fuel regardless of their location or physical limitations.

**Efficiency:** The application aims to streamline the fuel delivery process and optimize efficiency. By utilizing advanced logistics and route planning algorithms, it ensures that fuel deliveries are made in a timely manner. The application aims to minimize waiting times and provide users with a reliable and efficient fuel delivery service.

**Safety and Security:** The On-Demand Fuel Delivery Application prioritizes the safety and security of both users and fuel delivery providers. Stringent safety protocols are implemented to ensure that the fuel delivery process adheres to industry standards and regulations. Users can have peace of mind knowing that their fuel delivery is handled safely and securely.

**Transparency:** The application aims to provide users with transparent and real-time information regarding their fuel orders. Users can track the status of their deliveries, including estimated arrival times and delivery updates. This transparency enhances the user experience and instills confidence in the fuel delivery process.

**Environmental Sustainability:** The On-Demand Fuel Delivery Application recognizes the importance of environmental sustainability. By providing a platform for users to choose alternative fuels and promoting greener fuel options, the application contributes to reducing carbon emissions and supporting a more sustainable transportation ecosystem.

### **2.1.2 Project Scope**

The scope of the On-Demand Fuel Delivery Application project encompasses the development of a comprehensive and user-friendly mobile application that provides seamless fuel ordering and delivery services. The key components and functionalities within the project scope include:

**User Registration and Authentication:** The application will have a user registration system where users can create an account by providing their credentials, such as name, email, and password. Once registered, users can log in securely using their credentials for subsequent access to the application.

**Location-based Services:** The application will leverage geolocation technology to determine the user's current location or allow them to manually enter a specific address for fuel delivery. This feature enables users to receive fuel at their preferred location, whether it is their home, workplace, or any other specified address.

**Fuel Price Comparison:** The application will provide users with real-time information on fuel prices from nearby gas stations or fuel suppliers. This feature empowers users to make informed decisions by comparing prices and selecting the most cost-effective fuel option.

**Fuel Ordering:** Users will have the ability to select the type and quantity of fuel they require through the application's intuitive user interface. They can place an order for fuel delivery by specifying the desired fuel type (e.g., gasoline, diesel, or alternative fuels) and the required quantity.

**Order Tracking:** The application will include a tracking system that allows users to monitor the status of their fuel order in real-time. Users can view the progress of their delivery, including the estimated time of arrival and the location of the delivery person, ensuring transparency and providing peace of mind.

**Digital Receipts:** Once the fuel delivery is completed, users will receive digital receipts for their orders. These receipts will contain detailed information such as the order ID, fuel type, quantity, total cost, and transaction date. Users can easily access and store these receipts within the application for future reference.

**Communication and Notifications:** The application will facilitate communication between users and fuel delivery providers. Users may receive notifications regarding order confirmation, estimated delivery time, and updates on the status of their delivery. Similarly, delivery providers can communicate with users to ensure a smooth and efficient delivery process.

It is important to note that while the primary focus of the project is to develop the On-Demand Fuel Delivery Application with the features, the project scope may also encompass additional functionalities based on specific requirements and project constraints. The scope may be further refined during the development process to ensure the successful delivery of the application.

### **III. REQUIREMENTS ANALYSIS**

#### **3.1 Functional Requirements**

Functional Requirement for On-Demand Fuel Delivery Application are:

- **User Registration:** The application should allow users to create an account and register themselves by providing their basic information such as name, address, phone number, and email address.
- **Fuel Ordering:** The application should allow users to order fuel (gasoline or diesel) on-demand, specifying the type and quantity of fuel required.
- **Location Services:** The application should be able to identify the user's location using GPS and provide them with a list of fuel delivery providers operating in the area.
- **Payment Processing:** The application should allow users to pay for the fuel delivery using a variety of payment methods such as credit/debit card, net banking, mobile wallet, or cash on delivery.
- **Real-time Tracking:** The application should provide users with real-time tracking of their fuel delivery, showing the location of the fuel delivery provider and the estimated time of delivery.
- **Delivery Confirmation:** The application should provide users with a confirmation of delivery once the fuel has been delivered to their location.
- **User History:** The application should maintain a record of the user's fuel delivery history, including details such as the date and time of delivery, the quantity of fuel delivered, and the total cost of the delivery.

### 3.2 Non-Functional Requirements

Non-functional requirements for an on-demand fuel delivery application built using Flutter may include:

- **Performance:** The application should be optimized for performance to provide a smooth and responsive user experience. This includes fast loading times, quick response to user inputs, and efficient use of device resources.
- **Security:** The application should be designed with security in mind, including secure data transmission, secure storage of user data, and protection against common security threats such as SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF).
- **Reliability:** The application should be reliable, with a high degree of uptime and minimal downtime. It should also be able to handle large volumes of user requests without crashing or slowing down.
- **Usability:** The application should be easy to use and navigate, with a clear and intuitive user interface that requires minimal training for users to use effectively.
- **Scalability:** The application should be designed to scale easily to accommodate growing user demand, with the ability to handle many users and requests without performance degradation.
- **Compatibility:** The application should be compatible with a range of devices and operating systems, including Android and iOS, and should be able to run on devices with varying screen sizes and resolutions.
- **Accessibility:** The application should be designed to be accessible to users with disabilities, including support for assistive technologies such as screen readers and voice commands.
- **Maintainability:** The application should be easy to maintain and update, with clean and well-organized code that is easy to understand and modify.
- **Interoperability:** The application should be designed to integrate with other systems and services, such as payment gateways and fuel delivery providers, to provide a seamless and integrated user experience.
- **Performance Monitoring:** The application should be designed to provide performance metrics, logs, and alerts to monitor the performance of the application and identify issues that may affect the user experience.

### 3.3 Use-Case Specification

Use Case Name: Place Fuel Order

Primary Actor: User

Goal in Context: The user wants to order fuel on-demand and have it delivered to their location.

Preconditions:

- The user is registered and logged in to the application.
- The user has provided their location information.

Main Success Scenario:

1. The user opens the application and selects the "Place Order" option.
2. The application displays a list of fuel delivery providers operating in the user's area.
3. The user selects a fuel delivery provider from the list based on factors such as price, proximity, and delivery time.
4. The user selects the type and quantity of fuel they require and specifies their delivery location.
5. The user confirms the order and makes payment using a variety of payment methods such as credit/debit card, net banking, mobile wallet, or cash on delivery.
6. The application sends a notification to the selected fuel delivery provider, indicating the user's order details and delivery location.
7. The fuel delivery provider accepts the order and dispatches a delivery person to the user's location.
8. The application provides real-time tracking of the fuel delivery person's location and estimated time of arrival.
9. The fuel delivery person delivers the fuel to the user's location and confirms delivery through the application.
10. The application displays a confirmation of delivery to the user and updates the user's order history.

Postconditions:

- The user receives the ordered fuel at the specified location.
- The user's payment is processed and confirmed.
- The user's order history is updated with the details of the order and delivery.

## **IV. DESIGN**

### **4.1 Database Design**

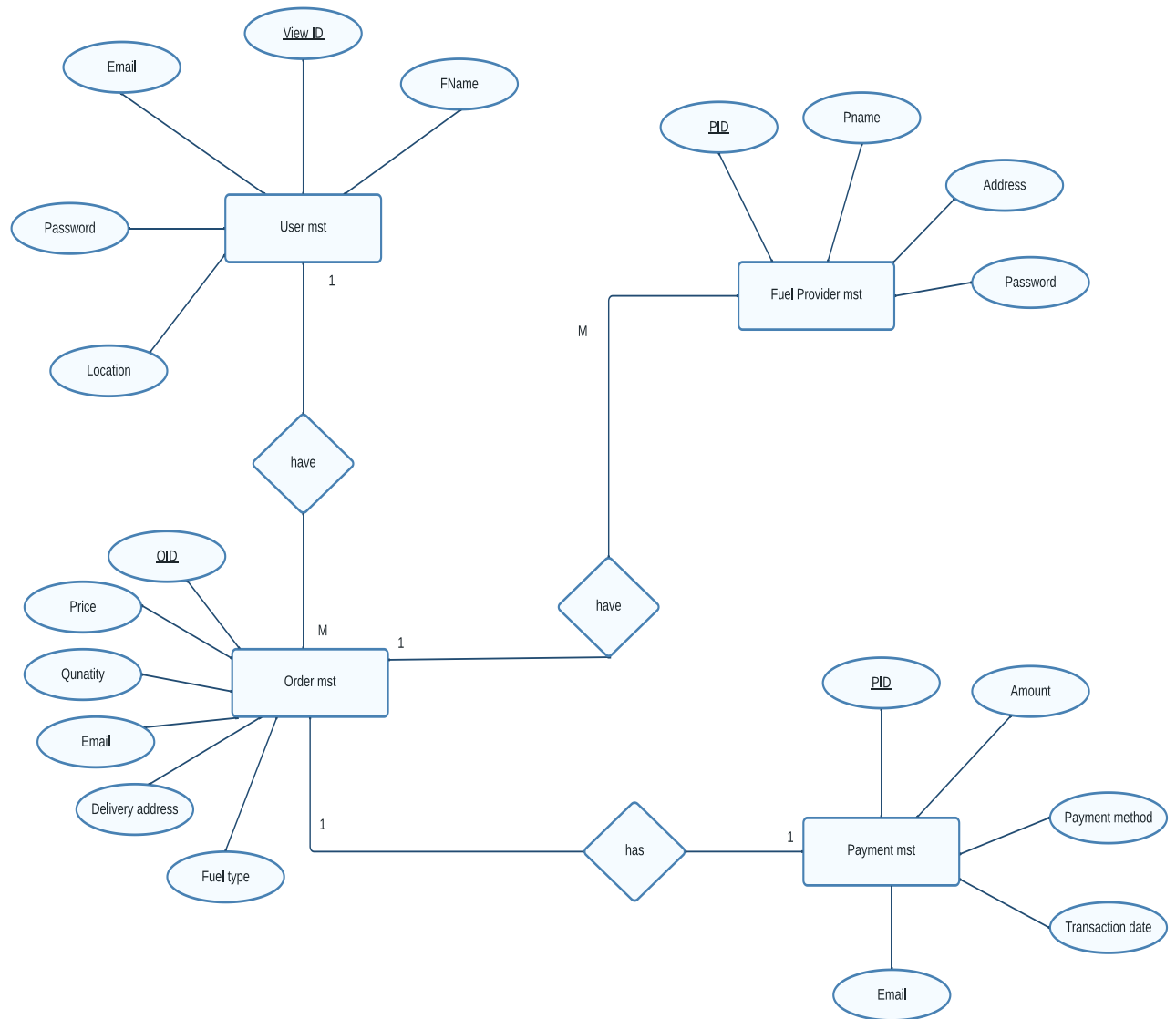
#### **4.1.1 ER Diagram**

The following tables will be designed based on the ER diagram:

The ER diagram for the fuel delivery application will include the following entities:

1. **User:** This entity will contain information about the user such as name, email, password, phone number, and location.
2. **Fuel Delivery Provider:** This entity will contain information about the fuel delivery provider such as name, email, password, phone number, and address.
3. **Order:** This entity will contain information about the order such as order ID, user ID, fuel delivery provider ID, fuel type, quantity, delivery address, delivery status, and delivery date.
4. **Payment:** This entity will contain information about the payment such as payment ID, user ID, order ID, payment method, amount, and transaction date.





