SOFTWARE DESIGN DOCUMENT

SSE3304-1

MINI PROJECT TITLE:

ONLINE PETROL STATION PAYMENT SYSTEM (OPSPS)

PREPARED ON:

31 JANUARY 2022

PREPARED BY:

NO.	STUDENT NAME	MATRIC NUMBER
1.	IZZAT HAKIMI BIN IKMAL HISHAM	208335
2.	DANISH AL-MUHAIMIN BIN MUSTAFFA	205647
3.	JULIA NURFADHILAH BINTI MOHAMAD FAUZI	208256
4.	NOOR SYAHIRAH BINTI MOHD SABRI	207542
5.	MUHAMAD ZAIMI BIN MOHAMED ZAIN	208229
6.	MUHAMMAD HARITH THAQIF BIN MOHD HAIRUDDIN	205823

Version	1.0
Print Date	
Release Date	31 JANUARY 2022
Release State	Final
Approval State	Pending
Approved By	
Prepared By	Group 1
Reviewed By	
Confidentiality Category	Public

CONTENT

1. Introduction

The Software Design Document is a document that will be used to offer documentation that will help in software development by describing how the software should be created. This design document details the designs that are being utilized or are proposed to be used in the project's implementation. The designs mentioned conform to the criteria outlined in the project's Software Requirements Specifications document.

1.1. Purpose

This Software Design Document comprises the design specifications for the Online Petrol Station Payment System (OPSPS), which was designed for the class. The objective of the Software Design Document is to offer a detailed description of a system's design so that software development may begin with a clear knowledge of what is to be developed and how it is anticipated to be built. In general, the designs in this document are meant to serve as guides for our group while we carry out the project. This document might be useful for designers who are intending to upgrade or modify the system's current design.

1.2. Scope

This document includes a detailed overview of the OPSPS design. This document provides a thorough overview of the inventory system's software architecture. It describes the structure and design of several of the modules covered in the SRS. It also shows some of the use cases that were converted into sequential and activity diagrams. The emphasis for this particular Software Design Document is on document production and document modification. The appointed group members in charge of the OPSPS will have the authority to make adjustments where they see necessary.

1.3. Definition and acronyms (to be defined as the project goes)

Acronyms	Definition
OPSPS	Online Petrol Station Payment System.
SRS	Software Requirements Specification.

2. Refinement Diagram

2.1. Use Case Diagram

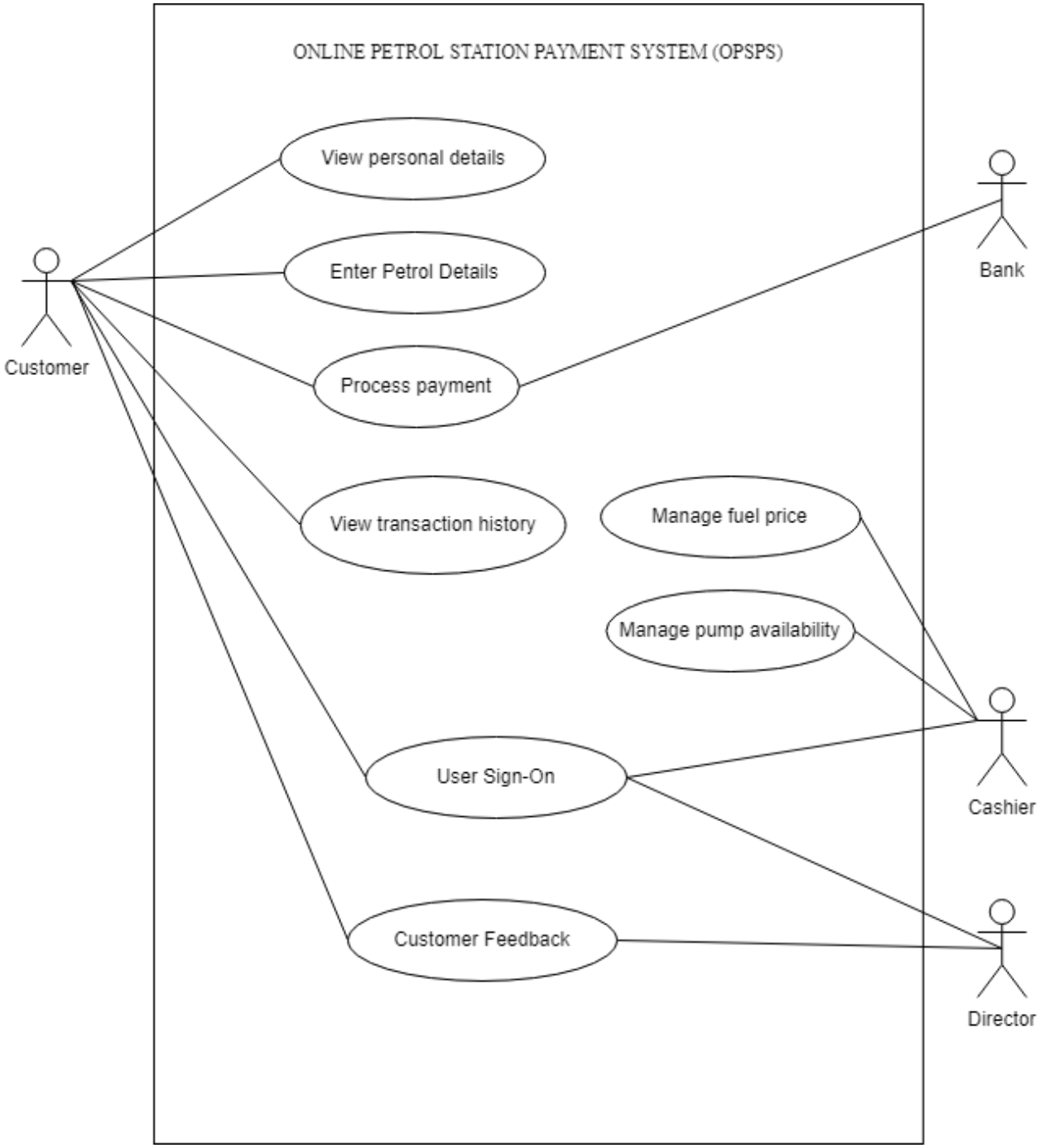


Figure 2.1 shows the use case diagram of online petrol station payment system (OPSPS).

2.2. **Use Case Description Refinement**

2.2.1. Use Case Description for User Sign On:

Brief Description	A user who wishes to access the OPSPS needs to provide to the system a valid login details.
Actors	Customer.
Preconditions	● Login details inserted must be valid. ● User does not input the log in details wrongly. ● Users must be registered to the system.
Main Flow	When the user opens the OPSPS application, a window prompts the user to input his/her phone number and password. The system will then verify that user's identity by referring to the data in customer class. Once a match is found, a message window will appear, informing the user that sign-on is successful.
Alternative Flows	In a case where the user fails to log into the system, there will be two conditions: 1. The user inputs a phone number that exists in the system and the wrong password. 2. The user inputs a non-existing phone number in the system. For the first condition, a message will be prompted showing that the user entered an incorrect password. Meanwhile, an input error message will be prompted if the user entered a non-existing phone number in the system.
Postconditions	Users will be redirected to the system menu user interface. The system will identify the user's identity as a customer.

2.2.2. Use Case Description for View Personal Detail

Brief Description	The customer can view their personal information that has been input to the system.
Actor	Customer
Preconditions	The customer has signed on to the system.
Main Flow	After the customer has successfully signed on to the system. There will be an option for customers to see their personal details. If the customer has chosen to view their personal details, the system will show the information of the customer which consists of Full Name, birthday, e-mail address, home address, phone number. There will be a ‘back’ button if the customer is done looking at their personal details which will return them to the main menu of OPSPS.
Alternative Flow	If the customer's details fail to load, the loading symbol will stay on the screen until the user clicks the back button and retry to access their personal details again.
Postconditions	Users will be redirected to the system menu user interface.

2.2.3. Use Case Description for Enter Petrol Detail

Brief Description	Customers must enter their petrol details (e.g. amount of fuel needed & payment amount) if they want to proceed with purchasing petrol at a petrol station.
Actors	Customer
Preconditions	Customer has logged in to the system.
Main Flow	**Choose a Petrol Station.** The system can track the user's location and would simply give the user a list of recommended locations of the nearest petrol stations to the user once the user opens the OPSPS. The user can pick a petrol station that they want to go to based on the given list. **Pick Pump number.** If the user has picked the petrol station, the system will then show available pumps at the particular petrol station. The user has to enter a pump number where they have parked their vehicle. The changes will be updated in the system. **Enter purchase amount.** The system will then ask the user the amount of fuel required. The user has to input their desired amount, where it needs to be above

	RM4. The amount will be updated in the system.
Alternative Flows	**Pick Pump number.** If the user picks the unavailable petrol pump, the system will show an error message and require the user to choose a different petrol pump number. **Enter purchase amount.** If the user input amount is less than RM 4, the system will show an error message and prompt the user to input a higher purchase amount. In a case where the user wants to change any of the payment details, the user can simply click a 'Back' button on the screen and return the user to the desired menu.
Postconditions	All the payment details will be reserved for the user's use and will be used in the next payment process.

2.2.4. Use Case Description for Process Payment

Brief Description	All payments will be completed once the customers choose a bank, click the “Pay” button and all the requiries (username, password, tac number and other based on bank actor) are correct.
Actors	Customer.
Preconditions	Customer has to log in to the system and has entered all the payment details
Main Flow	After the customer has entered payment details, The system displays a “pay” button (as confirmation of purchase) and the customer needs to choose a bank to proceed to online payment.
Alternative Flows	If the customers want to change their payment details, customers can click the back button to go back and change payment details (such as pump’s number, fuel type). After successfully choosing the correct payment details, customers can click the “Pay” button and choose any bank listed and the system will direct the customer to the Bank Interface. The customers then need to log in to their online bank account (enter username and password) and enter the correct tac number (based on the bank actor).
Postconditions	Users will be redirected to the payment user interface with a message “Payment successful!” and the customer can start fueling their car. If the payment is unsuccessful, the users will be redirected to the payment details with an error message.

2.2.5. Use Case Description for View Transaction History

Brief Description	Transaction history can be viewed by the customers using the button “View Transaction History”. All order details will be displayed in transaction history.
Actors	Customer
Preconditions	Customer has made a payment and fuel petrol
Main Flow	If the customers click on the “View Transaction History” button, the system will display the location of the petrol station, time , date purchase fuel, total amount purchase and pump number. The system also can let a customer download the transaction history into PDF files.
Alternative Flows	If the customers chosen not to click “View Transaction History” button, the system display “X” button to close the page of Transaction History and main menu will be displayed
Postconditions	Transaction history will be save at the page of payment history

2.2.6. Use Case Description for Manage Fuel Price

Brief Description	Cashiers will need to update the price of fuel per litre according to the Malaysia Petrol Price regularly.
Actor	Cashier
Preconditions	Cashiers need to log into the system. Cashiers also need to check the standard price for petrol in Malaysia every day to ensure the price per litre is up-to-date.
Main Flow	Different rates of price per litre will be set by the cashier for diesel and petrol. A selection box for different types of petrol will be shown on the screen. Cashier needs to fill up the details of prices for each type of petrol. The ‘save’ button will be provided after all the information is filled by the cashier. The cashier saved all the information by pressing the ‘save’ button.
Alternative Flows	The system will notify the cashier about the change in rate of petrol price if the cashier did not update the details. An error message will be displayed if there are some details not completed. The system also allows cashiers to cancel the updating process by displaying a ‘cancel’ button.
Postconditions	The petrol price will be updated into the system database.

2.2.7. Use Case Description for Manage Pump Availability

Brief Description	Cashier able to update the pump number that is available in that petrol station.
Actor	Cashier
Preconditions	Cashier needs to log in to the system using a unique ID. The system needs to successfully identify the location of the petrol station where the cashier belongs.
Main Flow	After the system successfully identifies the specific location of a petrol station, the cashier needs to regularly update the pump number that is available into the system.
Alternative Flows	If there is no pump available in that station, the venue manager will update the unavailability of the pump number.
Postconditions	The availability of the pump's number will be updated in the system database. The customer will be able to look out the availability of the pump number before make payment

2.2.8. Use Case Description for Process Feedback

Brief Description	The customer can leave feedback regarding the application each time after a complete payment. The director of OPSPS can view the feedback given by customers and use this feedback to adjust and improve current and future actions and behaviours of this system.
Actors	Customer, Director
Preconditions	Customers or directors need to log into the system.
Main Flow	After receiving feedback from the customer, The system will notify the director when get the feedback and the director can view the feedback by pressing “View Feedback” to see the message given by the customer.
Alternative Flows	If the customer does not want to leave a feedback, the customer can click the cancel button. If the director chooses to not view feedback to the system, then the director can press the “X” button to close the system.
Postconditions	The feedback will be updated into the feedback database.

