

CS-586: Software System Architecture

Project Deliverable 2

MDA-EFSM Model for the Gas-Pump Components

1. List of Meta Events for the MDA-EFSM:

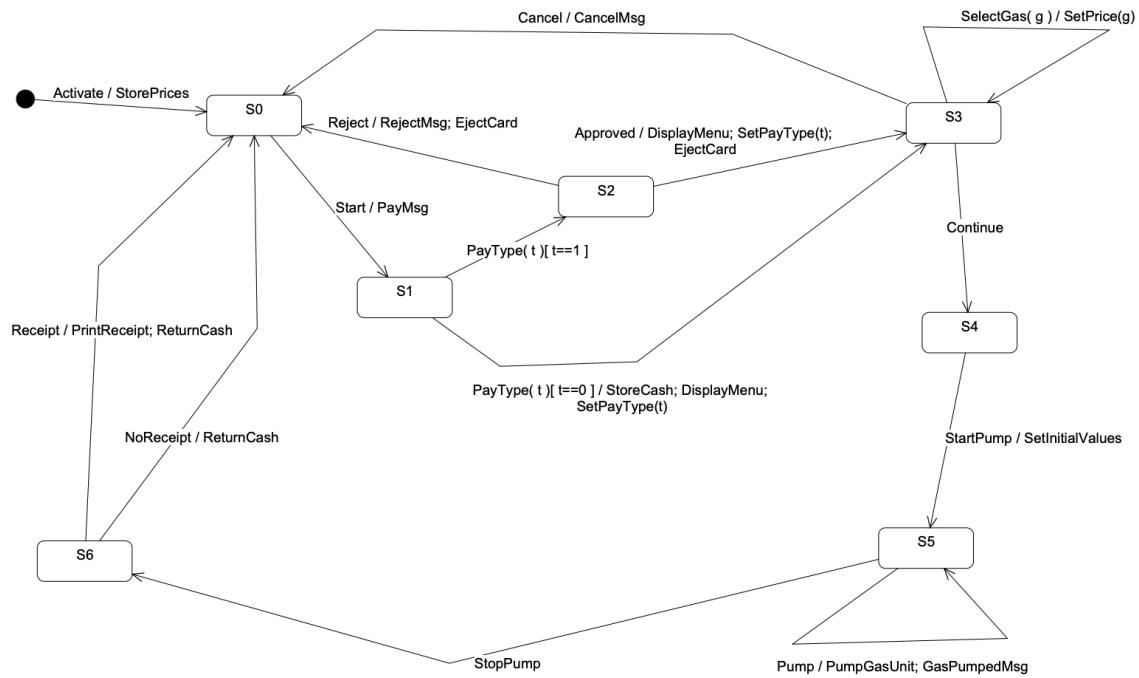
```
Activate()  
Start()  
PayType(int t)      //credit: t=1 ; cash: t=2  
Reject()  
Cancel()  
Approved()  
StartPump()  
Pump()  
StopPump()  
SelectGas(int g)  
Receipt()  
NoReceipt()
```

2. List of Meta Actions for the MDA-EFSM with descriptions

Action	Description
StoreData	Stores price(s) for the gas from the temp data store
PayMsg	Displays a type of payment method
StoreCash	Stores Cash from the temp data store
DisplayMenu	displays a menu with a list of selections
RejectMsg	displays credit card not approved message
SetPrice(int g)	set the price for the gas identified by 'g' identifier
ReadyMsg	displays the ready for pumping message
SetInitialValues	set G(or L) and total to 0
PumpGasUnit	dispose of units of gas and counts # of units disposed
GasPumpedMsg	displays the amount of disposed gas

StopMsg	stop pump message and receipt? Msg(optionally)
PrintReceipt	print a receipt
CancelMsg	displays the cancellation message
ReturnCash	returns the remaining cash

3. State Diagram of the MDA-EFSM for Gas Pumps:



4. Pseudo Code of all Operations of Input Processors of GasPump-1 and GasPump-2

- **GasPump-1 Input Processors**

```
Activate(int a)
{ if (a>0)
    { d->temp_a=a; m->Activate()
    }
}
```

```
Start()
{ m->Start(); }
```

```
PayCash(int c)
{ if (c>0)
    { d->temp_c=c; m->PayType(0)
    }
}
```

```
PayCredit()
{ m->PayType(1); }
```

```
Reject()
{ m->Reject(); }
```

```
Approved()
{ m->Approved(); }
```

```
Cancel()
{ m->Cancel(); }
```

```
StartPump()
{ m->Continue()
  m->StartPump();
}
```

```
Pump()
{ if (d->w==1)
    m->Pump()
  else if
      (d->cash < d->price*(d->L+1))
    { m->StopPump();
      m->Receipt(); }
```

```

else
    m->Pump()
}

StopPump()
{ m->StopPump();
  m->Receipt(); }

```

Notice:

cash: contains the value of cash deposited
 price: contains the price of the gas
 L: contains the number of liters already pumped
 w: pay type flag (cash: w=0; credit: w=1)
 cash, L, price, w: are in the data store
 m: is a pointer to the MDA-EFSM object
 d: is a pointer to the Data Store object

- **GasPump-2 Input Processors**

Activate(float a, float b, float c)

```

{ if ((a>0)&&(b>0)&&(c>0))
    { d->temp_a=a;
      d->temp_b=b;
      d->temp_c=c m->Activate()
    }
}

```

PayCash(int c)

```

{ if (c>0)
    { d->temp_cash=c;
      m->PayType(0)
    }
}

```

Start()

```
{ m->Start(); }
```

Cancel()

```
{ m->Cancel(); }
```

Diesel()

```
{ m->SelectGas(2);
  m->Continue();
}
```

```

Premium()
{ m->SelectGas(3);
  m->Continue();
}

Regular()
{ m->SelectGas(1);
  m->Continue();
}

StartPump()
{ m->StartPump(); }

PumpGallon()
{ if (d->cash < d->price*(d->G+1))
    m->StopPump();
else
    m->Pump()
}

Stop()
{ m->StopPump(); }

Receipt()
{ m->Receipt(); }

NoReceipt()
{ m->NoReceipt(); }

```

Notice:

cash: contains the value of cash deposited

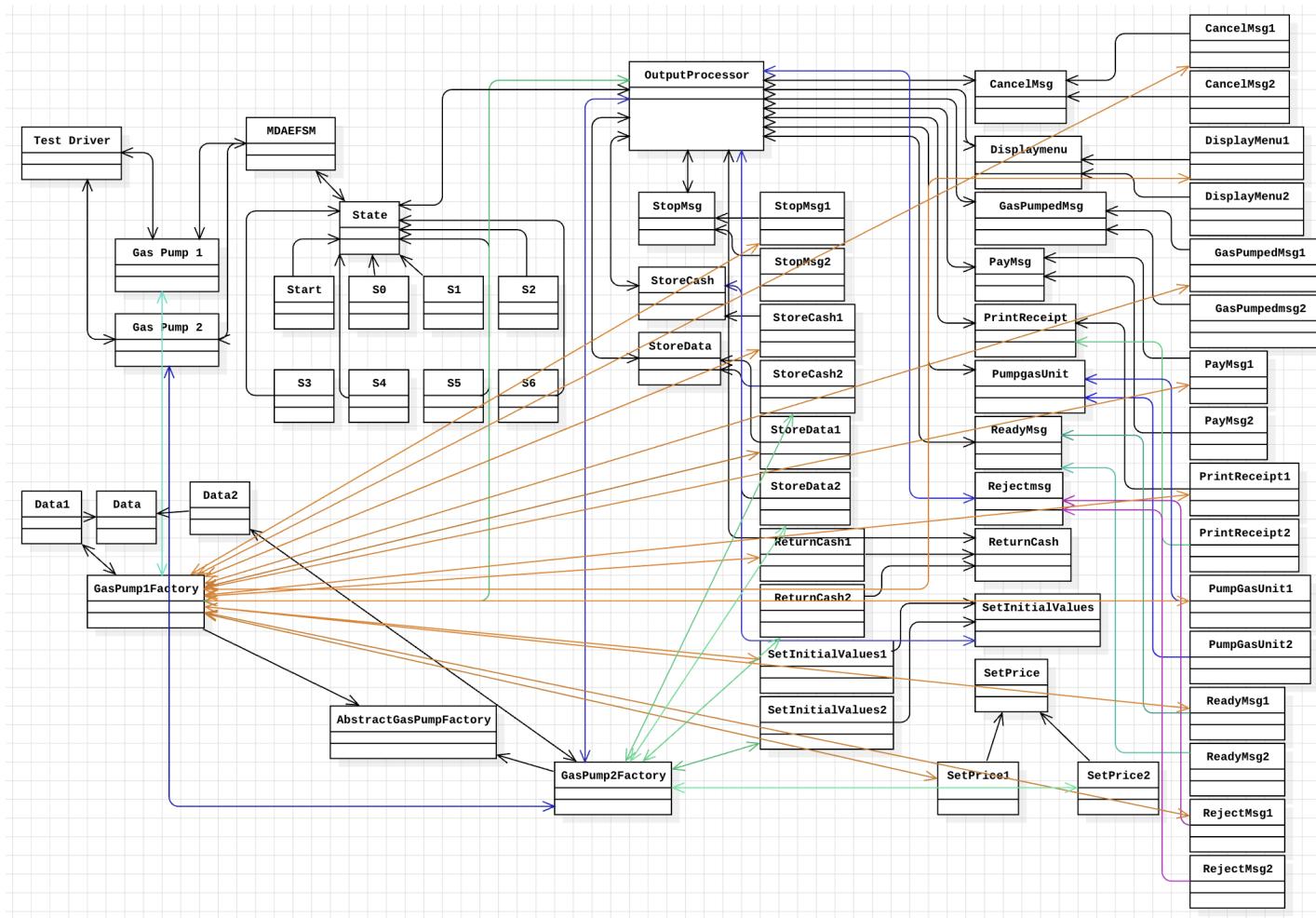
price: contains the price of the selected gas

