

# **Software Requirements Specification**

**for**

# **Online Petrol Delivery System**

**Version 1.0**

**Group No.: 6**

**Nur Iman binti Mohamad Idros**

**1231303620**

**Maryam binti Norazman**

**1211111809**

**Tan Jie Ting**

**1221102916**

**Fikrul Amsyar bin Azmin**

**241UC24167**

**Date: 8 December 2024**

## Contents

.....	2
.....	3
.....	4
1.1 TEAM MEMBERS .....	4
1.2 PROJECT PLAN .....	4
.....	5
2.1 DESCRIPTION .....	5
2.2 ACTORS.....	5
2.3 ASSUMPTIONS AND DEPENDENCIES .....	5
2.4 USE CASE DIAGRAM .....	7
.....	9
3.1 ACTOR 1 : USER .....	9
3.2 ACTOR 2 : DRIVER.....	11
3.3 ACTOR 3 : ADMIN.....	11
3.4 ACTOR 4 : SUPPLIER .....	12
.....	13
4.1 CLASS DIAGRAMS .....	13
4.2 SEQUENCE DIAGRAMS .....	13
.....	17
5.1 STATE DIAGRAMS .....	17
5.2 DATA FLOW DIAGRAMS .....	17
.....	18

## Revisions

Version	Primary Author(s)	Description of Version	Date Completed
Draft Type and Number	Full Name	Information about the revision. This table does not need to be filled in whenever a document is touched, only when the version is being upgraded.	00/00/00

## 1.1 Team Members

## 1.2 Project Plan

[illegible]

## 2 System Overview

### 2.1 Description

The Online Petrol Delivery System is designed to improve fuel delivery services for individuals, businesses, and industries. It has four primary users: Admin, User/Customer, Driver, and Supplier, each playing a distinct role for a smooth operation. The system facilitates user and driver registration with authentication to ensure safety and reliability. Customers can compare petrol prices based on delivery distance and volume, as well as evaluate delivery times across available drivers to make informed decisions. Once orders are placed, the system verifies petrol and delivery purchases, enabling real-time tracking to ensure customers have full visibility over their orders. Additionally, users can cancel orders and receive refunds, adding convenience to the service.

The system has efficient inventory management by keeping track of petrol stock levels and ensuring steady supply to meet demand from supplier. The supplier also responsible for providing petrol pricing data so that customer would be able to have a fast and well updated pricing data. Admins oversee the entire operation so that the system can runs smoothly without causing issue. Drivers can accept or reject delivery assignments and confirm completion, providing a seamless end-to-end service. The system outputs include registered user and driver accounts, price and delivery time comparisons, authenticated transactions, real-time tracking updates, inventory status, and order cancellation with refunds. By combining convenience, transparency, and efficiency, the Online Petrol Delivery System ensures a reliable and customer-centric fuel delivery experience.

### 2.2 Actors

Actor	Use Cases
Customer	Customer registration and profile management, order placement, order confirmation, order cancellation and refund
Driver	Driver registration and authentication, order confirmation
Admin	Driver registration and authentication, order placement, order cancellation and refund
Supplier	Inventory management, order confirmation