2.3. Activity Diagram

2.3.1. User Login

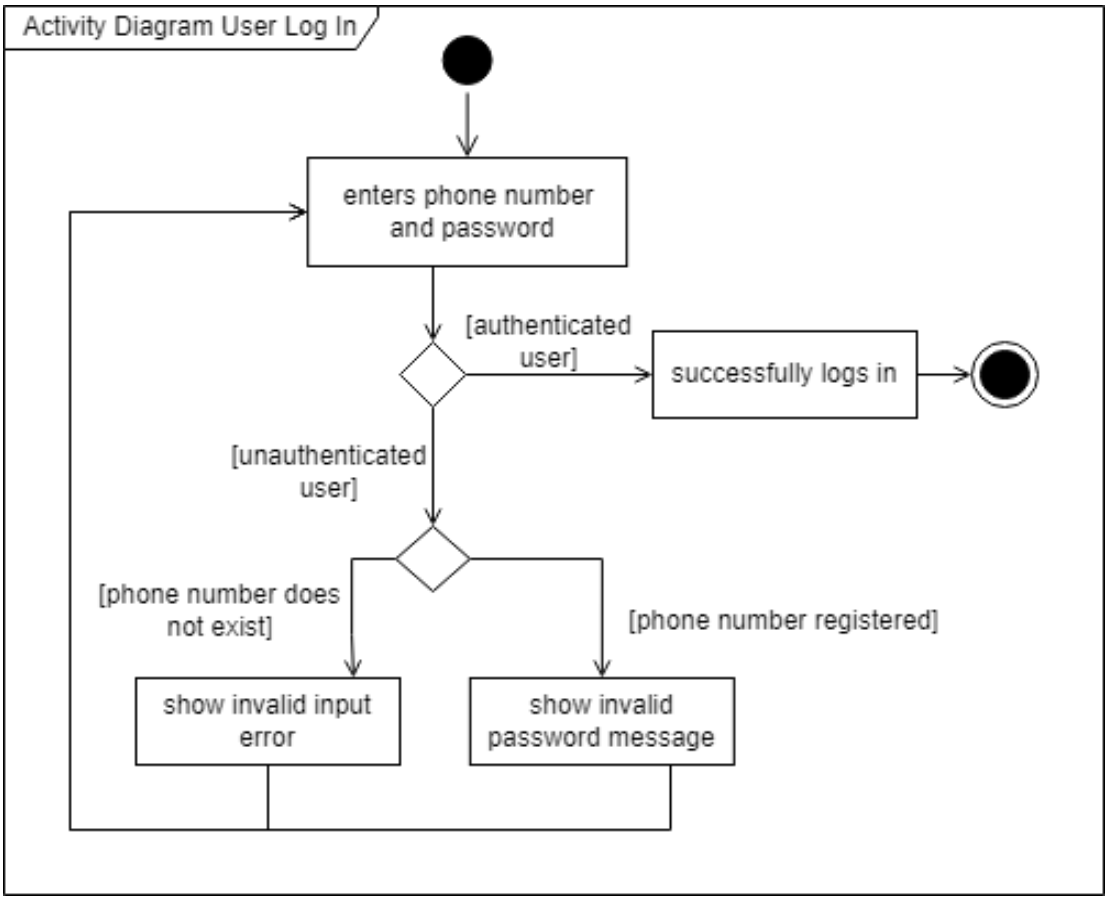


Figure 2.3.1 shows the flow when a user wants to log in into the system. The system will validate the phone number and password inserted by the user. If the system recognizes the login details, the user will be redirected to the main menu. Conversely, if the user inputs an invalid credential, the system will print out invalid input message.

2.3.2. View Personal Detail

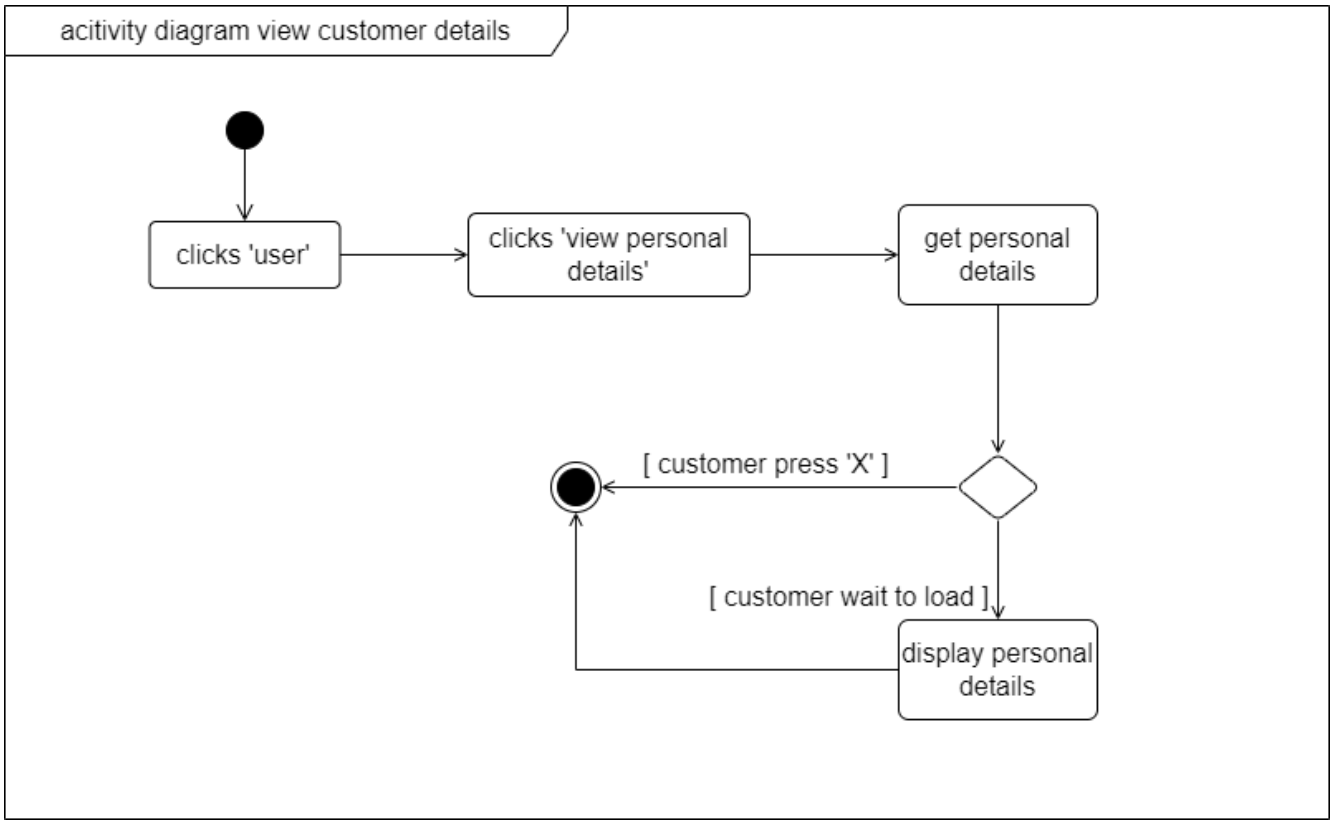


Figure 2.3.2 shows the flow when a customer wants to view their personal details. If the customer decides to cancel and press the 'X' button, the customer will be brought back to the main menu.

2.3.3. Enter Petrol Detail

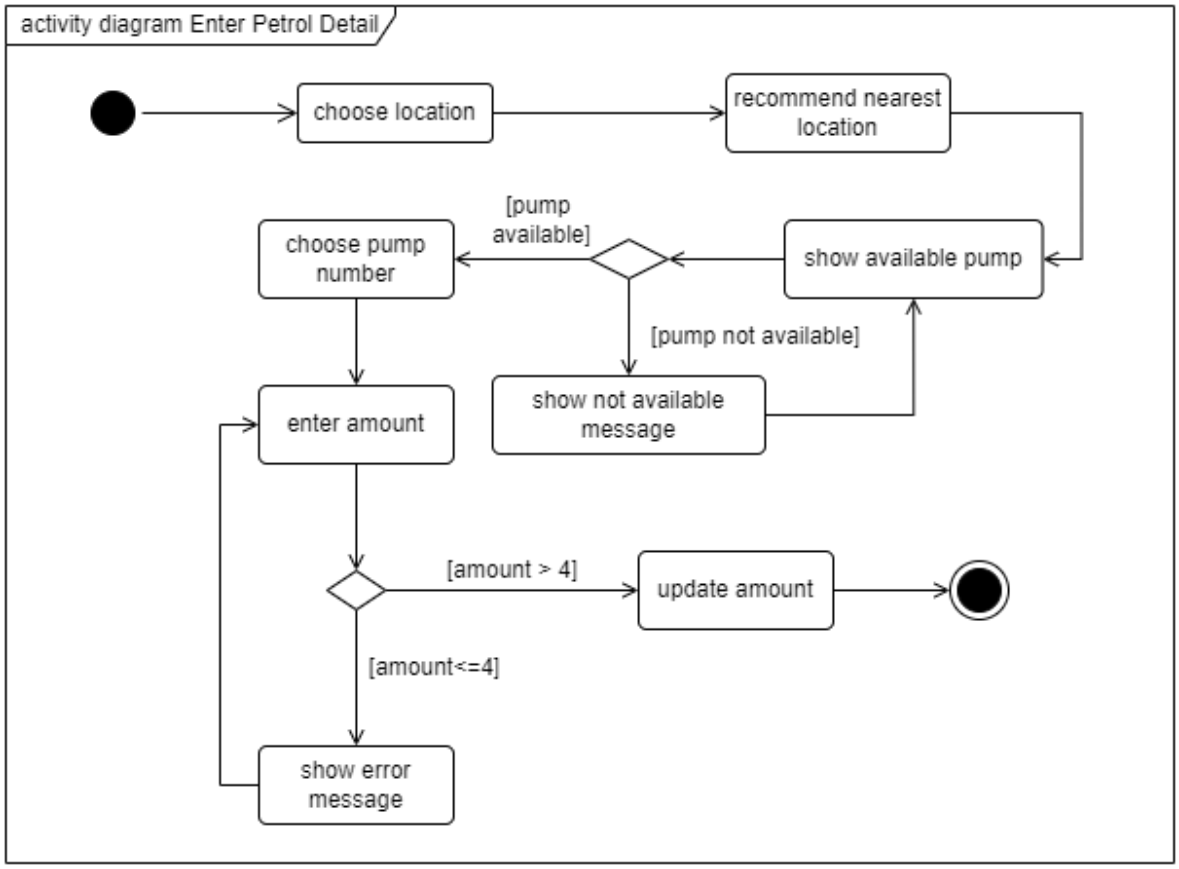


Figure 2.3.3 shows the process for the customer to purchase petrol. The system will ask the customer to choose the nearest location and fill the required details such as pump number and fuel amount before proceeding to the payment system.

2.3.4. Process Payment

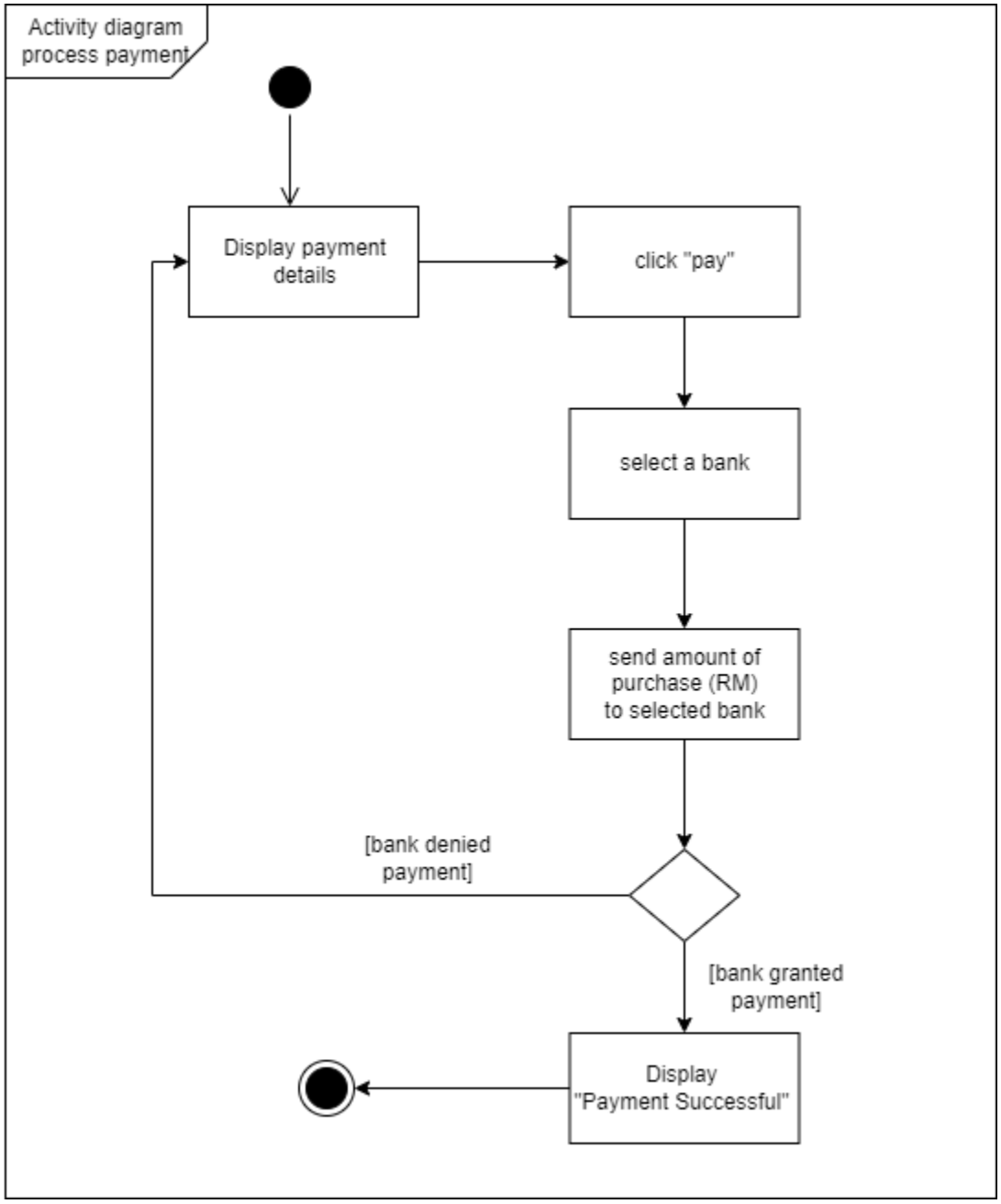


Figure 2.3.4 shows the flow of the process of the payment. Firstly, after inserting all the petrol details required, the customers will need to confirm all the payment details by clicking the “pay” button. The customers are then required to select a bank listed and they will be redirected to the bank interface based on the chosen bank. Lastly, if the payment succeeds, it will display “Payment Successful” else the customers will be directed back to the payment details with an error message.