G: contains the number of Gallons already pumped

cash, G, price are in the data store m: is a pointer to the MDA-EFSM object

d: is a pointer to the Data Store object

2. Class Diagram of the MDA of the Gas-Pump Components

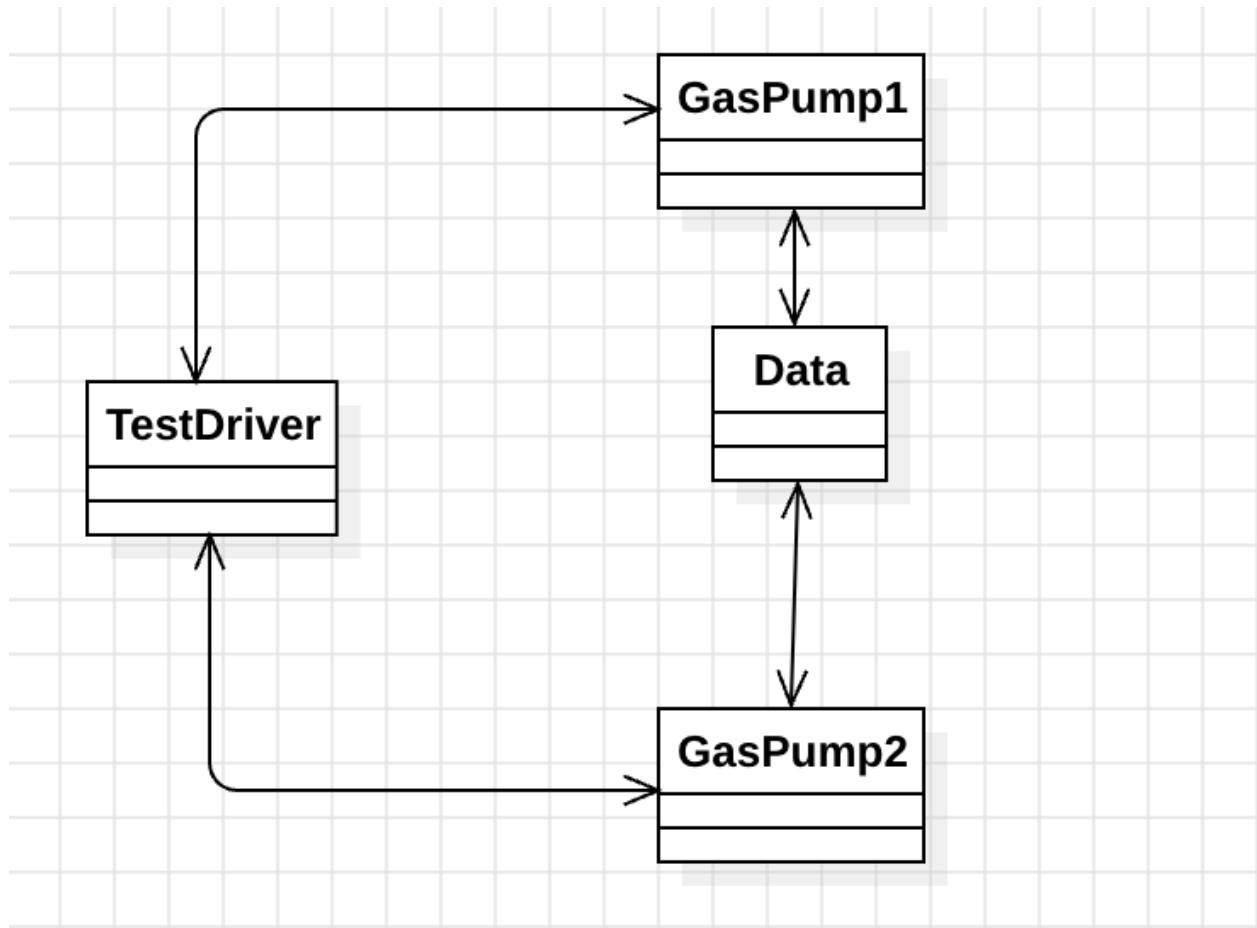


Note: In the above class diagram there will be associations from **GasPump2factory** to all of these classes: **StopMsg2**,
StoreCash2,
RejectMsg2,
PumpGasUnit2,
PrintReceipt2,
PayMsg2,
GasPumpedMsg2,
DisplayMenu2,
CancelMsg2 and **StoreCash2**.

3. Purpose, Description, and Responsibilities of each class diagram

a. Description of class purpose and responsibilities

Input Processor:



TestDriver:

Here User can select either of the Gas-pumps and the gaspump1Factory or gasPump2factory is created along with the objects and it accepts the input data given by the user and passes it to the specific gas pump selected by the user.

GasPump1: Contains Methods related to GasPump1

GasPump2: Contains Methods related to GasPump2

ConcreteFactory:

It returns the objects of strategy classes Data Store of respective GasPump1 and GasPump2.

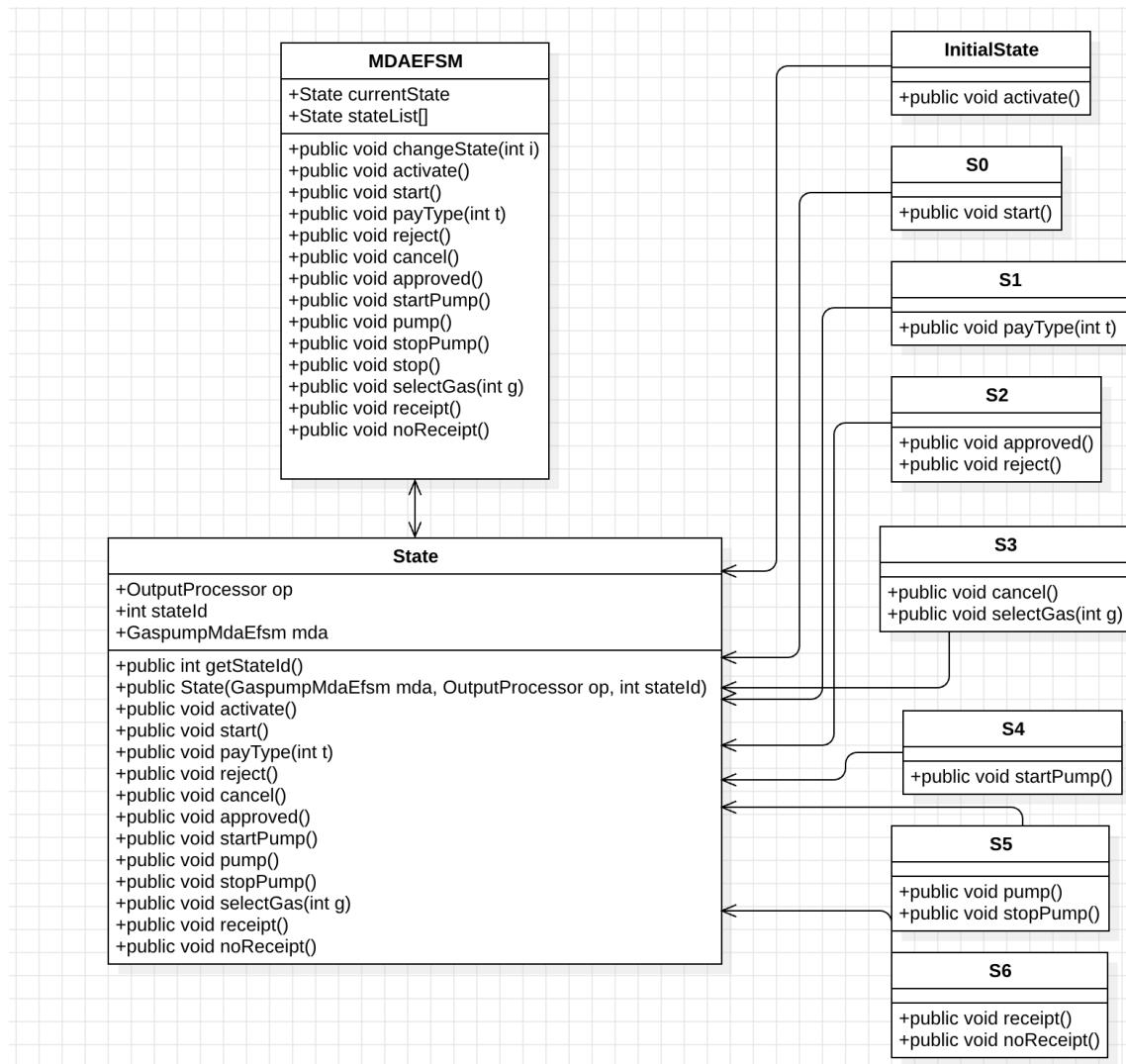
GasPump1Factory	GasPump2Factory
<pre>+public Data getData() +public StoreData getStoreData() +public PayMsg getPayMsg() +public StoreCash getStoreCash() +public DisplayMenu getDisplayMenu() +public RejectMsg getRejectMsg() +public SetPrice getSetPrice() +public ReadyMsg getReadyMsg() +public SetInitialValues getSetInitialValues() +public GasPumpedMsg getGasPumpedMsg() +public StopMsg getStopMsg() +public PrintReceipt getPrintReceipt() +public CancelMsg getCancelMsg() +public ReturnCash getReturnCash()</pre>	<pre>+public Data getData() +public StoreData getStoreData() +public PayMsg getPayMsg() +public StoreCash getStoreCash() +public DisplayMenu getDisplayMenu() +public RejectMsg getRejectMsg() +public SetPrice getSetPrice() +public ReadyMsg getReadyMsg() +public SetInitialValues getSetInitialValues() +public GasPumpedMsg getGasPumpedMsg() +public StopMsg getStopMsg() +public PrintReceipt getPrintReceipt() +public CancelMsg getCancelMsg() +public ReturnCash getReturnCash()</pre>

MDA-EFSM:

For all state changes, this is the main class and it contains a list of states its current state and operations are explained in the source code as same and explained in detail in the below class diagram attached.

List of States:

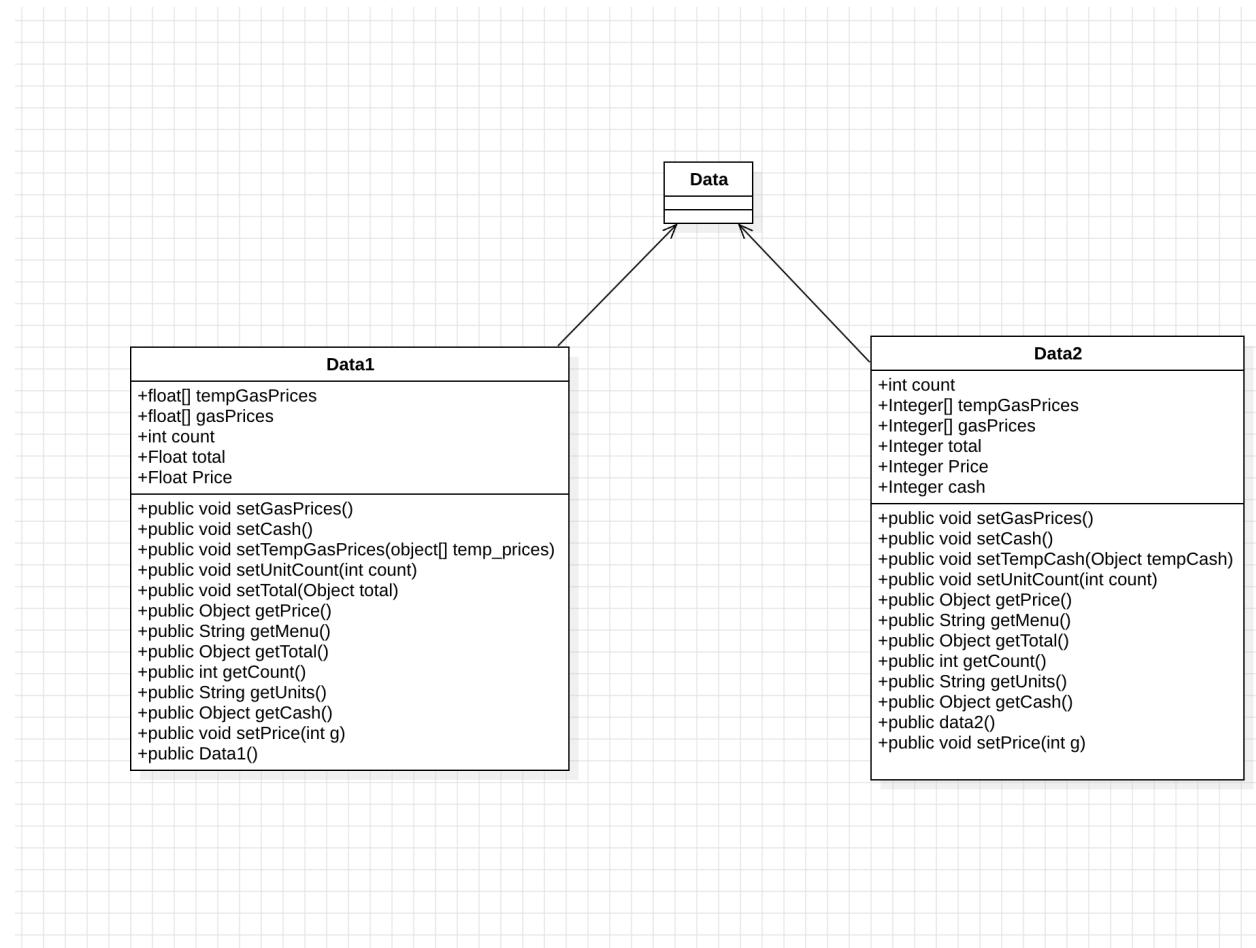
Initial State	-[0]
S0	-[1]
S1	-[2]
S2	-[3]
S3	-[4]
S4	-[5]
S5	-[6]
S6	-[7]



