

FUEL DELIVERY ON DEMAND APPLICATION

A Main Project Report

submitted by

SOORAJ M(MES20MCA-2052)

to

the APJ Abdul Kalam Technological University
in partial fulfillment of the requirements for the award of the Degree

of

Master of Computer Applications



Department of Computer Applications

MES College of Engineering
Kuttippuram, Malappuram - 679 582

July 2022

DECLARATION

I undersigned hereby declare that the project report **FUEL DELIVERY ON DEMAND APPLICATION**, submitted for partial fulfillment of the requirements for the award of degree of Master of Computer Applications of the APJ Abdul Kalam Technological University, Kerala, is a bona fide work done by me under supervision of Mr. Muhammad Jabir C, Assistant Professor, Department of Computer Applications. This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Place:Kuttippuram

SOORAJ M(MES20MCA-2052)

Date:06-07-2022

**DEPARTMENT OF COMPUTER APPLICATIONS
MES COLLEGE OF ENGINEERING, KUTTIPPURAM**



CERTIFICATE

This is to certify that the report entitled **FUEL DELIVERY ON DEMAND APPLICATION** is a bona fide record of the Main Project work carried out by **SOORAJ M(MES20MCA-2052)** submitted to the APJ Abdul Kalam Technological University, in partial fulfillment of the requirements for the award of the Master of Computer Applications, under my guidance and supervision. This report in any form has not been submitted to any other University or Institution for any purpose.

Internal Supervisor(s)

External Supervisor(s)

Head Of The Department

Acknowledgements

At the very outset I would like to thank the almighty's mercy towards me over the years... I wish to express my sincere thanks to my project guide Ms. Priya J.D, Assistant Professor, Dept. of Computer Applications who guided me for the successful completeness of this project. I also thank her for valuable suggestions, guidance, constant encouragement, boundless corporation, constructive comments and motivation extended to me for completion of this project work. I would express my sincere thanks to my internal guide Mr. Muhammad Jabir C, Assistant Professor, for his immense guidance to complete the project successfully. I would like to express my sincere thanks to all the faculty members of Department of Computer Applications for their support and valuable suggestion for doing the project work. Last but not least my graceful thanks to my parents, friends and also the persons who supported me directly and indirectly during the project.

SOORAJ M(MES20MCA-2052)

Abstract

Due to the growth of vehicles, fuel consumption became more. In our daily life, if vehicle stops, unfortunately, due to lack of fuel, the owner have to push the vehicle to the nearest fuel station which will be very difficult to the owner. So this is a problem which can cause loss of time and hard work by the owner. For senior citizens and people with medical issues, it will get even harder. In this cases the Fuel delivery on demand application can be used to deliver the fuel to those who need to refuel vehicles at any location at any time. This system uses GPS to get the location of the user. The Fuel Delivery on Demand application helps in the delivery of fuel depends on the user order or request. By using this application the user can select the type of fuel required, order and get the fuel in the place they are standing, which makes the process easier for the user. Admin approves the Fuel Station by verifying the details. Fuel Station can sell the fuel to the user by the demand of the user. User can search the fuel station on the basis of locality and order the fuel on online app which can be delivered by delivery agents.

Keywords:Fuel delivery, GPS

Contents

Declaration	i
Certificate	ii
Acknowledgements	iii
Abstract	iv
Contents	v
List of Figures	vii
List of Tables	ix
1 Introduction	1
1.1 Background	1
1.1.1 Motivation	2
1.2 Objective	2
1.3 Report Organization	2
2 Literature Survey	3
3 Methodology	4
3.1 Introduction	4
3.2 Modules	5
3.2.1 Admin	5
3.2.2 Fuel Station	5
3.2.3 User	6
3.2.4 Delivery Agent	6
3.3 Working	7
3.4 Developing Environment	8
3.5 Agile Methodology	9
3.5.1 User Story	9
3.5.2 Product Backlog	10
3.5.3 Project Plan	11

3.5.4	Sprint Plan	12
3.5.5	Sprint Actual	13
4	Results and Discussions	14
4.0.1	Screenshot	14
5	Conclusions	23
	References	24
Appendix		25
	Source Code	25

List of Figures

4.1	Login	14
4.2	Registration	15
4.3	Admin Home Page	15
4.4	Manage Fuel Station	16
4.5	Manage Delivery Agent	16
4.6	Complaints	17
4.7	Fuel Orders	17
4.8	Assign Delivery Agent	18
4.9	User Login	18
4.10	Sign Up	19
4.11	User Registration	19
4.12	Search Fuel Station	20
4.13	Order Fuel	20
4.14	Payment	21
4.15	Post Complaint	21
4.16	Delivery Agent Registration	22
4.17	Assigned Delivery	22
A.1	DFD-Level 0	31
A.2	DFD-Level 1	31
A.3	DFD-Level 2	32
A.4	DFD-Level 3	32

A.5 DFD-Level 4	33
---------------------------	----

List of Tables

3.1	User Story	9
3.2	Product Backlog	10
3.3	Project Plan	11
3.4	Sprint Plan	12
3.5	Sprint Actual	13
A.1	User Registration	27
A.2	Login	27
A.3	Fuel Station Registration	28
A.4	Delivery Agent Registration	28
A.5	Fuel Order	29
A.6	Payment	29
A.7	Complaint	30

Chapter 1

Introduction

1.1 Background

Fuel is one of the important factor that influencing our daily life. No one can imagine a day without using any kind of fuel. Fuels like petrol and diesel are widely used by vehicles in our day-to-day life. By considering past several years, the growth in the usage of vehicle is increasing. Thus due to the growth in the usage of vehicles, fuel consumption became more. While traveling, we see, people pushing vehicle to the fuel station or vehicle got stuck by roadside, which is a usual scenario, due to the lack of fuel. There is no other way rather than reaching fuel station for refueling vehicles. This makes every difficult for the people to refuel the vehicle. This makes even harder for the older, disabled or people with medical issues. The whole current scenario can be changed by using a mobile application in which the refueling of vehicles can be done without going to the fuel station. It will be more easy if a refueling mechanism available at people's demand. The mentioned system helps people to refuel vehicle at their location. By this system users can select the type of fuel and quantity needed. The system works with the help of the GPS, which will be more easy to track the user's location and there will be no need of guiding the delivery agent. This is a mobile application which will be easy for the users to use, as the majority of people uses the smart phone.

1.1.1 Motivation

Fuel delivery on demand by the users at their location makes vehicle refueling process more easier when they are away from the fuel station. The application mainly aims to help the people who gets trapped by lack of fuel for their vehicle. By providing this service a lot of time and effort can be saved.

1.2 Objective

Fuel delivery on demand application provides the user a refueling system, if the vehicle stops by lack of fuel at any location. The major objectives are:-

- Reduce time and effort.
- Handling emergencies.
- Delivery of fuel at user's location.
- Users can view nearest fuel stations

1.3 Report Organization

The rest of the project report is divided into four sections. Section 2 describes literature survey. Section 3 describes the methodology used for implementing the project. Section 4 gives the results and discussions. Finally Section 5 gives the conclusion.

Chapter 2

Literature Survey

Fuel delivery on demand application is the perfect way to deliver the fuel to the customers. As we saw that in real time food delivery app is available and is business growing fast and we think that as face a problem for fuel delivery is a big critical problem in emergency time for demand of fuel. It's services and functionality all are same as they perform in food ordering system. They are app-based services and work like Uber for fuel. That is, a person who needs to refuel his vehicle without visiting a fuelstation, downloads and registers on the app.