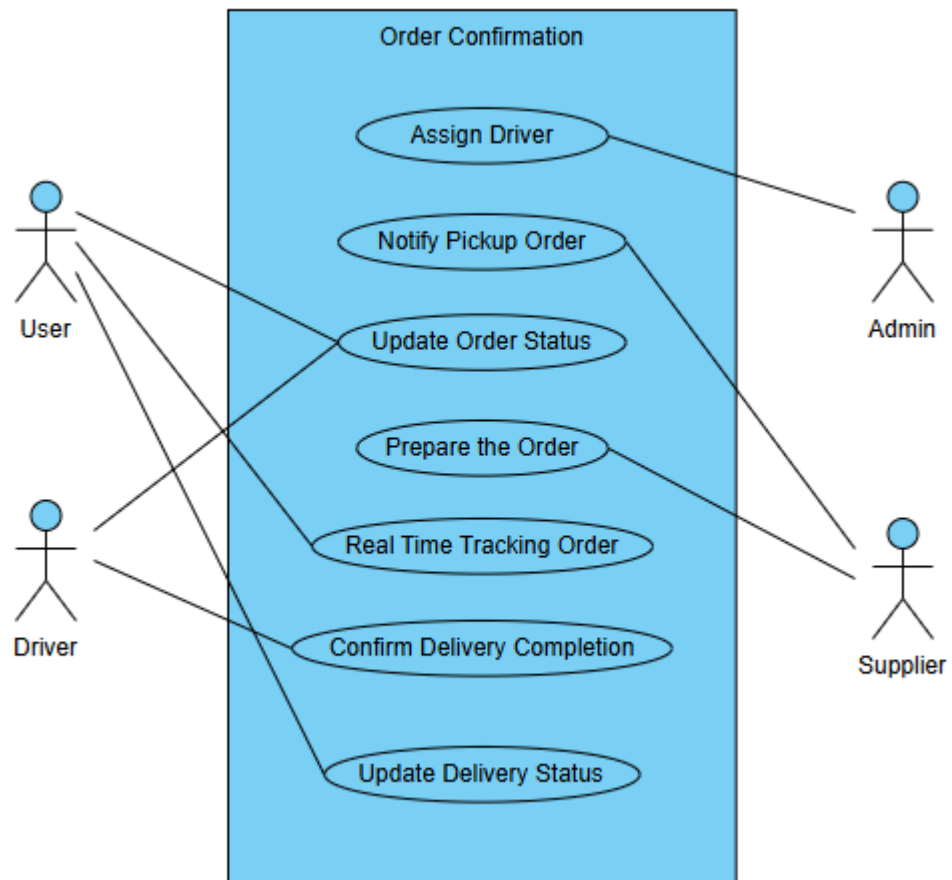
### 2.3 Assumptions and Dependencies

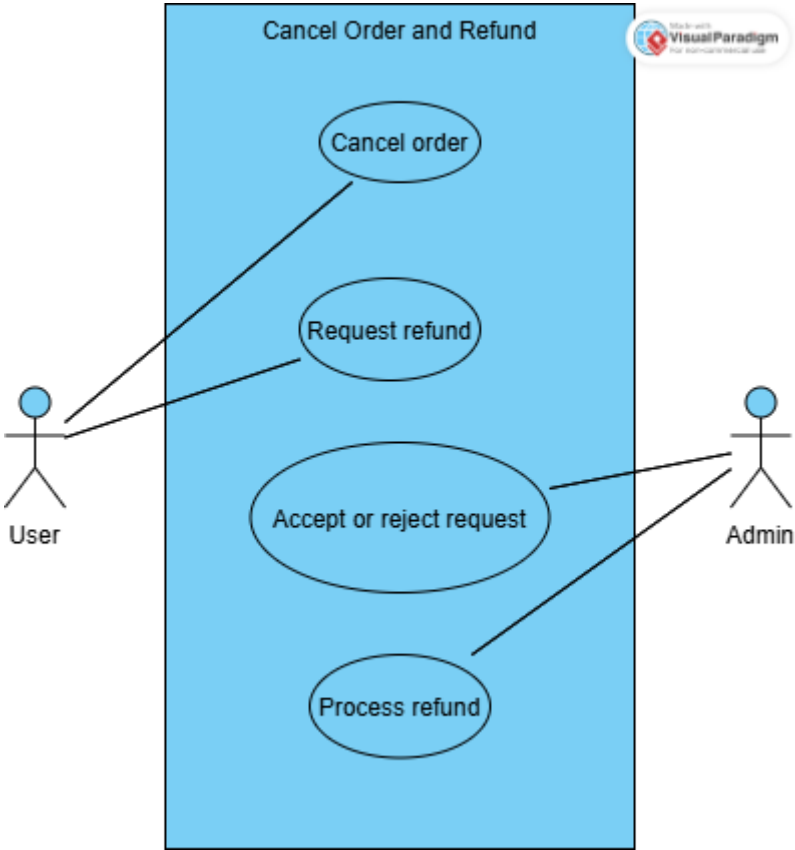
The system assumes the availability of reliable third-party APIs, such as payment gateways, GPS tracking, and mapping services, as these are critical for smooth operation. If these APIs become unavailable and unreliable the system's functionality and user experience could be negatively impacted. Additionally, stable internet connectivity is presumed for all users, drivers, and suppliers.