Dataclass:

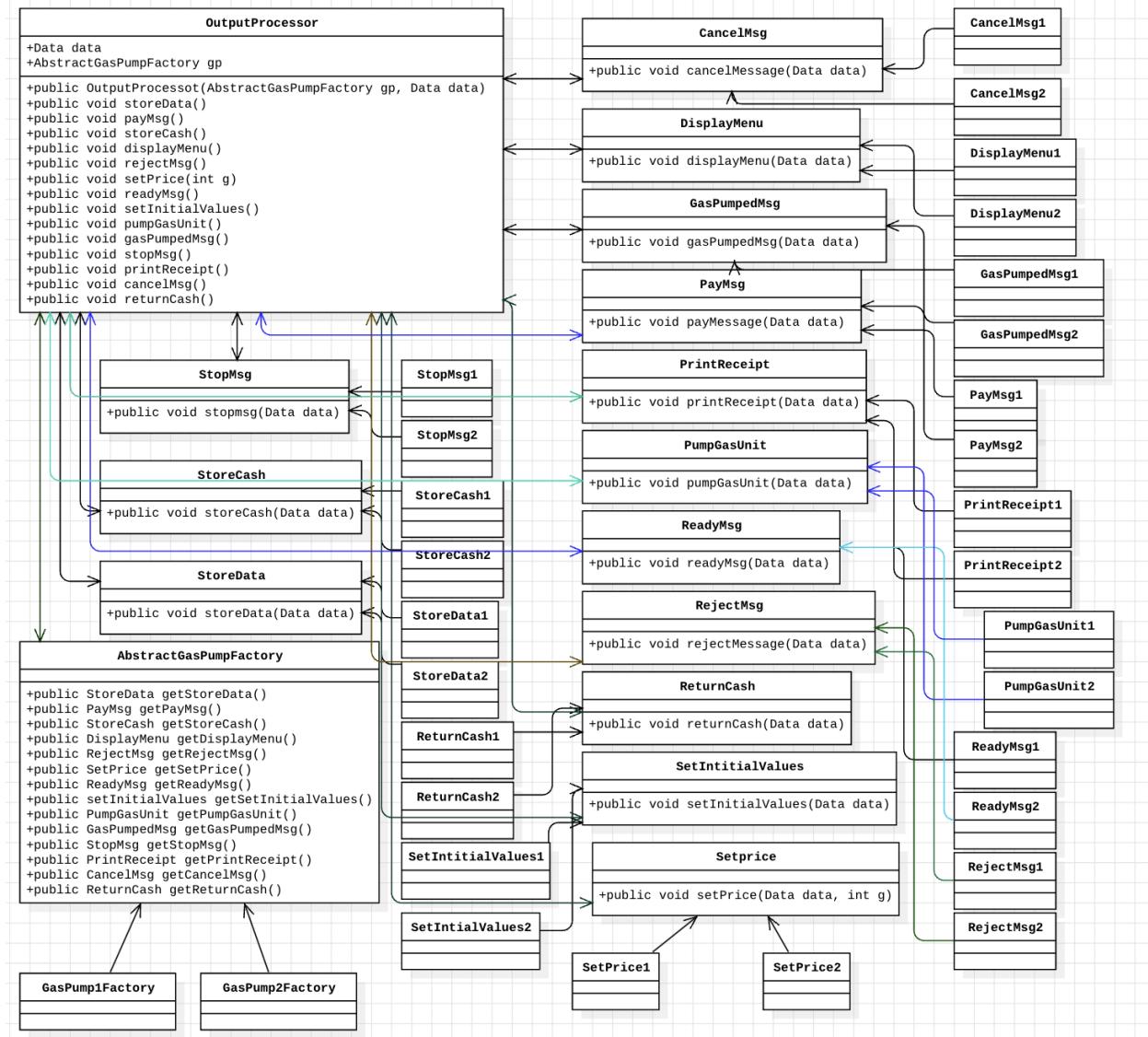
The methods and attributes are listed in the class diagram as below.

Data 1 & Data 2 – This will contain variables related to GasPump1 and GasPump2



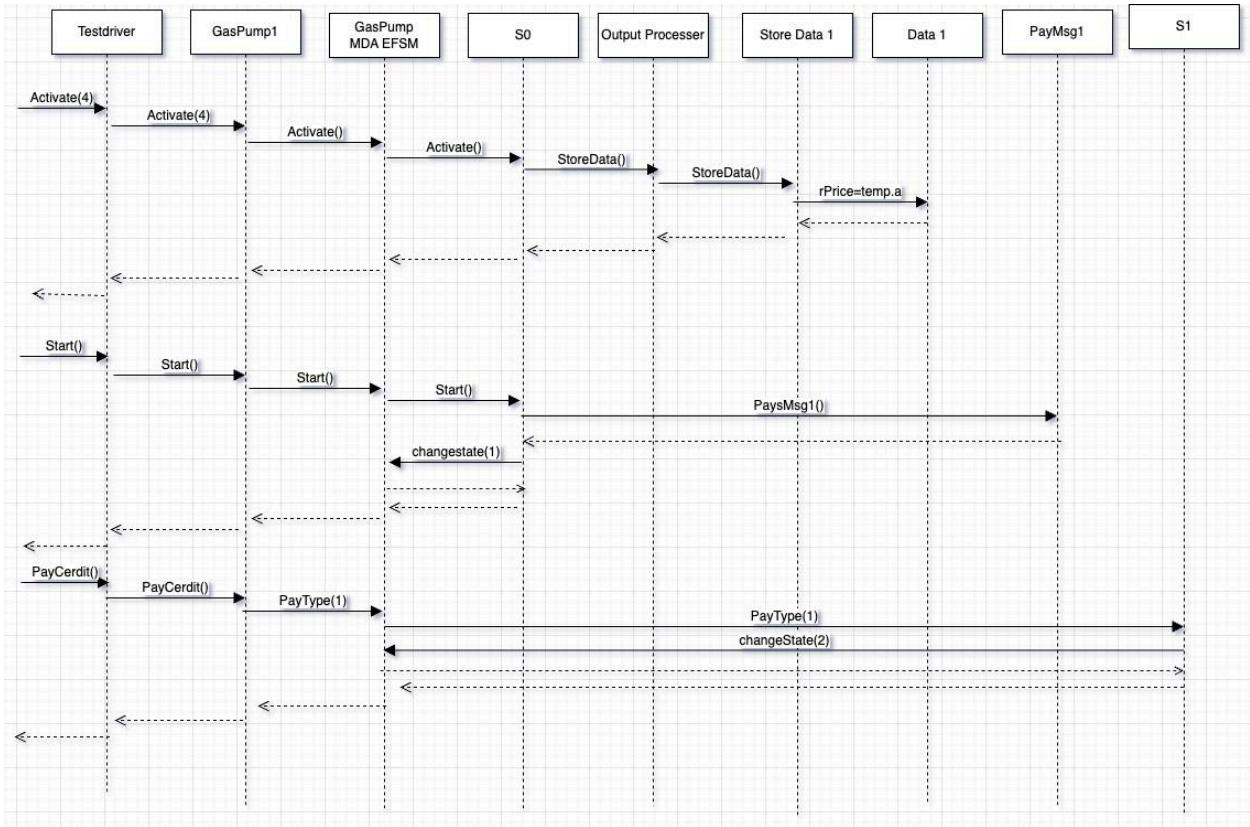
Output Processor:

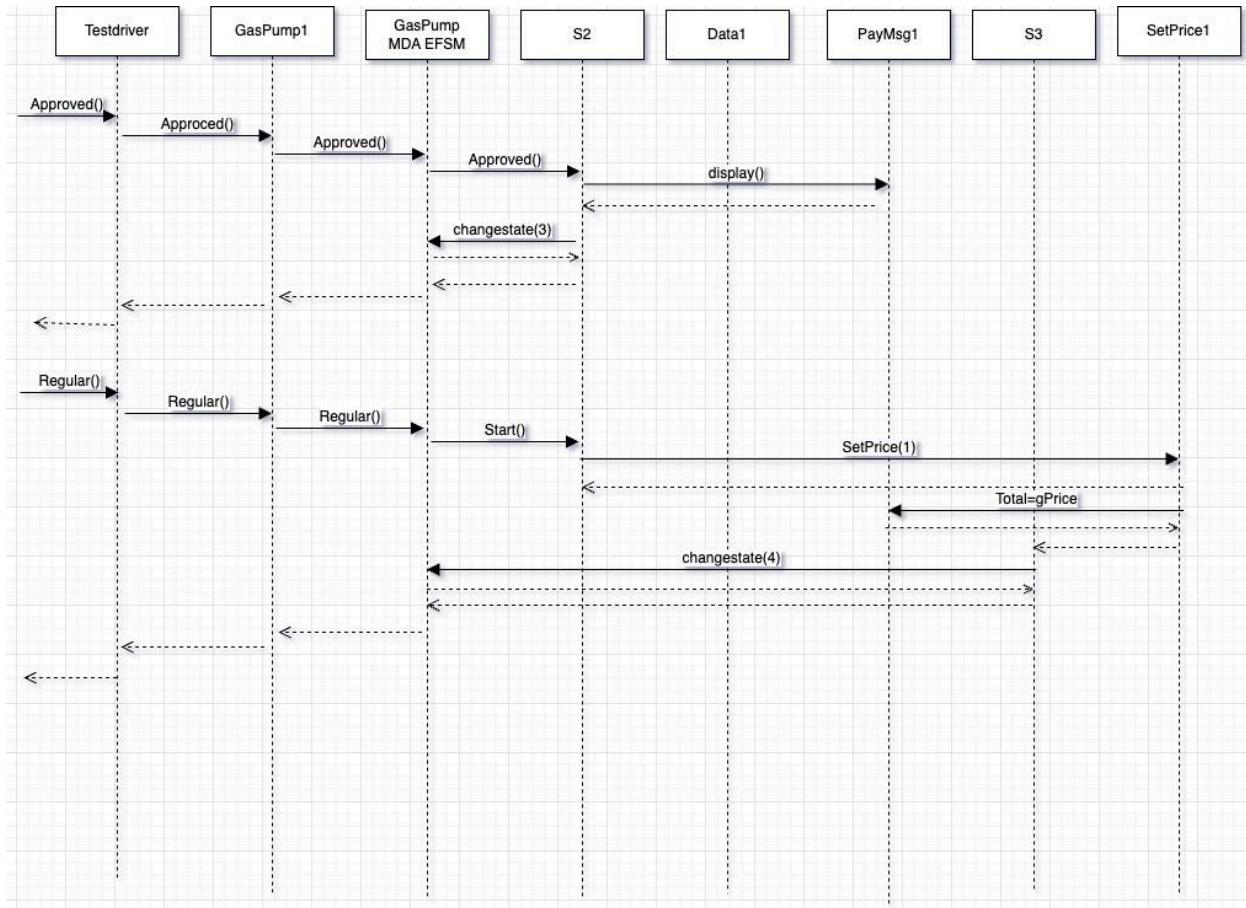
The Operations of OutputProcessor are explained in the source code below is the class diagram with the list of attributes and methods. This class gets the object from the concrete factory(GasPump1Factory, GasPump2factory) and performs the associated operation in the Strategy Pattern classes w.r.t Gaspump1 and gasPump2.