This study used a sample data set provided by a San Francisco Bay Area company to explore the potential impacts on vehicle miles traveled (VMT), carbon dioxide (CO₂) emissions, and traffic congestion. An analysis of vehicle travel associated with fuel station trips was conducted to establish a basis for comparison. Future scenarios comparing the potential impacts of scaled-up fuel delivery services in 2030 were also developed. The study concluded that fuel delivery services compared favorably to fuel stations in relation to environmental and traffic benefits in the longer term, even though personal fueling trips tended to generate low VMT. Benefits stemmed from efficiencies achieved by fueling multiple vehicles per delivery trip, replacing car share vehicle fueling trips, and removing trips from the network during peak hours. This analysis estimated that total annual CO₂ emissions associated with fuel delivery operations in the Bay Area were 76 metric tons, which is less than a typical gas station with 97 metric tons. Under assumptions of declining demand for fuels and significantly fewer fuel stations, and with highly efficient optimized operations, mobile delivery could gain up to 5 percentage market share for fuel and not add additional VMT over the business as usual scenario.

Chapter 3

Methodology

3.1 Introduction

Fuel delivery on demand application is a mobile application that helps user to refuel vehicle on road, unfortunately, if vehicle stops due to the lack of fuel. User can locate the nearest fuel station and place order for the fuel which will be delivered by a delivery agent. The system basically depends on GPS service. The system helps the user to view the fuel stations which registered in the system and place order for fuel. The system helps the delivery agent to get the location of the user who placed the order. The system consist of modules such as Admin, Fuel Station, User and Delivery Agent. The Admin and Fuel Station modules are represented as web application, created using python, django. The rest two modules are done as mobile application by considering the experience of the user, done using dart, flutter.

Django is a high-level open source Python web framework that encourages rapid development and clean, pragmatic design and it is also versatile in nature which allows to build applications for different-different domains. It have simple syntax, HTTP libraries, Middleware support and a Python unit test framework. PyCharm IDE is used for doing the web framework in this project. It is a dedicated Python Integrated Development Environment (IDE) providing a wide range of essential tools for developers.

Flutter framework is used for creating the mobile application part. Flutter is Google's free and open-source UI framework for creating native mobile applications. This framework contains SDK and their widget based UI library. Android Studio is the official Integrated Development Environment (IDE) for android app development which is used in this project.

Dart is a programming language designed for client development which is used in this framework. This language mainly focus on the development of front end. Dart is much easier to use so that it helps the developers to get tremendous control over the system.

The database system used in this project is the MySQL relational database management system. This database management system is based on SQL(Structured Query Language) queries. It can handle almost any amount of data, up to as much as 50 million rows or more.

3.2 Modules

3.2.1 Admin

- Login
- Verify fuel station
- Verify delivery agent
- View complaint
- Post reply

3.2.2 Fuel Station

- Register
- Login
- Receive order
- Assign delivery
- Receive payment

3.2.3 User

- Register
- Login
- Search Fuel Station
- Place order
- Make payment
- Post complaint
- View reply
- Get delivery

3.2.4 Delivery Agent

- Register
- Login
- View delivery request
- Update delivery Status
- Locate user
- Delivery

3.3 Working

In this project I am developing a web and mobile application for fuel delivery. The basic objective of the system is to deliver fuel to the people who got stuck by the road due to the lack of fuel. The project mainly contain four modules as follows.

- 1) Admin 2)Fuel Station 3)User 4)Delivery Agent

Admin controls the overall system which is done by web. Admin login to the the system to track or initiate activities happening in the system. Admin is the person who have the control over the software by already presetting the application. Admin accepts the service of fuel stations and delivery agents through a registration procedure. Admin allows them only after verifying the details uploaded by them. Admin have all the power to block or unblock both the fuel station and delivery agent according to their need for the service. The same is done with users too. Admin can view the complaints posted by the users and add a reply to them. By this action the evaluation of the system can be done.

The next module in the system is Fuel Station which sell fuel to the user,done as web application. Each and every fuel station which is intended to join the system has to register in to the system by providing basic details and authorised documents. Later this will be verified by the admin. After admins approval fuel station can enter into the system. Every fuel station can add their fuel price which can be viewed by the user at the time of order. After getting user's order fuel station can assign delivery agent to deliver the fuel. At the time of user placing a order and after delivery the fuel station get an sms alert. Order is placed after receiving the payment from the user. By assigning a delivery agent, fuel station shares the location details of the user to the respected delivery agent.

The third module in the system is the User which is done as mobile application. Users are the normal people who orders fuel when they got stuck by road. User who want to access the system has to register and login to the system .Users can view the registered fuel station in the system by searching and they can find the fuel station near them by verifying location details. They can order the fuel by selecting fuel station and by selecting the fuel type and quantity. They can finally place the order by completing payment online. They will be alerted by an sms after the payment that the order is placed. They have to wait for certain minutes to get the fuel which will be delivered by the delivery agent. User can post complaint, if any,

which will be replied and solved by the admin.

The fourth and final module is Delivery Agent which is also done as mobile application. They are the ones who delivers fuel from fuel station to the user. They will get an sms alert whenever they got assigned for a delivery by a fuel station. They can locate the user using GPS system. After the delivery the agent should update the status as delivered which will be known by fuel station.

3.4 Developing Environment

- Operating System - Windows 7 or Above, Android
- Front End - HTML5, CSS3, Dart
- Backend - Python, MySql
- Platform used - PyCharm , Android Studio
- Web Browser - Microsoft Edge
- Frame work -Flutter,Django

3.5 Agile Methodology

This project was developed using Agile Development Model. The Agile methodology is a way to manage a project by breaking it up into several phases. It involves constant collaboration with stakeholders and continuous improvement at every stage. The entire project was divided into five sprints. Designing of web application was done in first two sprints. Designing of back end was done in the next three sprints.

3.5.1 User Story

User StoryID	As a <type of user>	I want to	So that I can
1	Admin	Login	Access Home Page
2	Admin	Manage Fuel Station	Reject/Accept Fuel Station
3	Admin	View Complaint	View the complaint posted
4	Admin	Post Reply	Give response
5	Fuel Station	Registration	Register into the system
6	Fuel Station	Login	Access Home Page
7	Fuel Station	View Order	View the user's order for fuel
8	Fuel Station	Assign Delivery	Put an agent for fuel delivery
9	Fuel Station	View Status	View the status of the delivery
10	Fuel Station	Accept Payment	Charge the user for the service
11	User	Registration	Register into the system
12	User	Login	Access Home Page
13	User	Search Fuel Station	View available fuel stations
14	User	Place Order	Get fuel stations
15	User	View Status	View status of the order
16	User	Post Complaint	Share complaint and view reply
17	User	Make Payment	Pay the amount
18	Delivery Agent	Register	Register into the system
19	Delivery Agent	Login	Access Home Page
20	Delivery Agent	View Delivery Request	Deliver the fuel
21	Delivery Agent	Update Delivery Status	Complete the delivery process

Table 3.1: User Story

3.5.2 Product Backlog

User Story ID	Priority <High/Medium/Low>	Size (Hours)	Sprint <#>	Status <Planned/In progress/Completed	Release Date	Release Goal
1	Medium	5	1	Completed	21/04/2022	Creation Of Login Page
2	High	5		Completed	26/04/2022	Manage Fuel Station
3	Medium	5	2	Completed	29/04/2022	View Complaint
4	Medium	5		Completed	30/04/2022	Post Reply
5	Medium	4	3	Completed	03/05/2022	Fuel Station Registration
6	Medium	2		Completed	03/05/2022	Fuel Station Login
7	Medium	4		Completed	11/05/2022	View User's Order
8	High	2	4	Completed	23/05/2022	Assign Delivery
9	Medium	1		Completed	27/05/2022	View Delivery Status
10	Medium	1		Completed	31/06/2022	Accept Payment
11	Medium	2		Completed	08/06/2022	User Registration
12	Medium	1		Completed	13/06/2022	User Login
13	High	2		Completed	15/06/2022	Search Fuel Station
14	Medium	1		Completed	17/06/2022	Order Placement
15	High	1	5	Completed	21/06/2022	View Order Status
16	Medium	1		Completed	22/06/2022	Add Complaint
17	Medium	2		Completed	24/06/2022	Make Payment
18	Medium	1		Completed	27/06/2022	Delivery Agent Registration
19	Medium	1		Completed	29/06/2022	Delivery Agent Login
20	High	3		Completed	04/07/2022	View Delivery Request
21	Medium	1		Completed	06/07/2022	Update Delivery Status

