

Project Report

1. MDA-EFSM model for the *ACCOUNT* components

a. A list of events for the MDA-EFSM

MDA-EFSM Events:

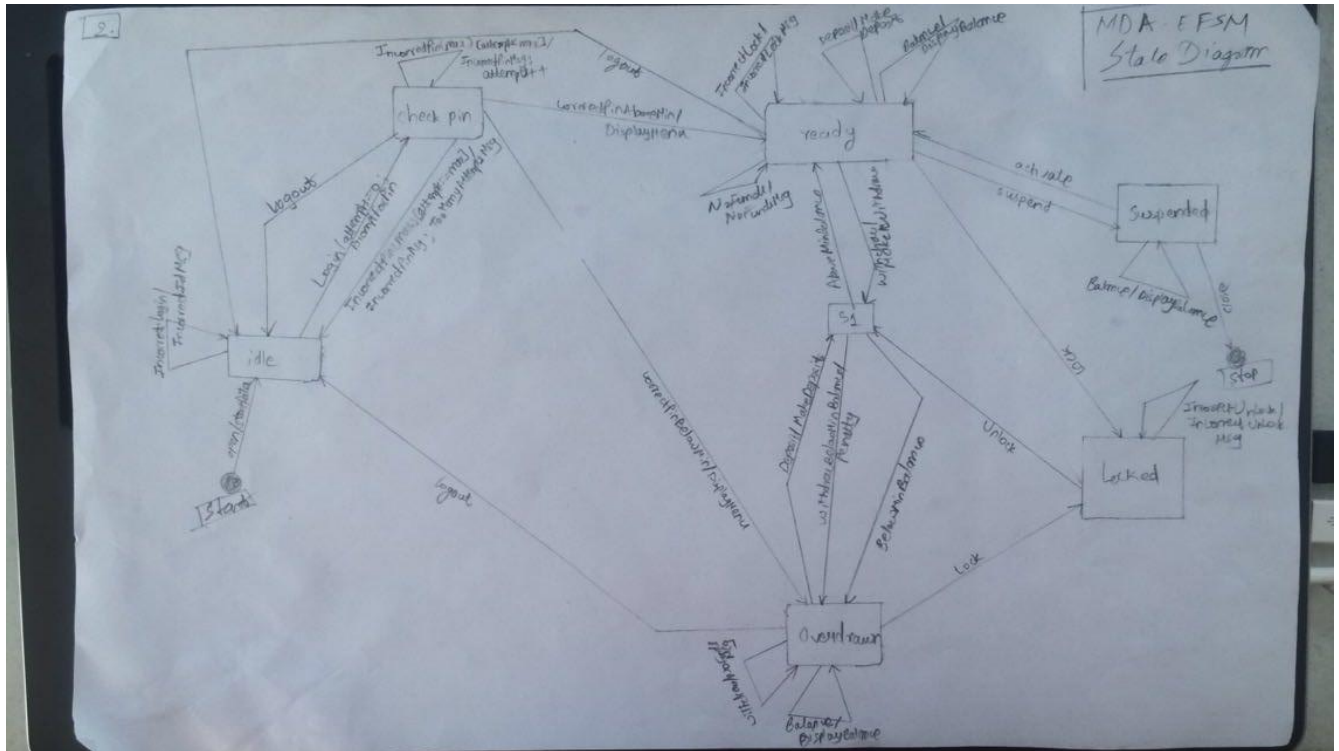
- E1. Open()
- E2. Login()
- E3. IncorrectLogin()
- E4. IncorectPin(int max)
- E5. CorrectPinBelowMin()
- E6. CorrectPinAboveMin()
- E7. Deposit()
- E8. BelowMinBalance()
- E9. AboveMinBalance()
- E10. Logout()
- E11. Balance()
- E12. Withdraw()
- E13. WithdrawBelowMinBalance()
- E14. NoFunds()
- E15. Lock()
- E16. IncorrectLock()
- E17. Unlock()
- E18. IncorrectUnlock()
- E19. Suspend()
- E20. Activate()
- E21. Close()

b. A list of actions for the MDA-EFSM with their descriptions

MDA-EFSM Actions:

- A1: StoreData() // stores pin from temporary data store to *pin* in data store
- A2: IncorrectIdMsg() // displays incorrect ID message
- A3: IncorrectPinMsg() // displays incorrect pin message
- A4: TooManyAttemptsMsg() // display too many attempts message
- A5: DisplayMenu() // display a menu with a list of transactions
- A6: MakeDeposit() // makes deposit (increases balance by a value stored in temp. data store)
- A7: DisplayBalance() // displays the current value of the balance
- A8: PromptForPin() // prompts to enter pin
- A9: MakeWithdraw() // makes withdraw (decreases balance by a value stored in temp. data store)
- A10: Penalty() // applies penalty (decreases balance by the amount of penalty)
- A11: IncorrectLock Msg() // displays incorrect lock msg
- A12: IncorrectUnlock Msg() // displays incorrect unlock msg
- A13: NoFundsMsg() // Displays no sufficient funds msg

c. A state diagram of the MDA-EFSM



d. Pseudo-code of all operations of Input Processors of *ACCOUNT-1* and *ACCOUNT-2*