Poor or inconsistent internet connections could lessen the system's performance, hinder real-time tracking, and reduce user satisfaction. Another important assumption is hardware and device compatibility. It is expected that users, drivers, and administrators will have access to devices capable of running the application or web interface. If users rely on incompatible or outdated devices, it could severely affect system performance or prevent the system from functioning on the device.

The system also has key dependencies that influence its operation. It relies on secure and efficient payment gateway services, such as FPX Net, to handle transactions. GPS and mapping services are essential to enable real-time delivery tracking and optimizing routes for drivers. Furthermore, the system is heavily reliant on the availability of drivers and suppliers. Any shortage or lack of availability in these roles could disrupt operations and lessen the system's ability to meet user demands effectively.

## 2.4 Use Case Diagram





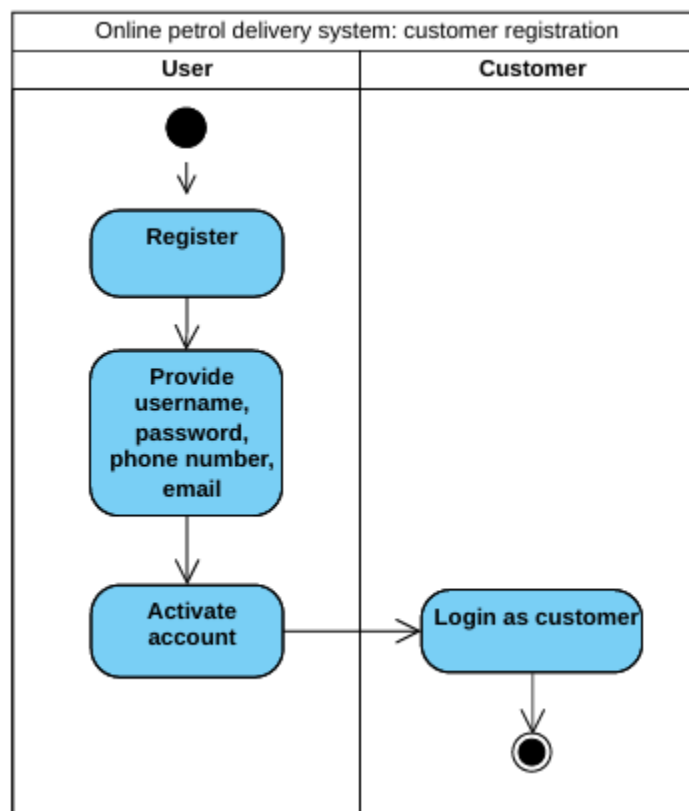


## 3 Basic Requirements

### 3.1 Actor 1 : User

#### 3.1.1 Use Case : User Registration and Profile Management

To start to use the system as a customer, the user will have to register an account first. To register, the user must provide their name as username, password, phone number, email address. A verification email will be sent to the user's email and require user to do activation. After the user activate the email, then user will be able to login to the system.



### **3.1.2 Use Case : Order Placement**

User should be able to place an order by specifying the required petrol volume and delivery location. The system then calculates delivery costs based on the petrol quantity and delivery distance, allowing users to compare prices. Additionally, users can compare delivery times for available drivers. After reviewing the details, users confirm the order and make payment through the system. The system processes the payment, and then sends user an order confirmation that includes a detailed receipt, estimated delivery time, and options for tracking. Admin then ensures that the order is valid.

---

### **3.1.3 Use Case : Order Cancellation and Refund**

User would be able to cancel an order through the system before delivery if they changed their plans by using the cancel order button. The system then would cancel the order accordingly. User should be refunded during this process. If approved, the admin initiates the refund process, and the system credits the user's account or payment method. Notifications are sent to the user and admin to track the refund process.



## **3.2 Actor 2 : Driver**

### **3.2.1 Use Case : Driver Registration and Authentication**

To start to use the system as a driver, the user will also have to register an account. To register, the user will have to also provide their name as username, password, phone number, email address, softcopy of their license, car details, photo of the car. Users will also receive verification email to do activation. Then the admin will check their license and car details, if the license is still valid and the car is in an ideal condition, then the admin will approve the registration. Users will only be able to login after they receive confirmation from admin sent through email.

