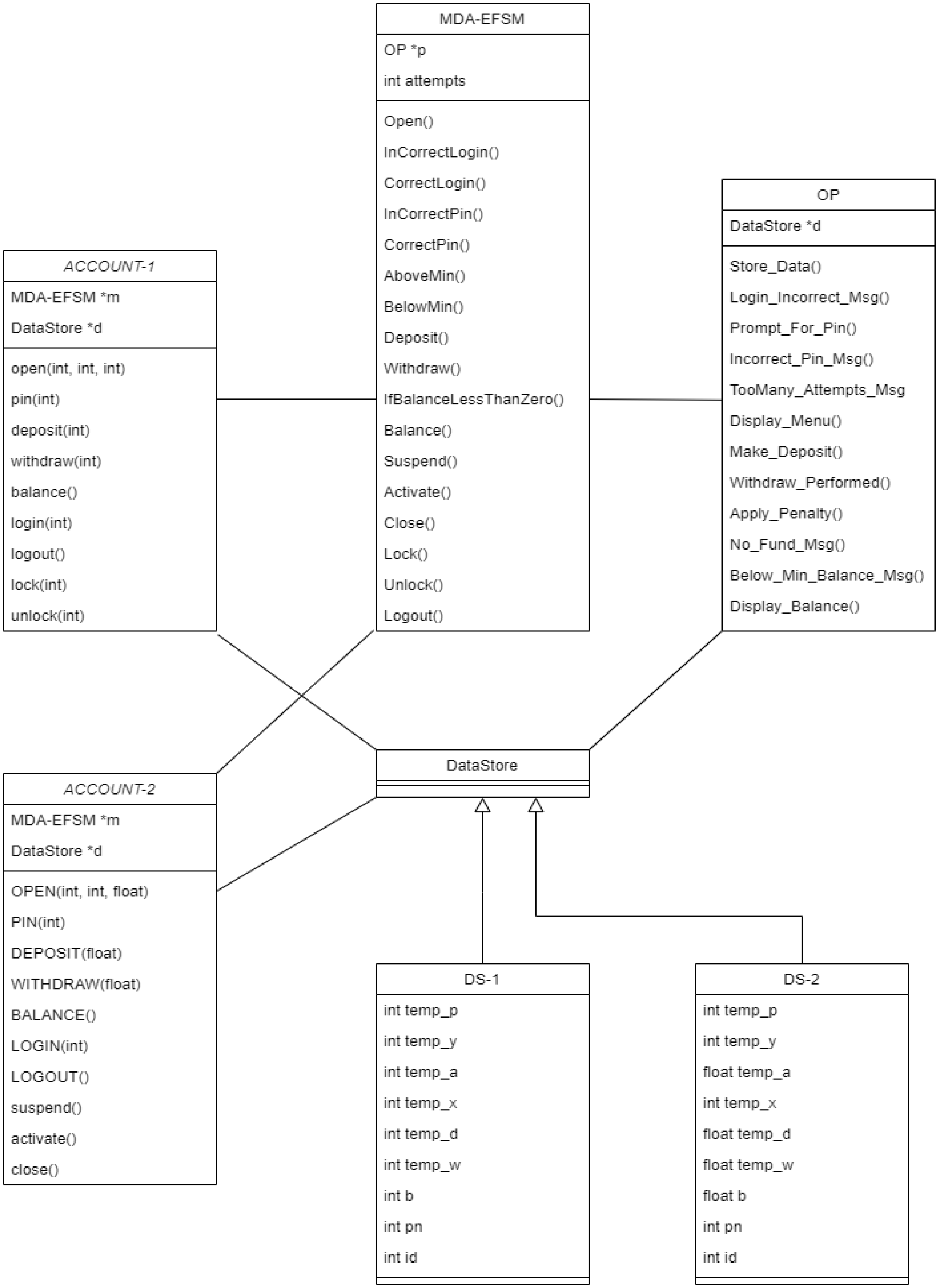
CS 586 SOFTWARE SYSTEM ARCHITECTURE

Rutul Mehta (A20476293)

Table of Contents

1. Class Diagram



2. A List of meta events for MDA-EFSM

- Open()
- InCorrectLogin()
- CorrectLogin()
- InCorrectPin()
- CorrectPin()
- AboveMin()
- BelowMin()
- Deposit()
- Withdraw()
- IfBalanceLessThanZero()
- Balance()
- Suspend()
- Activate()
- Close()
- Lock()
- Unlock()
- Logout()

3. A List of meta actions for the MDA-EFSM

- Store_Data()
  Store Pin, balance, and client identification from temporary data store to pn, b, and id respectively

- Login_Incorrect_Msg()
  Display Incorrect Login Message

- Prompt_For_Pin()
  Prompt to enter pin.