Table 3.2: Product Backlog

3.5.3 Project Plan

User StoryID	Task Name	Start Date	End Date	Days	Status
1	Sprint 1	20/04/2022	21/04/2022	2	Completed
2		25/04/2022	26/04/2022	2	Completed
3	Sprint 2	29/05/2022	29/05/2022	1	Completed
4		30/05/2022	30/05/2022	1	Completed
5	Sprint 3	02/05/2022	03/05/2022	1	Completed
6		03/05/2022	03/05/2022	1	Completed
7		09/05/2022	11/05/2022	3	Completed
8	Sprint 4	20/05/2022	23/05/2022	4	Completed
9		26/05/2022	27/06/2022	2	Completed
10		30/05/2022	31/05/2022	2	Completed
11		06/06/2022	08/06/2022	3	Completed
12		13/06/2022	13/06/2022	1	Completed
13		14/06/2022	15/06/2022	2	Completed
14		17/06/2022	17/06/2022	1	Completed
15	Sprint 5	20/06/2022	21/06/2022	2	Completed
16		22/05/2022	22/06/2022	1	Completed
17		23/06/2022	24/06/2022	2	Completed
18		27/06/2022	27/06/2022	1	Completed
19		29/06/2022	29/06/2022	1	Completed
20		04/07/2022	04/07/2022	1	Completed
21		06/07/2022	06/07/2022	1	Completed

Table 3.3: Project Plan

3.5.4 Sprint Plan

Backlog Item	Status & completion date	Original estimate in hours	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14
UserStory #1,#2																
UI Designing	20/04/2022	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0
Database Connectivity	21/04/2022	2	0	2	0	0	0	0	0	0	0	0	0	0	0	0
Coding	26/04/2022	4	0	0	2	0	0	0	2	0	0	0	0	0	0	0
Testing	26/04/2022	2	0	0	0	0	0	0	2	0	0	0	0	0	0	0
UserStory #3,#4																
UI Designing	29/04/2022	2	0	2	0	0	0	0	0	0	0	0	0	0	0	0
Database Connectivity	29/04/2022	2	0	2	0	0	0	0	0	0	0	0	0	0	0	0
Coding	30/04/2022	4	0	0	4	0	0	0	0	0	0	0	0	0	0	0
Testing	30/04/2022	2	0	0	0	2	0	0	0	0	0	0	0	0	0	0
UserStory #5,#6,#7																
UI Designing	03/05/2022	3	3	0	0	0	0	0	0	0	0	0	0	0	0	0
Database Connectivity	03/04/2022	2	0	2	0	0	0	0	0	0	0	0	0	0	0	0
Coding	03/05/2022	3	0	0	3	0	0	0	0	0	0	0	0	0	0	0
Testing	11/05/2022	2	0	0	0	2	0	0	0	0	0	0	0	0	0	0
UserStory #8,#9,#10,#11,#12,#13,#14																
UI Designing	27/05/2022	4	2	1	1	0	0	0	0	0	0	0	0	0	0	0
Database Connectivity	27/05/2022	2	0	2	0	0	0	0	0	0	0	0	0	0	0	0
Coding	17/06/2022	2	0	2	0	0	0	0	0	0	0	0	0	0	0	0
Testing	17/06/2022	2	0	2	0	0	0	0	0	0	0	0	0	0	0	0
UserStory #15,#16,#17,#18,#19,#20,#21																
UI Designing	22/06/2022	4	0	0	0	0	0	0	0	0	4	0	0	0	0	0
Database Connectivity	24/06/2022	2	0	0	0	0	0	0	0	0	0	2	0	0	0	0
Coding	06/07/2022	2	0	0	0	0	0	0	0	0	0	0	0	2	0	0
Testing	06/07/2022	2	0	0	0	0	0	0	02	0	0	0	0	0	0	0
Total		50	5	6	6	4	0	2	7	0	4	2	8	6	0	0

Table 3.4: Sprint Plan

3.5.5 Sprint Actual

Backlog Item	Status & completion date	Original estimate in hours	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14	Completed Y or N
UserStory #1,#2																	
UI Designing	20/04/2022	2	2	0	0	0	0	0	0	0	0	0	0	0	0	Y	
Database Connectivity	21/04/2022	2	0	2	0	0	0	0	0	0	0	0	0	0	0	Y	
Coding	26/04/2022	4	0	0	2	0	0	0	2	0	0	0	0	0	0	Y	
Testing	26/04/2022	2	0	0	0	0	0	0	2	0	0	0	0	0	0	Y	
UserStory #3,#4																	
UI Designing	29/04/2022	2	0	2	0	0	0	0	0	0	0	0	0	0	0	Y	
Database Connectivity	29/04/2022	2	0	2	0	0	0	0	0	0	0	0	0	0	0	Y	
Coding	30/04/2022	4	0	0	4	0	0	0	0	0	0	0	0	0	0	Y	
Testing	30/04/2022	2	0	0	0	2	0	0	0	0	0	0	0	0	0	Y	
UserStory #5,#6,#7																	
UI Designing	03/05/2022	3	3	0	0	0	0	0	0	0	0	0	0	0	0	Y	
Database Connectivity	03/04/2022	2	0	2	0	0	0	0	0	0	0	0	0	0	0	Y	
Coding	03/05/2022	3	0	0	3	0	0	0	0	0	0	0	0	0	0	Y	
Testing	11/05/2022	2	0	0	0	2	0	0	0	0	0	0	0	0	0	Y	
UserStory #8,#9,#10,#11,#12,#13,#14																	
UI Designing	27/05/2022	4	2	1	1	0	0	0	0	0	0	0	0	0	0	Y	
Database Connectivity	27/05/2022	2	0	2	0	0	0	0	0	0	0	0	0	0	0	Y	
Coding	17/06/2022	2	0	2	0	0	0	0	0	0	0	0	0	0	0	Y	
Testing	17/06/2022	2	0	2	0	0	0	0	0	0	0	0	0	0	0	Y	
UserStory #15,#16,#17,#18,#19,#20,#21																	
UI Designing	22/06/2022	4	0	0	0	0	0	0	0	0	4	0	0	0	0	Y	
Database Connectivity	24/06/2022	2	0	0	0	0	0	0	0	0	0	2	0	0	0	Y	
Coding	06/07/2022	2	0	0	0	0	0	0	0	0	0	0	0	2	0	Y	
Testing	06/07/2022	2	0	0	0	0	0	0	02	0	0	0	0	0	0	Y	
Total			50	5	6	6	4	0	2	7	0	4	2	8	6	0	

Table 3.5: Sprint Actual

Chapter 4

Results and Discussions

Using Fuel delivery on demand application, fuel can be delivered successfully. This system is helpful for the people who got stuck due to lack of fuel. By using GPS it found to be easy to locate the exact location of the user. The system became more effective as people uses smartphone with GPS facility. This is a better system at nowadays because the usage of fuel increased and price varies certain days.