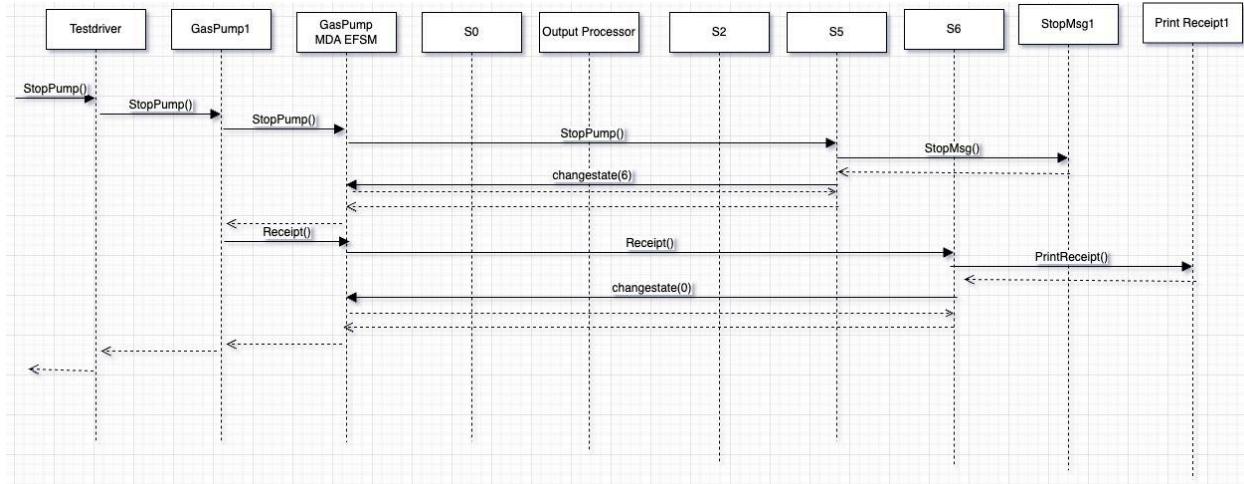
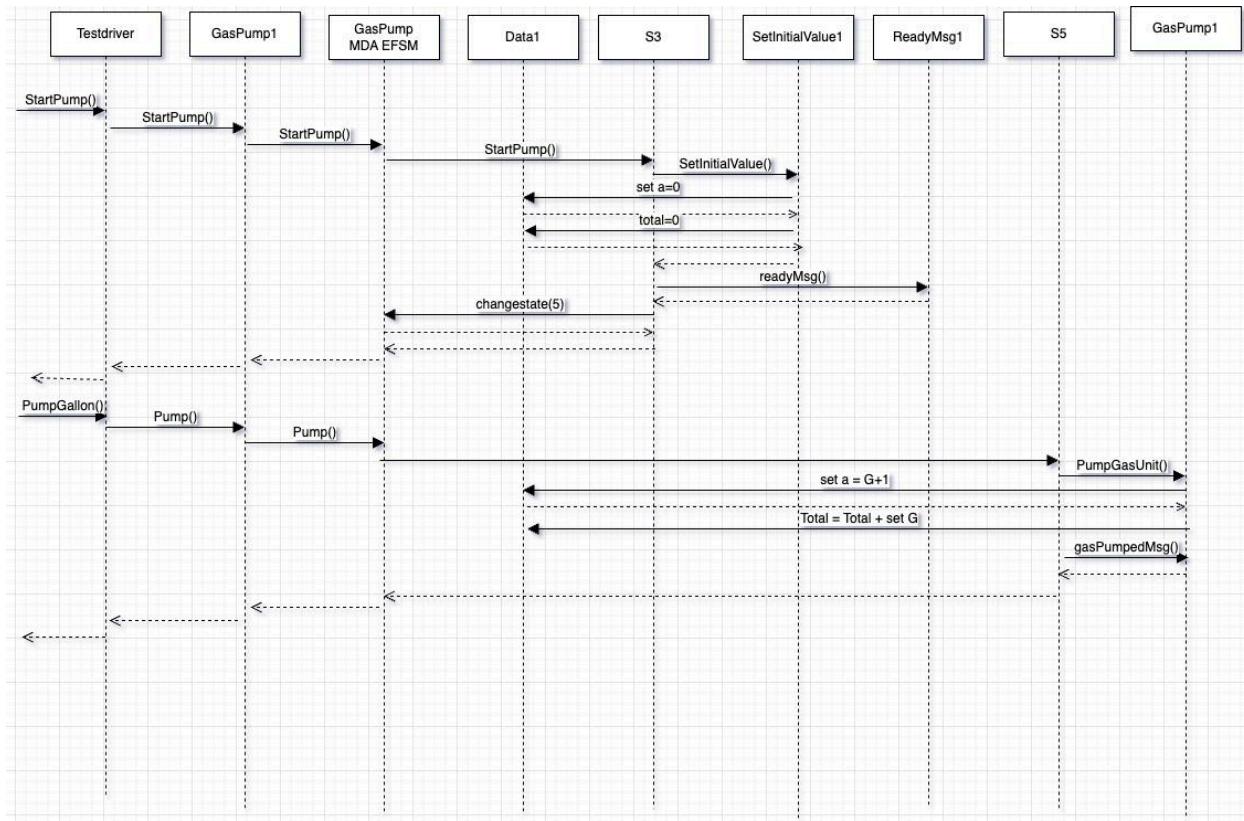


