**MDA-EFSM Model for the Gas-Pump Component**



**General MDA-EFSM Architecture - Class Diagram**



Notice:

DataStore1 is for the GasPump-1

DataStore2 is for the GasPump-2

**List of Events**

|  |
| --- |
| **Events (Meta Events)** |
| Activate() |
| Start() |
| PayType(int t) // credit: t=1; cash: t=2 |
| Reject() |
| Approved() |
| Cancel() |
| SelectGas(int g) |
| StartPump() |
| PumpGas() |
| StopPump() |
| Receipt() |
| NoReceipt() |

**List of Actions**

|  |  |
| --- | --- |
| **Actions (Meta Actions)** | **Responsibility** |
| StoreData() | Stores prices for gas types in data store |
| PayMsg() | Displays type of payment |
| DisplayMenu() | Displays selection menu for types of gas (super, premium etc) |
| RejectMsg() | Displays a reject message if credit card is rejected |
| CancelMsg() | Displays a cancel message if purchase is cancelled |
| StoreCash() | Stores cash in data store |
| StorePrice(int g) | Store price for the selected gas type in data store |
| InitializeValues() | Sets the total units of gas to 0 (e.g: G=0 or L=0) and sets the total amount to 0 (e.g: total=0) |
| ReadyMsg() | Displays ‘ready for pumping’ message |
| StopMsg() | Displays stop pumping message & Receipt or NoRecipt message (optional) |
| PumpGasUnit() | Pumps 1 unit of gas and counts total units of gas pumped |
| DisplayGas() | Displays the total units of gas pumped |
| ReturnCash() | Returns the remaining cash |
| PrintReceipt() | Prints a receipt |

**Gas-Pump 1 Pseudocode**

Notice:

***d*** *is a pointer to the DataStore object*

***m*** *is a pointer to MDA-EFSM object*

***temp\_a****,* ***temp\_b*** *are used to store temporary data in the DataStore*

1. Activate(float a, float b)

if(a>0 && b>0)

d.temp\_a = a;

d.temp\_b = b;

m.Activate()

2. Start()

m.Start()

3. PayCredit()

m.PayType(1)

4. Reject()

m.Reject()

5. Cancel()

m.Cancel()

6. Approved()

m.Approved()

7. Super()

m.SelectGas(2)

8. Regular()

m.SelectGas(1)

9. StartPump()

m.StartPump()

10. PumpGallon()

m.PumpGas()

11. StopPump()

m.StopPump()

m.Receipt()

**Gas-Pump 2 Pseudocode**

Notice:

***d*** *is a pointer to the DataStore object*

***m*** *is a pointer to MDA-EFSM object*

***cash*** *contains the inserted cash in the DataStore*

***price*** *contains the selected gas type’s price in the DataStore*

***L*** *contains the total no. of liters pumped*

***temp\_a****,* ***temp\_b****,* ***temp\_c****,* ***temp\_cash*** *are used to store temporary data in the DataStore*

1. Activate(int a, int b, int c)

if(a>0 && b>0 && c>0)

d.temp\_a = a;

d.temp\_b = b;

d.temp\_c = c;

m.Activate()

2. Start()

m.Start()

3. PayCash(int c)

if(c>0)

d.temp\_cash = c

m.PayType(2)

4. Cancel()

m.Cancel()

5. Premium()

m.SelectGas(3)

6. Regular()

m.SelectGas(1)

7. Super()

m.SelectGas(2)

8. StartPump()

m.StartPump()

9. PumpLiter()

if(d.cash < (d.L+1 \* d.price))

m.StopPump()

else

m.PumpGas()

10. Stop()

m.StopPump()

11. Receipt()

m.Receipt()

12. NoReceipt()

m.NoReceipt()