4.0.1 Screenshot

1. The Admin and Fuel Station can login to the system.

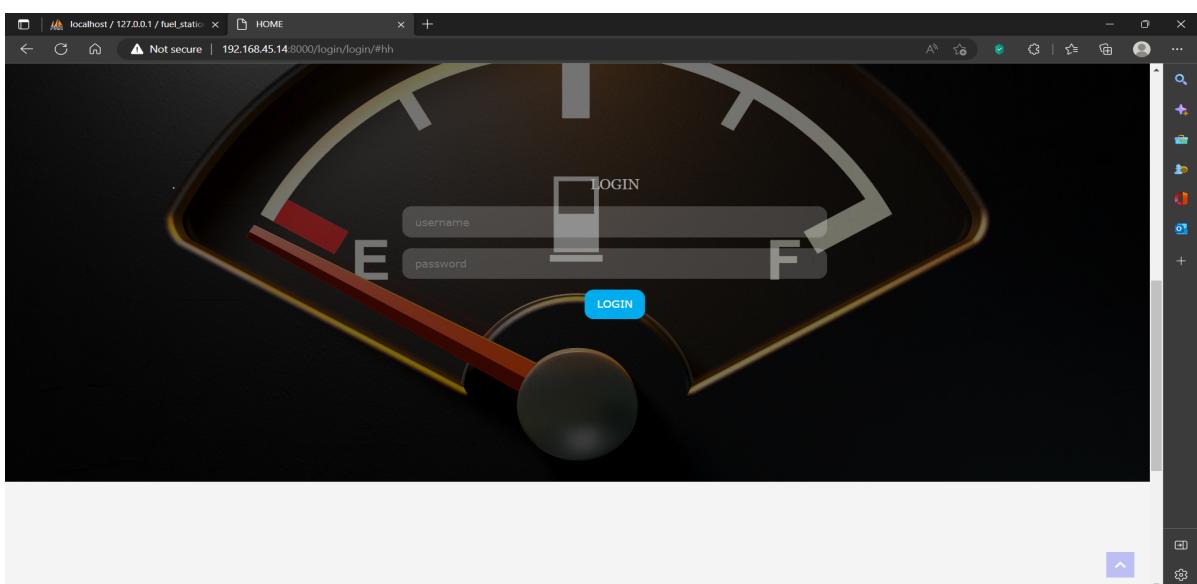


Figure 4.1: Login

2. Fuel Station can register here if they are accessing the system for the first time

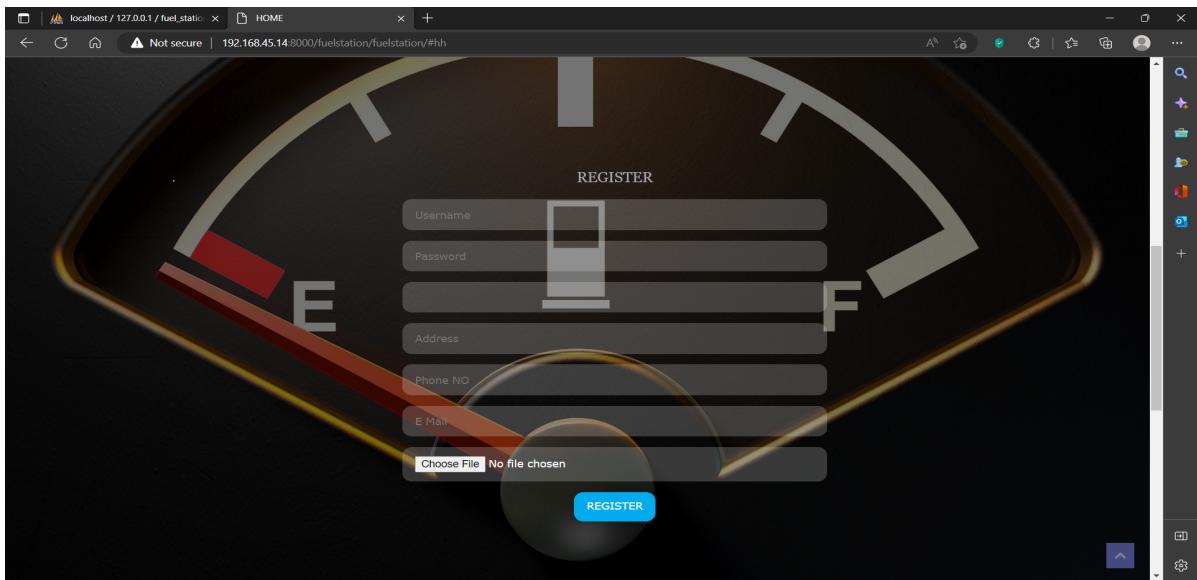


Figure 4.2: Registration

3. Admin goes to the home page after login.

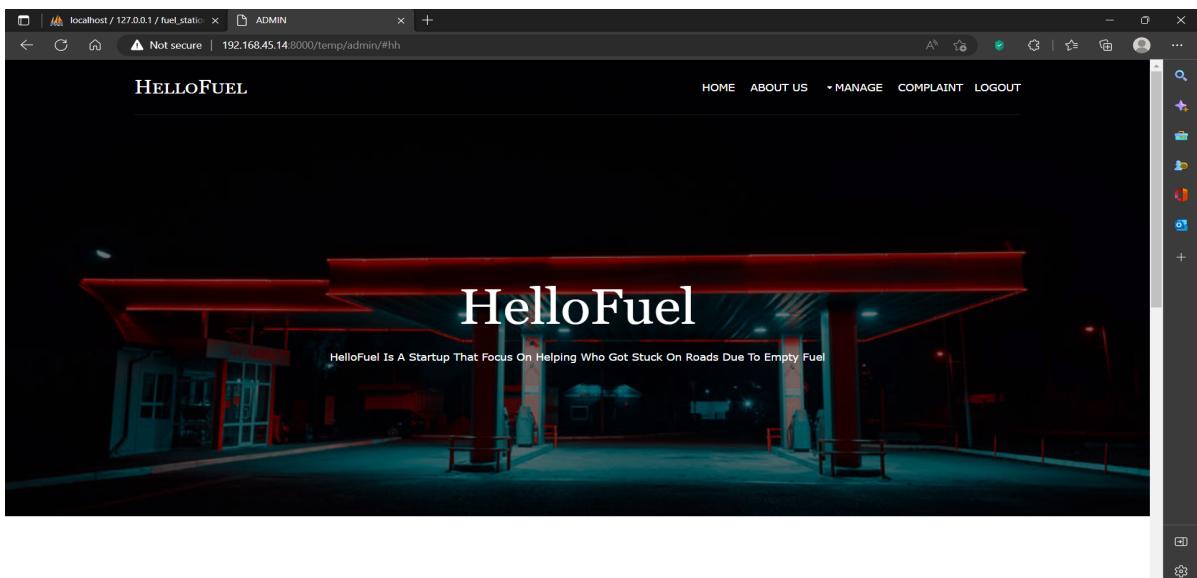


Figure 4.3: Admin Home Page

4. Admin manages fuel station after verifying the information uploaded by the fuel station.

fs_id	Station Name	Address	Mail Id	Phone No	Status	Document		
1	INDIANOIL	kuttipuram	indianoil@gmail.com	8075172060	approved	<button>DOWNLOAD</button>	<button>APPROVE</button>	<button>REJECT</button>
2	f2	abbc	indianoil@gmail.com	9245259582	approved	<button>DOWNLOAD</button>	<button>APPROVE</button>	<button>REJECT</button>
3	f3	abc	indianoil@gmail.com	8075172060	approved	<button>DOWNLOAD</button>	<button>APPROVE</button>	<button>REJECT</button>
6	F4	malappuram	ghag@gmail.com	8594051426	approved	<button>DOWNLOAD</button>	<button>APPROVE</button>	<button>REJECT</button>
7	f10	abc	a@gmail.com	9876543210	approved	<button>DOWNLOAD</button>	<button>APPROVE</button>	<button>REJECT</button>
8	HP Petroleum	Kuttippuram	hppktp12@gmail.com	6524371089	approved	<button>DOWNLOAD</button>	<button>APPROVE</button>	<button>REJECT</button>
9	Hindusthan Petroleum Pump	kuttipuram	hp123@gmail.com	9635451278	rejected	<button>DOWNLOAD</button>	<button>APPROVE</button>	<button>REJECT</button>
10	Hp Pump	kuttipuram	hp123@gmail.com	9635451278	approved	<button>DOWNLOAD</button>	<button>APPROVE</button>	<button>REJECT</button>
11	Pump	edappal	pump123@gmail.com	6928021536		<button>DOWNLOAD</button>	<button>APPROVE</button>	<button>REJECT</button>