#### 4.1.2 Design Tables and Normalization:

The following tables will be designed based on the ER diagram:

- User Table:

Column Name	Data Type	Constraints
User_ID	int	Primary Key
Name	varchar	Not Null
Email	varchar	Not Null, Unique
Password	varchar	Not Null
Phone_Number	varchar	Not Null
Address	varchar	Not Null

- Fuel Provider Table:

Column Name	Data Type	Constraints
Fuel _Prov_ID	int	Primary Key
Name	varchar	Not Null
Email	varchar	Not Null, Unique
Password	varchar	Not Null
Phone_Number	varchar	Not Null
Address	varchar	Not Null

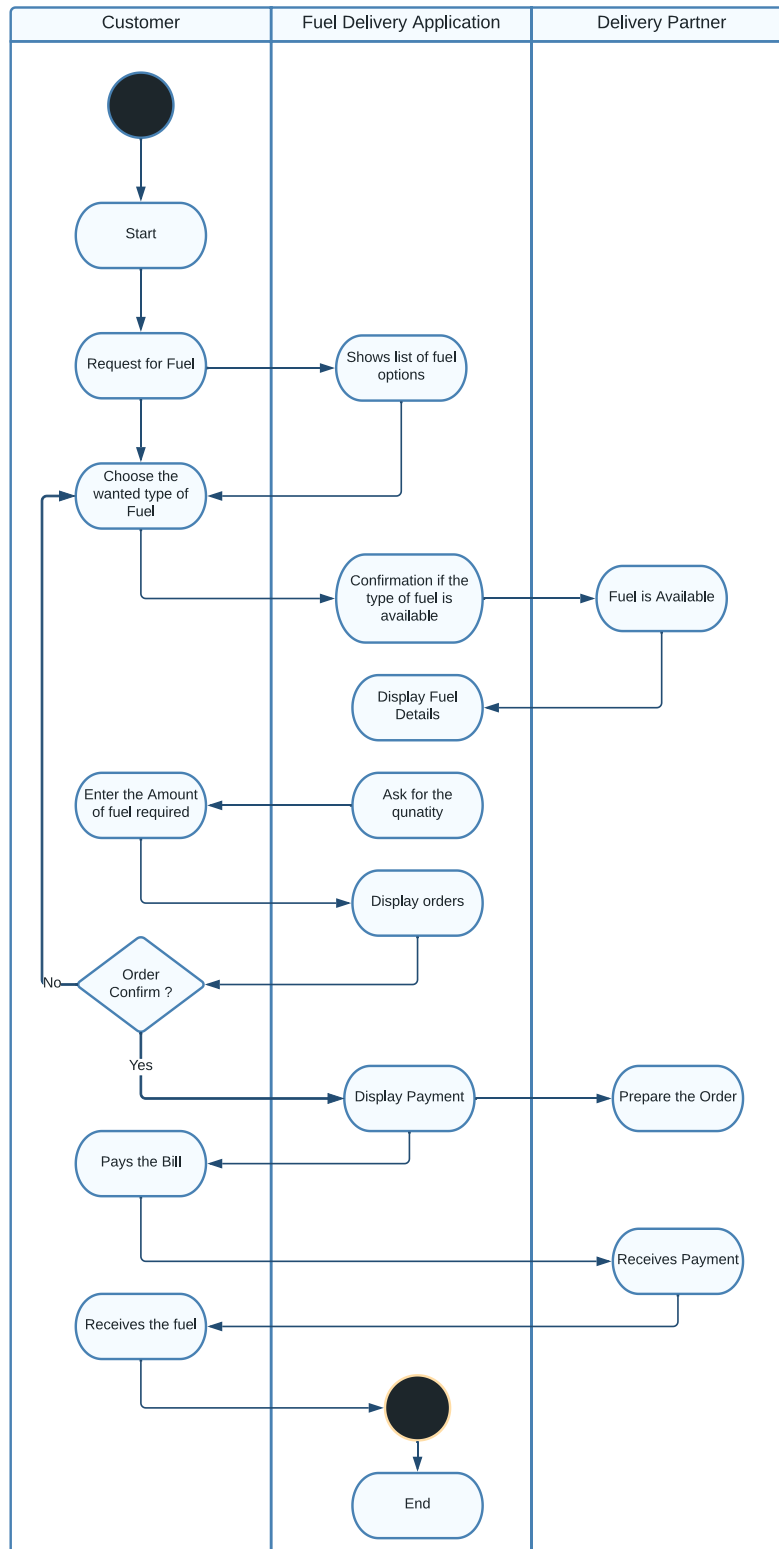
- Order Table:

Column Name	Data Type	Constraints
Order_ID	int	Primary Key
User_ID	int	Foreign Key
Fuel_Prov_ID	int	Foreign Key
Fuel_Type	varchar	Not Null
Quantity	int	Not Null
Delivery_Address	varchar	Not Null
Delivery_Status	varchar	Not Null
Delivery_Date	datetime	Not Null

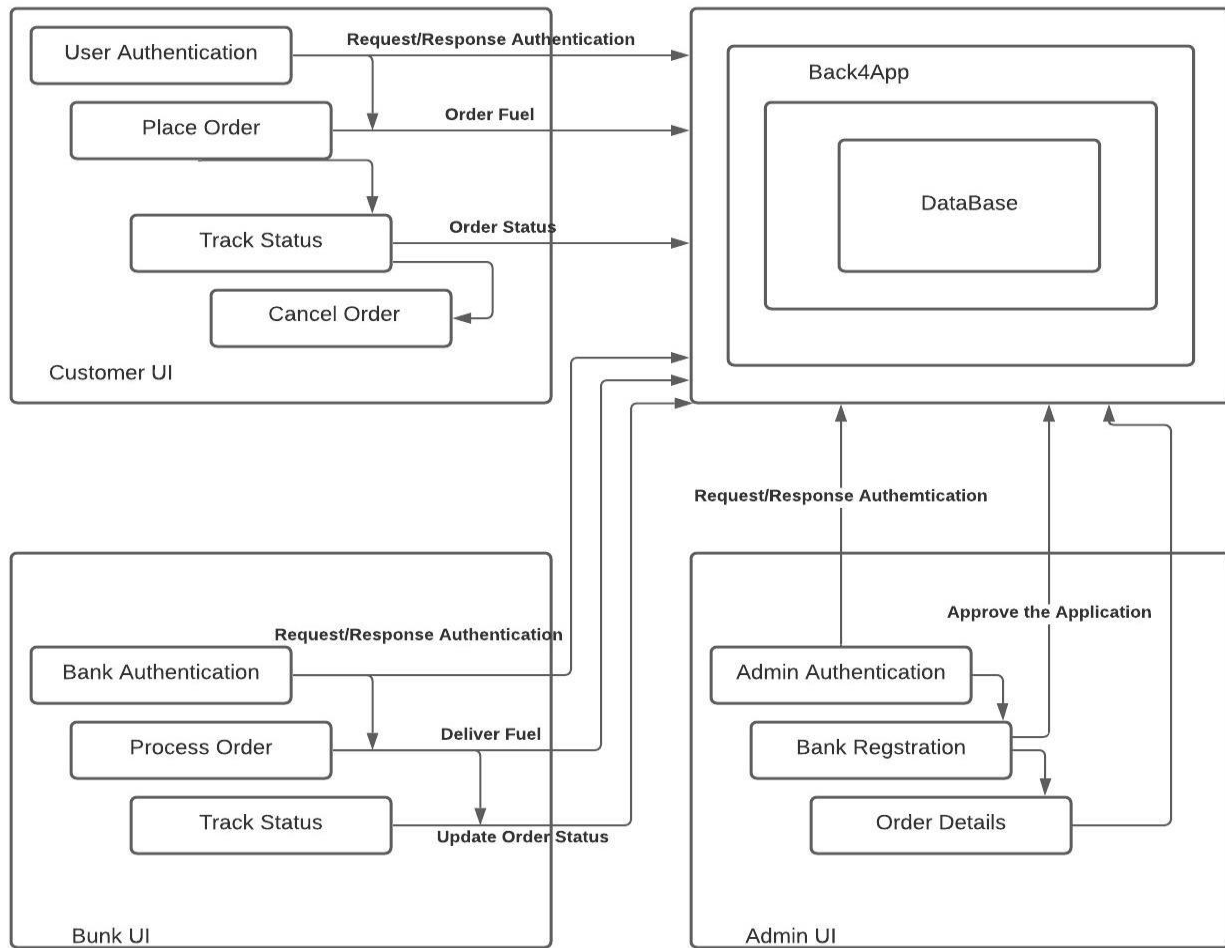
- Payment Table:

Column Name	Data Type	Constraints
Payment_ID	int	Primary Key
User_ID	int	Foreign Key
Order_ID	int	Foreign Key
Payment_Method	varchar	Not Null
Amount	float	Not Null
Transaction_Date	datetime	Not Null