4. Sequence Diagrams:

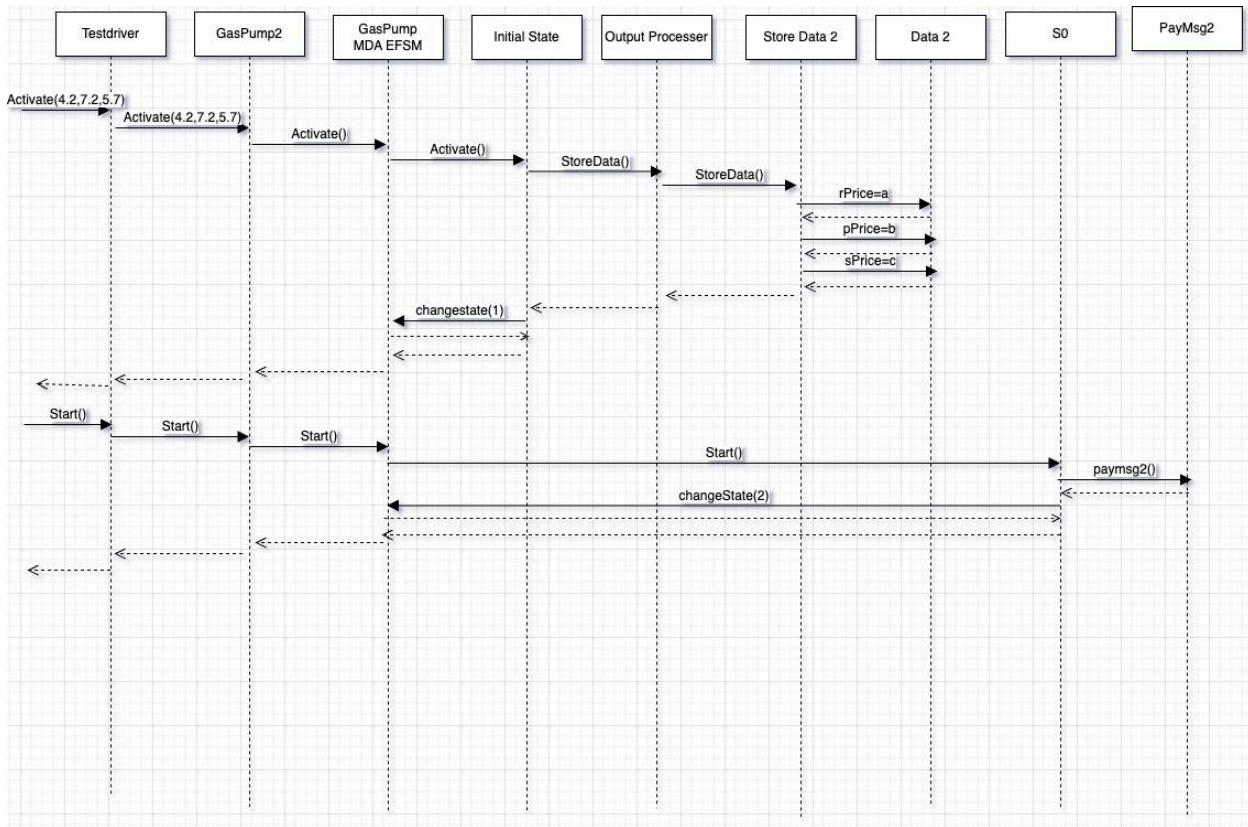
- GasPump1 Sequence Diagram

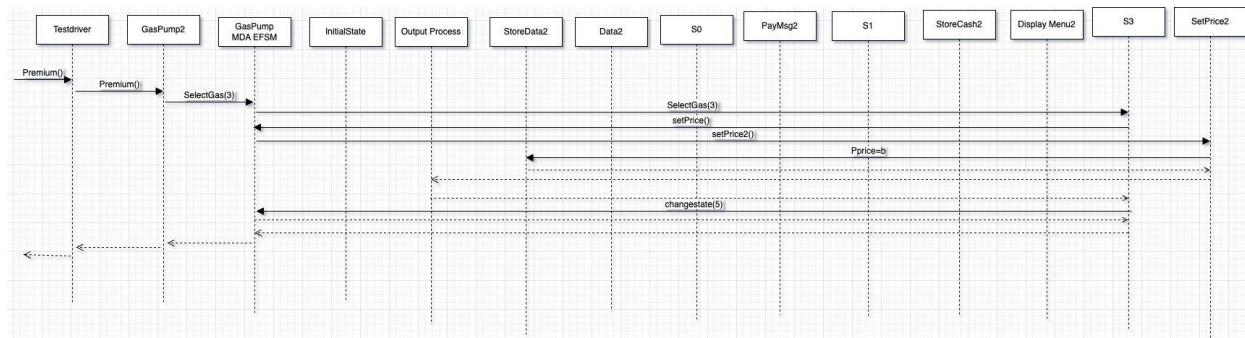
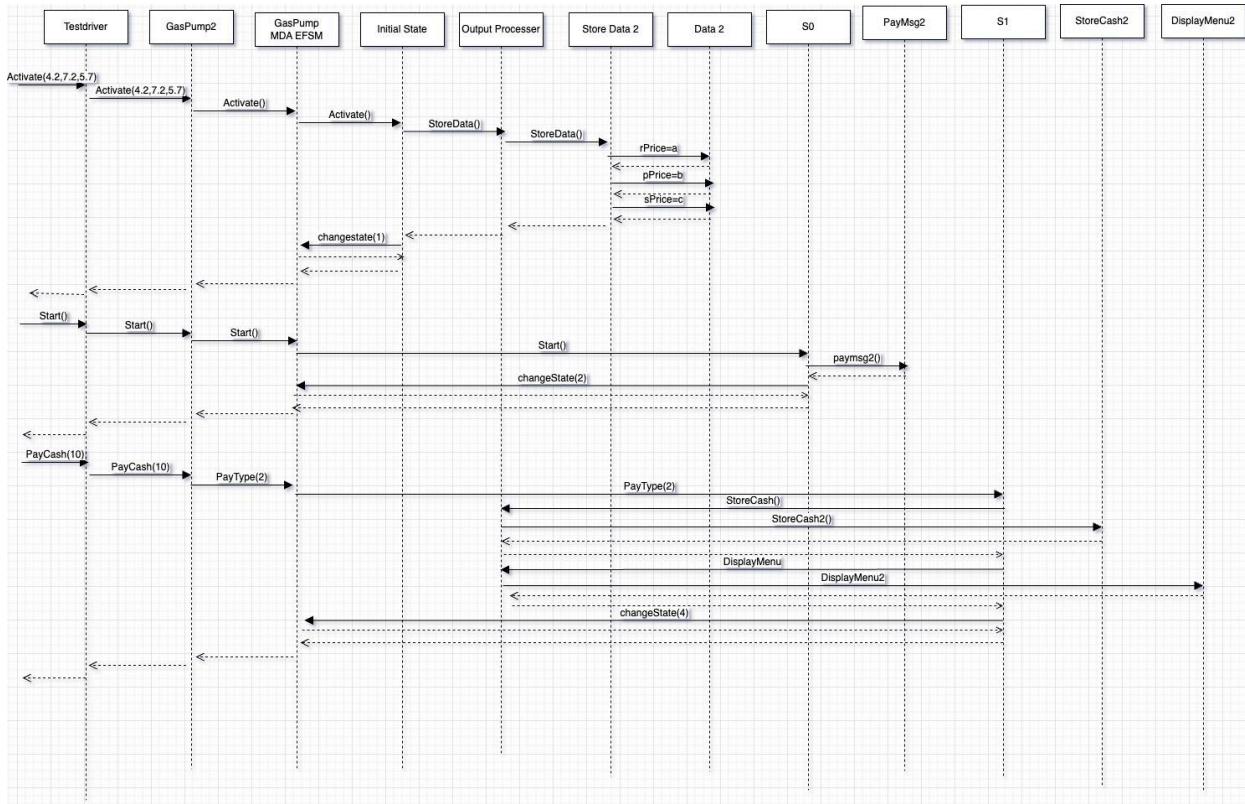


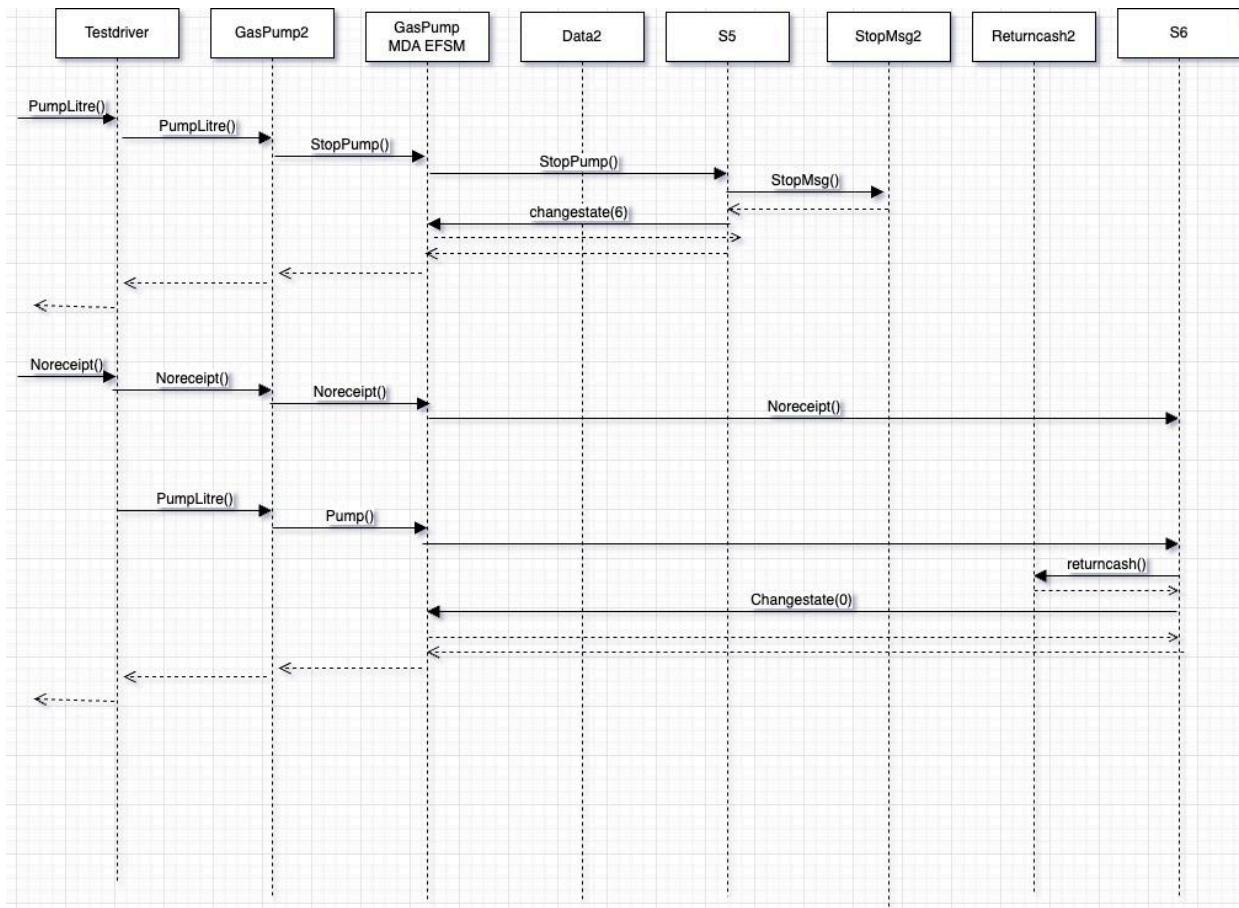
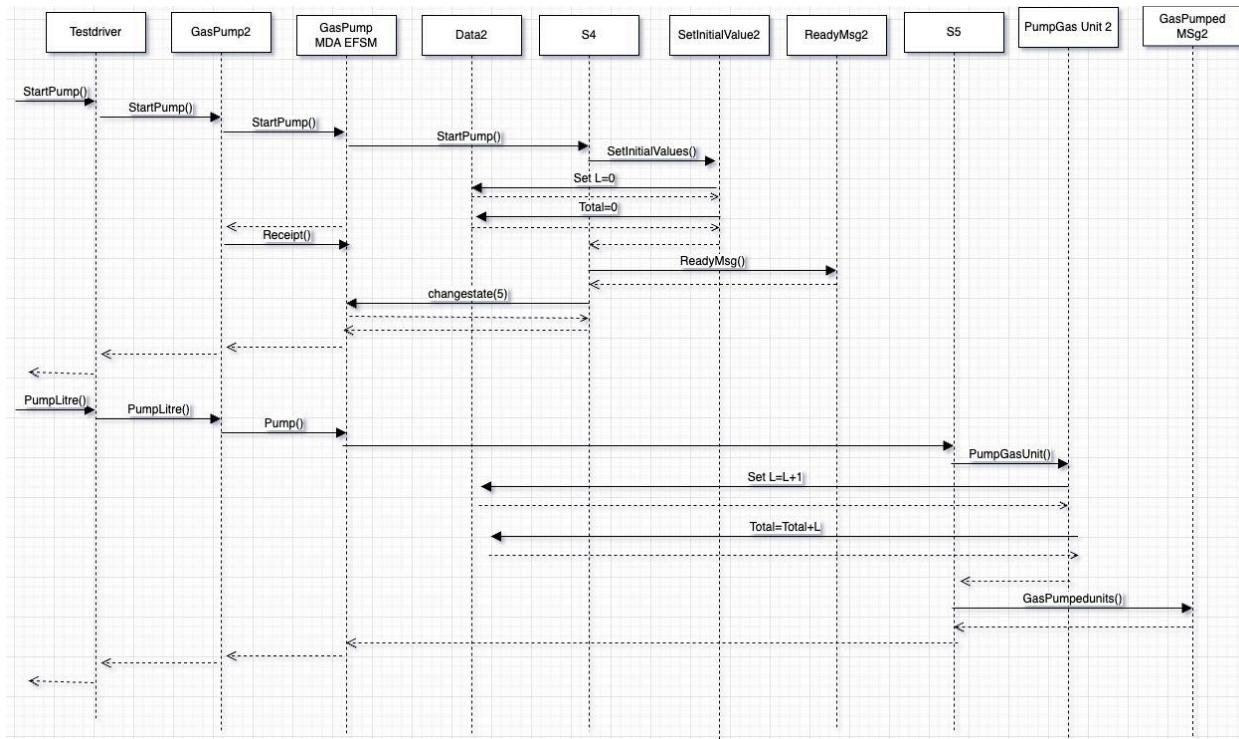