Figure 4.4: Manage Fuel Station

5. Admin manages delivery agent after verifying the information uploaded by the delivery agent.

d_id	Username	Address	Mail Id	Phone No	Status	Document		
1	d1	abc	delivery1@gmail.com	9876543210	rejected	<button>DOWNLOAD</button>	<button>APPROVE</button>	<button>REJECTED</button>
2	d2	kozhikode	d2@gmail.com	9876543212	approved	<button>DOWNLOAD</button>	<button>APPROVE</button>	<button>REJECTED</button>
3	ass	aaa	aa@aa.c	1234567890	approved	<button>DOWNLOAD</button>	<button>APPROVE</button>	<button>REJECTED</button>
4	mashood	kuttipuram	mashood123@gmail.com	9756768910	approved	<button>DOWNLOAD</button>	<button>APPROVE</button>	<button>REJECTED</button>

Figure 4.5: Manage Delivery Agent

6. Admin views the complaint posted by user and replies

The screenshot shows a web browser window titled 'ADMIN' with the URL 'localhost / 127.0.0.1 / fuel_static'. The page displays a table titled 'VIEW COMPLAINT' containing ten rows of data. Each row includes columns for com_id, user_id, username, complaint, date, and reply. A 'REPLY' button is present in each row under the 'reply' column.

com_id	user_id	username	complaint	date	reply
1	1	aswin	demo	June 17, 2022	<button>REPLY</button>
2	1	abc	aaaa	June 25, 2022	<button>REPLY</button>
3	3	abc	new	June 25, 2022	<button>REPLY</button>
4	3	abc	aaanew	June 25, 2022	<button>REPLY</button>
5	3	a	akash	June 25, 2022	<button>REPLY</button>
6	4	us1	babin	June 26, 2022	<button>REPLY</button>
7	9	Abhijith	Problem occurred while using the app	June 30, 2022	<button>REPLY</button>
8	9	Abhijith	Problem occurred while using the app	June 30, 2022	<button>REPLY</button>
9	9	Abhijith	Problem occurred while using the app	June 30, 2022	<button>REPLY</button>
10	9	Abhijith	Problem occurred while using the app	June 30, 2022	<button>REPLY</button>

Figure 4.6: Complaints

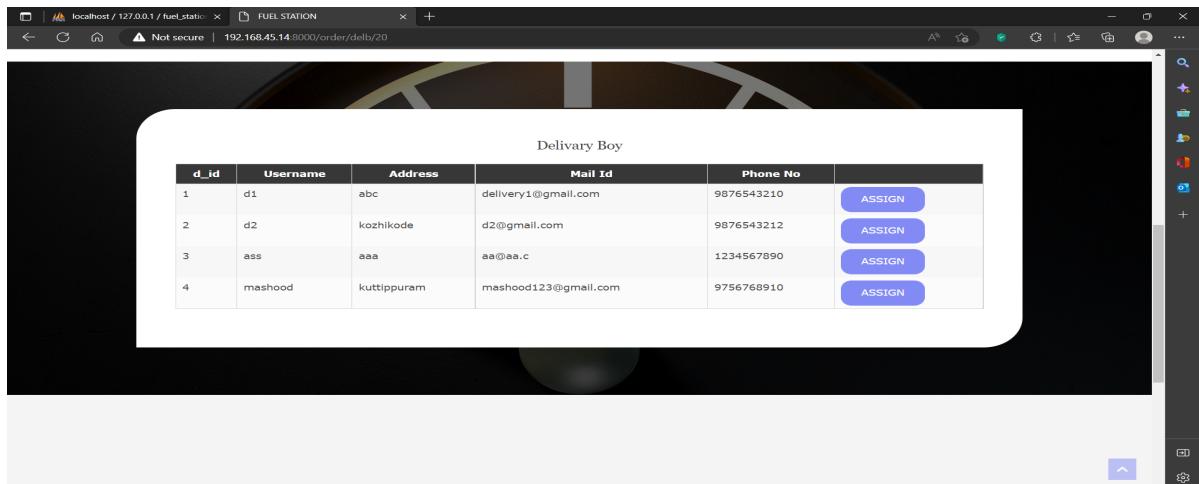
7. Fuel Station can view the orders and assign delivery agent to deliver fuel.

The screenshot shows a web browser window titled 'FUEL STATION' with the URL 'localhost / 127.0.0.1 / fuel_station'. The page displays a table titled 'FUEL ORDER' containing one row of data. The row includes columns for User, Fuel, Qty, Date, Phone No, and Status. An 'ASSIGN' button is present in the last column.

User	Fuel	Qty	Date	Phone No	Status	ASSIGN
9	petrol	2	July 4, 2022	9745237645	pending	<button>ASSIGN</button>

Figure 4.7: Fuel Orders

8. Fuel Station select delivery agent to deliver the fuel.



Delivery Boy					
d_id	Username	Address	Mail Id	Phone No	
1	d1	abc	delivery1@gmail.com	9876543210	<button>ASSIGN</button>
2	d2	kozhikode	d2@gmail.com	9876543212	<button>ASSIGN</button>
3	ass	aaa	aa@aa.c	1234567890	<button>ASSIGN</button>
4	mashood	kuttipuram	mashood123@gmail.com	9756768910	<button>ASSIGN</button>