2.3.5. View Transaction History

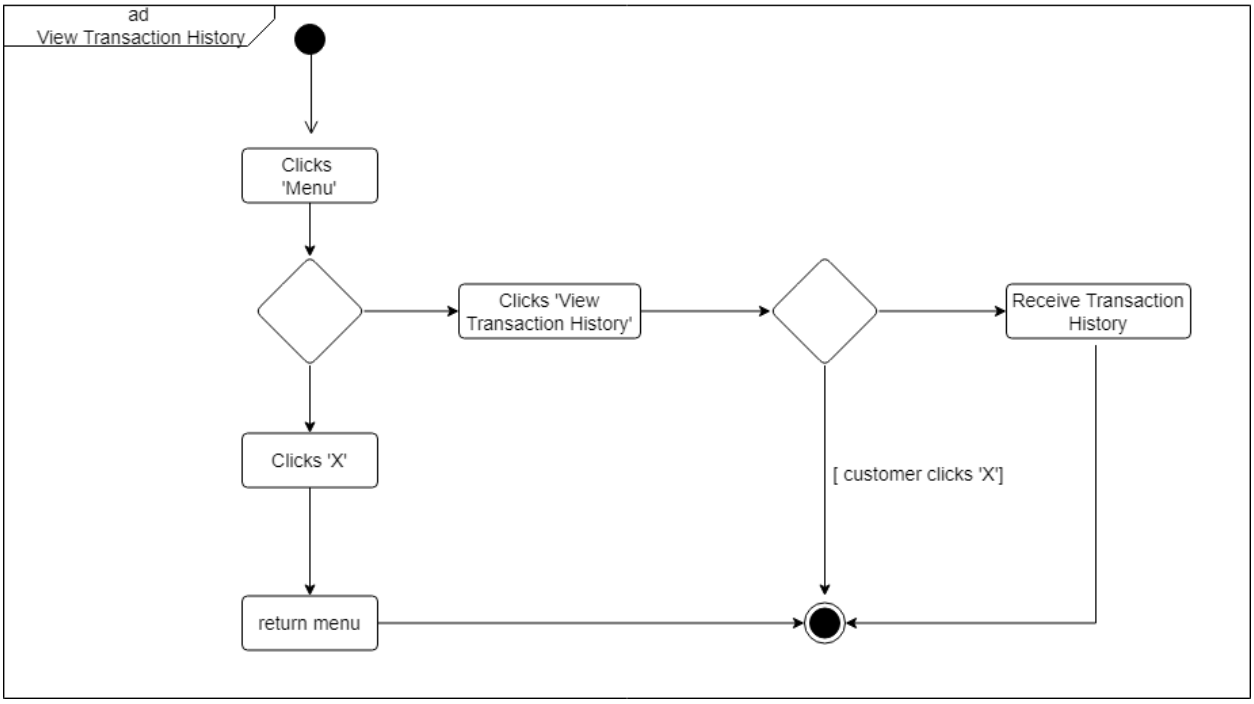


Figure 2.3.5 shows the flow when a customer wants to view their transaction history. If the customer decides to cancel and press the 'X' button, the customer will be brought back to the main menu.

2.3.6. Manage Fuel Price

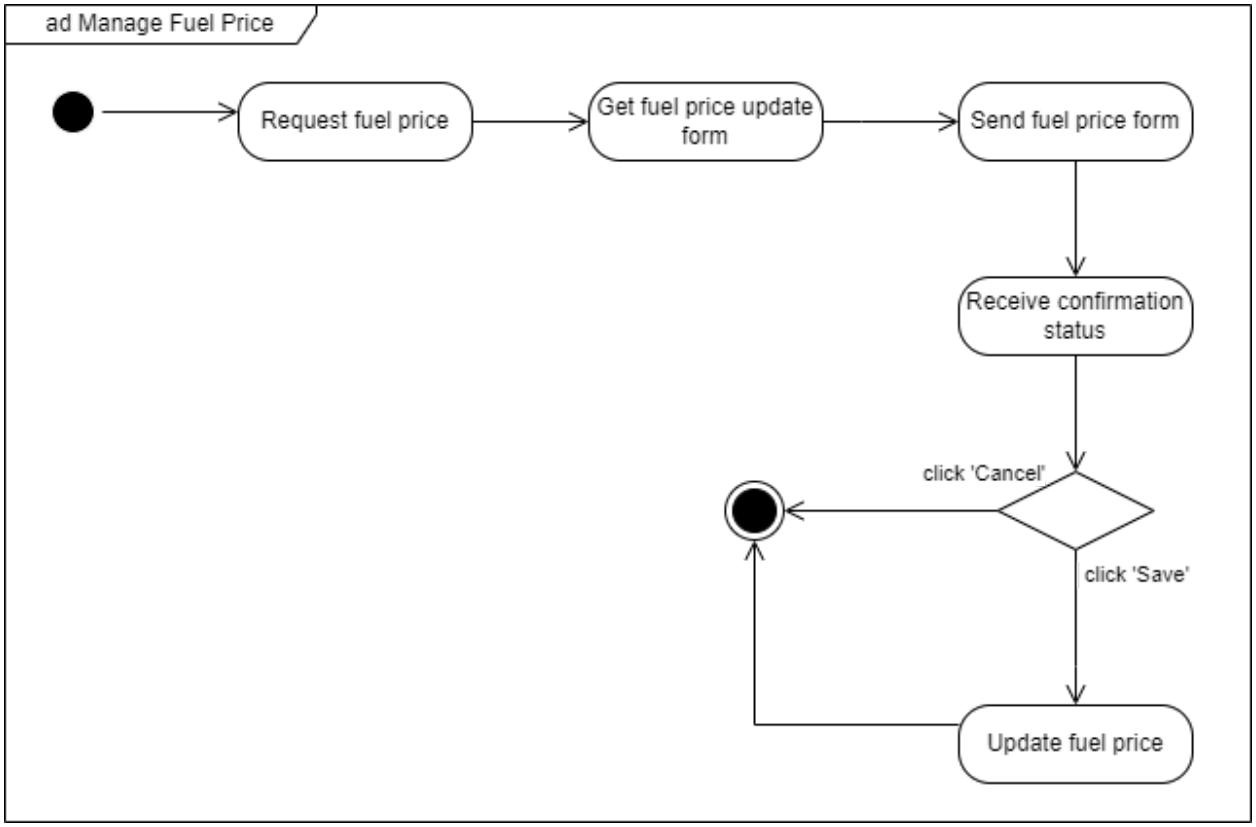


Figure 2.3.6 portrays the flow of updating fuel prices from start point to finish point with one decision path that may be faced while the updating process is being executed.

2.3.7. Manage Pump Availability

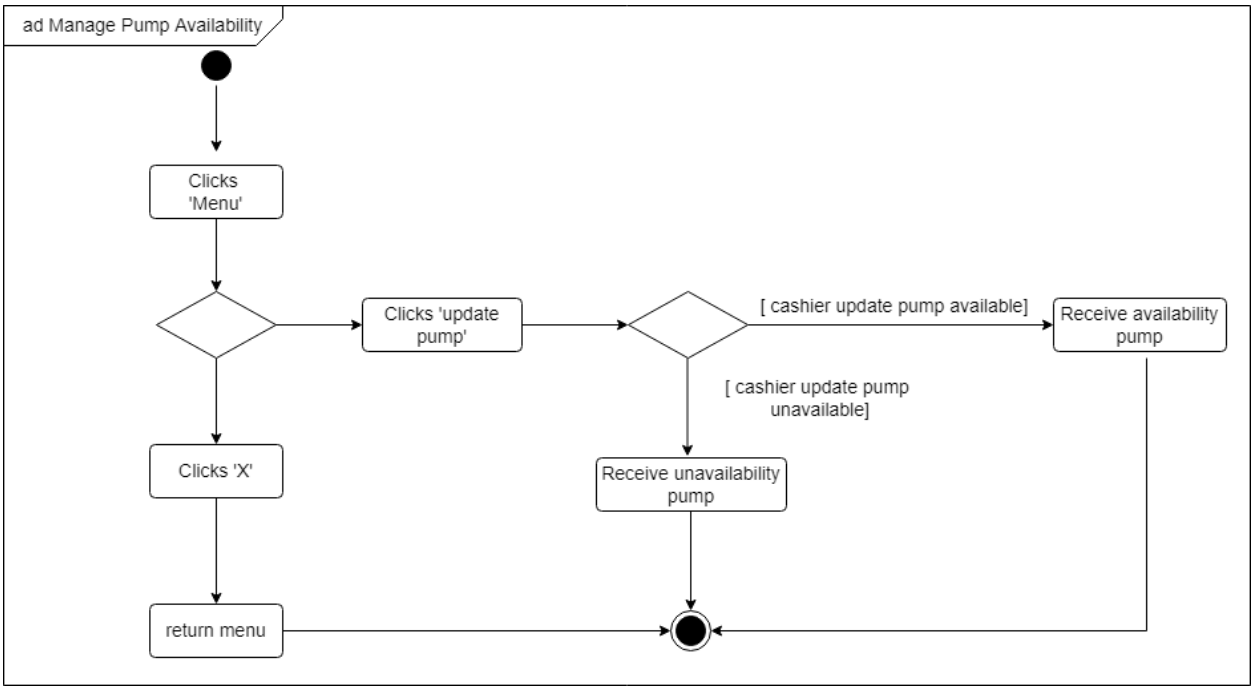


Figure 2.3.7 shows the flow of view transaction history from start point to finish point. When the cashier wants to update a pump , the cashier needs to click the menu and click update pump.The cashier are then required to select which pump and press available or unavailable pumps. If the cashier decides to cancel and press the ‘X’ button , the cashier will be brought back to the main menu.