---

## **3.3 Actor 3 : Admin**

### **3.3.1 Use Case : Order Confirmation**

Once an order is placed, admin then assigned available drivers for delivery and the system then notifies the supplier. Drivers can accept or reject the delivery task, and upon acceptance, the system updates the order status. If driver reject the order, the admin will find the user with a new driver. The supplier prepares the petrol for delivery. By providing each order with unique order number or tracking ID, user would be able to track order in real – time using GPS technology. For the driver, as they move along the delivery route, the system updates the server with their current GPS coordinates. Updated location on a map, estimated time of arrival (ETA), and notifications about delivery progress are all provided by the system using this data with map services. A visually displayed map that shows the driver's location and status updates is the user output such as “Out for delivery”, “The driver has picked up your order”. After successful delivery, the driver confirms completion, and the system marks the order as delivered, notifying the user.

---

## **3.4 Actor 4 : Supplier**

### **3.4.1 Use Case : Inventory Management**

The supplier provides petrol stock information, including stock available in the area and pricing data, into the system. The system processes this information, updating the inventory database.

--	--	--

## **4 Specific Requirements**

### **4.1 Class Diagrams**



### **4.2 Sequence Diagrams**

#### **4.2.1 Use Case 1 : User Registration and Profile Management**

1. To register, provide username, password, phone number, email address
2. System will send out activation account
3. User need to activate it via the email
4. Then the information will be saved in the database
5. Then the user will be able to login as customer

#### **4.2.2 Use Case 2 : Driver Registration and Authentication**

1. To register, provide username, password, phone number, email address, softcopy of license, car details
2. System will send out activation account
3. User need to activate it via the email
4. Then registered information will be sent to admin
5. Admin either approve or reject
6. System will save the information if approve is receive
7. System will send out approval or rejection to the user based on admin reply
8. User will be able to login if approval is received
  - a. If rejection is received, then the user can retry



### **4.2.3 Use Case 3 :Order Placement**

1. User logs into the system.
2. User navigates to the "Place Order" page.
3. User enters the following details:
  - Type of petrol
  - Quantity required.
  - Delivery address.
  - Preferred delivery time.
4. The system validates the entered details (e.g., checks if the stock is available and the address is valid).
5. If valid, the system calculates the total cost, including taxes and delivery fees.
6. The system displays the order summary to the user.
7. User reviews and confirms the order.
8. The system creates a new order record in the database and assigns an order ID.
9. The system sends a notification to the Admin for further processing.
10. The system displays an order placement confirmation message to the user.



#### **4.2.4 Use Case 4 : Order Confirmation**

1. Admin logs into the system and views pending orders.
2. Admin selects an order and assigns an available driver.
3. System updates the order status to "Assigned" and notifies the driver.
4. Driver accepts or rejects the task.
  - If accepted, the system updates the status to "Out for Delivery."
  - If rejected, the system marks the order as "Driver Rejected," and admin reassigns a new driver.
5. System notifies the supplier to prepare the petrol for delivery.
6. Driver's location is tracked in real-time via GPS and displayed on the map.
7. User can track the order's ETA and progress on the map.
8. Once delivered, the driver marks the order as "Delivered."
9. System updates the status and notifies the user of successful delivery.



#### **4.2.5 Use Case 5 : Inventory Management**

1. Supplier will need to provide pricing and remaining stock to the system
2. Then system will update the database based on the information receive

#### **4.2.6 Use Case 6 : Order Cancellation and Refund**

1. User requests order cancellation by pressing the cancel order button
2. The system then would cancel the order accordingly
3. The system requests the admin for order cancellation approval
4. Admin either approve or reject the order cancellation
5. User receives the notification for the order cancellation
  - User will receive the refund credits if approved
  - User will not receive the refund credits if not approved
6. Admin receives the notification about the successful refund process





## **5 Behavioural Requirements**

### **5.1 State Diagrams**

*<TO DO: Describe the states, events, and transitions and place the state diagrams.>*

### **5.2 Data Flow Diagrams**

*<TO DO: Describe the data flows and place the data flow diagrams.>*

## 6 Other Requirements

<This section is **Optional**. Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>