## 4.2 Activity Diagram

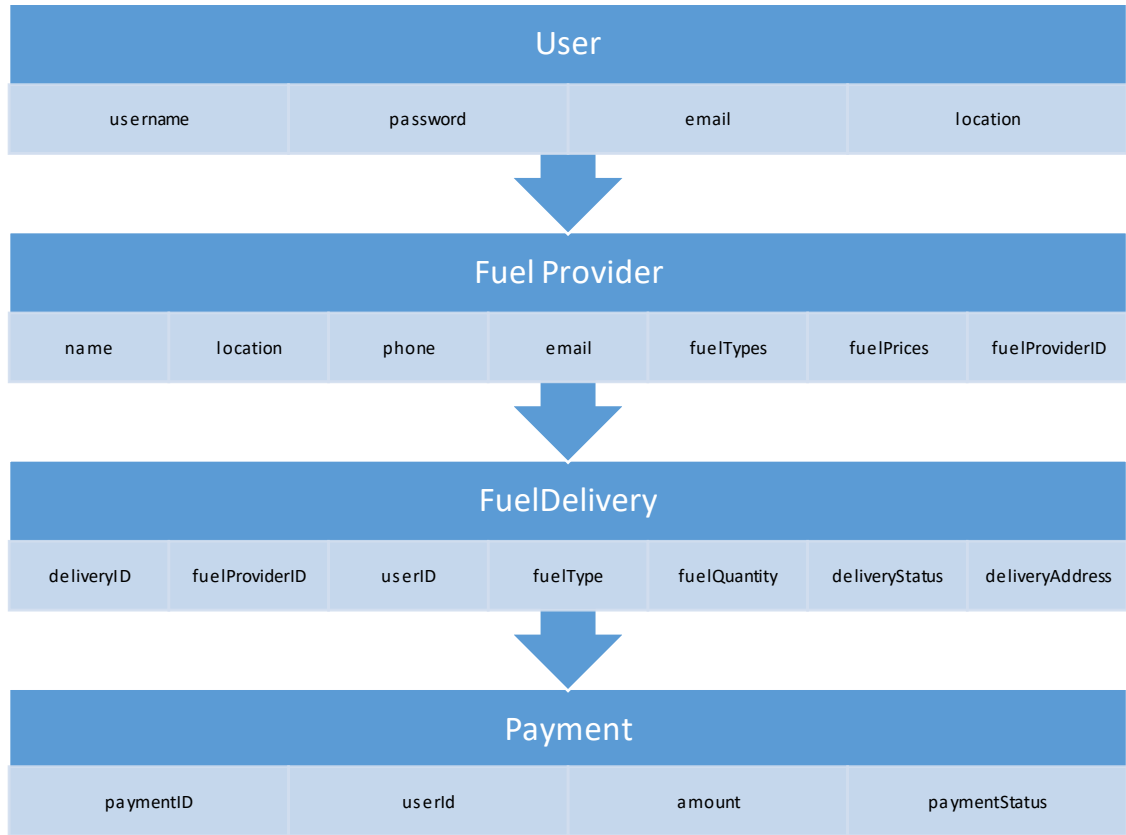


### 4.3 Architecture and Flow Diagram



- The Customer UI includes the interactive environment developed for the customer. The entire section contains all the section present directly to the customer. First Section in the Customer UI is the User Authentication section in which the user must enter their credentials through the Request/Response Authentication in which one module sends request to other module and waits for a response. The user credentials are sent to the Database of the Application for Authentication. Once the response is received from the Database Section the process is processed accordingly. Once User Authentication is completed the customer then can move forward to the Place Order Section which enables the customer to be able to Place Order for the Fuel Delivery. Once the user has placed the order the record is stored in the database. Once the user has placed the order, they will be able to track their order status where they can see the current status of their order and can track the location of the delivery. The customer would also have access to Cancel Order where the customer can cancel their order and once the cancel request is completed the process the aborted and an entry is made in the database regarding the entire case.
- The Bunk UI will handle all the transaction related process where we intend to hire a third party for safer, faster and reliable transactions. Bunk is the placeholder for the third-party service we intend to utilize in our application. The Bunk UI contains Bank Authentication, Process Order, Track status. First the Bank Authentication will include verification of the bank details the customer provides with their respective bank, once the verification is completed the service will be providing the customer with number of options for the transaction that their bank provides for smoother user experience. If bank details are stored in the database the service will request the details from the database module and wait for the response and if the details are not present the process is proceeded as mentioned above and the user bank details are stored in the database. Once the Authentication and the transaction is completed then the order is processed forward and a message is sent to the database regarding the process and on successful user transaction the customer is send the details of the transaction. In the Track Status section, the Bunk UI constantly monitors and sends the updates to the database. If the order is cancelled the Bunk UI starts the process of returning the money to the customer by requesting the details from the database.
- The Admin UI contains all Admin tools providing several productivity features. The architecture focuses on the three section of the Admin UI namely Admin Authentication, Bank Registration and Order Details. The admin module manages the entire system, and only authorized personnel can access it. The admin can view all registered users and fuel stations, track orders, and oversee payments. The admin module also provides analytical reports, including sales reports and customer insights, to help the fuel delivery service to make informed decisions.

#### 4.4 Class Diagram



The class diagram includes four classes: User, FuelProvider, FuelDelivery, and Payment.

The User class represents a user of the application and contains attributes such as username, password, email, and location.

The FuelProvider class represents a fuel delivery provider and includes attributes such as name, location, phone, email, fuelTypes, and fuelPrices.

The FuelDelivery class represents a fuel delivery and includes attributes such as deliveryId, fuelProviderId, userId, fuelType, fuelQuantity, deliveryStatus, and deliveryAddress.

The Payment class represents a payment made for a fuel delivery and includes attributes such as paymentId, userId, amount, and paymentStatus.

Overall, this class diagram provides a basic representation of the entities and relationships in the Fuel Delivery Application.

## V. EXPERIMENT AND TESTING

### 5.1 Test cases developed.

- Test case: User login

Description: Verify that the user can login to the application using valid credentials.

Steps:

1. Launch the application.
2. Enter valid username and password.
3. Click on the "Login" button.
4. Expected result: User should be logged in successfully and taken to the dashboard page.

- Test case: Place an order

Description: Verify that the user can place an order for fuel delivery.

Steps:

1. Login to the application.
2. Click on the "Place Order" button.
3. Select a fuel delivery provider from the list.
4. Select the type and quantity of fuel required.
5. Specify the delivery location.
6. Confirm the order and make payment.
7. Expected result: User should receive a confirmation of the order, and the fuel delivery provider should receive the order details.

- Test case: Real-time tracking

Description: Verify that the user can track the fuel delivery person in real-time.

Steps:

1. Login to the application.
2. Place an order for fuel delivery.
3. Open the order details page.
4. Check the real-time tracking information.
5. Expected result: User should be able to see the fuel delivery person's location and estimated time of arrival.



- Test case: Payment confirmation  
Description: Verify that the user receives payment confirmation after making a payment for an order.  
Steps:
  1. Login to the application.
  2. Place an order for fuel delivery.
  3. Make payment using a valid payment method.
  4. Check the payment confirmation page.
  5. Expected result: User should receive a confirmation of the payment and the order status should be updated accordingly.
- Test case: Order history  
Description: Verify that the user can view their order history.  
Steps:
  1. Login to the application.
  2. Navigate to the "Order History" page.
  3. Expected result: User should be able to see a list of their previous orders along with the order details and status.

## **5.2 Testing used in our project.**

- I. Functional testing: This type of testing is used to ensure that the application functions as intended and all the features work correctly. It includes testing of the user interface, user authentication, placing an order, payment processing, delivery tracking, and order history.
- II. Performance testing: This type of testing is used to measure the application's performance under various conditions, such as heavy load, peak usage times, and slow internet speeds. It includes testing of application response time, page load time, and server uptime.
- III. Security testing: This type of testing is used to ensure that the application is secure and protected from unauthorized access and attacks. It includes testing of user authentication, password protection, data encryption, and secure communication protocols.
- IV. Compatibility testing: This type of testing is used to ensure that the application works correctly on different devices, operating systems, and browsers. It includes testing on different screen sizes, hardware configurations, and software versions.
- V. Usability testing: This type of testing is used to measure the application's ease of use and user-friendliness. It includes testing of the user interface, navigation, and user experience.
- VI. Regression testing: This type of testing is used to ensure that the application's new features or updates do not impact the existing functionality negatively. It includes testing of the old and new features after updates or changes are made to the application.

## **VII. CONCLUSION**

Our On-demand fuel delivery application has the potential to revolutionize the traditional fueling process and provide customers with a convenient, time-saving, and efficient service. By leveraging technology and innovation, the application can address the challenges and limitations of traditional fueling methods, such as long queues, time-consuming commutes, and limited availability of fuel stations. Our On-Demand fuel delivery application represents an innovative and promising solution to the challenges and limitations of traditional fueling methods. By continuously improving and innovating the application, we can further enhance its benefits and promote a more sustainable and efficient fueling system for the future. Overall, the On-Demand Fuel Delivery Application developed using Flutter will help meet the growing demand for fuel while promoting sustainability and creating a more convenient and reliable fuel delivery system.

### **6.1 Problems and Issues in current system**

While fuel delivery applications offer many benefits to both customers and fuel delivery companies, there are also several problems and issues that can arise. Some of the common problems and issues in fuel delivery applications include:

- 1) **Safety and security:** As mentioned earlier, fuel is a hazardous substance, and there is a risk of spills or accidents during the delivery process. Fuel delivery companies need to ensure that their delivery vehicles and equipment are properly maintained and that their drivers are trained to handle fuel safely.
- 2) **Accuracy of delivery:** Fuel delivery applications rely on accurate GPS location tracking to ensure that fuel is delivered to the correct location. However, GPS location tracking can sometimes be inaccurate, leading to fuel being delivered to the wrong location.
- 3) **Compatibility with different devices:** Fuel delivery applications need to be compatible with a wide range of devices, including smartphones and tablets. However, not all devices may be compatible with the app, leading to compatibility issues and customer frustration.
- 4) **Reliability of the app:** Fuel delivery applications need to be always reliable and available to ensure that customers can order fuel when they need it. Any downtime or technical issues with the app can result in lost sales and customer dissatisfaction.