2.3.8. Process Feedback

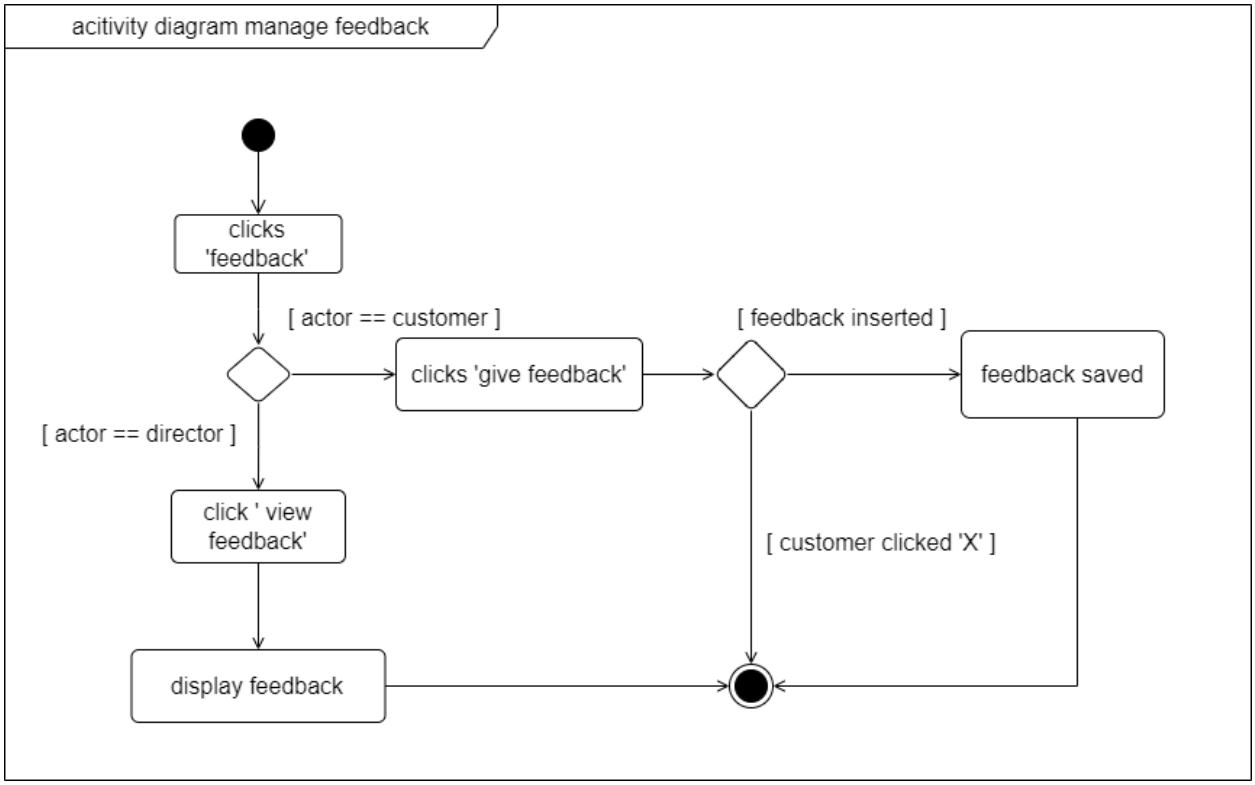


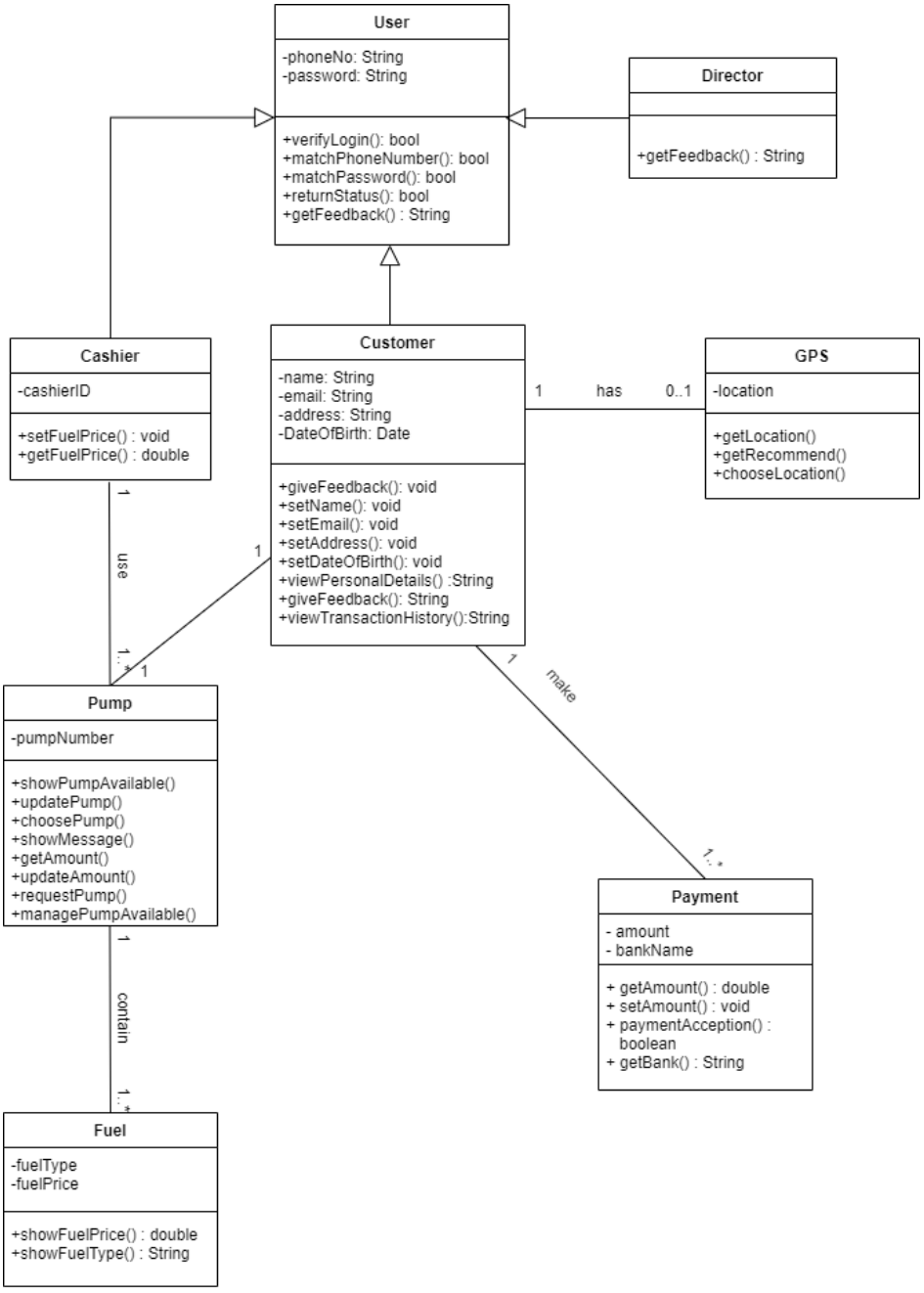
Figure 2.3.8 shows the flow of two types of user. Firstly, when a customer wants to insert feedback after each successful transaction. Secondly, when the director wants to view the feedback given by customers.

3. Class Diagram

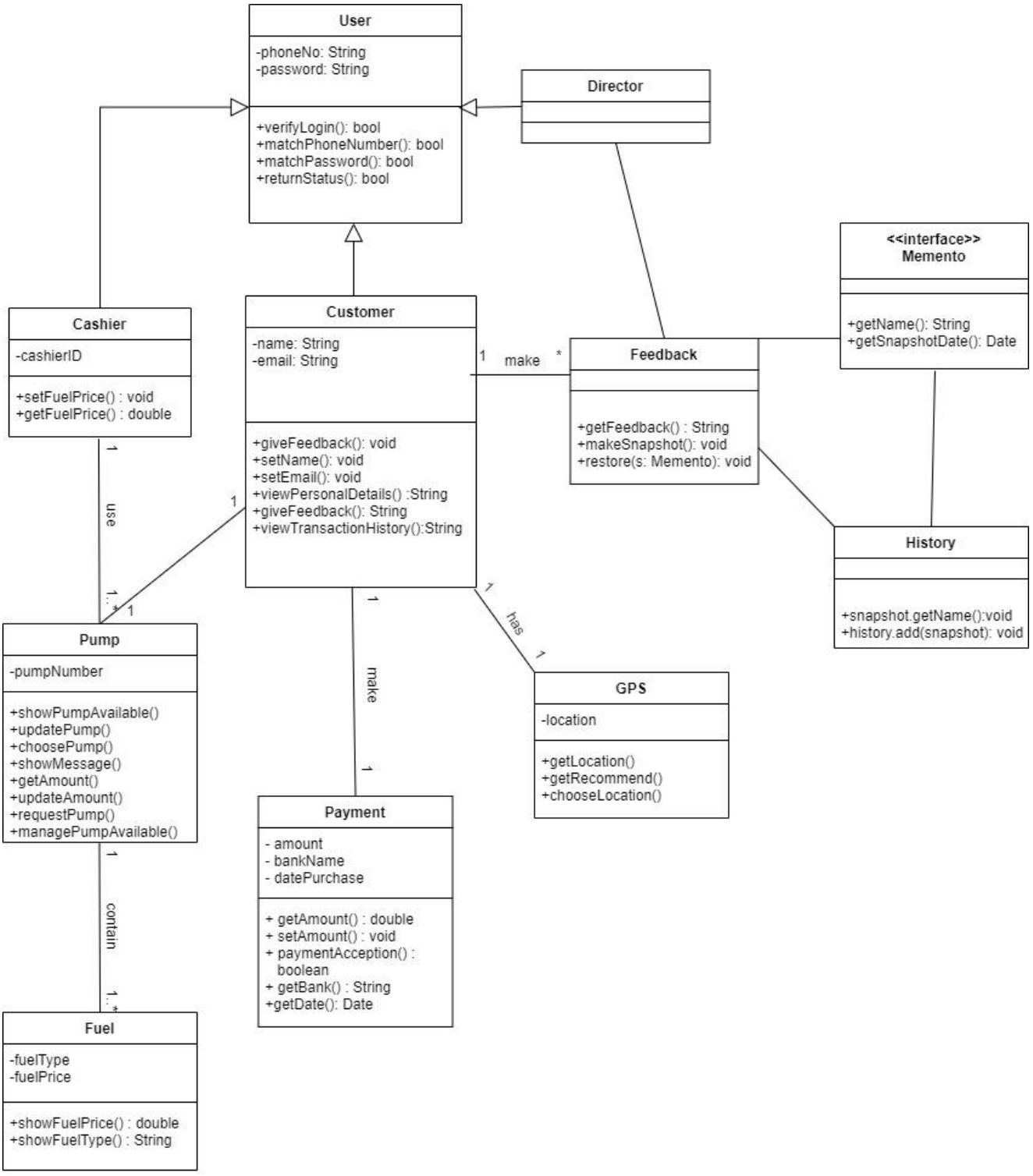
Class diagrams show the structure of a system. The static structure of Object-Oriented systems, or how they are constructed rather than how they behave, is captured by class diagrams. Architectural design is aided by class diagrams. The fundamentals of Object Oriented systems are represented by class diagrams. They identify the various classes, their interrelationships, and interactions.

3.1. Class Diagram Iteration

3.1.1. Iteration 1


User
-phoneNo: String
-password: String
+verifyLogin(): bool
+matchPhoneNumber(): bool
+matchPassword(): bool
+returnStatus(): bool
+getFeedback() : String
Director
+getFeedback() : String
Cashier
-cashierID
+setFuelPrice() : void
+getFuelPrice() : double
Customer
-name: String
-email: String
-address: String
-DateOfBirth: Date
+giveFeedback(): void
+setName(): void
+setEmail(): void
+setAddress(): void
+setDateOfBirth(): void
+viewPersonalDetails() :String
+giveFeedback(): String
+viewTransactionHistory():String
GPS
-location
+getLocation()
+getRecommend()
+chooseLocation()
1
has
0..1
1
use
1..*
1
Pump
-pumpNumber
+showPumpAvailable()
+updatePump()
+choosePump()
+showMessage()
+getAmount()
+updateAmount()
+requestPump()
+managePumpAvailable()
1
make
1..*
Payment
- amount
- bankName
+ getAmount() : double
+ setAmount() : void
+ paymentAcception() : boolean
+ getBank() : String
1
contain
1..*
Fuel
-fuelType
-fuelPrice
+showFuelPrice() : double
+showFuelType() : String

3.1.2. Iteration 2



4. Design Pattern

4.1. Memento design

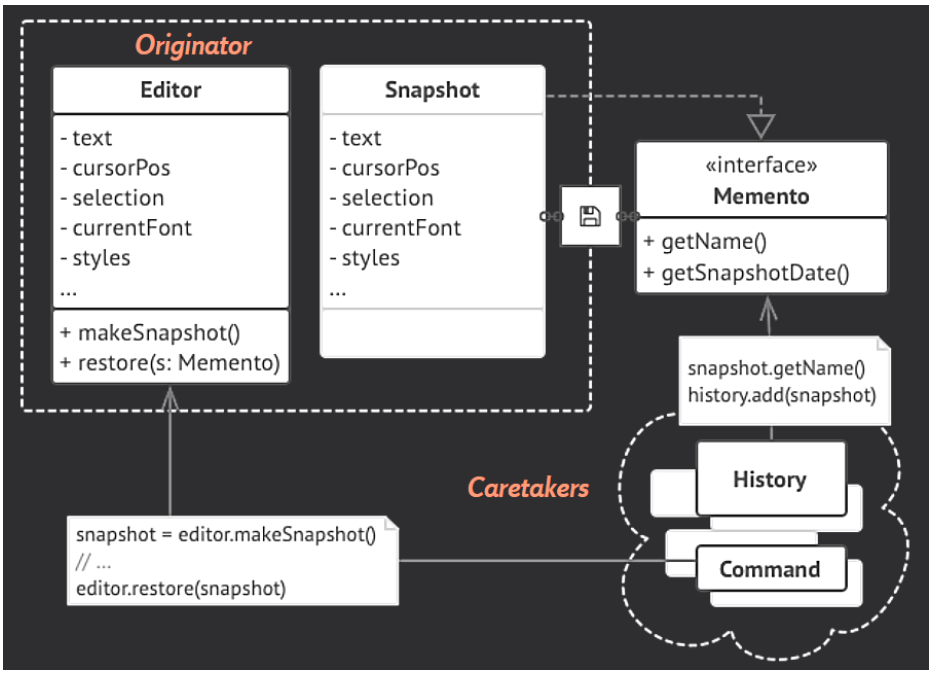
4.1.1. Definition

The Memento Design Pattern is implied to offer a solution to implement undoable actions. This is achieved by saving the states of an object at a given instant and restoring it if the actions performed needs to be undone.