- InCorrect_Pin_Msg()
  Display Incorrect Pin Message

- TooMany_Attempts_Msg()
  Display too many attempts message

- Display_Menu()
  Display menu

- Make_Deposit()
  Make deposit

- Withdraw_Performed()
  Make withdraw

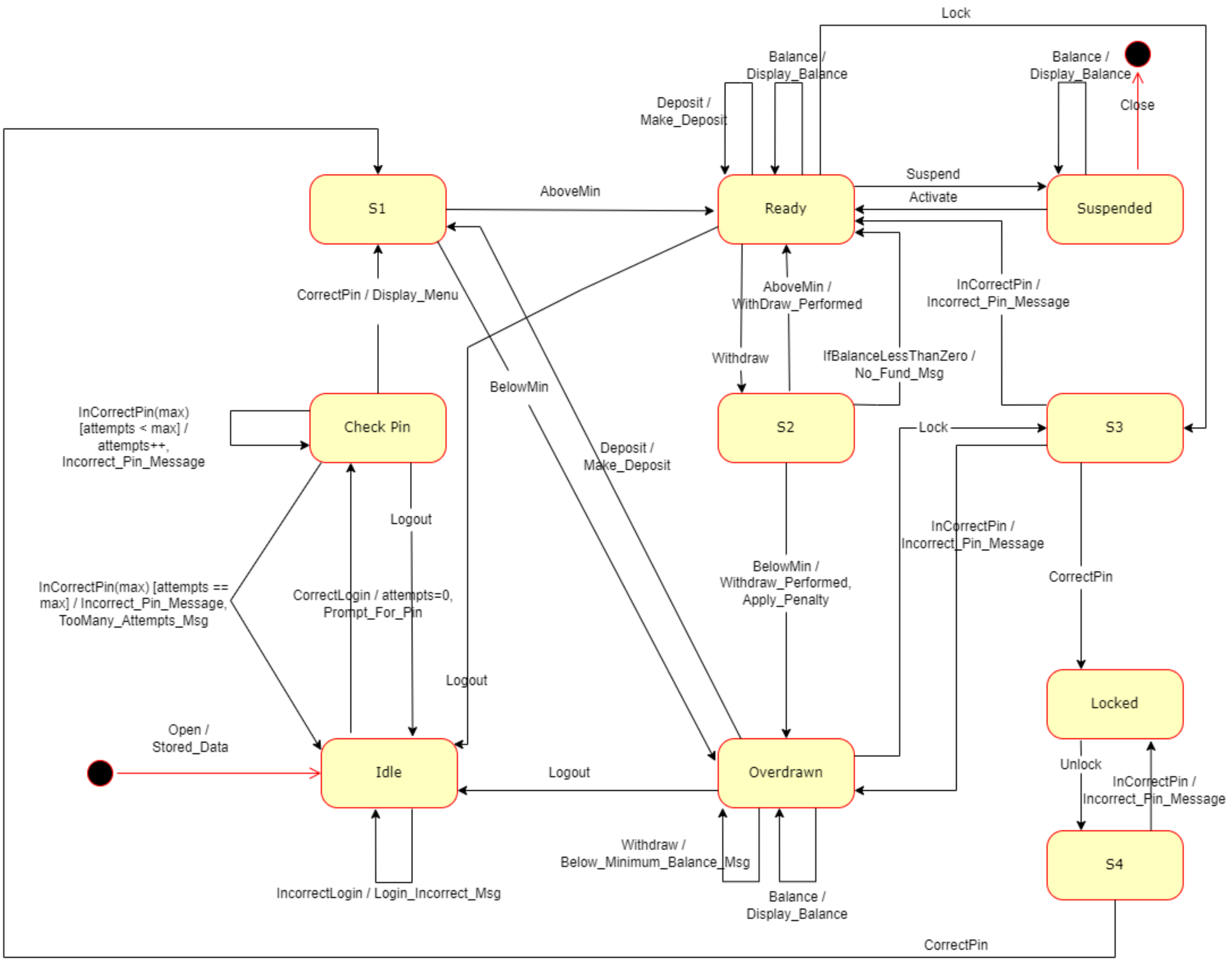
- Apply_Penalty()
  Applies penalty

- No_Fund_Msg()
  Display message when balance is null.