Figure 4.8: Assign Delivery Agent

9. User and Delivery agent can login to the system.

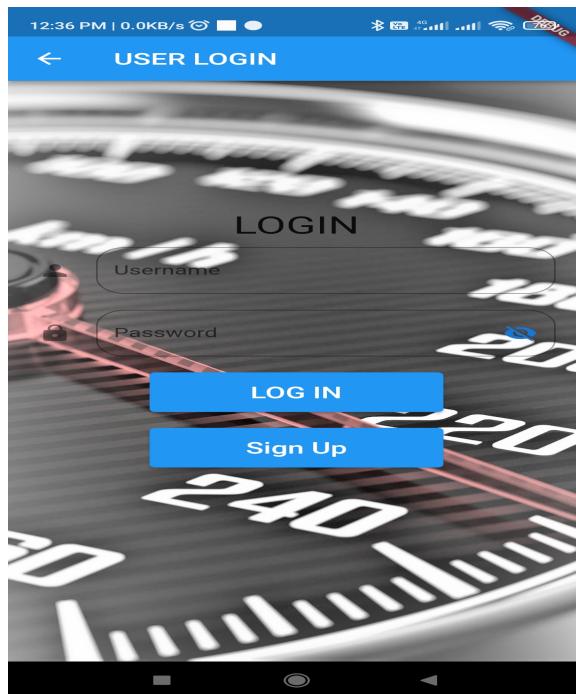


Figure 4.9: User Login

10. User or Delivery agent can select their registration page to register into the system

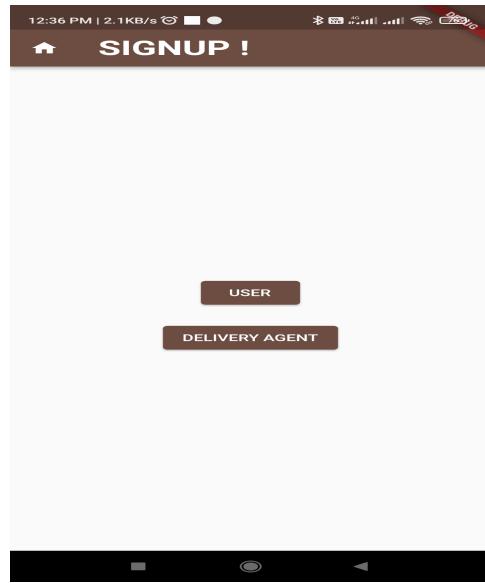


Figure 4.10: Sign Up

11. User can register here if they are accessing the system for the first time.

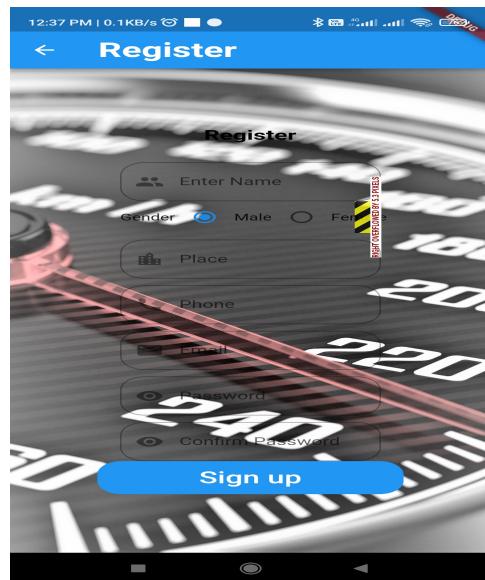


Figure 4.11: User Registration

12. User can search for fuel station based on location.

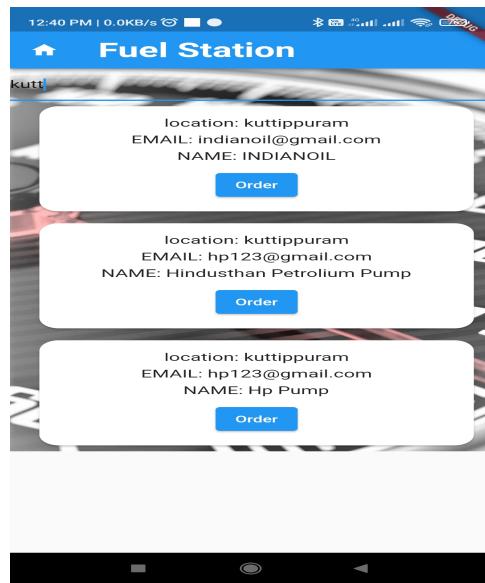


Figure 4.12: Search Fuel Station

13. User can order fuel by selecting type and quantity

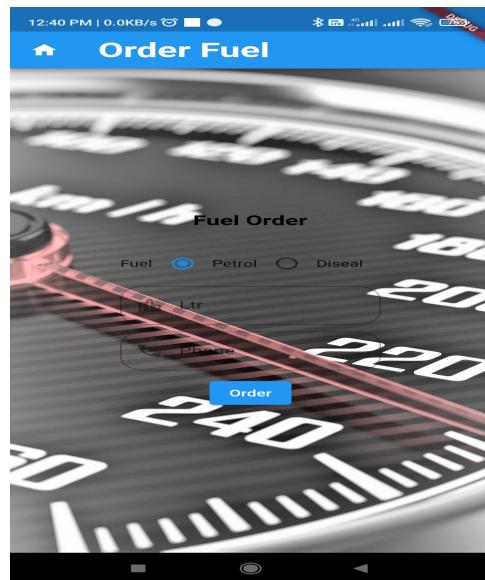


Figure 4.13: Order Fuel

14. User can confirm order by completing payment.

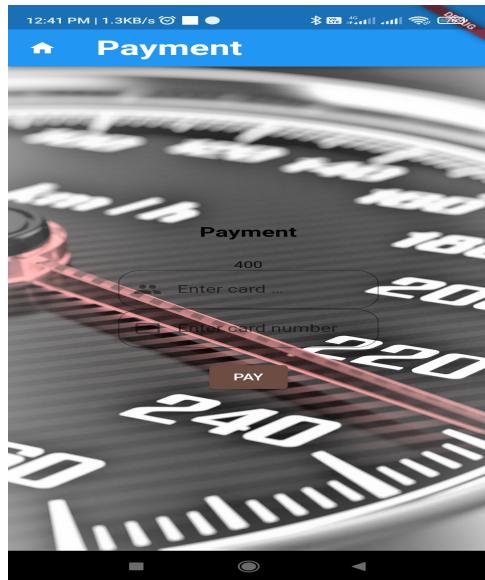


Figure 4.14: Payment

15. User can post complaint and view reply from admin.

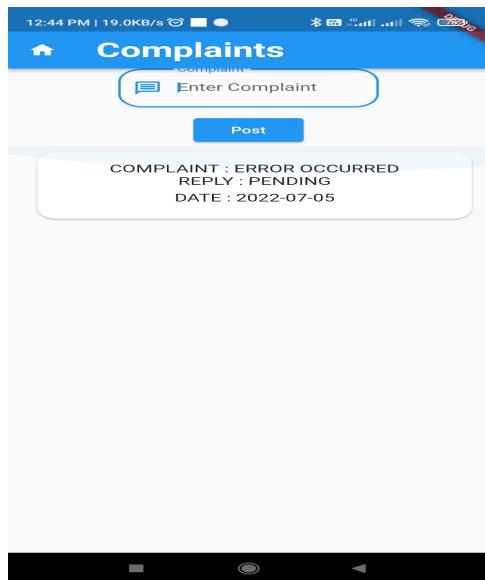


Figure 4.15: Post Complaint

16. Delivery agents can register here if they are accessing the system for the first time.

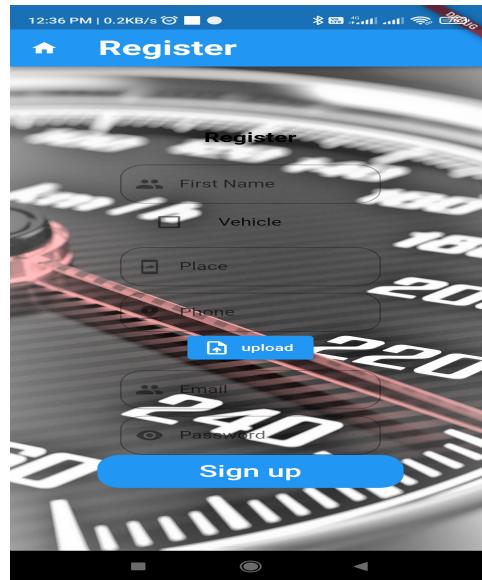


Figure 4.16: Delivery Agent Registration

17. Delivery agent can view the order assigned by the fuel station here and agent can locate the user to deliver the fuel.

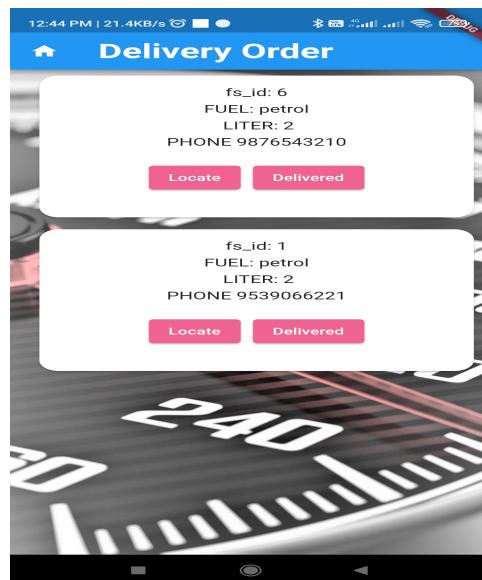


Figure 4.17: Assigned Delivery

Chapter 5

Conclusions

Fuel delivery on demand application helps the user to refuel the vehicle on their demand at their current location. The fuel station who sells, user who buys, delivery agent who delivers the fuel has to register into the system. Fuels stations and delivery agents are verified by the admin. User can select the fuel type and quantity to place order. Fuel gets delivered by tracking the GPS location of the user. This system helps the people who gets stuck by road due to the lack of petrol. The system is easy to use, and can order fuel quickly using any smartphones. The system is mainly developed as two parts, a web application and a mobile application. Web application is done using python django framework and mobile application is built using flutter framework. All necessary validations are carried out in this project where ever required and hence is a reliable system. The project has been developed, tested , documented successfully . This has been developed as versatile and user friendly as possible. At the final stage of this project with a proud feeling that some thing new had developed. In future, this system can be useful for all who face the issue mentioned and will create new expirience in life.