4.1.2. Software design component

1. Memento element to store in different internal states in special objects.
2. Originator element offers set and get values from the currently targeted Memento. It also creates new mementos and assigns current values to them.
3. Caretaker : Holds an ArrayList that contains all previous versions of mementos. It can store are retrieve stored mementos

4.1.3. Software design visualisation



4.1.4. Software design implementation

1. A Copy Of Originator's State Called Snapshot Are Created From Originator Object
2. The Snapshot Are Stored In A Special Object Called Memento
3. Mementos Are Then Stacked Inside Another Object, Called Caretakers
4. When A User Triggers The Undo, The History Grabs The Targeted Memento From The Stack And Passes It Back To The Originator, Requesting A Roll-Back
5. The Originator Changes Its Own State With The Values Taken From The Memento

4.1.5. Software design implementation in class diagram

In feedback class, customers are given the option to provide their feedback based on the application after each successful transaction. The feedback is inserted into a text field on the interface. Customers are given the 'undo' button to undo any changes they made to the text before submitting the form. This is useful when a customer accidentally highlights and deletes the whole text or simply wants to undo any changes.

4.1.6. Advantages of software design implementation

Firstly, to Produce snapshots of the object’s state without violating its encapsulation. Secondly, to restore an object to its previous state. Thirdly, to ensure the object's state data is safe and secure.

5. Sequence Diagram

5.1. User Login

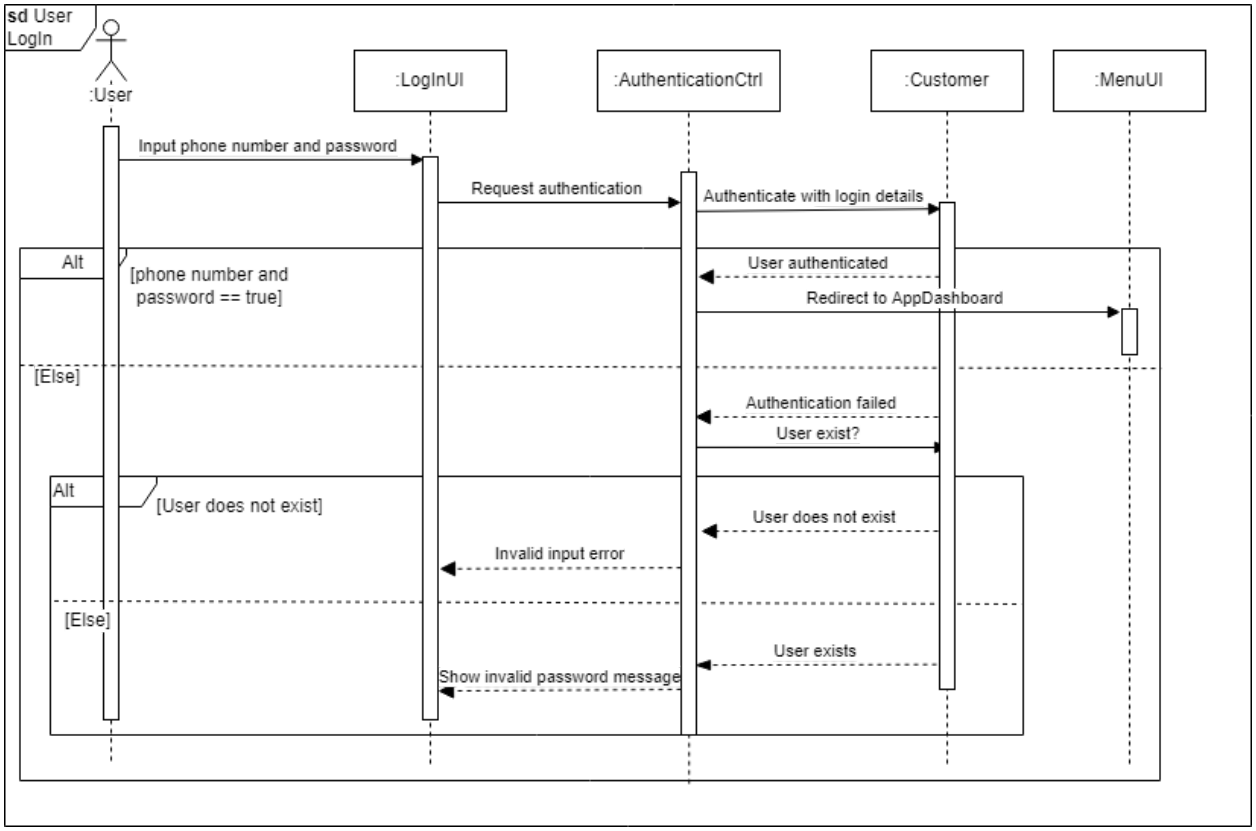


Figure 5.1 shows the sequence diagram of an user actor logging in into the system. The system will authenticate the login details inserted by the actor before redirecting the user to the main main.

5.2. View Personal Details

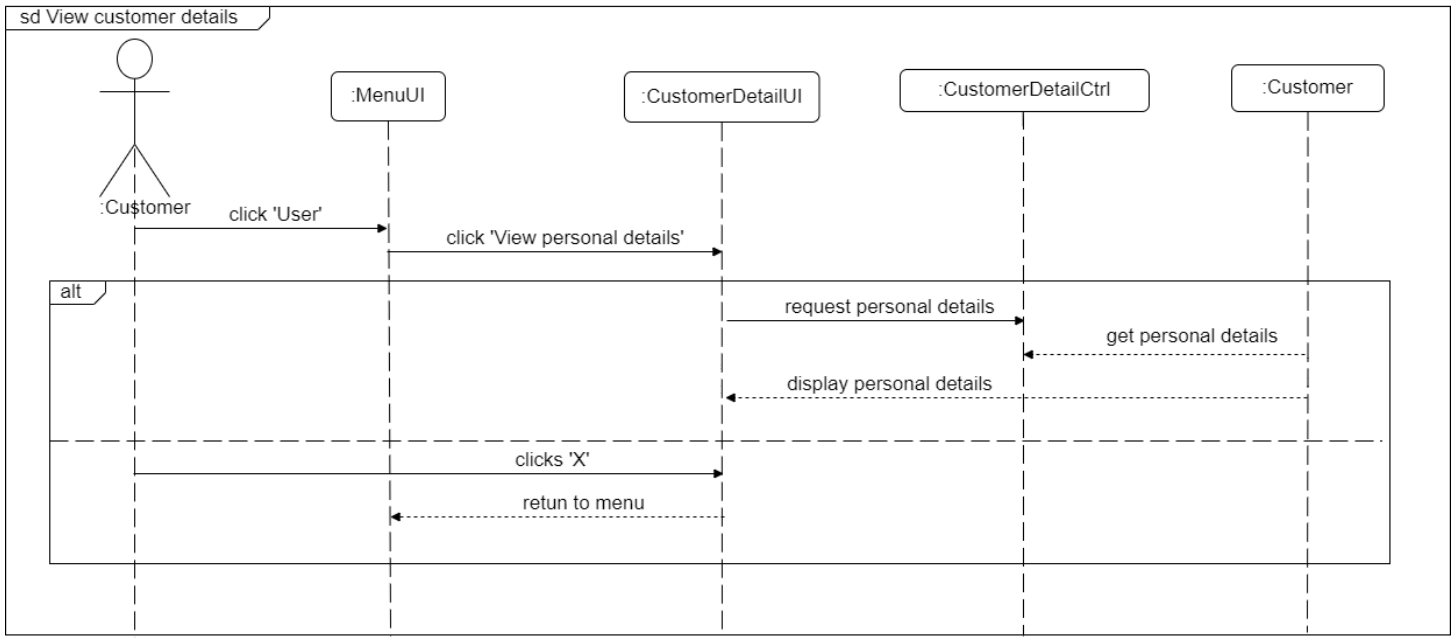


Figure 5.2 shows the sequence of a customer viewing their personal details.

5.3. Enter Petrol Details

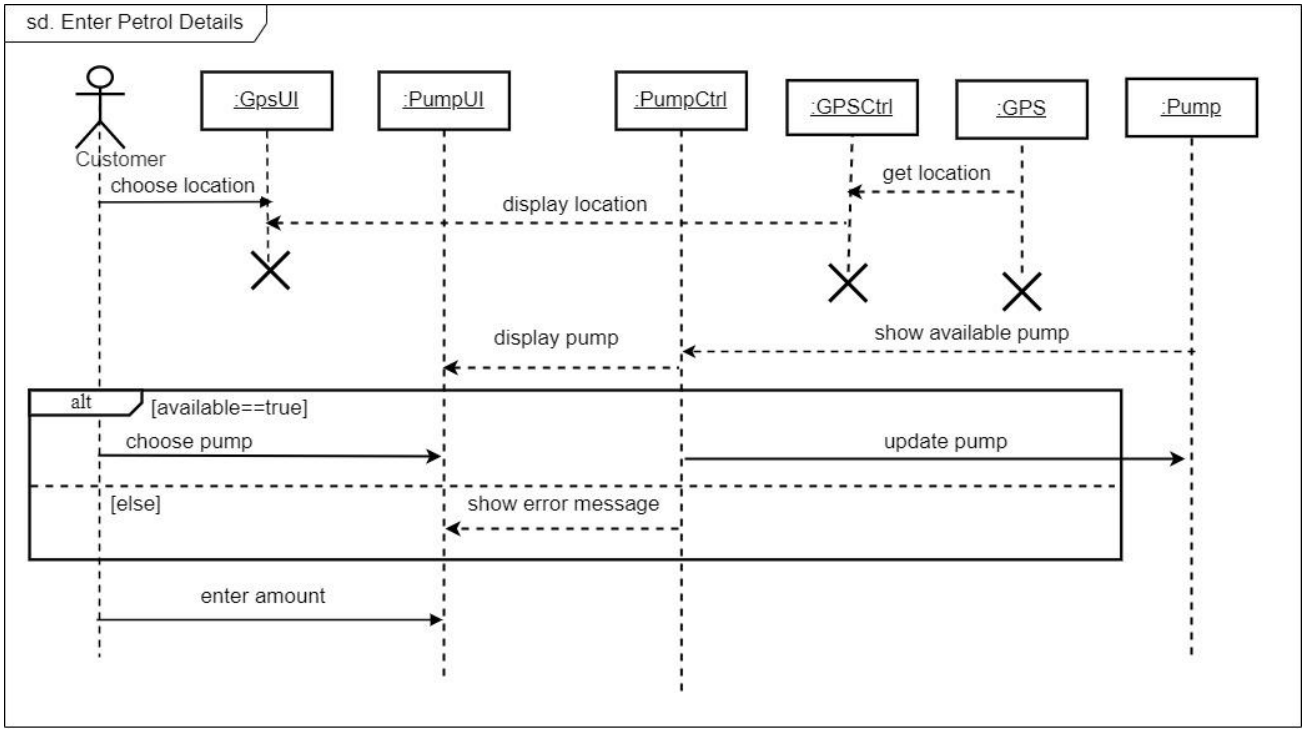


Figure 5.3 shows the sequence diagram of entering the petrol detail process to purchase petrol in a petrol station from choosing the nearest location until inserting the amount of petrol required.