- 5) User interface: The user interface of fuel delivery applications needs to be user-friendly and intuitive, especially for customers who may not be tech-savvy. If the app is difficult to use, customers may abandon it and seek other fuel delivery options.
- 6) Data privacy and security: Fuel delivery applications collect sensitive customer data, such as credit card information and location data. It is important to ensure that this data is stored securely and that customer privacy is protected.

In summary, while fuel delivery applications offer many benefits, there are also several problems and issues that can arise. Fuel delivery companies need to address these issues to ensure that their apps are reliable, secure, and user-friendly, and that they provide a positive customer experience.

## **6.2 Future extension**

As technology continues to evolve, there are several potential future extensions for fuel delivery applications that could further enhance the customer experience and improve the efficiency of fuel delivery operations. Here are a few possibilities:

- 1) **Integration with smart home technology:** As smart home technology becomes more prevalent, fuel delivery apps could be integrated with smart home devices, such as Amazon Alexa or Google Home. This would allow customers to order fuel using voice commands and receive notifications when their delivery has arrived.
- 2) **Predictive analytics:** By analyzing customer data, fuel delivery companies could use predictive analytics to anticipate when customers will need fuel and proactively schedule deliveries. This could help reduce the risk of running out of fuel and improve the efficiency of delivery operations.
- 3) **Integration with electric vehicle charging:** As electric vehicles become more common, fuel delivery apps could be integrated with electric vehicle charging infrastructure. This would allow customers to order both fuel and charging services using the same app.
- 4) **Use of alternative fuels:** As alternative fuels, such as hydrogen and biofuels, become more prevalent, fuel delivery apps could be updated to allow customers to order these fuels as well.
- 5) **In conclusion,** there are several potential future extensions for fuel delivery applications that could further improve the efficiency and convenience of fuel delivery operations. As technology continues to evolve, fuel delivery companies will need to stay up-to-date with the latest trends and innovations to remain competitive in the marketplace.

## APPENDIX:

# ON-DEMAND FUEL DELIVERY APPLICATION BUILT USING FLUTTER

---

Navneet Mishra  
School of CSIT,  
Symbiosis University of Applied  
Sciences,  
Indore, India  
2019BTCS050@student.suas.ac.in

Ritika Raghuvanshi  
School of CSIT,  
Symbiosis University of Applied  
Sciences,  
Indore, India  
2019BTCS066@student.suas.ac.in

Indrajeet Kumar  
Assistant Professor, School of CSIT,  
Symbiosis University of Applied  
Sciences,  
Indore, India  
indrajeet.kumar@suas.ac.in

**ABSTRACT-** THE INCREASING DEMAND FOR FUEL DUE TO THE GROWTH OF AUTOMOBILES IN THE MARKET HAS LED TO THE NEED FOR ON-DEMAND FUEL SUPPLY APPLICATIONS THAT DEPEND ON USER ORDERS AND REQUIREMENTS. WHEN A VEHICLE RUNS OUT OF FUEL, IT CAN BE A HASSLE FOR THE OWNER TO PUSH THE CAR OR SEEK HELP TO REACH THE NEAREST GAS STATION. FOR OLDER PEOPLE AND THOSE WHO ARE MEDICALLY ILL, THIS TASK CAN BE EVEN MORE DIFFICULT. ADDITIONALLY, PEOPLE MUST GO TO GAS STATIONS TO FILL UP GENERATORS. TO ADDRESS THESE ISSUES, WE INTRODUCE A NEW SOLUTION FOR VEHICLE REFUELING AND EMERGENCY POWER SUPPLIES THROUGH THE DEVELOPMENT OF AN ON-DEMAND FUEL DELIVERY APPLICATION. THIS APPLICATION PROVIDES DOOR-TO-DOOR COVERAGE AND ALLOWS END USERS TO CHOOSE THE TYPE OF FUEL THEY NEED, ORDER IT, AND RECEIVE IT WITH EASE. THE RESULT OF THE PROJECT WILL BE AN APPLICATION THAT ALLOWS CUSTOMERS TO ORDER FUEL, FIND AVAILABLE GAS STATIONS NEARBY, AND HELPS OWNERS TRACK ORDERS AND AVAILABLE STATIONS.

## I. INTRODUCTION

Our app-based service, On-Demand Fuel Delivery Application that is built using flutter, is similar to what we other On-Demand Delivery services but we aim to provide fuel to other customer. Our goal is to establish a system where the user can request for the fuel to be deliver to his footsteps. We aim to provide timely delivery of fuel to customers. In this modern fast paced world where demands are amplifying day-by-day, we aim to revolutionize of the least modernized area by introducing our On-Demand Fuel Delivery Application built using Flutter which removes the existing constraints and gives an easy to use, safe, reliable way to meet the user demands. On-Demand Fuel Delivery Application provides online fuel ordering services including an engaging and comprehensive online fuel ordering process such as ordering online, tracking order, and checking the fuel prices nearby. The future scalability of on-demand fuel services is enormous, and

several end-users can be targeted. For example, 36% of India's electricity in urban areas is generated solely by diesel generator sets. In addition, the number of gas stations currently in India needs to be more than doubled to meet the growing demand.

- The "Fuel Management System" paper [1] by Areeg Abubakr and Siddig Ali published in the IEEE Journal in January 2017 provides a review of various fuel management systems used in transportation, highlighting their importance in reducing fuel consumption and costs. The study covers the technical aspects of FMS, challenges in implementing FMS, and their potential benefits in the transportation industry.
- The research paper "Demand for Road Fuel in Small Developing Countries" [2] by Sunil Chandrasiri examines the challenges of meeting the growing demand for road fuel in these countries driven by economic development and population growth. The study analyses factors affecting demand and the effectiveness of policy interventions like fuel taxes and subsidies. The paper suggests a more comprehensive approach is needed to promote sustainable transportation that considers technological innovations and behavioural change. It offers recommendations for policymakers to address these challenges.
- The "All India Study on Sectoral Demand of Diesel & Petrol" [3] by Nielsen India Private Limited is a report that examines the sector-wise demand for diesel and petrol in India, with a focus on key drivers of demand. The study provides insights for policymakers and industry stakeholders and contributes to a better understanding of the energy sector in India.

- The research paper "Long-term Forecast of Energy and Fuel Demand for Ecuador's Sustainable Road Transport Sector (2016-2035): Applying the LEAP Model" [4] by Luis Rivera Gonzalez and David Bologna analyses the use of the LEAP model to forecast energy and fuel demand in Ecuador's road transport sector between 2016 and 2035. The study evaluates the impact of different policy scenarios on energy consumption and greenhouse gas emissions and recommends sustainable transport policies to reduce emissions and improve energy efficiency. The research offers valuable insights and recommendations for policymakers to promote sustainable development.
- "India's Oil Demand: Empirical Estimates and Future Forecasts" [5] by Pradeep Agarwal is a research paper that examines the factors driving oil demand in India, the impact of policy interventions on oil demand, and recommendations for promoting energy conservation and renewable energy sources. The study suggests that India's oil demand will continue to grow due to economic development and urbanization, and offers insights and recommendations for policymakers.
- The Ministry of Oil and Gas of India published a report on the "All India Survey on Diesel and Gasoline Demand by Sector" which outlines the consumption of diesel and gasoline in various sectors. The transportation sector is the largest consumer at 60%, followed by agriculture and industry. The report emphasizes the need to reduce dependency on fossil fuels and promote sustainable alternatives for a greener economy.
- During the forecast period of 2022-2032, the On-Demand fuel delivery market is likely to increase at a CAGR of 6.8%. In 2022, this market is expected to reach around \$4.8 billion. The On-Demand fuel delivery market value will likely be \$6.2 billion by 2026. On-demand application revenue is likely to generate \$935 billion in 2023.

## I. BACKGROUND

Flutter is an open-source UI toolkit developed by Google that enables developers to build natively compiled applications for mobile, web, and desktop from a single codebase. Flutter was first introduced in May 2017 at the Google I/O developer conference and has since gained popularity among developers due to its fast development cycle, high-performance, and cross-platform capabilities. Flutter uses a reactive programming model, where changes to the UI are automatically reflected in the app's state, and vice versa. This enables developers to build highly interactive and responsive apps with a smooth user experience. Flutter also comes with a rich set of customizable widgets and allows developers to

create their own widgets or modify existing ones to suit their needs.

Flutter's hot-reload feature enables developers to see the changes they make to the code in real-time, without having to restart the app. This makes the development process much faster and efficient. Flutter also provides a rich set of tools, including a powerful IDE called Flutter Studio, as well as command-line tools and plugins for popular development environments like Android Studio and Visual Studio Code. Overall, Flutter is a versatile and powerful toolkit that enables developers to create high-quality, cross-platform apps with ease. Its popularity is only expected to grow in the coming years as more developers discover its potential and adopt it for their projects.

## II. SYSTEM MODEL

Our On-Demand Fuel Delivery Application mainly consists of four modules:

### A. Register module

The registration module of our On-Demand Fuel Delivery Application requires users as well as the fuel station to enter their credentials and login first. User registration requires you to provide information such as your name, contact number, email ID, username, and password. Gas stations need to provide information such as the name of the gas station, contact number, email ID, user name, password, and location of the gas station.

### B. Information Module

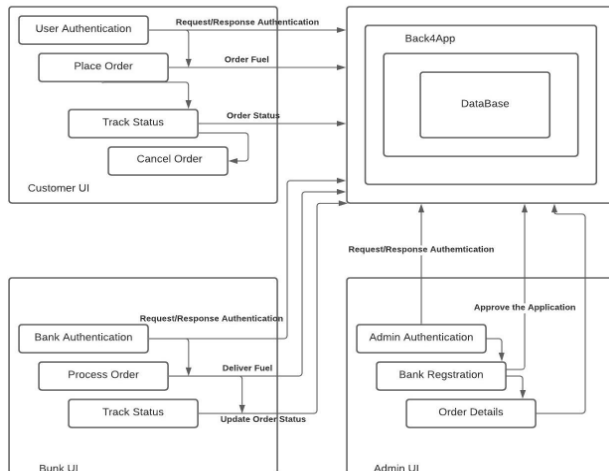
The fuel station must provide information about fuel availability, prices of different fuels, types of fuel available, and services. Since fuel is the most important factor for any vehicle, its price changes daily, and the price of fuel also changes depending on the location of the gas station. Therefore, it is the gas station's responsibility to update fuel prices daily.