Pseudo-code of all operations of Input Processors of Account-1:

```
int attempts = 0;

open (string p, string y, float a){

    //storing p, y and a into temp_store

    store p in d -> temp_p

    store y in d -> temp_y

    store a in d -> temp_a

    m -> open()
```

```
}  
pin (string x){  
    if(x == d -> pin){  
        if(d -> balance > 500)  
            m -> correctPinAboveMin()  
        else if(d -> balance <= 500)  
            m -> correctPinBelowMin()  
    }  
    else  
        m -> IncorrectPin(3)  
}  
deposit (float d){  
    store d in d -> temp_d;  
    m -> deposit()  
    if(d -> balance > 500)  
        m -> depositAboveMin()  
    else  
        m -> depositBelowMin()  
}  
withdraw (float w){  
    store w in d -> temp_w;  
    m -> withdraw()  
    if(d -> balance > 500)  
        m -> withdrawAboveMin()  
    else  
        m -> withdrawBelowMin()
```

```
}  
balance (){  
    m -> balance()  
}  
login(string y){  
    if(y == d -> id)  
        m -> login()  
    else  
        m -> incorrectLogin()  
}  
logout(){  
    m -> logout()  
}  
lock(string x){  
    if(x == d -> pin)  
        m -> lock()  
    else if(x != d -> pin)  
        m -> lockIncorrectPin()  
}  
unlock(string x){  
    if(x == d -> pin){  
        if(d -> balance > 500)  
            m -> unlockCorrectPinAboveMin()  
        else if(d -> balance <= 500)  
            m -> unlockCorrectPinBelowMin()  
    }  
    else if(x != d -> pin)  
        m -> lockIncorrectPin()
```

```
}
```

Notice:

m: is a pointer to the MDA-EFSM object

ds: is a pointer to the Data Store object

which contains the following data items:

balance: contains the current balance

pin: contains the correct pin #

uid: contains the correct user ID

temp_p, *temp_y*, *temp_a*, *temp_d*,

temp_w, *temp_x* are used to store values of
parameters

Pseudo-code of all operations of Input Processors of Account-2:

```
int attempts = 0;
```

```
OPEN (int p, int y, int a){
```

```
    //storing p, y and a into temp_store
```

```
    store p in d -> temp_p
```

```
    store y in d -> temp_y
```

```
    store a in d -> temp_a
```

```
    m -> open()
```

```
}  
PIN (int x){  
    if(x == d -> pin)  
        m -> correctPinAboveMin()  
    else  
        m -> IncorrectPin(2)  
  
}  
DEPOSIT (int d){  
    store d in d -> temp_d;  
    m -> Deposit()  
  
}  
WITHDRAW (int w){  
    store w in d -> temp_w;  
    m -> withdraw()  
    if(d -> balance > 0)  
        m -> withdraw()  
    else  
        m -> NoFunds()  
  
}  
BALANCE (){  
    m -> balance()  
}  
  
LOGIN(int y){  
    if(y == d -> id)  
        m -> login()
```

```
        else
            m -> incorrectLogin()

    }

    LOGOUT(){
        m -> logout()
    }

    suspend(){
        m -> suspend()
    }

    activate(){
        m -> activate()
    }

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

Notice:

m: is a pointer to the MDA-EFSM object

ds: is a pointer to the Data Store object
which contains the following data items:

balance: contains the current balance

pin: contains the correct pin #

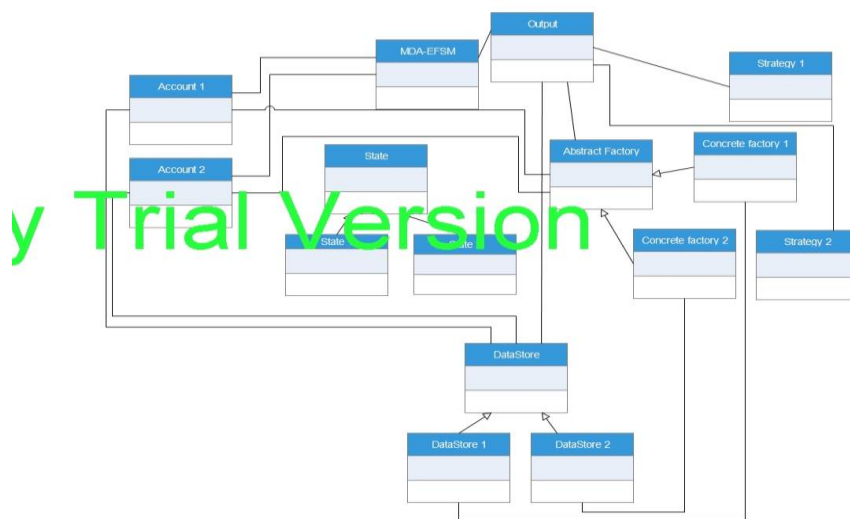
uid: contains the correct user ID

temp_p, *temp_y*, *temp_a*, *temp_d*,

temp_w, *temp_x* are used to store values of
parameters

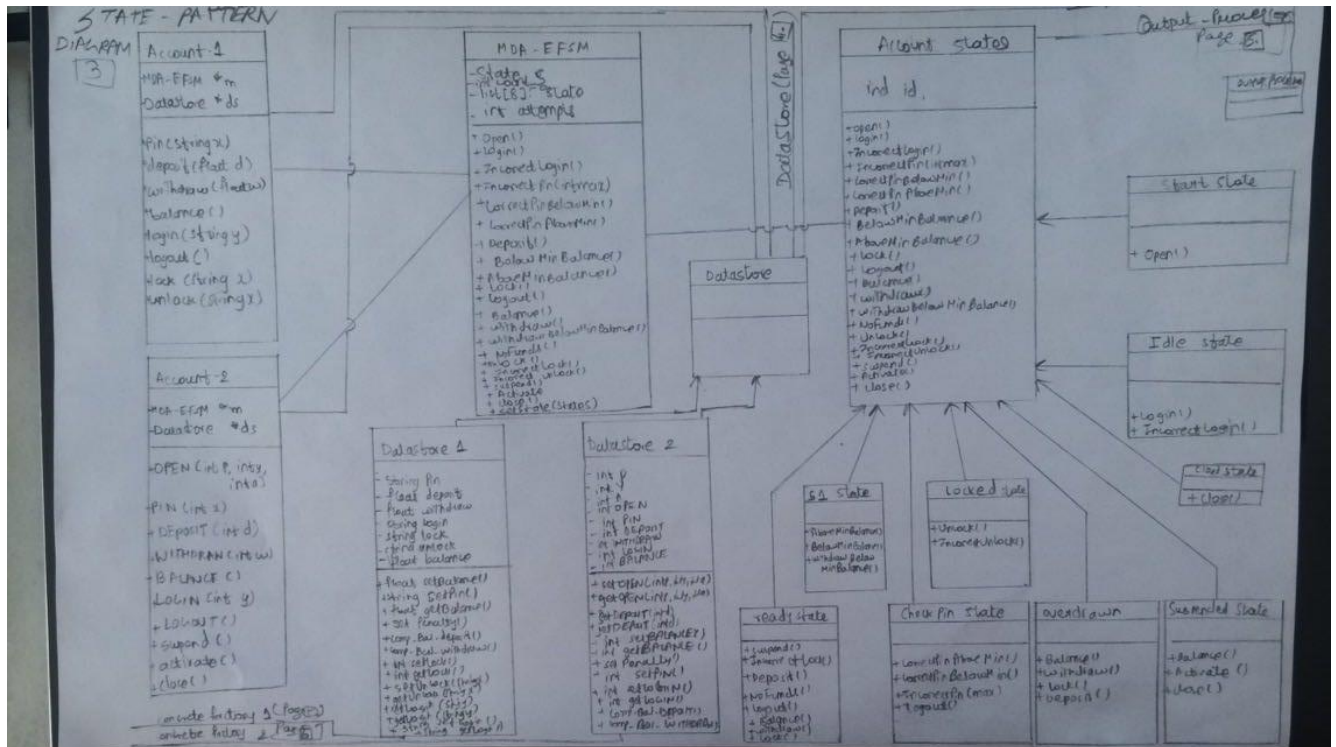
2. Class diagram(s) of the MDA-ACCOUNT components using three patterns namely, State, Strategy and Abstract factory pattern:

High-Level Diagram:

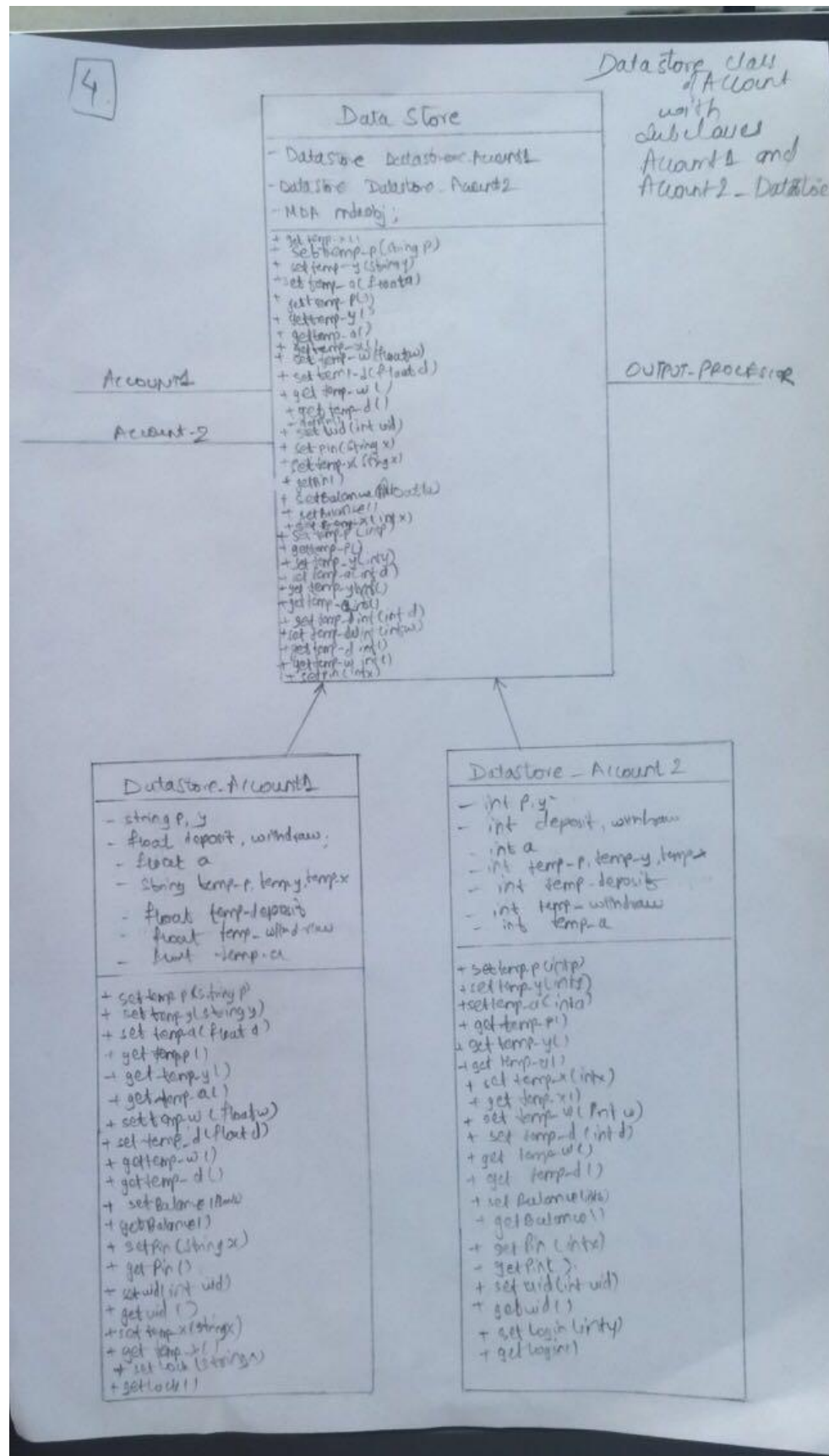


3. For each class in the above class diagrams purpose of the class, specific responsibilities of each operation for each class and purpose of main attributes of the class is mentioned as below:

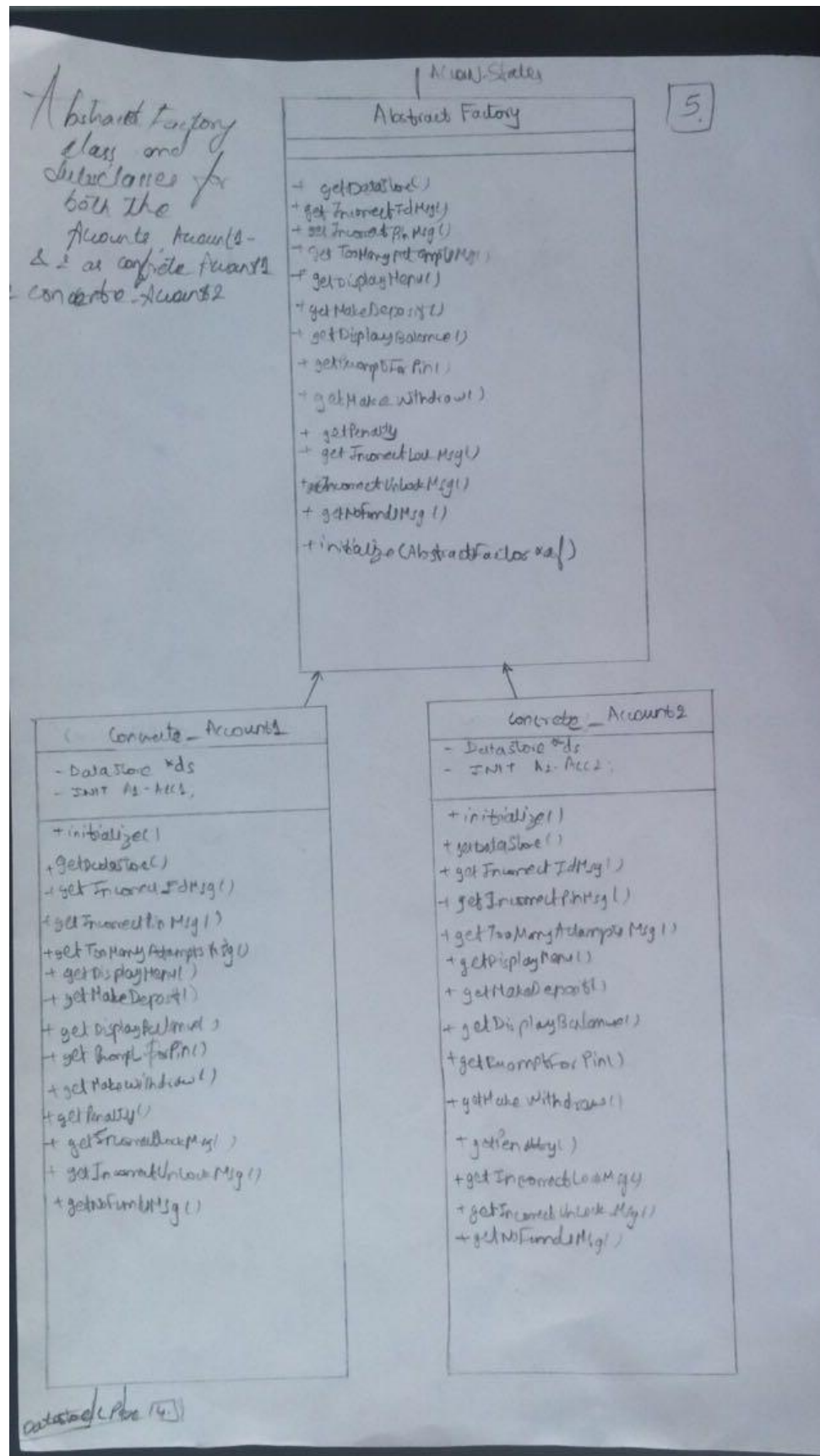