- GasPump2 Sequence Diagram







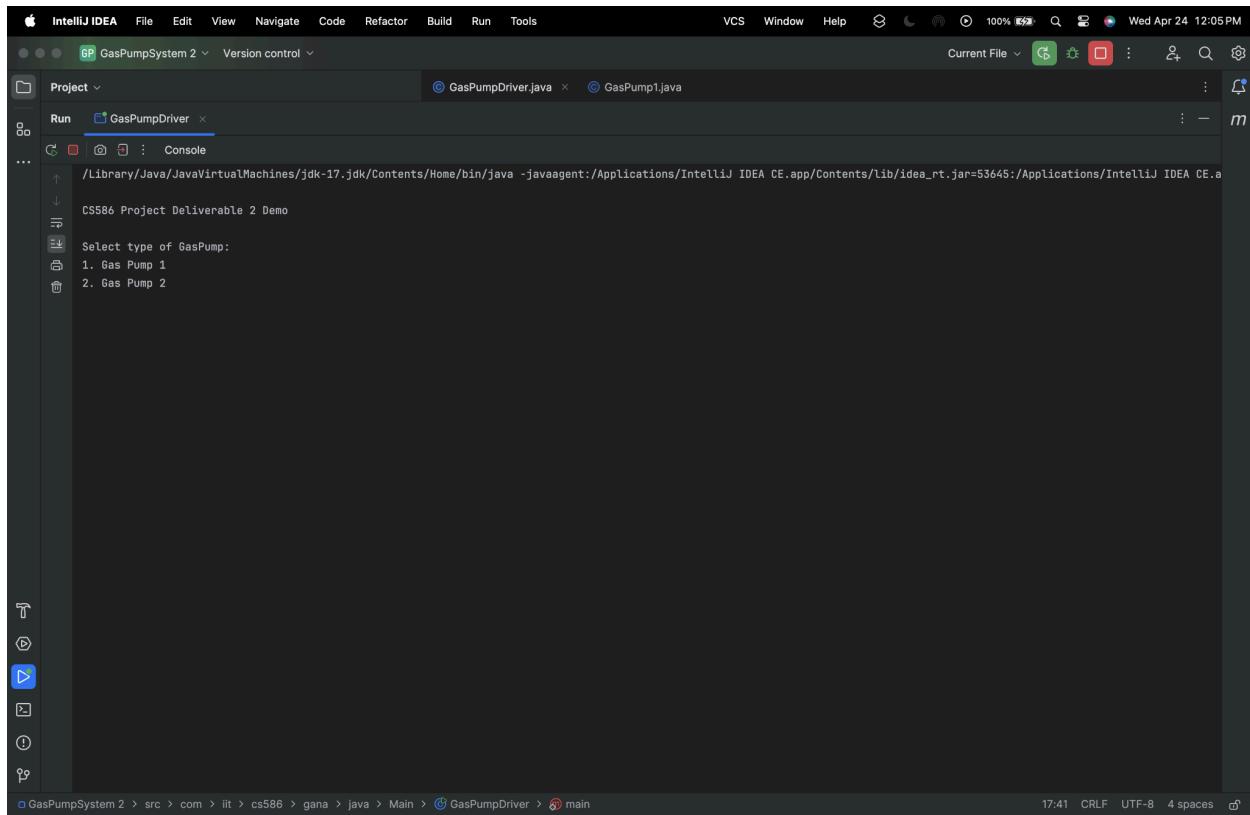
5. Project Executables

Runnable Jar file has been uploaded on the blackboard. Steps to run the jar file is as below:

- 1.Copy the .jar file to your desktop
- 2.Open “Command Prompt” and navigate to your desktop folder
- 3.Run the following command to make the jar file running as expected:

```
java -jar Testdriver.jar
```

Executing for GasPump1-



The screenshot shows the IntelliJ IDEA interface with the following details:

- Top Bar:** IntelliJ IDEA, File, Edit, View, Navigate, Code, Refactor, Build, Run, Tools, VCS, Window, Help, system icons, battery level (100%), and date (Wed Apr 24 12:06 PM).
- Project Bar:** GP GasPumpSystem 2, Version control.
- Toolbars:** Project, Run (GasPumpDriver), Console.
- Console Output:**

```
/Library/Java/JavaVirtualMachines/jdk-17.jdk/Contents/Home/bin/java -javaagent:/Applications/IntelliJ IDEA CE.app/Contents/lib/idea_rt.jar=53645:/Applications/IntelliJ IDEA CE.a
CS586 Project Deliverable 2 Demo

Select type of GasPump:
1. Gas Pump 1
2. Gas Pump 2
1
*****
GP-1 Menu Of Operations
0. Activate(int a)
1. Start
2. PayCredit
3. PayCash(int c)
4. Reject
5. Cancel
6. Approved
7. StartPump
8. Pump
9. StopPump
q. Quit the Program
*****
```
- Sidebar:** Project tree (GasPumpSystem 2), Run configurations (GasPumpDriver), and a list of recent files.
- Status Bar:** GasPumpSystem 2 > src > com > lit > cs586 > gana > java > Main > GasPumpDriver > main.

The screenshot shows the IntelliJ IDEA interface with the following details:

- Top Bar:** IntelliJ IDEA, File, Edit, View, Navigate, Code, Refactor, Build, Run, Tools, VCS, Window, Help.
- Toolbar:** Version control, Current File, and several icons for file operations.
- Project View:** Shows a project named "GP GasPumpSystem 2" with files "GasPumpDriver.java" and "GasPump1.java".
- Run Tab:** Set to "GasPumpDriver".
- Console Tab:** Active, showing the output of the Java application.
- Output in Console:**

```
/Library/Java/JavaVirtualMachines/jdk-17.jdk/Contents/Home/bin/java -javaagent:/Applications/IntelliJ IDEA CE.app/Contents/lib/idea_rt.jar=53645:/Applications/IntelliJ IDEA CE.a
CS586 Project Deliverable 2 Demo

Select type of GasPump:
1. Gas Pump 1
2. Gas Pump 2
1
*****
GP-1 Menu Of Operations
0. Activate(int a)
1. Start
2. PayCredit
3. PayCash(int c)
4. Reject
5. Cancel
6. Approved
7. StartPump
8. Pump
9. StopPump
q. Quit the Program
*****
0
Activate
Enter the price parameter a:
5
```

```
*****  
GP-1 Menu Of Operations  
0. Activate(int a)  
1. Start  
2. PayCredit  
3. PayCash(int c)  
4. Reject  
5. Cancel  
6. Approved  
7. StartPump  
8. Pump  
9. StopPump  
q. Quit the Program  
*****  
0  
Activate  
Enter the price parameter a:  
5  
GasPump1 activated successfully!  
*****  
GP-1 Menu Of Operations  
0. Activate(int a)  
1. Start  
2. PayCredit  
3. PayCash(int c)  
4. Reject  
5. Cancel  
6. Approved  
7. StartPump  
8. Pump  
9. StopPump  
q. Quit the Program  
*****
```

```
GasPump1 activated successfully!
*****
GP-1 Menu Of Operations
0. Activate(int a)
1. Start
2. PayCredit
3. PayCash(int c)
4. Reject
5. Cancel
6. Approved
7. StartPump
8. Pump
9. StopPump
q. Quit the Program
*****
1
Start
Thank you for choosing GasPump-1
Please select payment type
*****
GP-1 Menu Of Operations
0. Activate(int a)
1. Start
2. PayCredit
3. PayCash(int c)
4. Reject
5. Cancel
6. Approved
7. StartPump
8. Pump
9. StopPump
q. Quit the Program
*****
```

The screenshot shows the IntelliJ IDEA interface with the project 'GasPumpSystem 2' open. The 'Run' tool window is active, displaying the output of the 'GasPumpDriver' application. The console output is as follows:

```
Thank you for choosing GasPump-1
Please select payment type
*****
GP-1 Menu Of Operations
0. Activate(int a)
1. Start
2. PayCredit
3. PayCash(int c)
4. Reject
5. Cancel
6. Approved
7. StartPump
8. Pump
9. StopPump
q. Quit the Program
*****
2
PayCredit
PLEASE WAIT -- AUTHENTICATING CREDIT CARD
*****
GP-1 Menu Of Operations
0. Activate(int a)
1. Start
2. PayCredit
3. PayCash(int c)
4. Reject
5. Cancel
6. Approved
7. StartPump
8. Pump
9. StopPump
q. Quit the Program
*****
```

```
PLEASE WAIT -- AUTHENTICATING CREDIT CARD
*****
GP-1 Menu Of Operations
0. Activate(int a)
1. Start
2. PayCredit
3. PayCash(int c)
4. Reject
5. Cancel
6. Approved
7. StartPump
8. Pump
9. StopPump
q. Quit the Program
*****
6
Approve
CREDIT CARD APPROVED
Gasoline [$5/liter]
Otherwise, select (S) to cancel
*****
GP-1 Menu Of Operations
0. Activate(int a)
1. Start
2. PayCredit
3. PayCash(int c)
4. Reject
5. Cancel
6. Approved
7. StartPump
8. Pump
9. StopPump
q. Quit the Program
*****
7
```

The screenshot shows the IntelliJ IDEA interface with the following details:

- Top Bar:** IntelliJ IDEA, File, Edit, View, Navigate, Code, Refactor, Build, Run, Tools, VCS, Window, Help.
- Right Top Corner:** 100%, battery icon, search icon, date: Wed Apr 24 12:07 PM.
- Project Bar:** GP GasPumpSystem 2, Version control.
- Run Tab:** Run, GasPumpDriver.
- Console Tab:** GasPumpDriver.
- Console Output:**

```
GP-1 Menu Of Operations
0. Activate(int a)
1. Start
2. PayCredit
3. PayCash(int c)
4. Reject
5. Cancel
6. Approved
7. StartPump
8. Pump
9. StopPump
q. Quit the Program
*****
7
StartPump
gasoline selected @ price of $5/gallon
Select (7) to start the pump
READY TO DISPENSE FUEL
Select (9) to dispense 1 gallon of gasoline
Otherwise, select (x) to stop
*****
GP-1 Menu Of Operations
0. Activate(int a)
1. Start
2. PayCredit
3. PayCash(int c)
4. Reject
5. Cancel
6. Approved
7. StartPump
8. Pump
9. StopPump
q. Quit the Program
*****
```
- Sidebar:** Project, Run, GasPumpDriver, Console.
- Bottom Status Bar:** GasPumpSystem 2 > src > com > lit > cs586 > gana > java > Main > GasPumpDriver > main.
- Bottom Right:** 17:41, CRLF, UTF-8, 4 spaces.

```
IntelliJ IDEA 2023.2.3 | Ultimate Edition (Community Edition)
Build #IU-232.5710.200, built on April 18, 2023
Runtime version: 17.0.8+0.1-b09-301-7588545 x86_64
VM: OpenJDK 64-Bit Server VM by JetBrains s.r.o
Windows 10 10.0
Java: Java(TM) SE Runtime Environment 17.0.8+0.1-b09-301-7588545
File | Settings | Help | View | Code | Refactor | Build | Run | Tools | VCS | Window | Help | ☰ | 100% | 🔍 | Wed Apr 24 12:07 PM
```