### C. Order Fuel Module

When user registers with the application, they can order fuel at his location in the required quantity. Users must enter their credentials after they have access to the services provided by the application before they can use the application. To order fuel, users must first find a nearby gas station and check the availability of fuel at that gas station. After checking the availability of fuel and services, users can order fuel as needed.

### D. Track Order Module

Once an order is placed, the user can track the order, whether the order was accepted, and whether the order was delivered. To receive order updates, the gas station must approve or reject the order and update the order status.



**Fig.1:** Architecture & Flow Diagram

**Fig.1:** In this figure we can see the entire architecture and the flow of the entire application:

- The Customer UI includes the interactive environment developed for the customer. The entire section contains all the section present directly to the customer. First Section in the Customer UI is the User Authentication section in which the user must enter their credentials through the Request/Response Authentication in which one module sends request to other module and waits for a response. The user credentials are sent to the Database of the Application for Authentication. Once the response is received from the Database Section the process is processed accordingly. Once User Authentication is completed the customer then can move forward to the Place Order Section which enables the customer to be able to Place Order for the Fuel Delivery. Once the user has placed the order the record is stored in the database. Once the user has placed the order, they will be able to track their order status where they can see the current status of their order and can track the location of the delivery. The customer would also have access to Cancel Order where the customer can cancel their order and once the cancel request is completed the process is aborted and an entry is made in the database regarding the entire case.
- The Bunk UI will handle all the transaction related process where we intend to hire a third party for safer, faster and reliable transactions. Bunk is the placeholder for the third-party service we intend to utilize in our application. The Bunk UI contains Bank Authentication, Process Order, Track status. First the Bank Authentication will include verification of the bank details the customer provides with their respective bank, once the

verification is completed the service will be providing the customer with number of options for the transaction that their bank provides for smoother user experience. If bank details are stored in the database the service will request the details from the database module and wait for the response and if the details are not present the process is proceeded as mentioned above and the user bank details are stored in the database. Once the Authentication and the transaction is completed then the order is processed forward and a message is sent to the database regarding the process and on successful user transaction the customer is send the details of the transaction. In the Track Status section, the Bunk UI constantly monitors and sends the updates to the database. If the order is cancelled the Bunk UI starts the process of returning the money to the customer by requesting the details from the database.

The Admin UI contains all Admin tools providing several productivity features. The architecture focuses on the three section of the Admin UI namely Admin Authentication, Bank Registration and Order Details. The admin module manages the entire system, and only authorized personnel can access it. The admin can view all registered users and fuel stations, track orders, and oversee payments. The admin module also provides analytical reports, including sales reports and customer insights, to help the fuel delivery service to make informed decisions.

## I. IMPLEMENTATION

The implementation of our On-Demand Fuel Delivery Application will involve the following stages:

### A. Design phase:

During this stage, the user interface, features, and functionalities of the application will be designed. We will use Flutter to build the user interface, and we will incorporate Material Design principles to ensure a clean, modern look.

### B. Development phase:

This stage involves the actual coding of the application. We will use the Flutter framework to develop the application, and we will use Firebase for user authentication and database management. Google Maps API will be integrated for displaying the nearby gas stations and tracking the fuel delivery.

### C. Testing phase:

In this phase, the application will be tested to ensure that it is functioning correctly and meets the user requirements. We will conduct both unit and integration testing to ensure that all the components of the application are working together as intended.

### D. Deployment phase:

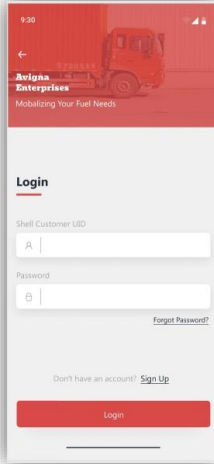
Once the application has been developed and tested, it will be deployed to the Google Play Store and the Apple App Store.



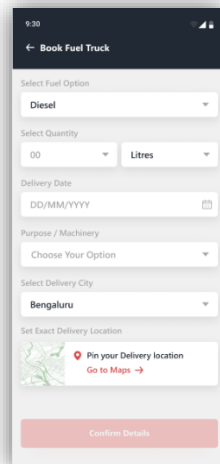
Users will be able to download and install the application on their smartphones.

## I. SIMULATION RESULTS

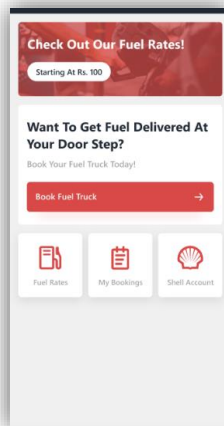
The following figures illustrates the key screens that the user will interact with while using Our On-Demand Application built using Flutter. Flutter is Google's SDK for crafting attractive, User-friendly environment for mobile, web, and desktop from a single codebase.



**Fig.2.** Login page



**Fig.3.** Home page



**Fig.4.** Place Order page

**Fig.2:** Login Page where registered users can input their pre-existing credentials stored in the database and login to their accounts. User can use the forgot password feature incase the user forgets their password. Users will be able to reset their password through email verification. Unregistered users can Sign Up in-order to use the Application.

**Fig.3:** Main Activity screen is displayed when the user has successfully has verified his credentials and has login to the application. In this screen the user can click on book fuel truck and proceed to the next step of the process.

**Fig.4:** Place Order Activity screen provides user with a lot of options starting with fuel option where user can choose the type of fuel he wants to be delivered, user can also choose the quantity of the fuel and the measurement units. User must provide the delivery date and purpose of his request for the fuel. The user then must provide his exact location and once the user has input the details, he can proceed by clicking on the Confirm Details.

## II. CHALLENGES AND OPPORTUNITIES

The On-Demand Fuel Delivery Application faces several challenges and opportunities. One of the challenges is building a robust and reliable delivery system. Fuel delivery requires special equipment and trained personnel, and the application must ensure that the delivery is safe, efficient, and timely. The application must also comply with regulatory requirements, such as fuel transportation regulations, to ensure the safety of the delivery personnel and users.

Another challenge is the adoption of the application by fuel stations and customers. The application must convince fuel stations to partner with the service, and customers must be willing to use the application instead of visiting gas stations physically. The application must also be user-friendly, reliable, and provide a seamless experience to ensure customer satisfaction.

The On-Demand Fuel Delivery Application also presents several opportunities. The first opportunity is providing convenience to users. Users can avoid the hassle of visiting gas stations and waiting in queues by ordering fuel through the application. The application can also provide users with real-time fuel prices, promotions, and discounts, helping them save time and money.

## III. CONCLUSION

Our On-demand fuel delivery application has the potential to revolutionize the traditional fuelling process and provide customers with a convenient, time-saving, and efficient service. By leveraging technology and innovation, the application can address the challenges and limitations of traditional fuelling methods, such as long queues, time-consuming commutes, and limited availability of fuel stations. Our On-Demand fuel delivery application represents an innovative and promising solution to the challenges and limitations of traditional fuelling methods. By continuously improving and innovating the application, we can further enhance its benefits and promote a more sustainable and efficient fuelling system for the future.

Overall, the On-Demand Fuel Delivery Application developed using Flutter will help meet the growing demand

for fuel while promoting sustainability and creating a more convenient and reliable fuel delivery system.

#### I. REFERENCES

1. A. A. I. Ahmed, S. A. E. Mohammed and M. A. M. H. Satte, "Fuel management system," 2017 International Conference on Communication, Control, Computing and Electronics Engineering (ICCCCEE), Khartoum, Sudan, 2017, pp. 1-7.
2. Chandrasiri, S. (2006). Demand for road-fuel in a small developing economy: The case of Sri Lanka. *Energy Policy*, 34(14), 1833-1840.
3. Nielsen India Private Limited. (2013). All India Study on Sectoral Demand of Diesel & Petrol. Ministry of Petroleum and Natural Gas.
4. Rivera-González, L., Bolonio, D., García-López, G. A., & Alvarez, M. (2019). Long-Term Forecast of Energy and Fuels Demand Toward a Sustainable Road Transport Sector in Ecuador (2016–2035): A LEAP Model Application. *Energies*, 12(20), 3849, doi: 10.3390/en12203849.
5. Agarwal, P. (2012). India's Petroleum Demand: Empirical Estimations and Projections for the Future. Institute of Economic Growth (IEG) University.
6. Abubakr, A., & Ali, S. (2017). Fuel Management System. *IEEE Journal*, 5(1), 23-30.
7. Rabinovich, A., Azuri, Y., & Shtilman, L. (2018). Assessment of fuel delivery system of a high-performance UAV engine. *Journal of Propulsion and Power*, 34(4), 880-888.
8. Gao, H., Liu, J., & Huang, Q. (2019). Fault diagnosis of fuel delivery system for diesel engine based on dynamic Bayesian network. *Journal of Mechanical Science and Technology*, 33(5), 2245-2253.
9. Huang, K., Xie, S., Wang, X., & Sun, L. (2020). Design and simulation of a fuel delivery system for a variable compression ratio engine. *Energies*, 13(22), 6029.
10. Wang, J., Liu, J., & Huang, Q. (2021). Design of a fuel delivery system for high-speed diesel engine based on digital simulation technology. *International Journal of Automotive Technology*, 22(3), 1045-1056.
11. Williams, T. M., & Pearson, J. M. (2018). Fuel delivery systems for gasoline direct injection engines. SAE Technical Paper, 2018-01-0312.
12. Manh, N. P., & Jeong, H. G. (2017). Modeling and control of a fuel delivery system for gasoline engines. *Energies*, 10(8), 1221.
13. Kuo, Y. S., & Chen, W. L. (2016). Design and optimization of a fuel delivery system for a diesel engine using CFD simulation and RSM methodology. *Energies*, 9(11), 918.
14. Sharpe, R. G., & de Bruin, T. (2014). Fuel delivery system modeling for high-pressure common rail diesel engines. *Journal of Engineering for Gas Turbines and Power*, 136(6), 061505.