5.4. Process Payment

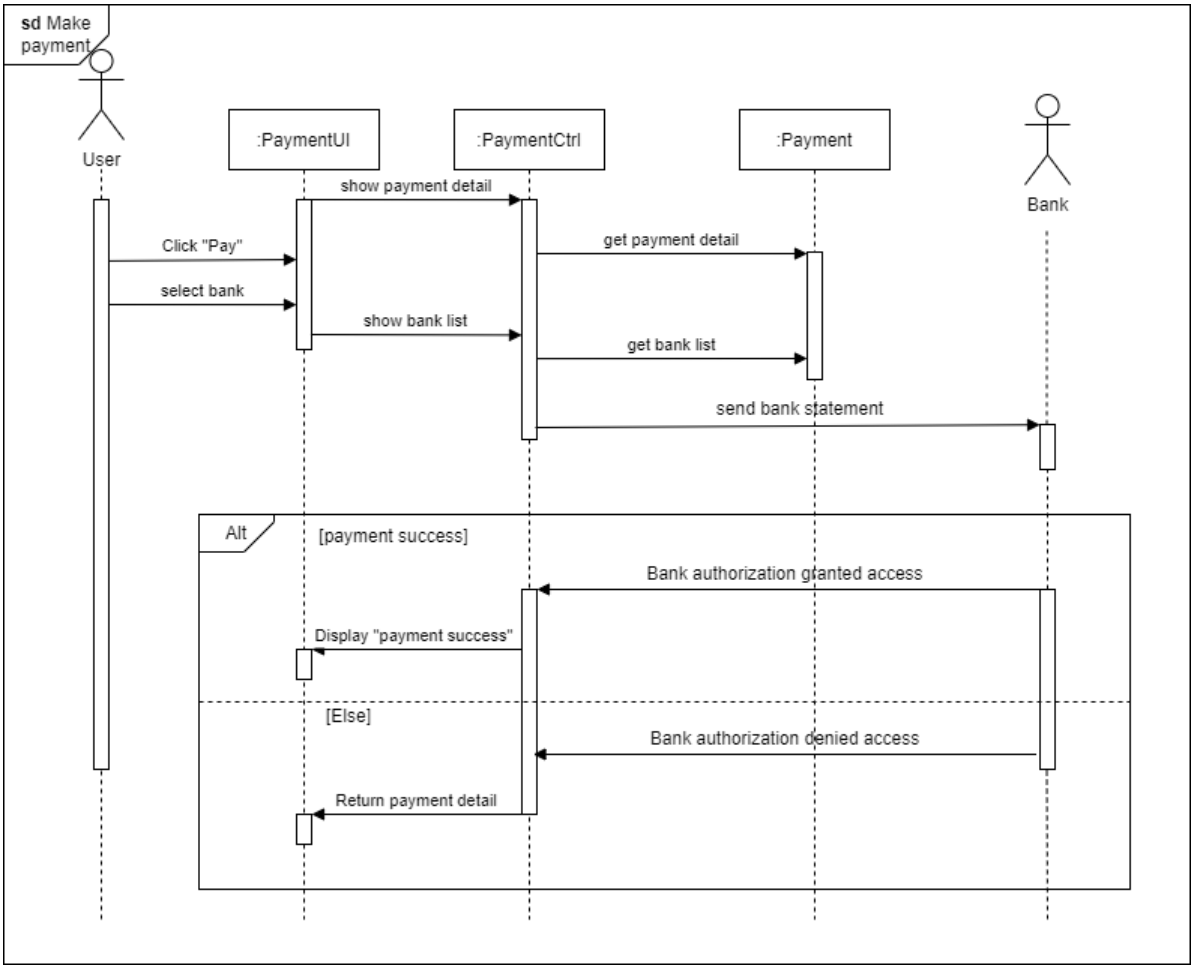


Figure 5.4 shows the sequence of a customer making their payment using online banking.

5.5. View Transaction History

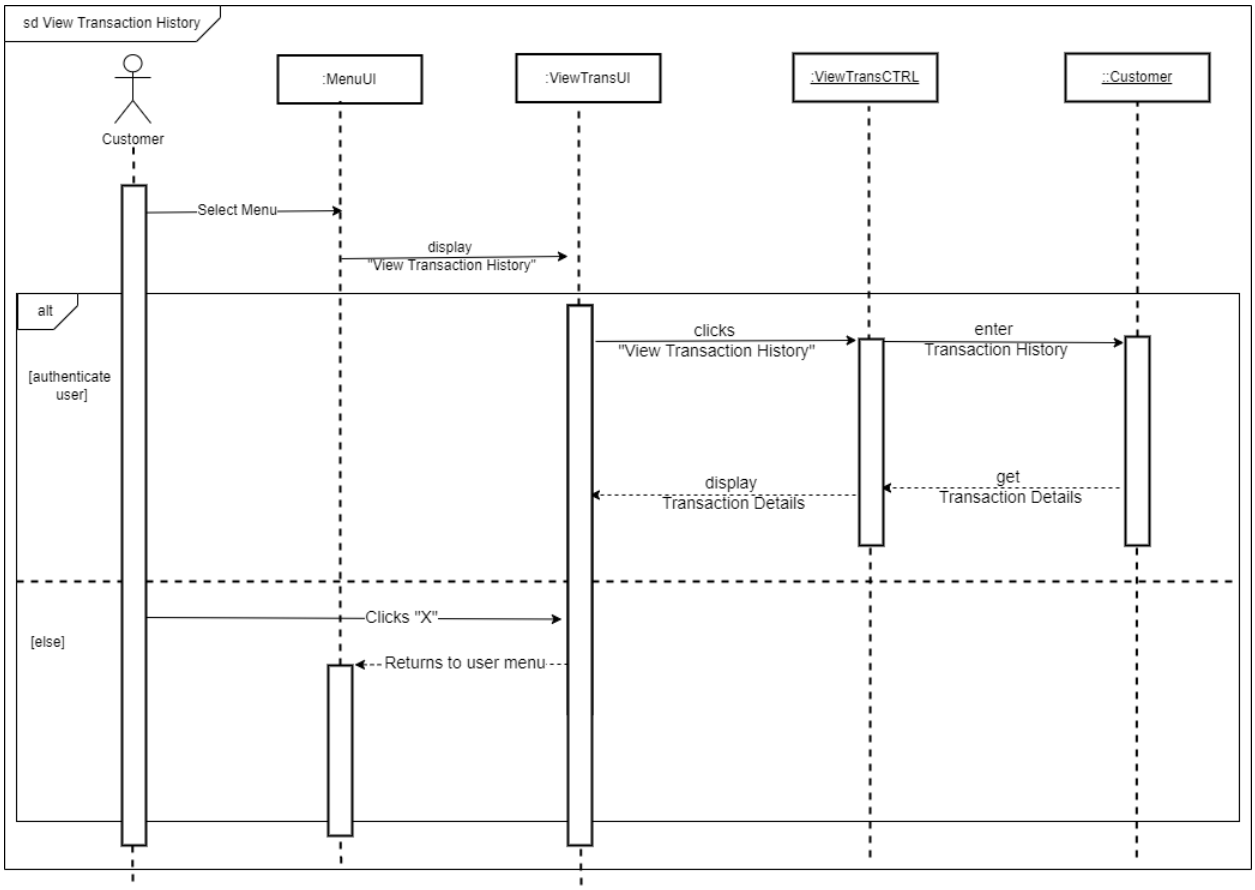


Figure 5.2 shows the sequence of a customer viewing their transaction history.

5.6. Manage Fuel Price

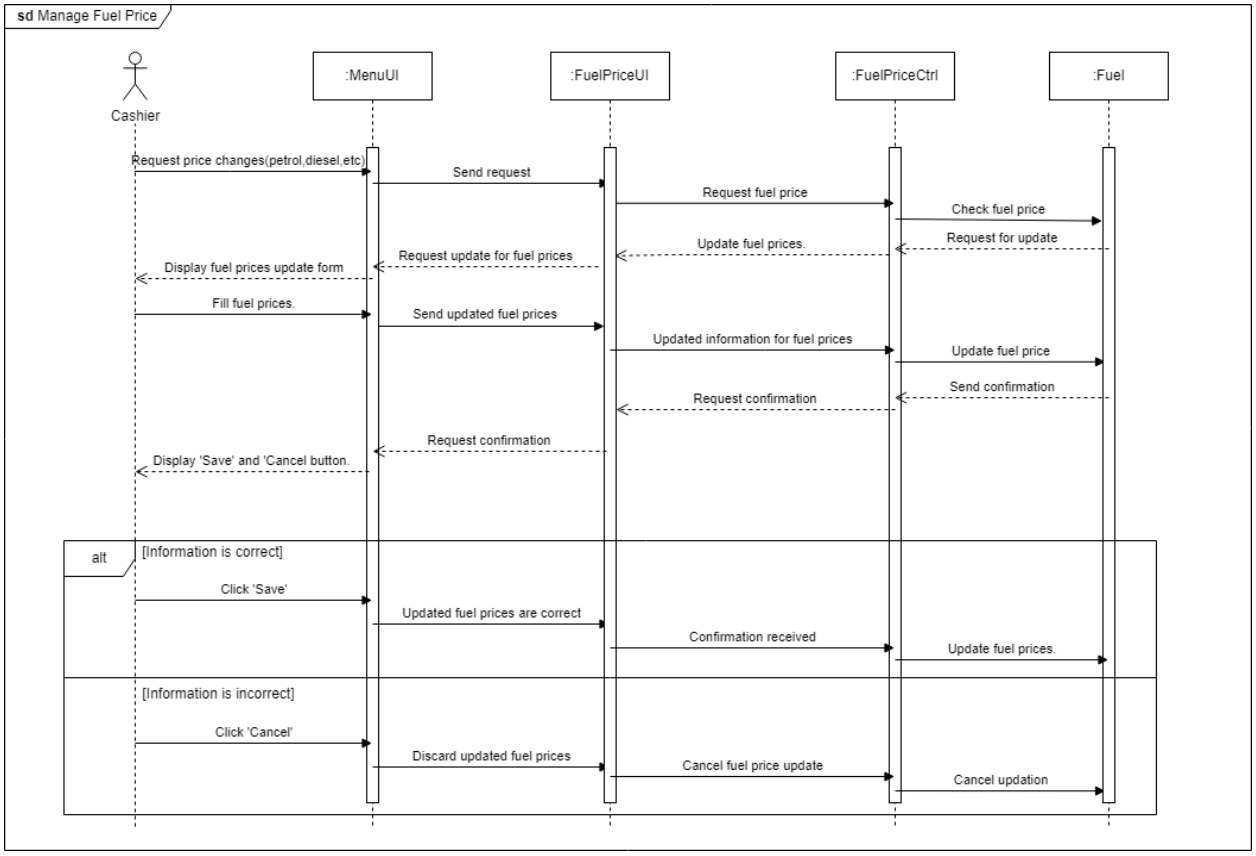


Figure 3.6 shows the flow of the cashier which is the actor, on updating the prices of each fuel type.

5.7. Manage Pump Availability

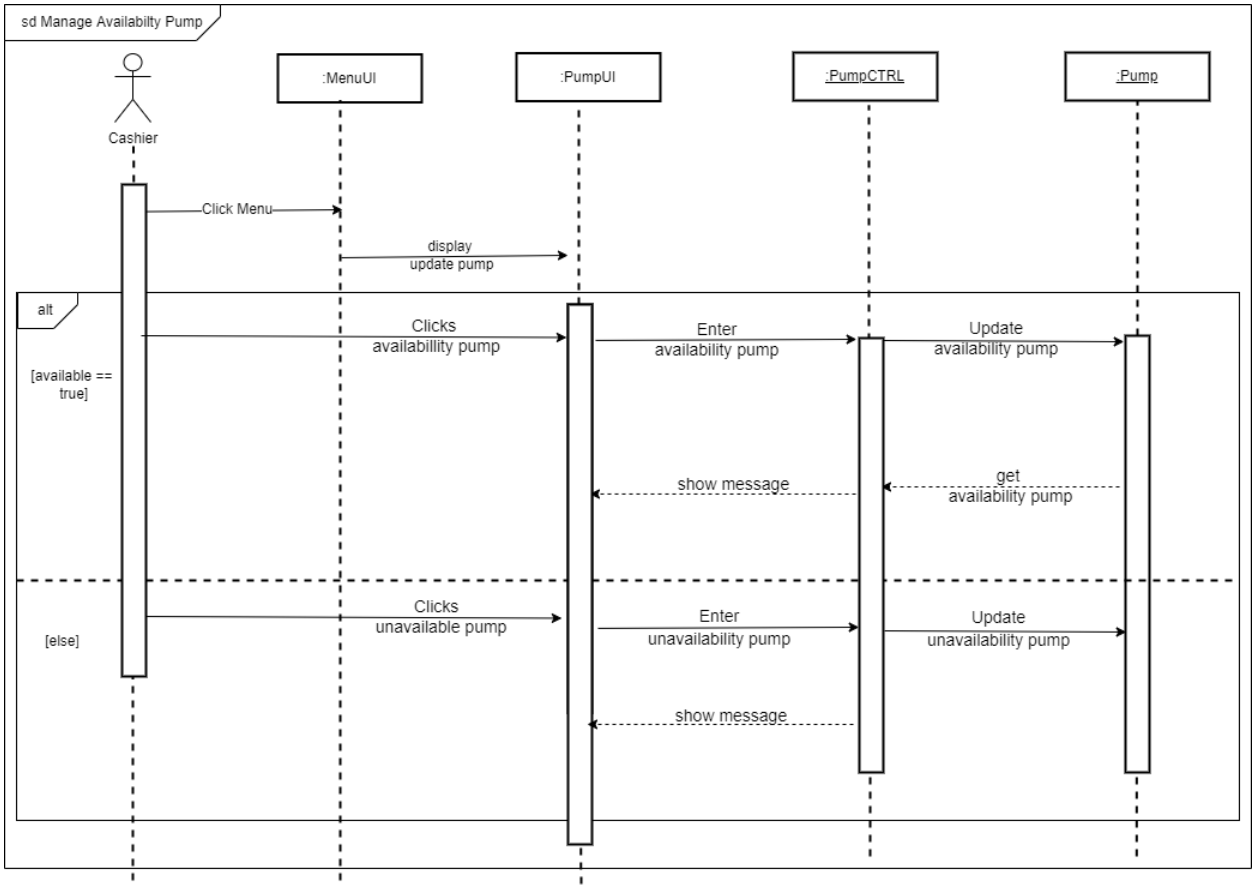


Figure 5.7 shows the sequence of a cashier updating available and unavailable pump.