Project GP GasPumpSystem 2 Version control Current File

Run GasPumpDriver

Console

```
Select (Y) to dispense 1 gallon of gasoline
Otherwise, select (X) to stop
*****
GP-1 Menu Of Operations
0. Activate(int a)
1. Start
2. PayCredit
3. PayCash(int c)
4. Reject
5. Cancel
6. Approved
7. StartPump
8. Pump
9. StopPump
q. Quit the Program
*****
8
Pump
Pumped 1 gallon of gasoline
Total # of gallons pumped: 1
*****
GP-1 Menu Of Operations
0. Activate(int a)
1. Start
2. PayCredit
3. PayCash(int c)
4. Reject
5. Cancel
6. Approved
7. StartPump
8. Pump
9. StopPump
q. Quit the Program
*****
```

IntelliJ IDEA File Edit View Navigate Code Refactor Build Run Tools VCS Window Help ⚡ 100% 🔍 Wed Apr 24 12:08 PM

GasPumpSystem 2 Version control

Project Run GasPumpDriver GasPump1.java

Console

```
pumpe 1 gallon of gasoline
Total # of gallons pumped: 1
*****
GP-1 Menu Of Operations
0. Activate(int a)
1. Start
2. PayCredit
3. PayCash(int c)
4. Reject
5. Cancel
6. Approved
7. StartPump
8. Pump
9. StopPump
q. Quit the Program
*****
8
Pump
Pumped 1 gallon of gasoline
Total # of gallons pumped: 2
*****
GP-1 Menu Of Operations
0. Activate(int a)
1. Start
2. PayCredit
3. PayCash(int c)
4. Reject
5. Cancel
6. Approved
7. StartPump
8. Pump
9. StopPump
q. Quit the Program
*****
```

GasPumpSystem 2 > src > com > lit > cs586 > gana > java > Main > GasPumpDriver > main

17:41 CRLF UTF-8 4 spaces ⌂

```
IntelliJ IDEA 2023.2.3 (Community Edition)
Build #IC-232.8510.2000, built on April 10, 2024
Runtime version: Java 17.0.10+10-b09-758-8458442 amd64
VM: OpenJDK 64-Bit Server VM by JetBrains s.r.o
OS: macOS 13.4.1
```

GasPumpSystem 2 > src > com > lit > cs586 > gana > java > Main > GasPumpDriver > main

Current File > GasPumpDriver.java > GasPump1.java

Project Run GasPumpDriver > Console

...
7. StartPump
8. Pump
9. StopPump
q. Quit the Program

9
StopPump
STOPPING PUMP ...
Printing receipt ...

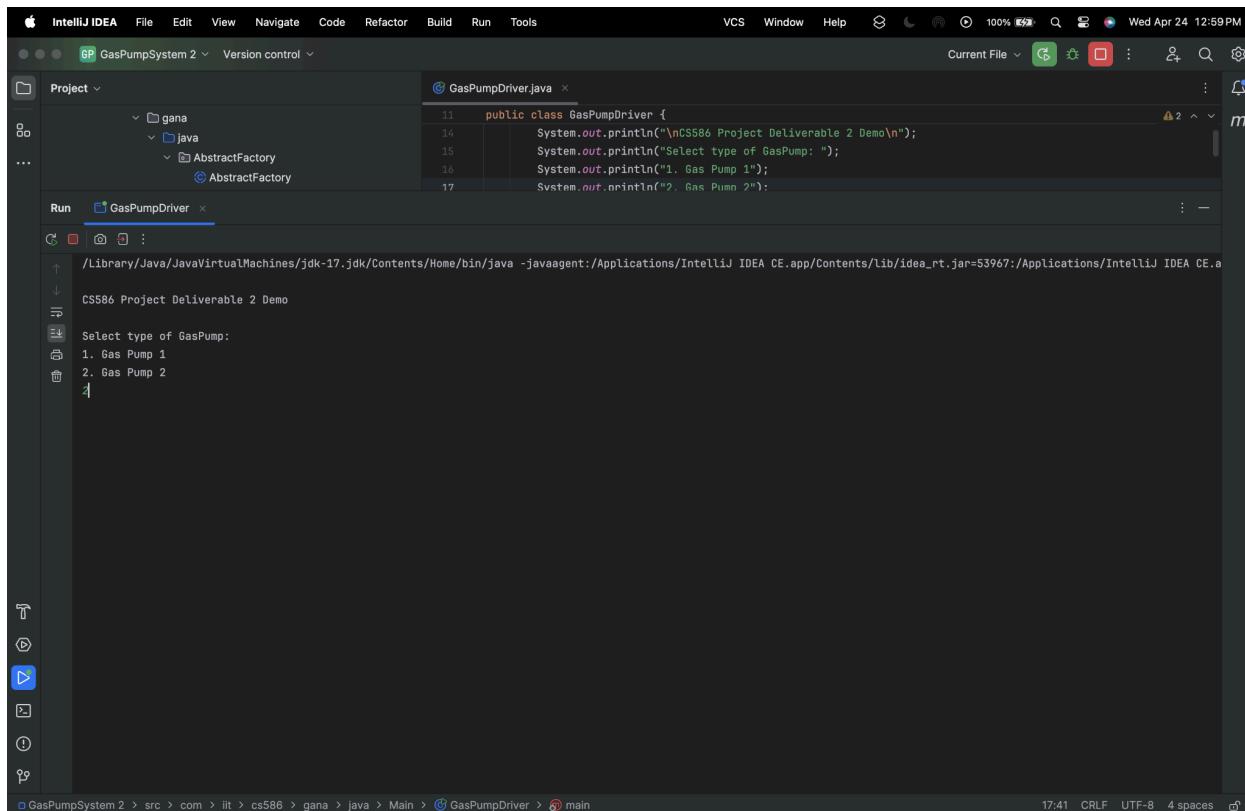
2 Liter of gasoline @ \$5/Liter
Total: \$10

Transaction complete
No cash to return
Transaction finished

GP-1 Menu Of Operations
0. Activate(int a)
1. Start
2. PayCredit
3. PayCash(int c)
4. Reject
5. Cancel
6. Approved
7. StartPump
8. Pump
9. StopPump
q. Quit the Program

q
Quitting ...
Process finished with exit code 0

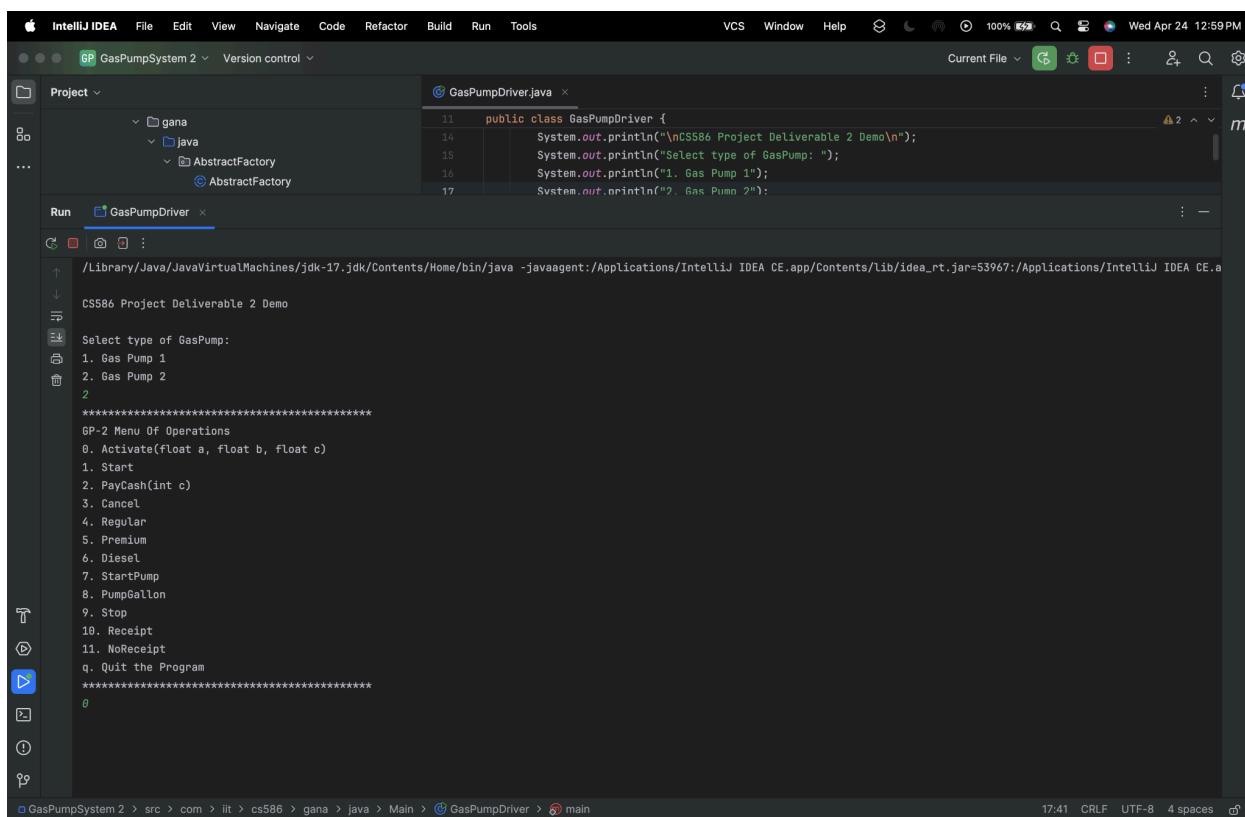
Executing for GasPump2-



```
GasPumpSystem 2 > src > com > iit > cs586 > gana > java > Main > GasPumpDriver > main
```

```
11  public class GasPumpDriver {
12
13      System.out.println("\nCS586 Project Deliverable 2 Demo\n");
14      System.out.println("Select type of GasPump: ");
15      System.out.println("1. Gas Pump 1");
16      System.out.println("2. Gas Pump 2");
17
```

The terminal output shows the program prompting for gas pump selection and listing two options: 1. Gas Pump 1 and 2. Gas Pump 2.



```
GasPumpSystem 2 > src > com > iit > cs586 > gana > java > Main > GasPumpDriver > main
```

```
11  public class GasPumpDriver {
12
13      System.out.println("\nCS586 Project Deliverable 2 Demo\n");
14      System.out.println("Select type of GasPump: ");
15      System.out.println("1. Gas Pump 1");
16      System.out.println("2. Gas Pump 2");
17
```

```
*****
GP-2 Menu Of Operations
0. Activate(float a, float b, float c)
1. Start
2. PayCash(int c)
3. Cancel
4. Regular
5. Premium
6. Diesel
7. StartPump
8. PumpGallon
9. Stop
10. Receipt
11. NoReceipt
q. Quit the Program
*****
```

The terminal output shows the user selecting option 2, which then displays a menu of operations including Activate, Start, PayCash, Cancel, Regular, Premium, Diesel, StartPump, PumpGallon, Stop, Receipt, NoReceipt, and Quit the Program.

```
public class GasPumpDriver {
    public static void main(String[] args) {
        System.out.println("\nNCSS86 Project Deliverable 2 Demo\n");
        System.out.println("Select type of GasPump: ");
        System.out.println("1. Gas Pump 1");
        System.out.println("2. Gas Pump 2");
    }
}
```

Run GasPumpDriver

GasPumpDriver

10. Receipt
11. NoReceipt
q. Quit the Program

0
Activate(float a, float b, float c)
Enter the price parameter a:
4
Enter the price parameter b:
5
Enter the price parameter c:
7
GasPump2 activated successfully!