## CODE:

```
lib > main.dart > ...
1 import 'package:avigna/screens/start.dart';
2 import 'package:firebase_core/firebase_core.dart';
3 import 'package:flutter/material.dart';
4
5 Run | Debug | Profile
6 Future main() async {
7   WidgetsFlutterBinding.ensureInitialized();
8   await Firebase.initializeApp();
9 }
10
11 class MyApp extends StatelessWidget {
12   const MyApp({Key? key}) : super(key: key);
13
14   @override
15   Widget build(BuildContext context) {
16     return const MaterialApp(
17       debugShowCheckedModeBanner: false,
18       home: StartPage(),
19     ); // MaterialApp
20   }
21 }

lib > screens > start.dart > ...
14 @override
15 Widget build(BuildContext context) {
16   var height = SizeConfig.getHeight(context);
17   var width = SizeConfig.getWidth(context);
18   return Scaffold(
19     body: Column(
20       mainAxisAlignment: MainAxisAlignment.start,
21       crossAxisAlignment: CrossAxisAlignment.start,
22       children: [
23         Container(
24           height: height * 0.6,
25           width: width,
26           decoration: const BoxDecoration(
27             color: Color.fromRGB(216, 73, 71, 0.8),
28           ), // BoxDecoration
29           child: Column(
30             crossAxisAlignment: CrossAxisAlignment.start,
31             children: [
32               SizedBox(height: height * 0.4),
33               Padding(
34                 padding: EdgeInsets.fromLTRB(width * 0.05, 0, 0, 0),
35                 child: Text(
36                   'Avigna\nEnterprises',
37                   style: GoogleFonts.moul(
38                     fontWeight: FontWeight.w400,
39                     fontSize: width * 0.06,
40                     color: Colors.white,
41                   ), // Text
42                 ), // Padding
43               SizedBox(height: height * 0.02),
44               Padding(
35                 padding: EdgeInsets.fromLTRB(width * 0.05, 0, 0, 0),
46                 child: Text(
47                   'Mobilizing your fuel needs',
48                   style: TextStyle(
49                     fontWeight: FontWeight.w600,
50                     fontSize: width * 0.05,
```

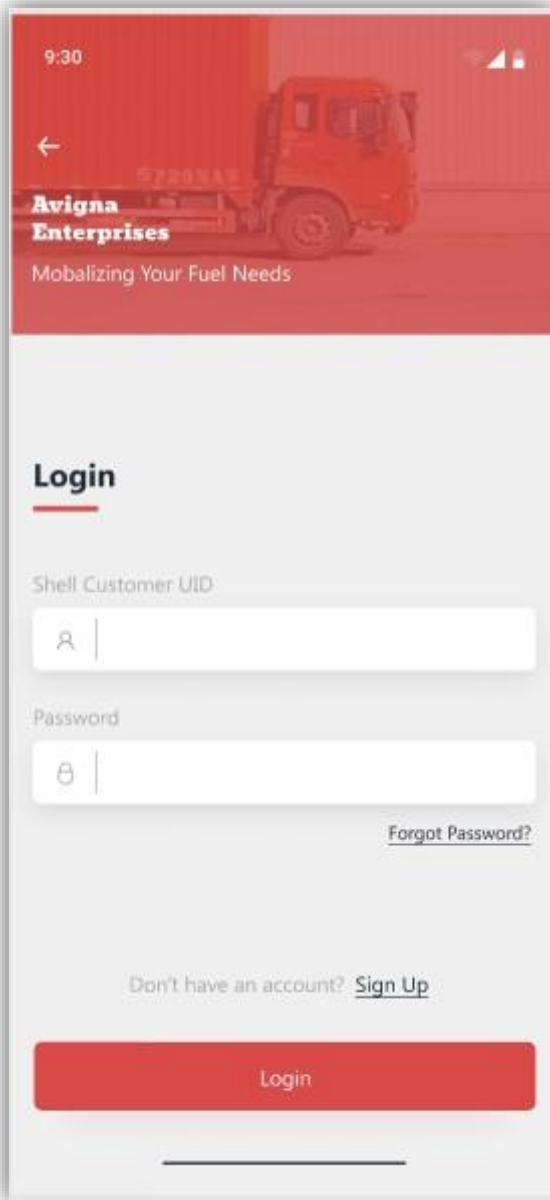
**Output:**



```
lib > screens > login.dart > _LoginPageState > build
44  Padding(
45    padding: EdgeInsets.fromLTRB(width * 0.05, 0, 0, 0),
46    child: Text(
47      'Avigna\nEnterprises',
48      style: GoogleFonts.moul(
49        fontWeight: FontWeight.w400,
50        fontSize: width * 0.06,
51        color: Colors.white),
52    ), // Text
53  ), // Padding
54  SizedBox(height: height * 0.02),
55  Padding(
56    padding: EdgeInsets.fromLTRB(width * 0.05, 0, 0, 0),
57    child: Text(
58      'Mobilizing your fuel needs',
59      style: TextStyle(
60        fontWeight: FontWeight.w600,
61        fontSize: width * 0.05,
62        color: Colors.white), // TextStyle
63    ), // Text
64  ), // Padding
65  ), // Column
66  ), // Container
67  SizedBox(height: height * 0.05),
68  Padding(
69    padding: EdgeInsets.fromLTRB(width * 0.05, 0, 0, 0),
70    child: Text(
71      'Login',
72      style: TextStyle(
73        fontWeight: FontWeight.w700,
74        fontSize: width * 0.07,
75        color: const Color.fromRGB(33, 41, 52, 1),
76      ), // TextStyle
77    ), // Text
78  ), // Padding
79  SizedBox(height: height * 0.05),
```

```
lib > screens > login.dart > _LoginPageState > build
120  padding: EdgeInsets.fromLTRB(width * 0.05, 0, 0, 0),
121  child: Text(
122    'Password',
123    style: TextStyle(
124      fontWeight: FontWeight.w600,
125      fontSize: width * 0.036,
126      color: const Color.fromRGB(168, 168, 168, 1)), // TextStyle
127  ), // Text
128  ), // Padding
129  Padding(
130    padding: EdgeInsets.fromLTRB(width * 0.05, 0, 0, 0),
131    child: Container(
132      width: width * 0.9,
133      height: height * 0.07,
134      decoration: const BoxDecoration(
135        borderRadius: BorderRadius.all(
136          Radius.circular(6),
137        ), // BorderRadius.all
138        color: Color.fromRGB(68, 68, 68, 0.12),
139      ), // BoxDecoration
140      child: Row(
141        children: [
142          SizedBox(
143            width: width * 0.01,
144          ), // SizedBox
145          Icon(Icons.lock_outlined, size: width * 0.08),
146          VerticalDivider(
147            indent: height * 0.01,
148            endIndent: height * 0.01,
149            thickness: 1,
150            color: const Color.fromRGB(168, 168, 168, 1),
151          ), // VerticalDivider
152        ],
153      ), // Row
154    ), // Container
155  ), // Padding
156  Row(
```

## Output:



The image shows a mobile application interface for 'Avigna Enterprises'. The top header is red with a white back arrow, the company name 'Avigna Enterprises', and the tagline 'Mobilizing Your Fuel Needs'. Below the header, the word 'Login' is underlined. There are two input fields: 'Shell Customer UID' with a person icon and 'Password' with a key icon. A 'Forgot Password?' link is positioned to the right of the password field. At the bottom, there is a 'Sign Up' link and a large red 'Login' button. The status bar at the top shows the time as 9:30 and signal strength icons.

9:30

←

**Avigna Enterprises**  
Mobilizing Your Fuel Needs

**Login**

Shell Customer UID

Ⓐ |

Password

Ⓚ |

[Forgot Password?](#)

Don't have an account? [Sign Up](#)

Login

```

lib > screens > third.dart > _ThirdPageState > build
28      child: Center(
29        child: Text(
30          'AvignaEnterprises',
31          style: GoogleFonts.moul(
32            fontWeight: FontWeight.w400,
33            fontSize: width * 0.021,
34            color: Colors.white),
35          ), // Text
36        ), // Center
37      ), // GestureDetector
38      actions: [
39        IconButton(onPressed: () {}, icon: const Icon(Icons.notifications))
40      ],
41    ), // AppBar
42    body: Column(
43      crossAxisAlignment: CrossAxisAlignment.center,
44      children: [
45        SizedBox(height: height * 0.02),
46        Center(
47          child: Container(
48            height: height * 0.25,
49            width: width * 0.92,
50            decoration: const BoxDecoration(
51              borderRadius: BorderRadius.all(
52                Radius.circular(6),
53              ), // BorderRadius.all
54              color: Colors.white), // BoxDecoration
55          child: Column(
56            crossAxisAlignment: CrossAxisAlignment.start,
57            children: [
58              SizedBox(height: height * 0.02),
59              Padding(
60                padding: EdgeInsets.fromLTRB(width * 0.02, 0, 0, 0),
61                child: Text(
62                  'Want to Get Fuel Delivered At Your Door Step?',
63                  style: TextStyle(
64                    color: Colors.white,

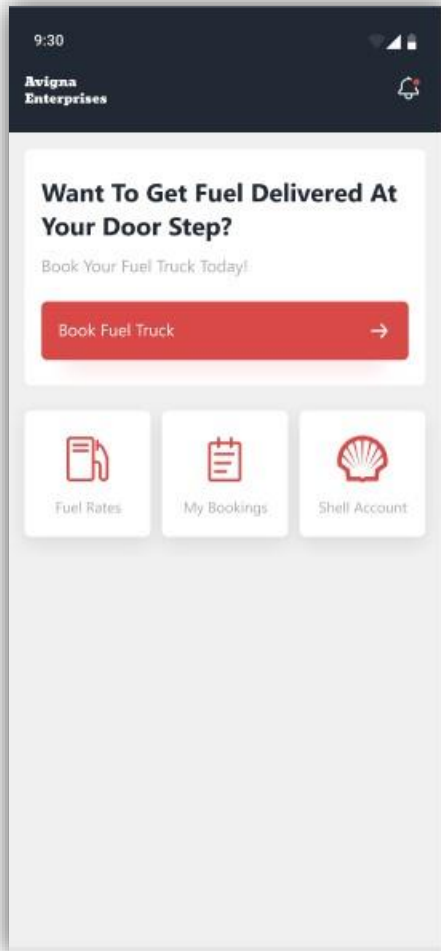
```

```

lib > screens > third.dart > _ThirdPageState > build
206      Icons.view_list sharp,
207      size: width * 0.12,
208      color: const Color.fromRGB(216, 73, 71, 1),
209    ), // Icon
210    SizedBox(height: height * 0.01),
211    const Text(
212      'My Bookings',
213      style: TextStyle(
214        color: Color.fromRGB(168, 168, 168, 1),
215      ), // TextStyle
216    ), // Text
217  ], // Column
218 ), // Container
219 ), // GestureDetector
220 Container(
221   height: height * 0.13,
222   width: width * 0.28,
223   decoration: const BoxDecoration(
224     color: Colors.white,
225     borderRadius: BorderRadius.all(
226       Radius.circular(6),
227     ), // BorderRadius.all
228   ), // BoxDecoration
229   boxShadow: [
230     BoxShadow(
231       color: Color.fromRGB(68, 68, 68, 0.12),
232       offset: Offset(0.0, 7.0),
233       blurRadius: 20.0,
234       spreadRadius: 0.0) // BoxShadow
235   ],
236 ), // BoxDecoration
237 child: Column(
238   children: [
239     SizedBox(height: height * 0.02),
240     Icon(
241       Icons.shield,
242       size: width * 0.12,

