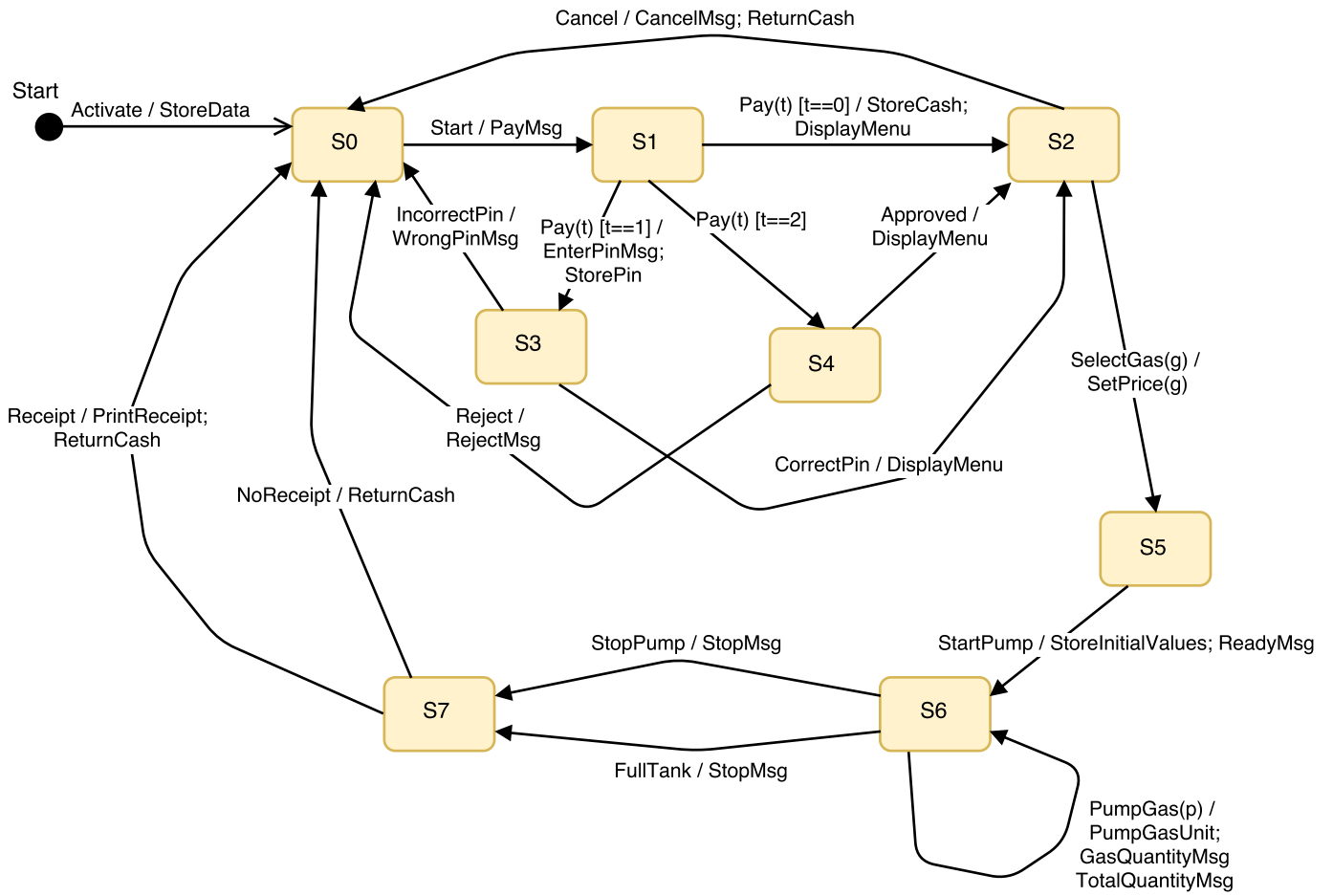


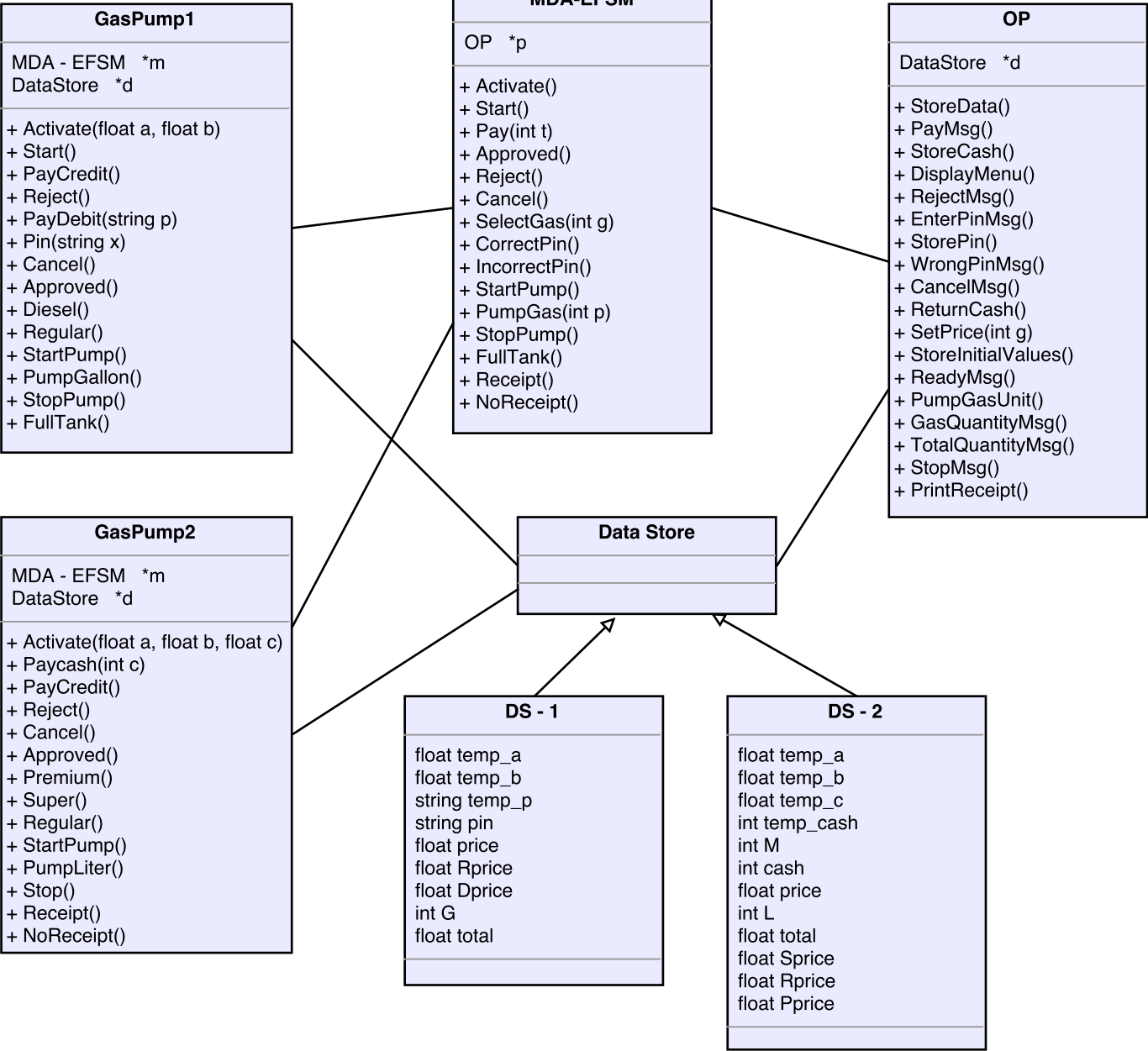
A thick dark gray vertical bar runs down the left side of the page. An orange arrow points to the right from this bar, containing the date. Below the bar, several thin, curved lines in black and gray sweep upwards from the bottom left corner.