5.8. Manage Feedback

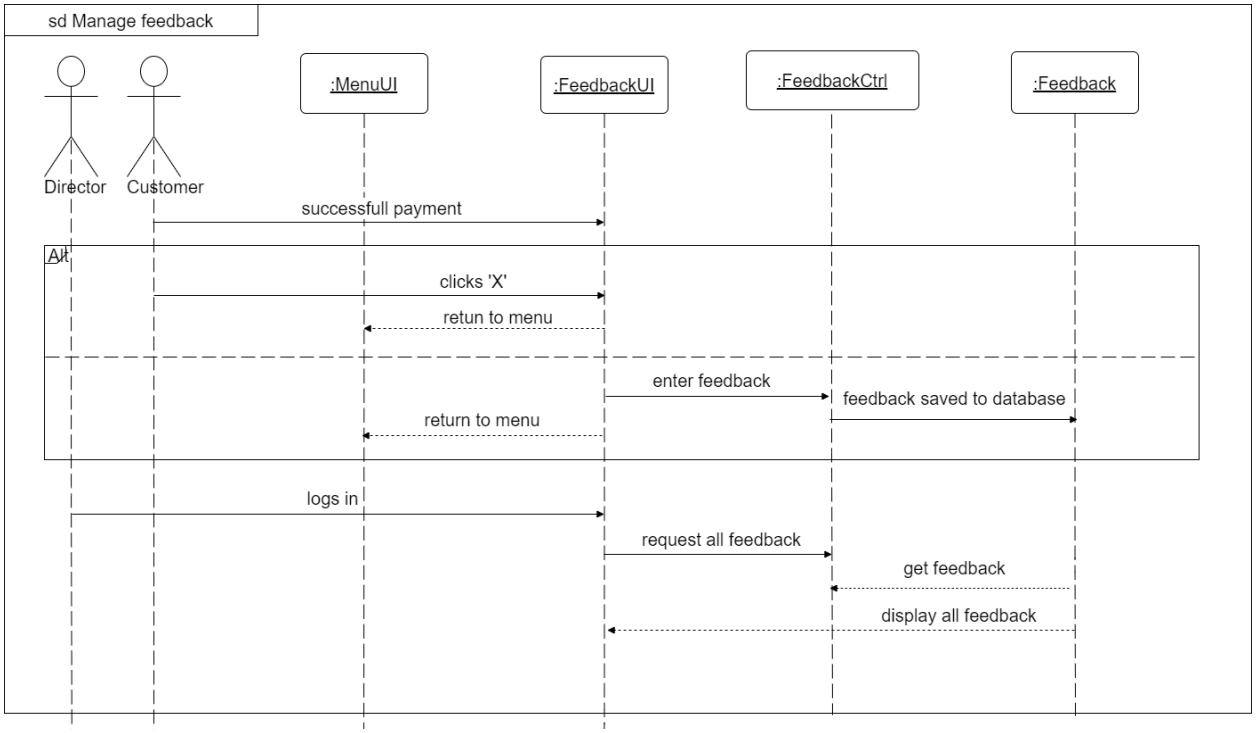


Figure 5.8 shows the sequence of two types of user. Firstly, when a customer wants to insert feedback after each successful transaction. Secondly, when the director wants to view the feedback given by customers.

6. State Diagram

6.1. State diagram for machine payment status

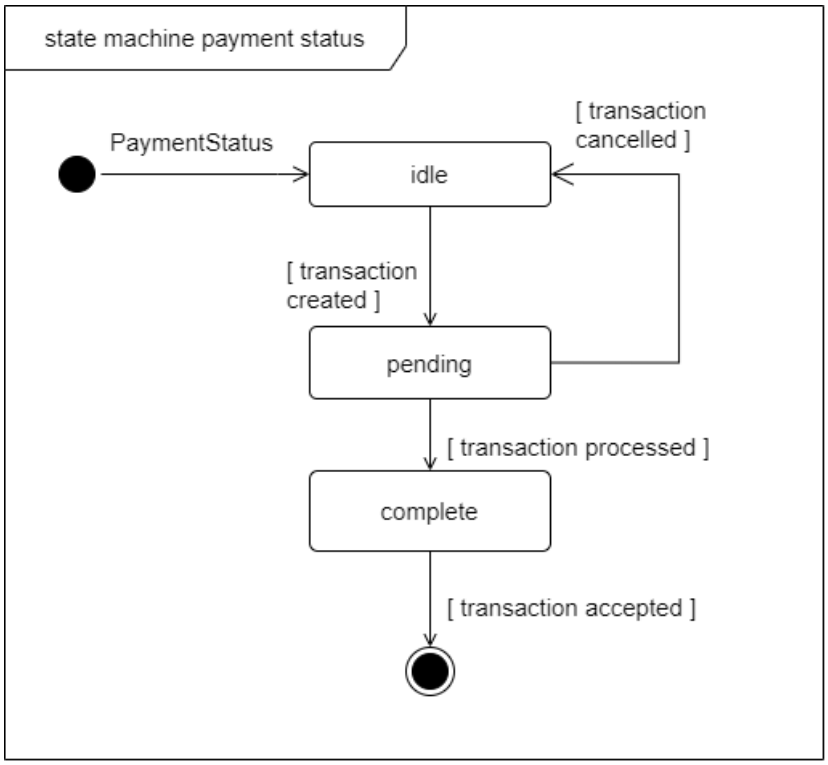


Figure 6.1 shows the changes of state in machine payment status. When no payment is made, the status is idle. When the payment is being processed by the bank, the state is pending. Finally, when the bank approves the transaction, the state is accepted.

6.2. State diagram for machine pump availability

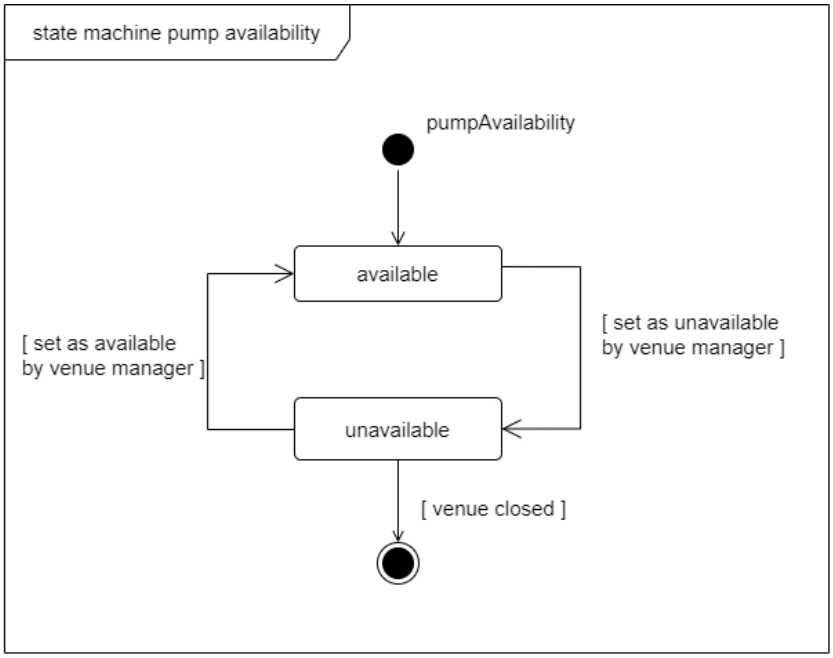


Figure 6.2 shows the changes of state in machine pump availability. When a machine is broken, the cashier will set the pump station as unavailable. Hence, customers are unable to choose the unavailable pump station for payment.

7. Algorithmic/ Non-Algorithmic Logic Specification

7.1. User Login

Conditions and actions	Rule 1	Rule 2	Rule 3
Conditions			
Is the phone number correct?	Y	Y	N
Is the password correct?	Y	N	N
Actions			
Redirect to menu user interface	X		
Show wrong password message		X	
Show input error message			X

Table 7.1 describes the sequence of internal logical steps when user logging in into the system and its result of different sets of inputs.

7.2. View Personal Details

Conditions and actions	**Rule 1**	**Rule 2**	**Rule 3**	**Rule 5**
Conditions				
Customer clicked 'View Personal Detail'?	Y	N	Y	-
Customer details successfully fetched?	Y	-	N	-
Actions				
Display personal details	X			
Display error message			X	
Stays on main menu interface		X		X

The table 7.2 shows non-algorithmic logic specification when the customer wants to view their personal details.

7.3. Enter Petrol Details

Conditions and actions	**Rule 1**	**Rule 2**	**Rule 3**	**Rule 4**	**Rule 5**
Conditions					
Is the location selected?	Y	Y	Y	Y	Y
Is the petrol number selected?	-	Y	N	N	Y
Is the fuel amount selected?	-	N	Y	N	Y
Actions					
Show the nearest location station	X				
Redirect to payment interface					X
Show error message		X	X	X	

Table 7.3 shows the condition and actions for entering the petrol details process. The system will proceed to the payment stage if all the three conditions are successfully filled. If one of the conditions is not fulfilled, the system will show an error message.

7.4. Process Payment

Conditions and actions	Rule 1	Rule 2	Rule 3
Conditions			
Is the bank listed chosen?	Y	Y	N
Is the payment successful?	Y	N	-
Actions			
Show payment successful message	X		
Redirect to payment details		X	
No action			X

Table 7.4 describes the conditions and action when a user makes a payment. The system will display a payment successful message if all the conditions required are filled correctly.

7.5. View Transaction History

Conditions and actions	Rule 1	Rule 2	Rule 3
Conditions			
Is the view transaction history selected?	Y	N	Y
Is the print transaction history selected?	-	-	Y
Actions			
Shows view transaction history	X		X
Back to the main menu		X	
Shows in PDF			X

Table 7.5 shows the conditions and action when a user selects a view transaction history . The system will display a transaction history if all the conditions required are filled correctly.

7.6. Manage Fuel Price

Conditions and actions	Rule 1	Rule 2	Rule 3
Conditions			
Is the fuel price update form filled?	Y	N	-
Is the button ‘Save’ clicked?	Y	Y	-
Is the button ‘Cancel’ clicked?	-	-	Y
Actions			
Show successful update	X		
Show invalid error message		X	
Redirect to main menu			X

Table 7.6 describes the sequence of internal logical steps for managing fuel price and its result of different sets of inputs.

7.7. Manage Pump Availability

Conditions and actions	Rule 1	Rule 2	Rule 3	Rule 4
Conditions				
Is the update pump selected?	Y	Y	Y	N
Is the availability pump selected?	Y	N	N	-
Is the unavailability pump selected?	N	Y	N	-
Actions				
Show successful update	X	X		
Redirect to main menu				X
Show invalid error message			X	

Table 7.7 shows the condition and actions for managing pump availability. The system will proceed to the payment stage if all the two conditions are successfully filled. If one of the conditions is not fulfilled, the system will show an error message.

7.8. Manage Feedback

Conditions and actions	**Rule 1**	**Rule 2**	**Rule 3**	**Rule 4**	**Rule 5**
Conditions					
The user is a customer?	Y	Y	Y	-	-
The feedback is inserted?	Y	N	-	-	-
The user is the director?	-	-	-	Y	Y
All feedback successfully fetched?	-	-	-	Y	N
Actions					
Update feedback to database	X				
Display all feedback				X	
Show error message					X
Brings to main menu interface		X	X		

8. Relations

8.1. User Table

User

PhoneNumber	password

The table 8.1 shows the relation table for user class. The user has its own primary key which is the phone number.

8.2. Customer Table

Customer

PhoneNumber	password	name	email

The table 8.2 shows the relation table for customer class. The customer has its own primary key which is the customer's phone number.

8.3. Director Table

Director

PhoneNumber	password

The table 8.3 shows the relation table for director class. The director has its own primary key which is the director's phone number.

8.4. Cashier Table

Cashier Table

PhoneNumber	CashierID

8.5. Feedback Table

Feedback

name	feedback

The table 8.5 shows the relation table for feedback class. All of the feedback will be recorded from the customer by referencing the customer's name as its primary key.

8.6. Pump Table

Pump Table

PhoneNumber	PumpNumber	CashierID

The table 8.6 shows the relation table for pump class. The user phone number will be recorded by referencing the customer's phone number and pump number is its primary key.

8.7. Payment Table

Payment

PhoneNumber	datePurchase	amount	bankName

The table 8.7 shows the relation table for payment class. All the payment details will be recorded by referencing the customer's phone number and date purchase as the primary key.

8.8. GPS Table

GPS Table

PhoneNumber	location

The table 8.8 shows the relation table for GPS class. Customer's phone number will be recorded by referencing the customer's table and location as its primary key.