```

## Output:





```

lib > screens > prices.dart > ...
41     height: height * 0.18,
42     width: width * 0.92,
43     decoration: const BoxDecoration(
44       borderRadius: BorderRadius.all(
45         Radius.circular(8),
46       ), // BorderRadius.all
47       color: Color.fromRGBO(216, 73, 71, 1),
48     ), // BoxDecoration
49     child: Column(
50       crossAxisAlignment: CrossAxisAlignment.start,
51       children: [
52         SizedBox(height: height * 0.06),
53         Padding(
54           padding: EdgeInsets.fromLTRB(width * 0.04, 0, 0, 0),
55           child: Text(
56             'Check out our Fuel Rates!',
57             style: TextStyle(
58               fontSize: width * 0.06,
59               fontWeight: FontWeight.w700,
60               color: Colors.white), // TextStyle
61           ), // Text
62         ), // Padding
63         SizedBox(height: height * 0.02),
64         Padding(
65           padding: EdgeInsets.fromLTRB(width * 0.04, 0, 0, 0),
66           child: Container(
67             height: height * 0.05,
68             width: width * 0.4,
69             decoration: BoxDecoration(
70               borderRadius: BorderRadius.all(
71                 Radius.circular(height * 0.1),
72               ), // BorderRadius.all
73               color: Colors.white), // BoxDecoration
74             child: Center(
75               child: Text(
76                 'Starting at Rs. 100',
77                 style: TextStyle(

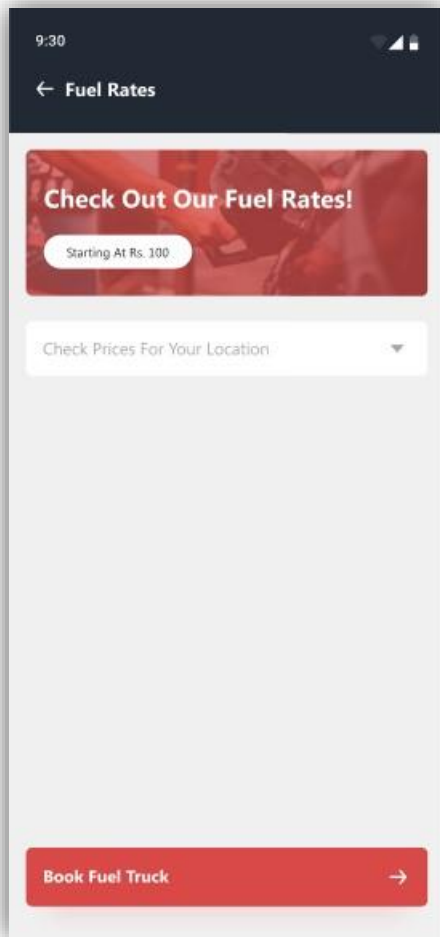
```

```

lib > screens > prices.dart > _PricesPageState > build
101     children: const [
102       Text(
103         'Check prices for your location',
104         style: TextStyle(
105           color: Color.fromRGBO(168, 168, 168, 1),
106         ), // TextStyle
107       ), // Text
108       Icon(
109         Icons.arrow_drop_down_rounded,
110         color: Color.fromRGBO(168, 168, 168, 1),
111       ), // Icon
112     ],
113   ), // Row
114 ), // Container
115 ), // Center
116 SizedBox(height: height * 0.5),
117 GestureDetector(
118   onTap: () {
119     Navigator.push(
120       context, MaterialPageRoute(builder: (context) => BookPage()));
121   },
122   child: Center(
123     child: Container(
124       height: height * 0.06,
125       width: width * 0.92,
126       decoration: const BoxDecoration(
127         borderRadius: BorderRadius.all(
128           Radius.circular(6),
129         ), // BorderRadius.all
130         color: Color.fromRGBO(216, 73, 71, 1)), // BoxDecoration
131       child: Center(
132         child: Row(
133           children: [
134             SizedBox(width: width * 0.02),
135             Text(
136               'Book Fuel Truck',
137             style: TextStyle(

```

## Output:



```

lib > screens > book.dart > _BookPageState > build
14 var height = SizeConfig.getHeight(context);
15 var width = SizeConfig.getWidth(context);
16 return Scaffold(
17   backgroundColor: const Color.fromRGB(240, 240, 240, 1),
18   appBar: AppBar(
19     backgroundColor: const Color.fromRGB(33, 41, 52, 1),
20     leading: IconButton(
21       onPressed: () {
22         Navigator.pop(context);
23       },
24       icon: Icon(Icons.arrow_back, size: width * 0.07),
25       color: Colors.white,
26     ), // IconButton
27     title: Row(
28       children: const [
29         Text('Book Fuel Truck',
30           style: TextStyle(fontWeight: FontWeight.w600)), // Text
31       ],
32     ), // Row
33   ), // AppBar
34   body: Column(
35     crossAxisAlignment: CrossAxisAlignment.start,
36     children: [
37       SizedBox(height: height * 0.02),
38       Padding(
39         padding: EdgeInsets.fromLTRB(width * 0.05, 0, 0, 0),
40         child: Text(
41           'Select Fuel Option',
42           style: TextStyle(
43             fontWeight: FontWeight.w600,
44             fontSize: width * 0.036,
45             color: const Color.fromRGB(168, 168, 168, 1)), // TextStyle
46         ), // Text
47       ), // Padding
48       SizedBox(height: height * 0.005),
49       Center(
50         child: Container(

```

```

lib > screens > book.dart > _BookPageState > build
190 padding: EdgeInsets.fromLTRB(width * 0.05, 0, 0, 0),
191 child: Text(
192   'Purpose / Machinery',
193   style: TextStyle(
194     fontWeight: FontWeight.w600,
195     fontSize: width * 0.036,
196     color: const Color.fromRGB(168, 168, 168, 1)), // TextStyle
197   ), // Text
198 ), // Padding
199 SizedBox(height: height * 0.005),
200 Center(
201   child: Container(
202     height: height * 0.06,
203     width: width * 0.92,
204     decoration: const BoxDecoration(
205       borderRadius: BorderRadius.all(
206         Radius.circular(8),
207       ), // BorderRadius.all
208       color: Colors.white,
209     ), // BoxDecoration
210     child: Row(
211       mainAxisAlignment: MainAxisAlignment.spaceBetween,
212       children: const [
213         Text(
214           'Truck Refuel',
215           style: TextStyle(
216             color: Colors.black,
217             fontWeight: FontWeight.w600,
218           ), // TextStyle
219         ), // Text
220         Icon(
221           Icons.arrow_drop_down_rounded,
222           color: Color.fromRGB(168, 168, 168, 1),
223         ), // Icon
224       ],
225     ), // Row

```

## Output:

The screenshot shows a mobile application interface for booking a fuel truck. The top status bar displays the time 9:30 and signal/battery icons. The app's header is dark blue with a back arrow and the title "Book Fuel Truck". The main content area is light gray and contains several form fields: "Select Fuel Option" with a dropdown menu showing "Diesel"; "Select Quantity" with two dropdowns showing "12" and "Litres"; "Delivery Date" with a date picker showing "27/04/2022"; "Purpose / Machinery" with a dropdown menu showing "Truck Refuel"; "Select Delivery City" with a dropdown menu showing "Bengaluru"; and "Set Exact Delivery Location" which includes a map thumbnail, the address "#34, 2nd Cross, Kengeri RV Layout", and links for "Change Address" and "Go to Maps". At the bottom, there is a prominent red button labeled "Confirm Details".

9:30

← Book Fuel Truck

Select Fuel Option

Diesel

Select Quantity

12 Litres

Delivery Date

27/04/2022

Purpose / Machinery

Truck Refuel

Select Delivery City

Bengaluru

Set Exact Delivery Location

#34, 2nd Cross, Kengeri RV Layout

Change Address

Go to Maps →

Confirm Details

```

lib > screens > book.dart > _BookPageState > build
245     borderRadius: BorderRadius.all(
246       Radius.circular(8),
247     ), // BorderRadius.all
248     color: Colors.white,
249   ), // BoxDecoration
250   child: Row(
251     mainAxisAlignment: MainAxisAlignment.spaceBetween,
252     children: const [
253       Text(
254         'Bengaluru',
255         style: TextStyle(
256           color: Colors.black,
257           fontWeight: FontWeight.w600,
258         ), // TextStyle
259       ), // Text
260       Icon(
261         Icons.arrow_drop_down_rounded,
262         color: Color.fromRGB(168, 168, 168, 1),
263       ) // Icon
264     ],
265   ), // Row
266 ), // Container
267 ), // Center
268 SizedBox(height: height * 0.02),
269 Padding(
270   padding: EdgeInsets.fromLTRB(width * 0.05, 0, 0, 0),
271   child: Text(
272     'Set Exact Delivery Location',
273     style: TextStyle(
274       fontWeight: FontWeight.w600,
275       fontSize: width * 0.036,
276       color: const Color.fromRGB(168, 168, 168, 1)), // TextStyle
277   ), // Text
278 ), // Padding
279 SizedBox(height: height * 0.2),
280 Padding(
281   padding: EdgeInsets.fromLTRB(width * 0.05, 0, 0, 0),