References

- [1] [https://ideausher.com/blog/how-to-start-a-fuel-delivery-business/.](https://ideausher.com/blog/how-to-start-a-fuel-delivery-business/)
- [2] [https://www.mindbowser.com/on-demand-fuel-delivery-app-development/.](https://www.mindbowser.com/on-demand-fuel-delivery-app-development/)

Appendix

Source Code

```

from django.http import HttpResponse
from django.shortcuts import render
from orderfuel.models import Orderfuel, Payment
from delivery.models import Delivary
# Create your views here.
def vieworder(request):
    uid=request.session["uid"]
    obb=Orderfuel.objects.filter(fs_id=uid)
    context={
        'obval':obb
    }
    return render(request,'orderfuel/orderview.html', context)

def viewdf(request,idd):
    request.session['oid']=idd
    obb=Delivary.objects.all()
    context={
        'obval':obb
    }
    return render(request,'orderfuel/viewdb.html', context)

from django.http import HttpResponseRedirect

def assign(request,idd):
    oid=request.session['oid']
    print(oid)
    print(idd)
    ord=Orderfuel.objects.get(order_id=oid)
    ord.d_id=idd
    ord.save()
    return HttpResponseRedirect('/order/order/')

# return render(request,'')
from rest_framework.views import APIView,Response
from orderfuel.serializers import android_serialiser, android_serialisers

import datetime
class ord_view(APIView):
    def get(self, request):
        ob =Orderfuel.objects.all()
        ser = android_serialiser(ob, many=True)

```

Appendix

```
        return Response(ser.data)

    def post(self, request):
        ob = Orderfuel()
        ob.date = datetime.datetime.now()
        ob.u_id = request.data['u_id']
        ob.fs_id = request.data['fsid']
        ob.fuel = request.data['fuel']
        ob.qty = request.data['qty']
        ob.status="pending"
        ob.latitude='11.2596128'
        ob.longitude='75.785404'
        ob.phone = request.data['phone']
        ob.d_id ='0'
        # ob. = request.data['gender']

        # //send sms here

        ob.save()

        return HttpResponse("yessss")

class payment(APIView):
    def get(self, request):
        ob =Payment.objects.all()
        ser = android_serialisers(ob, many=True)
        return Response(ser.data)

    def post(self, request):
        ob = Payment()
        ob.order_id="1"
        ob.cardnumber=request.data['cardnumber']
        ob.card_holdername=request.data['card_holdername']
        ob.amount=int(request.data['amount'])+50
        ob.save()

        return HttpResponse("yessss")
```

Database Design

Attribute Name	Datatype	Width	Description
User id	Integer	11	Primary Key
User name	Varchar	20	
Password	Varchar	20	
Gender	Varchar	10	
Address	Varchar	100	
Phone	Varchar	10	
E-mail	Varchar	20	

Table A.1: User Registration

Attribute Name	Datatype	Width	Description
login id	Integer	11	Primary Key
U id	Integer	11	Unique
User name	Varchar	20	
Password	Varchar	20	
Type	Varchar	20	

Table A.2: Login

Appendix

Attribute Name	Datatype	Width	Description
Fs id	Integer	11	Primary Key
User name	Varchar	20	
Password	Varchar	20	
Address	Varchar	100	
Phone	Varchar	10	
E-mail	Varchar	20	
Document	Varchar	100	
Status	Varchar	20	

Table A.3: Fuel Station Registration

Attribute Name	Datatype	Width	Description
D id	Integer	11	Primary Key
User name	Varchar	20	
Password	Varchar	20	
Address	Varchar	100	
Phone	Varchar	10	
E-mail	Varchar	20	
Document	Varchar	100	
Status	Varchar	20	
Vehicle	Varchar	20	

Table A.4: Delivery Agent Registration

Appendix

Attribute Name	Datatype	Width	Description
Order id	Integer	11	Primary Key
Fs id	Integer	20	Unique
D id	Integer	20	Unique
U id	Integer	20	Unique
Date	Date	20	
Fuel	Varchar	20	
Quantity	Varchar	20	
Latitude	Varchar	50	
Longitude	Varchar	50	
Phone	Varchar	10	
Status	Varchar	20	

Table A.5: Fuel Order

Attribute Name	Datatype	Width	Description
P id	Integer	11	Primary Key
Order id	Integer	11	
Amount	Integer	20	
Card holder name	Varchar	20	
Card number	Varchar	20	

Table A.6: Payment

Appendix

Attribute Name	Datatype	Width	Description
Com id	Integer	11	Primary Key
User id	Integer	11	Unique
Username	Varchar	20	
Date	Date	20	
Complaint	Varchar	200	
Reply	Varchar	200	