8.9. Fuel Table

Fuel

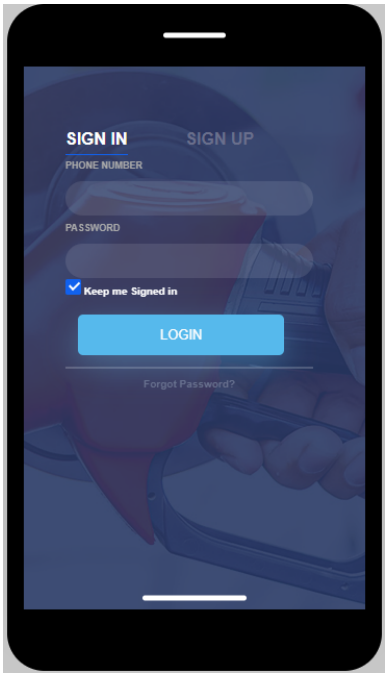
FuelType	PumpNumber

FuelPrice

FuelType	FuelPrice

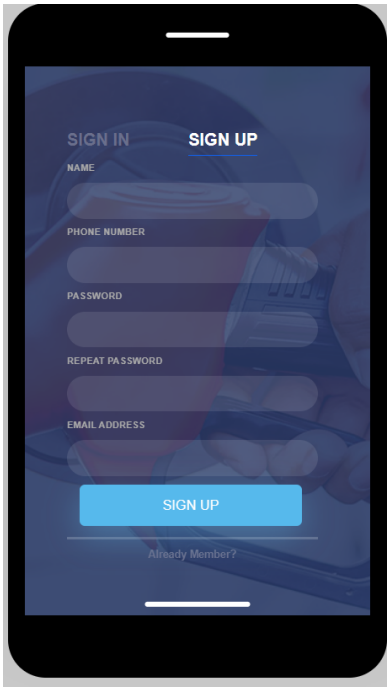
9. User Interface

9.1. Login interface for all user



The figure 9.1 shows the interface when all the users want to log into the system.

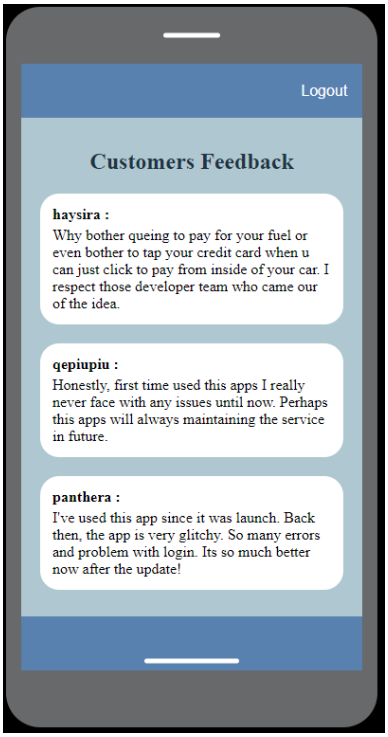
9.2. Register interface for customers



The figure 9.2. shows the interface when the user wishes to register into the system.

9.3. Case 1 : User Director

9.3.1. Director view feedback



The figure 9.2.1 shows the interface when the director logs in to view customer's feedback.

9.4. Case 2 : User Cashier

9.4.1. Cashier main menu

Figure 9.4.1 shows the profile picture of the cashier and if the profile picture is clicked, it will show the personal details of the cashier on duty. It also shows the log out button for the cashier.

9.4.2. Cashier update price

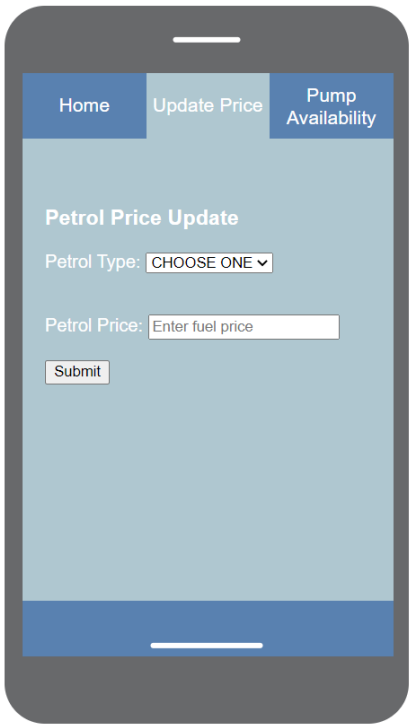
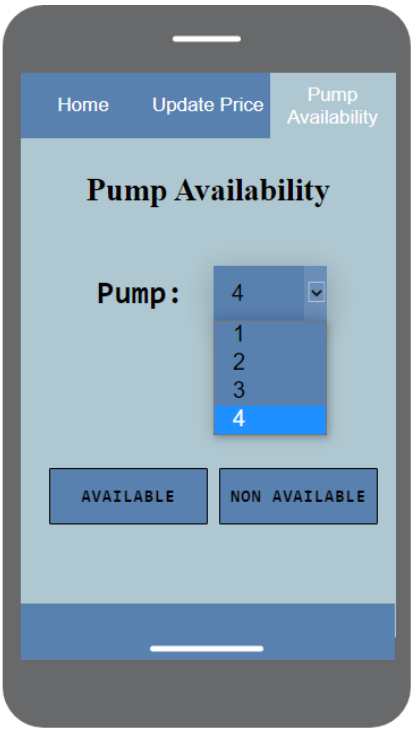


Figure 9.4.2 shows the interface of updating the price of petrol that was done by the cashier.

9.4.3. Cashier change pump availability



The figure 9.3.3 shows the interface when the cashier logs in to pump availability to manage available pump

.

9.5. Case 3 : User Customer

9.5.1. Customer Main Menu

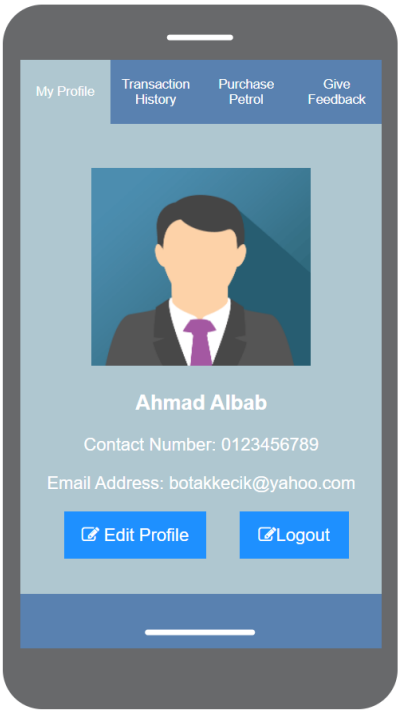


Figure 9.5.1 shows the homepage for the customer that would display a profile picture of the customer and if the customer decides not to insert a picture, the system would display an animated avatar. This interface would also display the name, contact number and the email address of the customer. This interface also contains the button “Edit Profile” which would allow users to edit their personal details. Lastly, “Logout” button would log users out of the account.

9.5.2. Customer Transaction History

The figure 9.4.2 shows the interface when the user logs in to view the customer's transaction history.

9.5.3. Customer Purchase Petrol

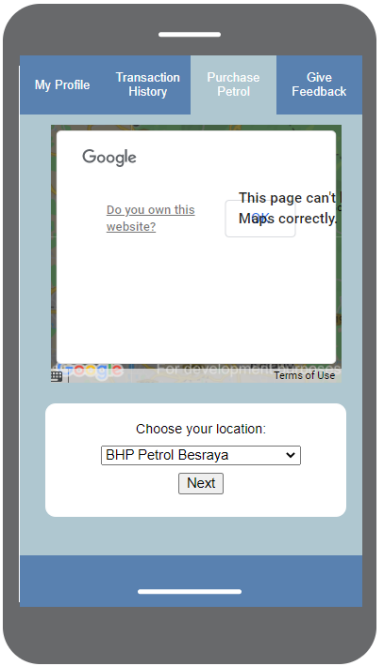


Figure 9.5.3.1 shows the map interface for the customer to choose the nearest location of the petrol station.

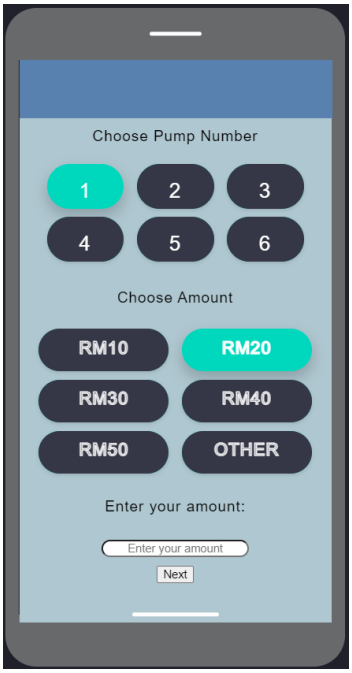


Figure 9.5.3.2 shows the interface for entering petrol detail. The customer needs to pick their pump number and amount required before proceeding to the next page.

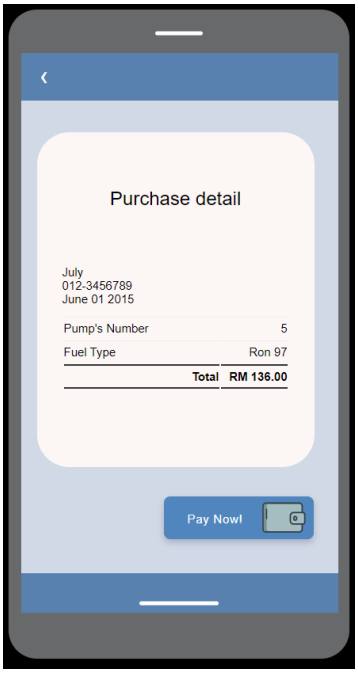


Figure 9.5.3.3 shows the interface of all the payment details. The customer will need to click the “pay” button to confirm the payment.

Figure 9.5.3.4 shows the interface of the bank listed. Customers are required to choose one bank and click proceed. The customer then will be directed to the bank interface for online payment.

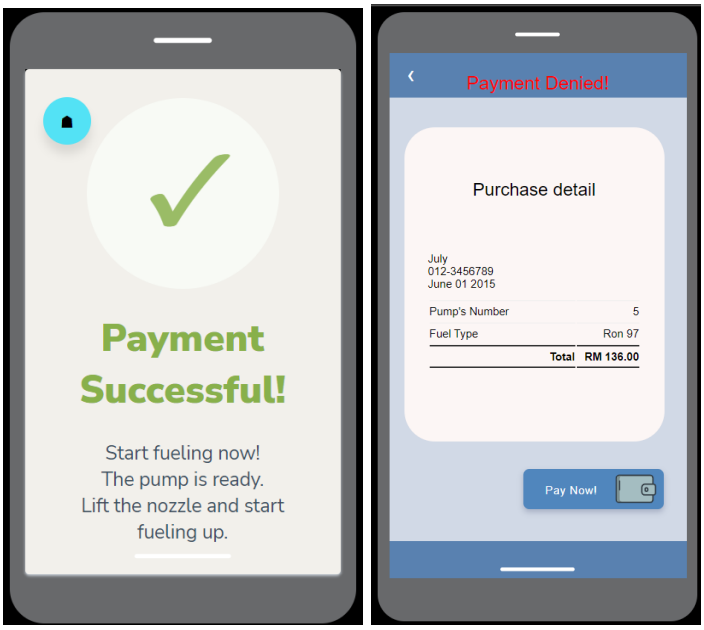
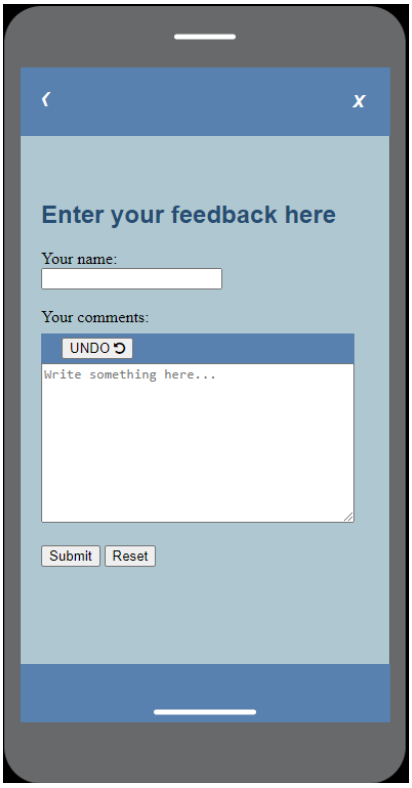


Figure 9.5.3.4 shows the interface after the payment process with the bank is done. The system will display “payment successful” if the bank grants the payment, else the customer will be redirected to the payment details with an error message “Payment denied!”.

9.5.4. Customer give feedback



The figure 9.4.4 shows the interface of customers giving feedback after each successful transaction.

10. CONCLUSION

The main idea of OPSPS is to make a new way of purchasing fuel which is easier and faster. Customers do not need to directly go to the counter to pay, customers can just purchase the fuel from inside the car before starting fueling. Customers who go alone to the petrol station do not need to leave their car which can increase the safety of their belongings. We believe that our project can change the way of purchasing fuel with a cashless method since we are moving with a more advanced technology in the future. Most systems are operating with cashless methods and OPSPS will be the new future of purchasing fuel.

11. REFERENCES