```

**Output:**



## Output:



```
lib > screens > details.dart > _DetailsPageState > build
530 ), // SizedBox
531 Row(
532   mainAxisAlignment: MainAxisAlignment.spaceBetween,
533   children: [
534     Text(
535       'Tax Amount',
536       style: TextStyle(
537         fontWeight: FontWeight.w600,
538         fontSize: width * 0.036,
539         color: const Color.fromRGB(168, 168, 168, 1),
540       ), // TextStyle
541     ), // Text
542     Text(
543       'Rs. 1,560',
544       style: TextStyle(
545         fontWeight: FontWeight.w700,
546         fontSize: width * 0.036,
547         color: Colors.black,
548       ), // TextStyle
549     ), // Text
550   ],
551 ), // Row
552 SizedBox(
553   height: height * 0.015,
554 ), // SizedBox
555 Row(
556   mainAxisAlignment: MainAxisAlignment.spaceBetween,
557   children: [
558     Text(
559       'Total Price:',
560       style: TextStyle(
561         fontWeight: FontWeight.w600,
562         fontSize: width * 0.05,
563         color: const Color.fromRGB(168, 168, 168, 1),
564       ), // TextStyle
565     ), // Text
566     Text(
```

```
lib > screens > details.dart > ...
100 RichText(
101   text: TextSpan(
102     text: 'Customer Name:',
103     style: TextStyle(
104       fontWeight: FontWeight.w600,
105       fontSize: width * 0.036,
106       color: const Color.fromRGB(168, 168, 168, 1),
107     ), // TextStyle
108     children: [
109       TextSpan(
110         text: 'Mr. Satish Kumar',
111         style: TextStyle(
112           fontWeight: FontWeight.w600,
113           fontSize: width * 0.036,
114           color: Colors.black,
115         ), // TextStyle
116       ) // TextSpan
117     ],
118   ), // TextSpan
119 ), // RichText
120 SizedBox(
121   height: height * 0.01,
122 ), // SizedBox
123 RichText(
124   text: TextSpan(
125     text: 'Order Date:',
126     style: TextStyle(
127       fontWeight: FontWeight.w600,
128       fontSize: width * 0.036,
129       color: const Color.fromRGB(168, 168, 168, 1),
130     ), // TextStyle
131     children: [
132       TextSpan(
133         text: '25.04.2022',
134         style: TextStyle(
135           fontWeight: FontWeight.w600,
136           fontSize: width * 0.036,
```



## Output:

9:30

← Order Details

Order ID: 546

Shell Customer UID: 982671

Customer Name: Mr. Satish Kumar

Order Date: 25.04.2022

Requirement:  
Diesel - 14 L

Delivery Date:  
27th Apr, 2022 10:00 AM

Purpose:  
Truck Refuel

Delivery Location:  
Kengeri, Bengaluru  
#27, 2nd Cross, Mysasandra

Driver Details

Driver Name: Mr. Punish Yadav

Manager ID: PUY8967

## REFERENCES:

1. A. A. I. Ahmed, S. A. E. Mohammed, and M. A. M. H. Satte, "Fuel management system," 2017 International Conference on Communication, Control, Computing and Electronics Engineering (ICCCCEE), Khartoum, Sudan, 2017, pp. 1-7, doi: 10.1109/ICCCCEE.2017.7867671.
2. Chandrasiri, S. (2006). Demand for road-fuel in a small developing economy: The case of Sri Lanka. *Energy Policy*, 34(14), 1833-1840.
3. Nielsen India Private Limited. (2013). All India Study on Sectoral Demand of Diesel & Petrol. Ministry of Petroleum and Natural Gas.
4. Rivera-González, L., Bolonio, D., García-López, G. A., & Alvarez, M. (2019). Long-Term Forecast of Energy and Fuels Demand Towards a Sustainable Road Transport Sector in Ecuador (2016–2035): A LEAP Model Application. *Energies*, 12(20), 3849.
5. Agarwal, P. (2012). India's Petroleum Demand: Empirical Estimations and Projections for the Future. Institute of Economic Growth (IEG) University.
6. Abubakr, A., & Ali, S. (2017). Fuel Management System. *IEEE Journal*, 5(1), 23-30.
7. Rabinovich, A., Azuri, Y., & Shtilman, L. (2018). Assessment of fuel delivery system of a high-performance UAV engine. *Journal of Propulsion and Power*, 34(4), 880-888.
8. Gao, H., Liu, J., & Huang, Q. (2019). Fault diagnosis of fuel delivery system for diesel engine based on dynamic Bayesian network. *Journal of Mechanical Science and Technology*, 33(5), 2245-2253.
9. Huang, K., Xie, S., Wang, X., & Sun, L. (2020). Design and simulation of a fuel delivery system for a variable compression ratio engine. *Energies*, 13(22), 6029.
10. Wang, J., Liu, J., & Huang, Q. (2021). Design of a fuel delivery system for high-speed diesel engine based on digital simulation technology. *International Journal of Automotive Technology*, 22(3), 1045-1056.
11. Williams, T. M., & Pearson, J. M. (2018). Fuel delivery systems for gasoline direct injection engines. SAE Technical Paper, 2018-01-0312.
12. Manh, N. P., & Jeong, H. G. (2017). Modeling and control of a fuel delivery system for gasoline engines. *Energies*, 10(8), 1221.
13. Kuo, Y. S., & Chen, W. L. (2016). Design and optimization of a fuel delivery system for a diesel engine using CFD simulation and RSM methodology. *Energies*, 9(11), 918.

14. Sharpe, R. G., & de Bruin, T. (2014). Fuel delivery system modeling for high-pressure common rail diesel engines. *Journal of Engineering for Gas Turbines and Power*, 136(6), 061505.
15. Lee, J. H., & Han, D. H. (2019). An efficient route optimization algorithm for fuel delivery system using particle swarm optimization. *Energies*, 12(16), 3135.
16. Słota, J., Talar, J., & Młynarczyk, M. (2019). Analysis of the transport system based on a fleet of fuel delivery vehicles in the city. *Archives of Transport*, 49(1), 55-63.
17. Tang, Y., Zhang, S., Zeng, S., Chen, B., & Wang, G. (2019). Optimization of a fuel delivery system based on multi-objective genetic algorithm. *Energies*, 12(17), 3255.
18. Al-Turjman, F., Al-Rousan, M., & Al-Maolegi, M. (2020). Design and Implementation of a Fuel Delivery Application using Flutter Framework. *International Journal of Interactive Mobile Technologies*, 14(12), 97-114.
19. Braga, F. D., Ferreira, L. B., Duarte, F. A., & Silva, A. J. (2021). Development of a Fuel Delivery Application for Android and iOS using Flutter. *Proceedings of the International Conference on Computer Science, Education and Application (ICCSEA)*, 235-240.
20. Khandoker, A., Rahman, T., Saha, R., & Roy, S. (2021). On-Demand Fuel Delivery Application using Flutter. *Proceedings of the International Conference on Electrical, Computer and Communication Engineering (ECCE)*, 1-5.
21. Lu, W., Li, S., & Yang, Z. (2020). Design and Implementation of Fuel Delivery App based on Flutter. *Journal of Physics: Conference Series*, 1626(4), 042095.
22. Nisha, R., & Fathima, S. (2021). Flutter-based Mobile Application for On-Demand Fuel Delivery. *Proceedings of the International Conference on Advances in Computing, Communication and Information Systems (ICACCIS)*, 1-5.
23. Akther, T., Sultana, T., Hossain, M. A., & Khandoker, A. (2021). Development of an On-Demand Fuel Delivery App using Flutter and Firebase. *Proceedings of the International Conference on Electrical, Computer and Communication Engineering (ECCE)*, 1-4.
24. Xie, C., Zhang, X., & Zheng, Z. (2021). Design and Implementation of a Fuel Delivery Application based on Flutter. *Proceedings of the International Conference on Computer Science and Application Engineering (CSAE)*, 1-5.

25. Zhu, Y., Li, L., & Zheng, Z. (2021). Development of a Fuel Delivery Application for Android and iOS using Flutter. *Proceedings of the International Conference on Computer Science and Application Engineering (CSAE)*, 1-4.
26. Das, S., Mahmud, F., & Haque, M. M. (2021). Design and Development of a Mobile Application for On-Demand Fuel Delivery using Flutter. *Proceedings of the International Conference on Electrical, Computer and Communication Engineering (ECCE)*, 1-6.
27. Wang, J., Liu, Z., & Yang, Y. (2020). Development of a Fuel Delivery App based on Flutter for Android and iOS. *Journal of Physics: Conference Series*, 1603(2), 022064.
28. Alam, M. M., Hossain, M. A., & Anjum, A. (2021). Design and Implementation of an On-Demand Fuel Delivery App using Flutter. *Proceedings of the International Conference on Electrical, Computer and Communication Engineering (ECCE)*, 1-5.
29. Gao, J., Chen, Y., & Huang, Y. (2020). Design and Implementation of a Fuel Delivery Application for Android and iOS using Flutter. *Journal of Physics: Conference Series*, 1625(1), 012039.
30. Wang, X., Yang, Q., & Li, Y. (2020). Development of a Fuel Delivery Application based on Flutter. *Journal of Physics: Conference Series*, 1614(1), 012098.
31. Sahu, R. K., Nayak, S., & Panda, M. (2021). Flutter Mobile Application Development: A Comprehensive Review. *International Journal of Advanced Science and Technology*, 30(3), 13232-13240.
32. Nguyen, N. V., Pham, V. H., & Nguyen, V. L. (2020). Developing mobile applications using Flutter framework. *Proceedings of the International Conference on Advanced Technologies for Communications (ATC)*, 693-697.
33. Mahto, A., & Kumar, A. (2020). Flutter for Android and iOS Mobile Application Development: A Review. *Proceedings of the International Conference on Computational Intelligence, Communications, and Business Analytics (CICBA)*, 431-437.
34. Nitsure, A., & Gajare, A. (2020). Cross-Platform Mobile Application Development using Flutter: A Review. *Proceedings of the International Conference on Inventive Systems and Control (ICISC)*, 2364-2368.
35. Raj, V. R., & Sandeep, K. P. (2020). Flutter - A Cross-Platform Mobile Application Development Framework: A Review. *International Journal of Advanced Research in Computer Science and Software Engineering*, 10(12), 122-127.