Table A.7: Complaint

Dataflow Diagram

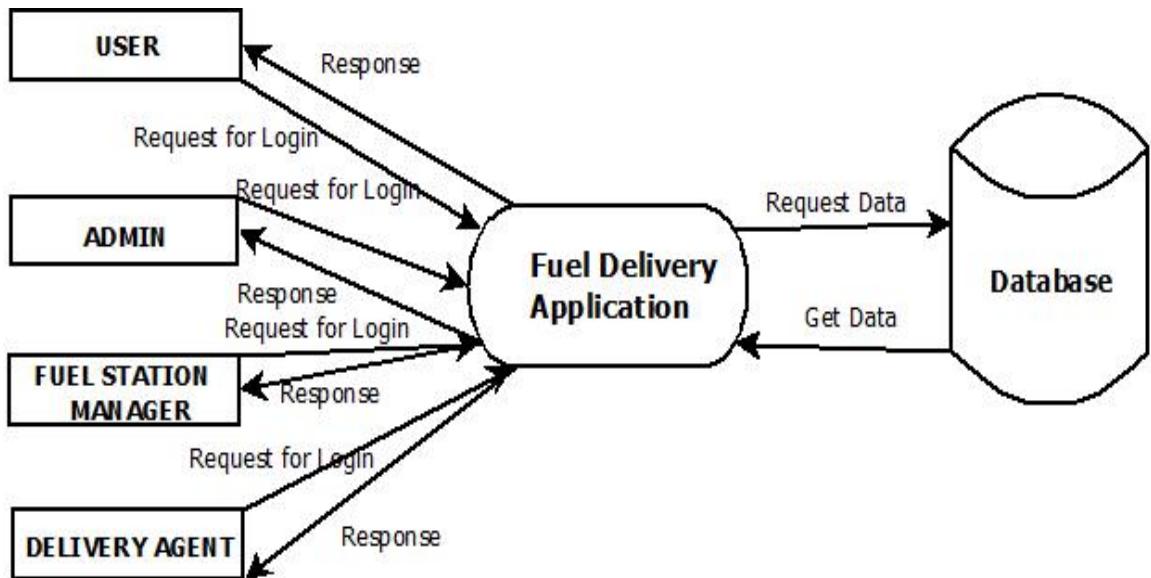


Figure A.1: DFD-Level 0

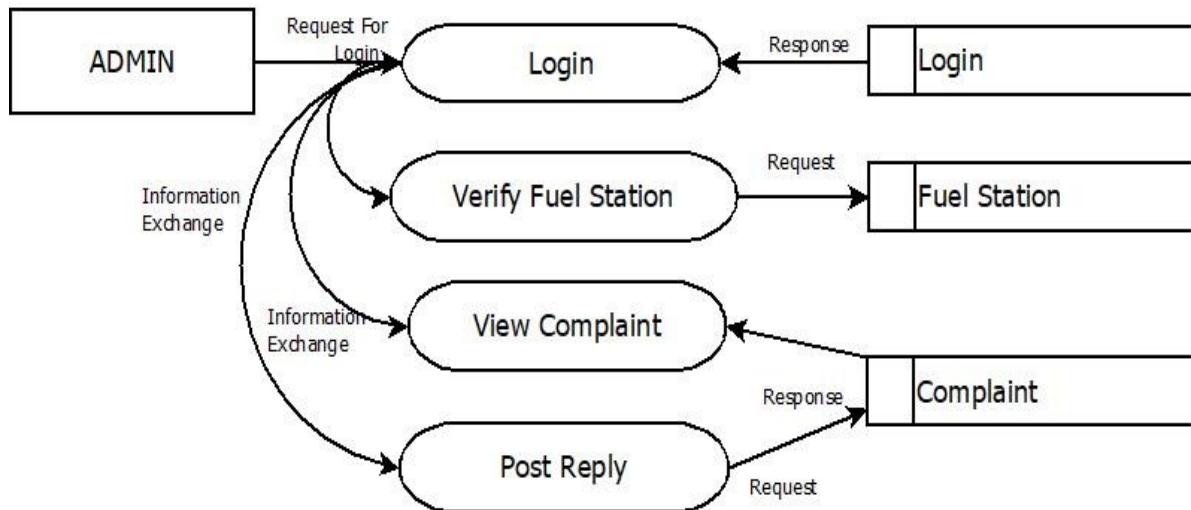


Figure A.2: DFD-Level 1

Appendix

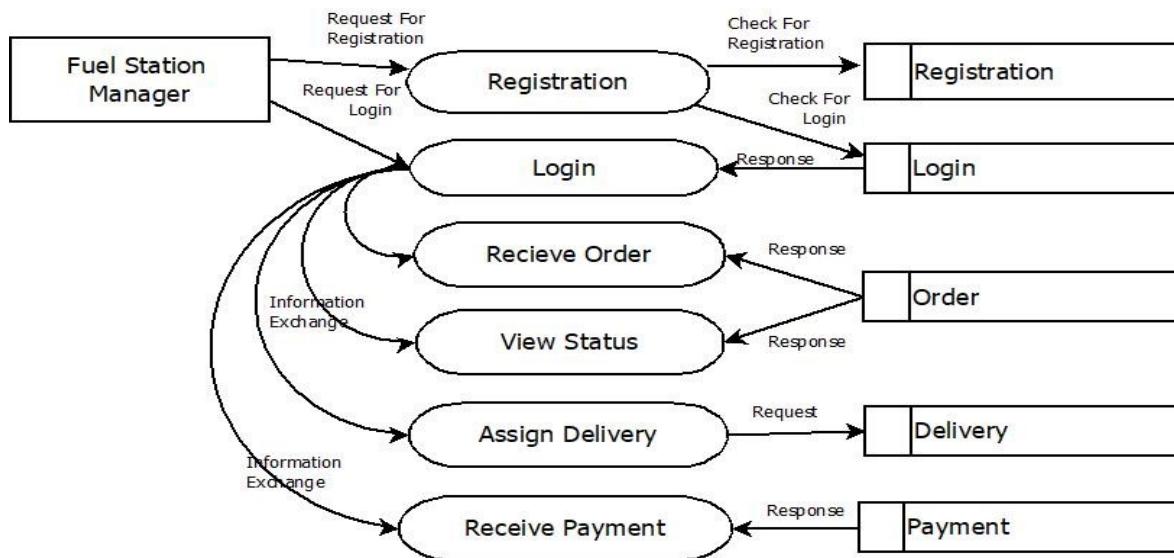


Figure A.3: DFD-Level 2

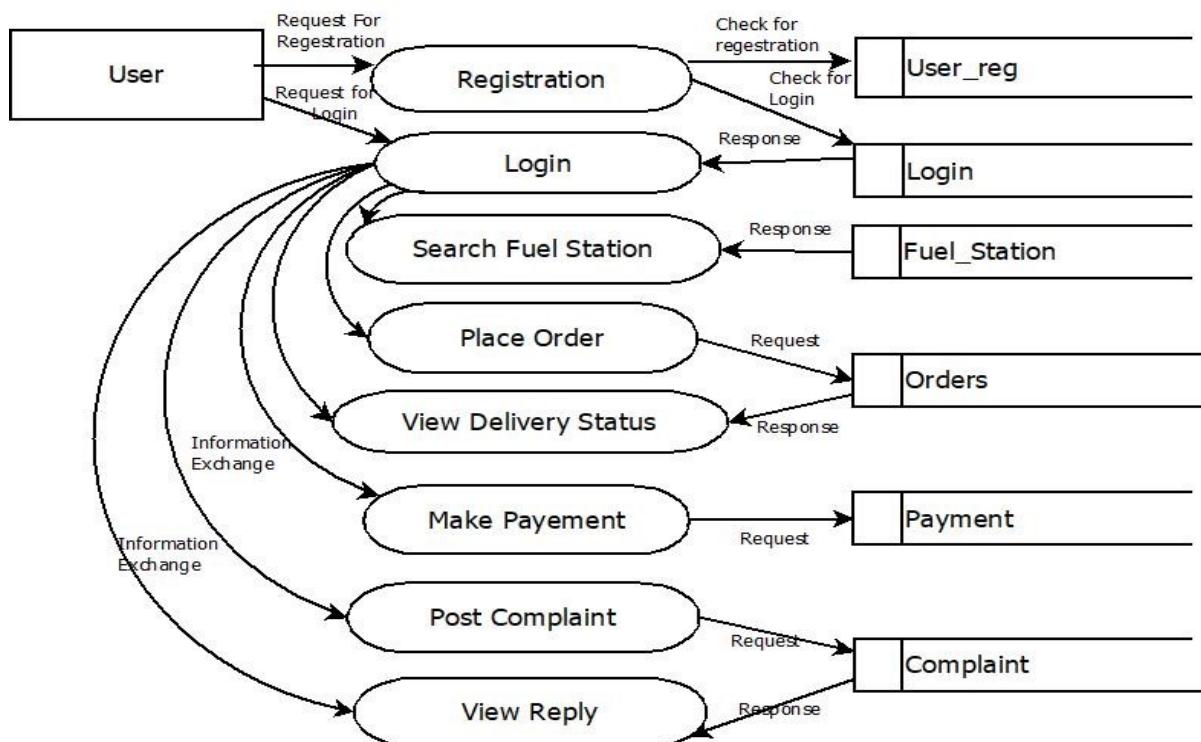


Figure A.4: DFD-Level 3

Appendix

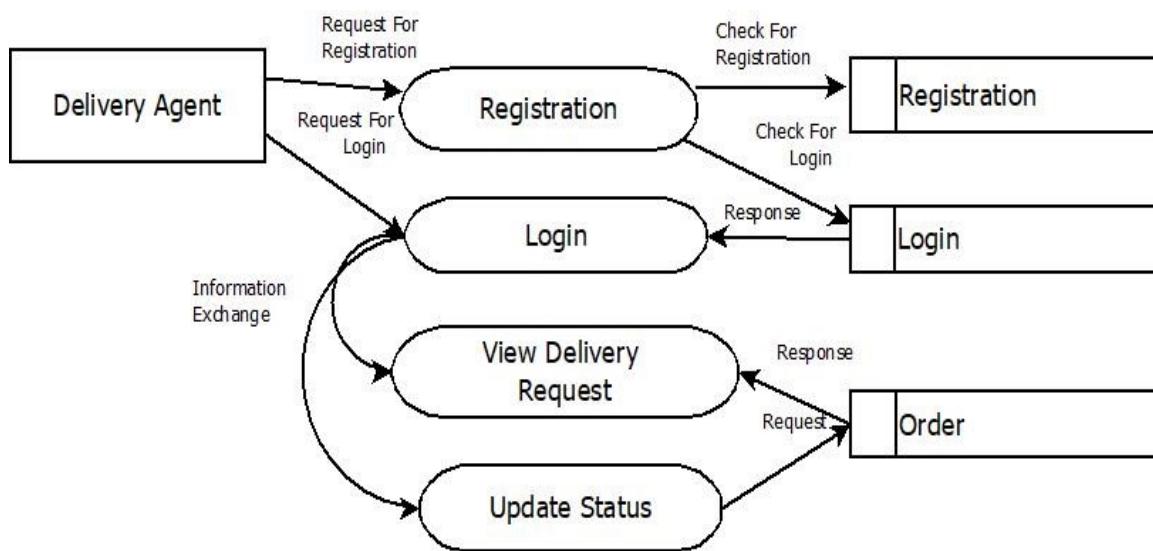


Figure A.5: DFD-Level 4