MDA-EFSM State Pattern Diagram Input Processor:



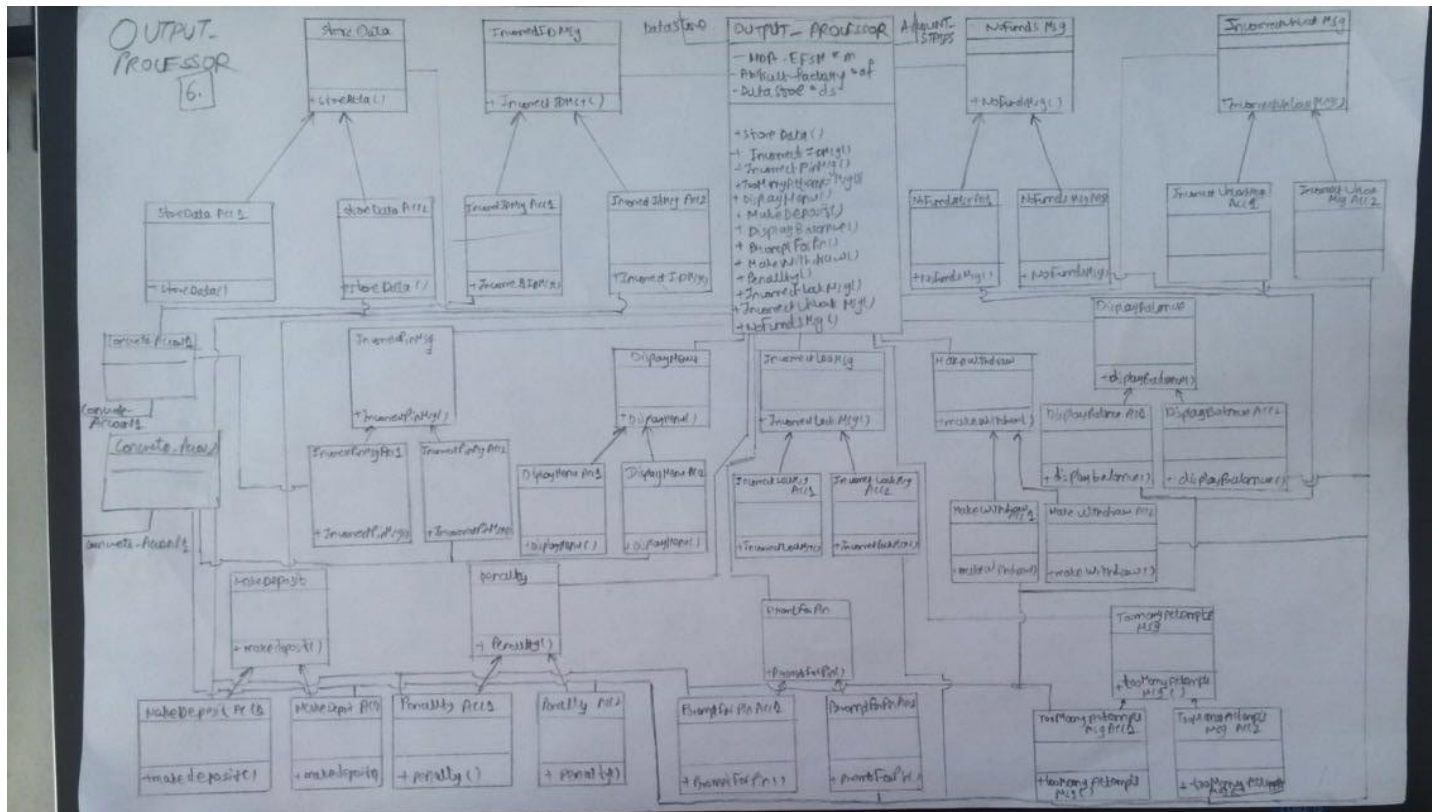
Data Storage of both the Accounts:



Abstract Factory Pattern diagram:

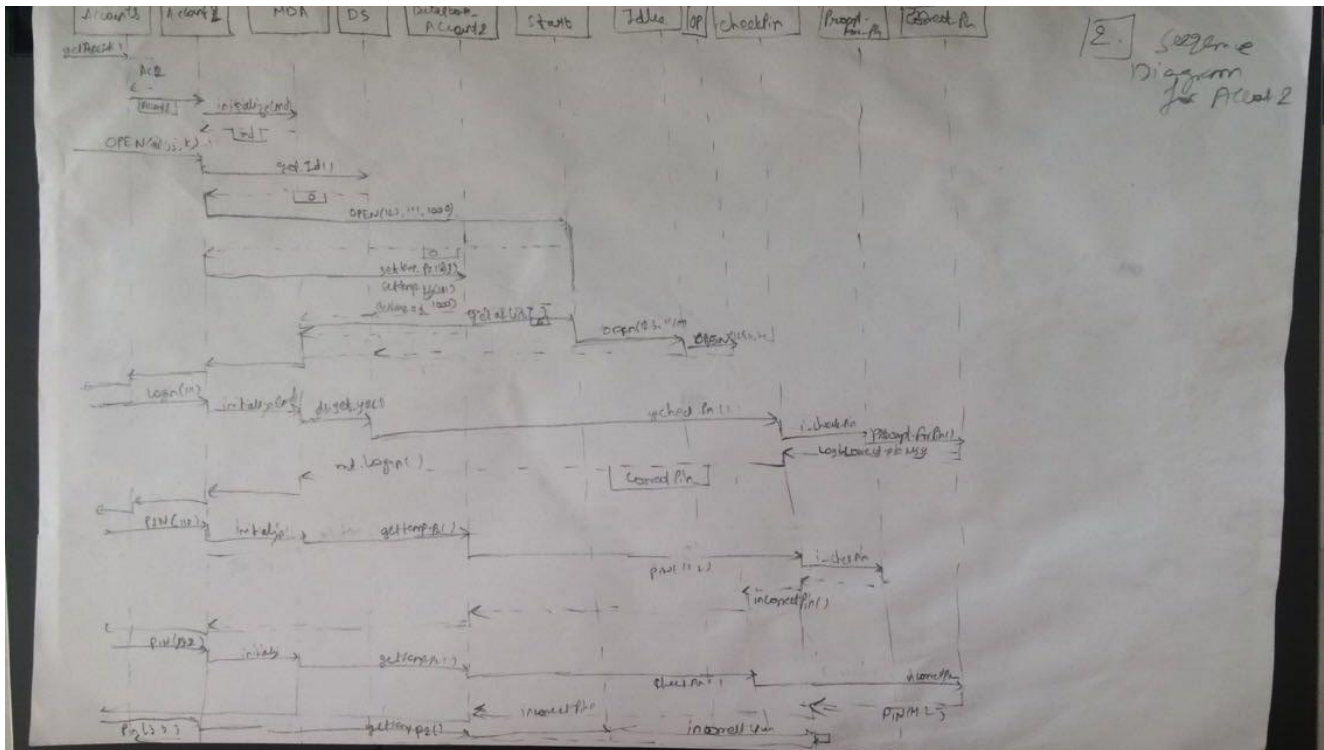


Output Processor diagram:

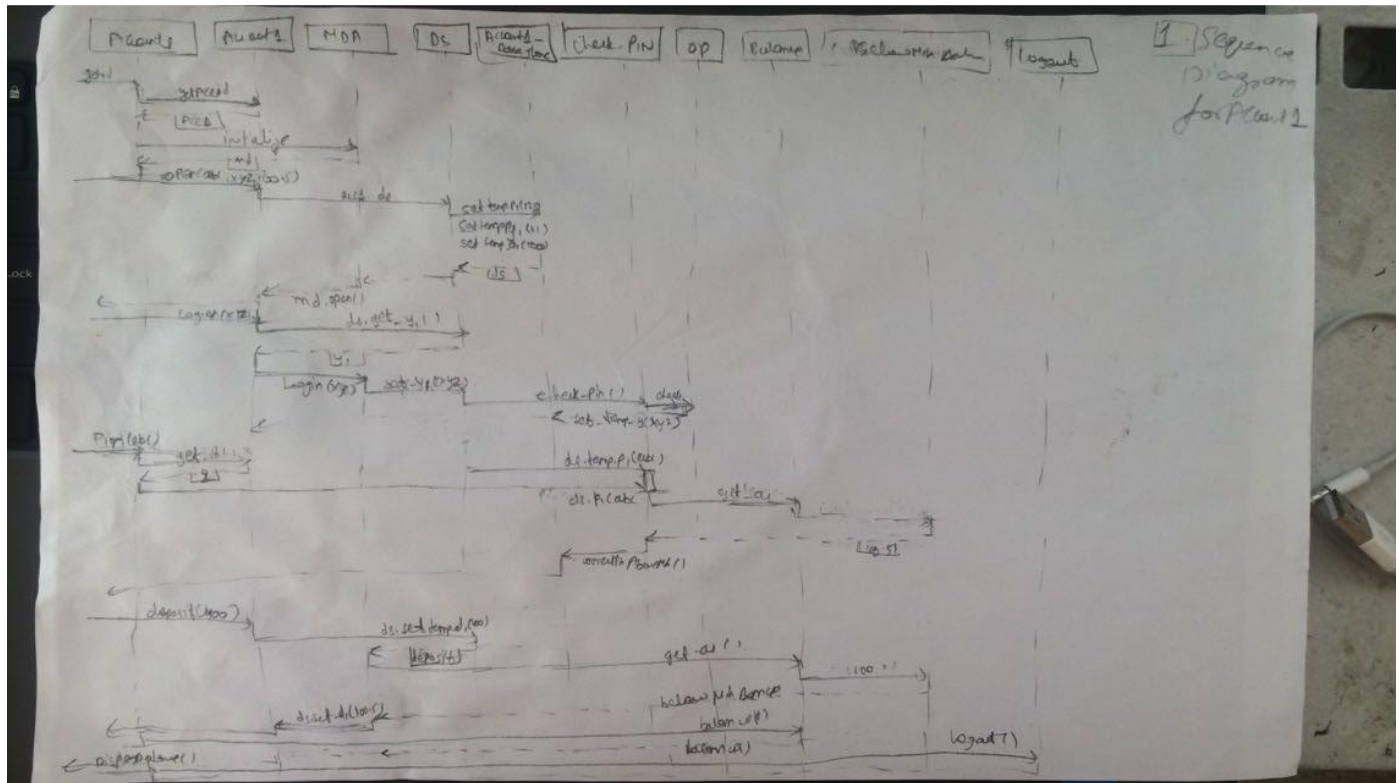


4. Dynamics. Provide two sequence diagrams for two Scenarios:

a. Scenario-I should show as to how the deposit is made in the *ACCOUNT-1* component, i.e., the following sequence of operations is issued: *open(abc,xyz,100.5)*, *login(xyz)*, *pin(abc)*, *deposit(400)*, *balance()*, *logout()*



- b. Scenario-II should show as to how an incorrect pin is entered three times in the *ACCOUNT-2* component, i.e., the following sequence of operations is issued: *OPEN(123,111,1000)*, *LOGIN(111)*, *PIN(112)*, *PIN(222)*, *PIN(333)*



5. Source-code and patterns

In this part of the report you should clearly indicate which parts of the source code are responsible for the implementation of the three required design patterns:

- state pattern
- strategy pattern
- abstract factory pattern.