GP-2 Menu Of Operations
0. Activate(float a, float b, float c)
1. Start
2. PayCash(int c)
3. Cancel
4. Regular
5. Premium
6. Diesel
7. StartPump
8. PumpGallon
9. Stop
10. Receipt
11. NoReceipt
q. Quit the Program

IntelliJ IDEA File Edit View Navigate Code Refactor Build Run Tools VCS Window Help ⚡ 100% 🔍 Wed Apr 24 1:00PM

GasPumpSystem 2 Version control

Project

gas

java

AbstractFactory

AbstractFactory

GasPumpDriver.java

```
11 public class GasPumpDriver {  
12     System.out.println("\nCS586 Project Deliverable 2 Demo\n");  
13     System.out.println("Select type of GasPump: ");  
14     System.out.println("1. Gas Pump 1");  
15     System.out.println("2. Gas Pump 2");  
16  
17 }
```

Run GasPumpDriver

7. StartPump
8. PumpGallon
9. Stop
10. Receipt
11. NoReceipt
q. Quit the Program

1

Start

Thank you for choosing GasPump-2

***** Of Operations

0. Activate(float a, float b, float c)

1. Start

2. PayCash(int c)

3. Cancel

4. Regular

5. Premium

6. Diesel

7. StartPump

8. PumpGallon

9. Stop

10. Receipt

11. NoReceipt

q. Quit the Program

2

PayCash(int c)

Insert cash (enter \$ amount):

1d

GasPumpSystem 2 > src > com > lit > cs586 > gana > java > Main > GasPumpDriver > main

17:41 CRLF UTF-8 4 spaces

IntelliJ IDEA File Edit View Navigate Code Refactor Build Run Tools VCS Window Help ⚡ 100% 🔍 Wed Apr 24 1:00PM

GasPumpSystem 2 Version control

Project

gas

java

AbstractFactory

AbstractFactory

GasPumpDriver.java

```
11 public class GasPumpDriver {  
12     System.out.println("\nCS586 Project Deliverable 2 Demo\n");  
13     System.out.println("Select type of GasPump: ");  
14     System.out.println("1. Gas Pump 1");  
15     System.out.println("2. Gas Pump 2");  
16  
17 }
```

Run GasPumpDriver

11. NoReceipt
q. Quit the Program

2

PayCash(int c)

Insert cash (enter \$ amount):

10

Amount of cash inserted: \$10

Please select gas type:

(4) Regular [\$4.0/Gallon]
(5) Premium [\$5.0/Gallon]
(6) Diesel [\$7.0/Gallon]

Otherwise, select (3) to cancel

***** Of Operations

0. Activate(float a, float b, float c)

1. Start

2. PayCash(int c)

3. Cancel

4. Regular

5. Premium

6. Diesel

7. StartPump

8. PumpGallon

9. Stop

10. Receipt

11. NoReceipt

q. Quit the Program

4

GasPumpSystem 2 > src > com > lit > cs586 > gana > java > Main > GasPumpDriver > main

17:41 CRLF UTF-8 4 spaces

IntelliJ IDEA File Edit View Navigate Code Refactor Build Run Tools VCS Window Help ⚡ 100% 🔍 Wed Apr 24 1:00PM

GasPumpSystem 2 Version control

Project

gana

java

AbstractFactory

AbstractFactory

Run GasPumpDriver

5. Premium
6. Diesel
7. StartPump
8. PumpGallon
9. Stop
10. Receipt
11. NoReceipt
q. Quit the Program

6
Diesel
Diesel gasoline selected @ price of \$7.0/Gallon
Select (7) to start the pump

GP-2 Menu Of Operations
0. Activate(float a, float b, float c)
1. Start
2. PayCash(int c)
3. Cancel
4. Regular
5. Premium
6. Diesel
7. StartPump
8. PumpGallon
9. Stop
10. Receipt
11. NoReceipt
q. Quit the Program

1

GasPumpSystem 2 > src > com > lit > cs586 > gana > java > Main > GasPumpDriver > main

17:41 CRLF UTF-8 4 spaces

IntelliJ IDEA File Edit View Navigate Code Refactor Build Run Tools VCS Window Help ⚡ 100% 🔍 Wed Apr 24 1:00PM

GasPumpSystem 2 Version control

Project

gana

java

AbstractFactory

AbstractFactory

Run GasPumpDriver

6. Diesel
7. StartPump
8. PumpGallon
9. Stop
10. Receipt
11. NoReceipt
q. Quit the Program

7
StartPump
READY TO DISPENSE FUEL
Select (8) to dispense 1 Gallon of Diesel gasoline
Otherwise, select (9) to stop

GP-2 Menu Of Operations
0. Activate(float a, float b, float c)
1. Start
2. PayCash(int c)
3. Cancel
4. Regular
5. Premium
6. Diesel
7. StartPump
8. PumpGallon
9. Stop
10. Receipt
11. NoReceipt
q. Quit the Program

|

GasPumpSystem 2 > src > com > lit > cs586 > gana > java > Main > GasPumpDriver > main

17:41 CRLF UTF-8 4 spaces

IntelliJ IDEA File Edit View Navigate Code Refactor Build Run Tools VCS Window Help ⚡ 100% 🔍 Wed Apr 24 1:01PM

GasPumpSystem 2 Version control

Project

gana

java

AbstractFactory

AbstractFactory

Run GasPumpDriver

5. Premium
6. Diesel
7. StartPump
8. PumpGallon
9. Stop
10. Receipt
11. NoReceipt
q. Quit the Program

8
PumpGallon
Pumped 1 liter of Diesel gasoline
Total # of liters pumped: 1.0

GP-2 Menu Of Operations
0. Activate(float a, float b, float c)
1. Start
2. PayCash(int c)
3. Cancel
4. Regular
5. Premium
6. Diesel
7. StartPump
8. PumpGallon
9. Stop
10. Receipt
11. NoReceipt
q. Quit the Program

17:41 CRLF UTF-8 4 spaces

```
public class GasPumpDriver {  
    public static void main(String[] args) {  
        System.out.println("\nCS586 Project Deliverable 2 Demo\n");  
        System.out.println("Select type of GasPump: ");  
        System.out.println("1. Gas Pump 1");  
        System.out.println("2. Gas Pump 2");  
    }  
}
```

IntelliJ IDEA File Edit View Navigate Code Refactor Build Run Tools VCS Window Help ⚡ 100% 🔍 Wed Apr 24 1:01PM

GasPumpSystem 2 Version control

Project

gana

java

AbstractFactory

AbstractFactory

Run GasPumpDriver

5. Premium
6. Diesel
7. StartPump
8. PumpGallon
9. Stop
10. Receipt
11. NoReceipt
q. Quit the Program

8
PumpGallon
NOT ENOUGH CASH
STOPPING PUMP ...

GP-2 Menu Of Operations
0. Activate(float a, float b, float c)
1. Start
2. PayCash(int c)
3. Cancel
4. Regular
5. Premium
6. Diesel
7. StartPump
8. PumpGallon
9. Stop
10. Receipt
11. NoReceipt
q. Quit the Program

17:41 CRLF UTF-8 4 spaces

```
public class GasPumpDriver {  
    public static void main(String[] args) {  
        System.out.println("\nCS586 Project Deliverable 2 Demo\n");  
        System.out.println("Select type of GasPump: ");  
        System.out.println("1. Gas Pump 1");  
        System.out.println("2. Gas Pump 2");  
    }  
}
```

```
public class GasPumpDriver {
    public static void main(String[] args) {
        System.out.println("\nNCSS86 Project Deliverable 2 Demo\n");
        System.out.println("Select type of GasPump: ");
        System.out.println("1. Gas Pump 1");
        System.out.println("2. Gas Pump 2");
    }
}
```

Run GasPumpDriver

4. Regular
5. Premium
6. Diesel
7. StartPump
8. PumpGallon
9. Stop
10. Receipt
11. NoReceipt
q. Quit the Program

9
Stop
OPERATION NOT ALLOWED IN THIS STATE

GP-2 Menu Of Operations
0. Activate(float a, float b, float c)
1. Start
2. PayCash(int c)
3. Cancel
4. Regular
5. Premium
6. Diesel
7. StartPump
8. PumpGallon
9. Stop
10. Receipt
11. NoReceipt
q. Quit the Program

IntelliJ IDEA File Edit View Navigate Code Refactor Build Run Tools VCS Window Help ⚡ 100% 🔍 Wed Apr 24 1:01PM

GasPumpSystem 2 Version control

Project

gana

java

AbstractFactory

AbstractFactory

GasPumpDriver.java

```
11 public class GasPumpDriver {  
12     System.out.println("\nCS586 Project Deliverable 2 Demo\n");  
13     System.out.println("Select type of GasPump: ");  
14     System.out.println("1. Gas Pump 1");  
15     System.out.println("2. Gas Pump 2");  
16  
17 }
```

Run GasPumpDriver

```
*****  
1. PrintReceipt  
2. Printing receipt ...  
*****  
1.0 Gallon of Diesel gasoline @ $7.0/liter  
Total: $7.0  
Cash inserted: $10  
*****  
Transaction complete  
Cash to return: $3.0  
Returning $3.0  
Transaction finished  
*****  
GP-2 Menu Of Operations  
0. Activate(float a, float b, float c)  
1. Start  
2. PayCash(int c)  
3. Cancel  
4. Regular  
5. Premium  
6. Diesel  
7. StartPump  
8. PumpGallon  
9. Stop  
10. Receipt  
11. NoReceipt  
q. Quit the Program  
*****
```

GasPumpSystem 2 > src > com > lit > cs586 > gana > java > Main > GasPumpDriver > main

IntelliJ IDEA File Edit View Navigate Code Refactor Build Run Tools VCS Window Help ⚡ 100% 🔍 Wed Apr 24 1:01PM

GasPumpSystem 2 Version control

Project

gana

java

AbstractFactory

AbstractFactory

GasPumpDriver.java

```
11 public class GasPumpDriver {  
12     System.out.println("\nCS586 Project Deliverable 2 Demo\n");  
13     System.out.println("Select type of GasPump: ");  
14     System.out.println("1. Gas Pump 1");  
15     System.out.println("2. Gas Pump 2");  
16  
17 }
```

Run GasPumpDriver

```
*****  
1.0 Gallon of Diesel gasoline @ $7.0/liter  
Total: $7.0  
Cash inserted: $10  
*****  
Transaction complete  
Cash to return: $3.0  
Returning $3.0  
Transaction finished  
*****  
GP-2 Menu Of Operations  
0. Activate(float a, float b, float c)  
1. Start  
2. PayCash(int c)  
3. Cancel  
4. Regular  
5. Premium  
6. Diesel  
7. StartPump  
8. PumpGallon  
9. Stop  
10. Receipt  
11. NoReceipt  
q. Quit the Program  
*****  
q  
Quitting ...  
  
Process finished with exit code 0
```

GasPumpSystem 2 > src > com > lit > cs586 > gana > java > Main > GasPumpDriver > main