03/28/2018

# SOFTWARE SYSTEM ARCHITECTURES PROJECT

Cristina Fernandez Garcia  
CWID: A20410992





## **MDA-EFSM Events:**

Activate()  
Start()  
Pay(int t) // cash: t=0; debit: t=1; credit: t=2;  
Approved()  
Reject()  
Cancel()  
SelectGas(int g) // regular: g=0; super: g=1; diesel: g=2; premium: g=3;  
CorrectPin()  
IncorrectPin()  
StartPump()  
PumpGas(int p) // gallon: p=0; liter: p=1;  
StopPump()  
FullTank()  
Receipt()  
NoReceipt()

## **MDA-EFSM Actions:**

StoreData() // stores prices of gas to data store  
PayMsg() // displays type of payment method  
StoreCash() // stores cash to data store  
DisplayMenu() // displays a menu with different selections  
RejectMsg() // displays credit card not accepted  
EnterPinMsg() // displays message to enter pin number  
StorePin() // stores pin to data store  
WrongPinMsg() // displays message of wrong pin number  
CancelMsg() // displays message of cancellation  
ReturnCash() // return the remaining cash  
SetPrice(int g) // sets the price of the selected gas 'g'  
StoreInitialValues() // stores the total and G or L to 0  
ReadyMsg() // displays a message that the system is ready  
PumpGasUnit() // disposes a unit of gas and add a unit to the count  
GasQuantityMsg() // displays the amount of gas disposed  
TotalQuantityMsg() // displays the total price  
StopMsg() // displays a message that the pump has stopped  
PrintReceipt() // prints the receipt of the purchase

## Operations of the Input Processor GasPump1:

```
Activate (float a, float b) {  
    if ((a>0)&&(b>0)) then  
        d.temp_a = a  
        d.temp_b = b  
        m.Activate()  
}
```

```
Start(){  
    m.Start()  
}
```

```
PayCredit(){  
    m.Pay(2)  
}
```

```
Reject(){  
    m.Reject()  
}
```

```
Approved(){  
    m.Approved()  
}
```

```
Cancel(){  
    m.Cancel ()  
}
```

```
PayDebit(string p){  
    d.temp_p = p  
    m.Pay(1)  
}
```

```
Diesel(){  
    m.SelectGas(2)  
}
```

```
Regular(){  
    m.SelectGas(0)  
}
```

```
Pin(string x){  
    if(d.pin == x)  
        mCorrectPin()  
    else  
        m.IncorrectPin()  
}
```

```
StartPump(){  
    if(d.price>0) then  
        m.StartPump()  
}
```

```
PumpGallon(){  
    m.PumpGas(0)  
}
```

```
StopPump(){  
    m.StopPump()  
    m.Receipt()  
}
```

```
FullTank(){  
    m.FullTank()  
    m.Receipt()  
}
```

Notice:

m: pointer to the MDA-EFSM

d: pointer to the Data Store

In the Data Store:

pin: contains the correct pin number

price: contains the price per gas unit

## Operations of the Input Processor GasPump2:

```
Activate(float a, float b, float c) {  
    if ((a>0)&&(b>0)&&(c>0)) then  
        d.temp_a = a  
        d.temp_b = b  
        d.temp_c = c  
        m.Activate()  
        m.Start()  
}  
  
PayCredit(){  
    m.Pay(2)  
}  
  
Reject(){  
    m.Reject()  
}  
  
Approved(){  
    m.Approved()  
}  
  
Cancel(){  
    m.Cancel ()  
}
```

```
PayCash(int c){  
    d.temp_cash = c  
    if (d.temp_cash>0)  
        m.Pay(0)  
}  
  
Premium(){  
    m.SelectGas(3)  
}  
  
Super(){  
    m.SelectGas(1)  
}  
  
Regular(){  
    m.SelectGas(0)  
}  
  
StartPump(){  
    m.StartPump()  
}  
  
Stop(){  
    m.StopPump()  
}
```

```
PumpLiter(){  
    if ((d.M==1) || ((d.M==0)&&(d.cash>=d.price*(d.L+1))))  
        m.PumpGas(1)  
    else if ((d.M==0)&&(d.cash<d.price*(d.L+1)))  
        m.StopPump()  
}  
  
Receipt(){  
    m.Receipt()  
}  
  
NoReceipt(){  
    m.NoReceipt()  
}  
  
Notice:  
m: pointer to the MDA-EFSM  
d: pointer to the Data Store  
  
In the Data Store:  
M: contains 0 or 1 depending on the payment method  
cash: contains the cash introduced to pay  
price: contains the price per gas unit  
L: contains the number of liters disposed
```