- Below_Min_Balance_Msg()
  Display message when balance is less than minimum limit

- Display_Balance()
  Display the current value of the balance

4. A state diagram/model of the MDA-EFSM


Lock
Balance / Display_Balance
Balance / Display_Balance
Close
Deposit / Make_Deposit
S1
AboveMin
Ready
Suspend
Activate
Suspended
CorrectPin / Display_Menu
AboveMin / WithDraw_Performed
InCorrectPin / Incorrect_Pin_Message
Withdraw
IfBalanceLessThanZero / No_Fund_Msg
BelowMin
InCorrectPin(max) [attempts < max] / attempts++, Incorrect_Pin_Message
Check Pin
S2
Lock
S3
Deposit / Make_Deposit
Logout
InCorrectPin / Incorrect_Pin_Message
BelowMin / Withdraw_Performed, Apply_Penalty
CorrectPin
InCorrectPin(max) [attempts == max] / Incorrect_Pin_Message, TooMany_Attempts_Msg
CorrectLogin / attempts=0, Prompt_For_Pin
Logout
Locked
Open / Stored_Data
Idle
Logout
Overdrawn
Unlock
InCorrectPin / Incorrect_Pin_Message
Withdraw / Below_Minimum_Balance_Msg
S4
IncorrectLogin / Login_Incorrect_Msg
Balance / Display_Balance
CorrectPin

5. Pseudo-code of all operations of Input Processor of Account-1 and Account-2

5.1 Operations of the input processor (Account-1)

5.1.1 Open(int,int,int)

```
open(int p, int y, int a)
{
    d->temp_p = p
    d->temp_y = y
    d->temp_a = a
    m->open()
}
```

5.1.2 Pin(int)

```
Pin(int x)
{
    If(x==d->pn)
    {
        m->CorrectPin()
        If(d->b > 100)
        {
            m->AboveMin()
        }
        Else
        {
            m->BelowMin()

        }
    }
    Else
    {
        m->InCorrectPin()
    }
}
```

5.1.3 Deposit(int)

```
Deposit(int d)
{
    d->temp_d = d;
    m->Deposit()
    if(m->b > 0)
    {
        m->AboveMin()
    }
    Else
    {
        m->BelowMin()
    }
}
```

5.1.4 Withdraw(int)

```
Withdraw(int w)
{
        d->temp_w = w
        if(m->b > 100)
        {
                m->Withdraw()
                m->AboveMin()
        }
        Else
        {
                m->BelowMin()
        }
}
```

5.1.5 Balance()

```
Balance()
{
        m->Balance()
}
```

5.1.6 Login(int)

```
Login(int y)
{
        If(y==d->id)
        {
                m->CorrectLogin()
        }
        Else
        {
                m->InCorrectLogin()
        }
}
```

5.1.7 Logout()

```
Logout()
{
        m->Logout()
}
```

5.1.8 Lock(int)

```
Lock(int x)
{
        If(x==d->pn)
        {
                m->Lock()
        }
        Else
        {
                m->InCorrectPin()
        }
}
```

5.1.9 Unlock(int)

```
Unlock(int x)

{
        If(x==d->pn)
        {
                m->Unlock()
                If(d->b >100)
                {
                        m->AboveMin()
                }
                Else
                {
                        m->BelowMin()
                }
        }
        Else
        {
                m->InCorrectPin()
        }
}
```

5.2 Operations of the input processor (Account-2)

5.2.1 Open(int,int,float)

```
open(int p, int y, float a)
{
        d->temp_p = p
        d->temp_y = y
        d->temp_a = a
        m->open()
}
```

5.2.2 Pin(int)

```
Pin(int x)
{
        If(x==d->pn)
        {
                m->CorrectPin()
        }
        Else
        {
                m->InCorrectPin()
        }
}
```

5.2.3 Deposit(float)

```
Deposit(float d)
{
        d->temp_d = d;
        m->Deposit()
}
```

5.2.4 Withdraw(float)

```
Withdraw(float w)
{
        d->temp_w = w
        if(m->b <= 0)
        {
                m->IfBalanceLessThanZero()
        }
        Else
        {
                m->Withdraw()
        }
}
```

5.2.5 Balance()

```
Balance()
{
        m->Balance()
}
```

5.2.6 Login(int)

```
Login(int y)
{
        If(y==d->id)
        {
                m->CorrectLogin()
        }
        Else
        {
                m->InCorrectLogin()
        }
}
```

5.2.7 Logout()

```
Logout()
{
        m->Logout()
}
```

5.2.8 activate()

```
activate()
{
	m->Activate()
}
```

5.2.9 suspend()

```
suspend()
{
	m->Suspend()
}
```

5.2.10 close()

```
close()
{
	m->Close()
}
```