```
//Driver class Accounts.java

import java.util.Scanner;

//import atm_machine.*;

public class Accounts {

    static int account;
    private static Scanner readInput;
    static MDA_EFSM_Account md=new MDA_EFSM_Account(); // creating the object
of the mda EFSM

    public Accounts() {
        // TODO Auto-generated constructor stub
    }

    public static void main(String[] args) {
        // TODO Auto-generated method stub1

        readInput = new Scanner(System.in);

        System.out.println("Select the type of the Account, 1 for
Account-1 and 2 for Account-2");

        account = readInput.nextInt();

        switch(account){

            case 1:

                Account1 ac1 = new Account1();

                ac1.initialize(md);

                break;

            case 2:

                Account2 ac2 = new Account2();

                ac2.initialize(md);

                break;

            default:

                System.out.println("Wrong selection made");
```

```

        break;
    }
}

}

//Account1.java
import java.util.Scanner;

public class Account1 {

    static int select = '1';
    private static Scanner readInput;
    float a, d, w;
    String x, p, y, uid;

    DataStore ds;
    MDA_EFSM_Account md;

    public void initialize(MDA_EFSM_Account mdl) {
        Concrete_Account1 catm = new Concrete_Account1();
        catm.initialize(catm);

        /*
         * output_processor op=new output_processor();
op.initialize(catm);
         */

        ds = new ACCOUNT1_DataStore();

        md = mdl;

        Ac1();
    }

    public void Ac1() {
        // TODO Auto-generated method stub

        System.out.println("Account 1:");

        System.out.println("Menu of operations: ");

        System.out.println("0. open(string, string, float)");

        System.out.println("1. login(string)");

        System.out.println("2. pin(string)");

        System.out.println("3. deposit(float)");

        System.out.println("4. withdraw(float)");

        System.out.println("5. balance()");
    }
}

```

```

System.out.println("6. logout()");

System.out.println("7. lock(string)");

System.out.println("8. unlock(string)");

System.out.println("q. Quit the Accounts program");

System.out.println("Please make a note of this operations");

readInput = new Scanner(System.in);
//Scanner out = new Scanner(System.in);

while (select != 'q') {

    System.out.println("Select operation: ");

    System.out.println("0-open,1-login,2-pin,3-deposit,4-
withdraw,5-balance,6-logout,7-lock,8-unlock");

    select = readInput.nextInt();

    switch (select) {

        case 0:

            System.out.println("Operation:  open(string p, string
y, float a");

            System.out.println("Enter value of the parameter
p:");

            p = readInput.next();

            // ds.setPin1(p);

            System.out.println("Enter value of the parameter
y:");

            y = readInput.next();

            // ds.setTemp_y1(y);

            System.out.println("Enter value of the parameter
a:");

            a = readInput.nextFloat();

            // ds.setTemp_a1(a);

            open(p, y, a);

            break;

        case 1:

            System.out.println("Operation:  login(string y)");

```

```
        System.out.println("Enter value of the parameter  
y:");  
        y = readInput.next();  
        login(y);  
        break;  
    case 2:  
        System.out.println("Operation:  pin(string x)");  
        System.out.println("Enter value of pin wit the  
parameter x:");  
        x = readInput.next();  
        pin(x);  
        break;  
    case 3:  
        System.out.println("Operation:  deposit(float d)");  
        System.out.println("Enter value of the parameter  
d:");  
        d = readInput.nextFloat();  
        deposit(d);  
        break;  
    case 4:  
        System.out.println("Operation:  withdraw(float w)");  
        System.out.println("Enter value of the parameter  
w:");  
        w = readInput.nextInt();  
        withdraw(w);  
        break;  
    case 5:  
        System.out.println("Operation:  balance()");  
        balance();  
        break;  
    case 6:  
        System.out.println("Operation:  LOGOUT()");  
        logout();  
        break;  
    case 7:
```

```

        System.out.println("Operation:  lock(string x)");
        System.out.println("Enter the pin p:");
        x = readInput.nextLine();
        lock(x);
        break;

    case 8:

        System.out.println("Operation:  unlock(string x)");
        System.out.println("Enter the pin p:");
        x = readInput.nextLine();
        unlock(x);
        break;

    default:

        System.out.println("Wrong selection made");

        break;

    }

}

}

public void unlock(String x2) {
    // TODO Auto-generated method stub
    // md.unLock();
    md.unLock();
    if (x.equals(ds.get_p1())) {

        md.unLock();
        if (ds.get_a1() > 500) {
            md.aboveMinBalance();
        }

        else{

            md.belowMinBalance();

        }

    }

    else{

        md.incorrectUnLock();

    }

}

public void lock(String x2) {
    // TODO Auto-generated method stub
    //md.lock();
    System.out.println(ds.get_p1());

```

```
System.out.println(x2);
if (x2.equals(ds.get_p1())) {
    System.out.println("its correct pin");
    md.lock();
}

else{
    md.incorrectLock();
}
}

public void logout() {
    // TODO Auto-generated method stub
    md.logout();
}

public void balance() {
    // TODO Auto-generated method stub
    md.balance();
}

private void withdraw(float w2) {
    // TODO Auto-generated method stub

    ds.setTemp_w1(w);

    md.withdraw();

    if (ds.get_a1() > 500) {

        md.aboveMinBalance();
        //System.out.println("-----" + ds.get_a1());
        //if (ds.get_a1() <= 500)
        //md.withdrawBlowMinBalance();
    }

    else{
        //md.withdraw();
        md.withdrawBlowMinBalance();
    }
}

public void deposit(float d2) {
    // TODO Auto-generated method stub

    ds.setTemp_d1(d2);

    md.deposit();

    if (ds.get_a1() > 500)

        md.aboveMinBalance();
}
```

```
        else
//////////
            //md.belowMinBalance();
            md.belowMinBalance();
    }

    public void pin(String x2) {
        // TODO Auto-generated method stub

        if (x2.equals(ds.get_p1())) {

            if (ds.get_a1() > 500)

                md.correctPinAboveMin();

            else if (ds.get_a1() <= 500)

                md.correctPinBelowMin();

        }

        else {

            md.incorrectPin(3);

        }

    }

    public void login(String y2) {
        // TODO Auto-generated method stub
        System.out.println(ds.get_y1());
        System.out.println(y2);

        if (y2.equals(ds.get_y1()))
            md.login();

        else

            md.incorrectLogin();

    }

    public void open(String p2, String y2, float a2) {
        // TODO Auto-generated method stub

        ds.setTemp_p1(p2);
        ds.setTemp_y1(y2);
        ds.setTemp_a1(a2);

        md._---_();

    }

}

//Account2.java
```



```
import java.util.Scanner;

public class Account2 {

    static int select = '1';
    private static Scanner readInput;
    int p, y, a, x, d, w;

    DataStore ds;
    MDA_EFSM_Account md;

    public void initialize(MDA_EFSM_Account md2)
    {
        Concrete_Account2 catm=new Concrete_Account2();
        catm.initialize(catm);

        /*output_processor op=new output_processor();
        op.initialize(catm);*/

        ds=new ACCOUNT2_DataStore();

        md=md2;

        Ac2();
    }

    public void Ac2() {

        System.out.println("Account 2:");

        System.out.println("Menu of operations: ");

        System.out.println("0. OPEN(int,int,int)");
        System.out.println("1. LOGIN(int)");
        System.out.println("2. PIN(int)");
        System.out.println("3. DEPOSIT(int)");
        System.out.println("4. WITHDRAW(int)");
        System.out.println("5. BALANCE()");
        System.out.println("6. LOGOUT()");
        System.out.println("7. suspend()");
        System.out.println("8. activate()");
        System.out.println("9. close()");
        System.out.println("q. Quit the Accounts2 program");
        System.out.println("Please make a note of this operations");
    }
}
```

```
readInput = new Scanner(System.in);

while(select!= 'q'){

    System.out.println("Select operation: ");

    System.out.println("0-OPEN,1-LOGIN,2-PIN,3-DEPOSIT,4-  
WITHDRAW,5-BALANCE,6-LOGOUT,7-suspend,8-activate,9-close");

    select = readInput.nextInt();

    switch(select){

    case 0:

        System.out.println("Operation:  OPEN(int p, int y, int a");

        System.out.println("Enter value of the parameter p:");
        p = readInput.nextInt();

        System.out.println("Enter value of the parameter y:");
        y = readInput.nextInt();

        System.out.println("Enter value of the parameter a:");
        a = readInput.nextInt();

        OPEN(p,y,a);

        break;

    case 1:

        System.out.println("Operation:  LOGIN(int y)");

        System.out.println("Enter value of the parameter y:");
        y = readInput.nextInt();

        LOGIN(y);

        break;

    case 2:

        System.out.println("Operation:  PIN(int x)");

        System.out.println("Enter value of the parameter x:");
        x = readInput.nextInt();
```

```
        PIN(x);  
        break;  
  
    case 3:  
  
        System.out.println("Operation:  DEPOSIT(int d)");  
  
        System.out.println("Enter value of the parameter d:");  
        d = readInput.nextInt();  
  
        DEPOSIT(d);  
        break;  
  
    case 4:  
  
        System.out.println("Operation:  WITHDRAW(int w)");  
  
        System.out.println("Enter value of the parameter w:");  
        w = readInput.nextInt();  
  
        WITHDRAW(w);  
        break;  
  
    case 5:  
  
        System.out.println("Operation:  BALANCE()");  
  
        BALANCE();  
        break;  
  
    case 6:  
  
        System.out.println("Operation:  LOGOUT()");  
  
        LOGOUT();  
        break;  
  
    case 7:  
  
        System.out.println("Operation:  suspend()");  
  
        suspend();  
        break;  
  
    case 8:  
  
        System.out.println("Operation:  activate()");
```

```
        activate();
        break;

    case 9:

        System.out.println("Operation:   close()");

        close();
        break;

    default:

        System.out.println("Wrong selection made");

        break;

    }

}

}

public void close() {
    // TODO Auto-generated method stub
    md.close();
}

public void activate() {
    // TODO Auto-generated method stub
    md.activate();
}

public void suspend() {
    // TODO Auto-generated method stub
    md.suspend();
}

public void LOGOUT() {
    // TODO Auto-generated method stub
    md.logout();
}

public void BALANCE() {
    // TODO Auto-generated method stub
    md.balance();
}

}

private void WITHDRAW(int w2) {
    // TODO Auto-generated method stub

    ds.setTemp_w2(w2);
}
```

```
md.withdraw();

if(ds.get_a2() > 0)

    md.withdraw();

else

    md.noFunds();

}

public void DEPOSIT(int d2) {
    // TODO Auto-generated method stub

    ds.setTemp_d2(d2);

    md.deposit();

}

public void PIN(int x2) {
    // TODO Auto-generated method stub

    if(x2 == ds.get_p2())

        md.correctPinAboveMin();

    else

        md.incorrectPin(2);

}

public void LOGIN(int l) {
    // TODO Auto-generated method stub

    System.out.println(ds.get_y2());
    System.out.println(l);

    if (l == ds.get_y2())
        md.login();

    else

        md.incorrectLogin();

}

public void OPEN(int i, int j, int k) {
    // TODO Auto-generated method stub

    ds.setTemp_p2(i);
    ds.setTemp_y2(j);
    ds.setTemp_a2(k);
    md.open();
}
```

```

    }

    }

//Input MDA EFSM class

public class MDA_EFSM_Account {

    // count is responsible for storing the value of the number of attempts
    public static int count = 0;

    private final Account_States list[] = { new Start(), new Idle(), new
Check_Pin(), new Ready(), new S1(),
        new Overdrawn(), new Locked(), new Suspended(), new Stop()
};

    // private final DataStore list2[] = {new ACCOUNT1_DataStore(), new
// ACCOUNT2_DataStore()};

    public Account_States as = list[0];
    // public DataStore as2 = list2[0];

    // object of different states are created and the object is initialized
with
    // the idle state.

    public void open() {

        as.open();
        switch (as.get_id()) {
            case 0:
                as = list[1];
                //System.out.println(as.get_id());
                System.out.println("State changed to Idle");
                break;

            default:
                System.out.println("Operation not permitted in this
state");
                break;
        }

    }

    public void login() {

        as.login();
        switch (as.get_id()) {
            case 1:
                as = list[2];
                System.out.println("State changed to Check pin");
                break;

            default:
                System.out.println("Operation not permitted in this
state");
                break;
        }
    }
}

```

```

    }

}

public void incorrectLogin() {

    as.incorrectLogin();
    switch (as.get_id()) {
    case 1:
        as = list[1];
        System.out.println("State remains in the Idle");
        break;

    default:
        System.out.println("Operation not permitted in this
state");
        break;
    }
}

public void incorrectPin(int max) {

    as.incorectPin(max);
    switch (as.get_id()) {
    case 2:
        if (count == max) {
            as = list[1];
            System.out.println("State changed to Idle");
        } else if (count < max) {
            as = list[2];
            System.out.println("State remains in the Check pin");
        }
        break;
    default:
        System.out.println("Operation not permitted in this
state");
        break;
    }
}

public void correctPinBelowMin() {

    as.correctPinBelowMin();
    switch (as.get_id()) {
    case 2:
        as = list[5];
        System.out.println("State changed to Overdrawn");
        break;
    // case 4:
    //     as = list[5];
    //     System.out.println("State changed to Overdrawn");
    //     break;

    default:
        System.out.println("Operation not permitted in this
state");

```

```
                break;
            }
        }

    public void correctPinAboveMin() {

        as.correctPinAboveMin();
        switch (as.get_id()) {
            case 2:
                as = list[3];
                System.out.println("State changed to READY");
                break;

            default:
                System.out.println("Operation not permitted in this
state");
                break;
        }
    }

    public void deposit() {

        as.deposit();
        switch (as.get_id()) {
            case 3:
                as = list[3];
                System.out.println("State remains in READY");
                break;
            case 5:
                as = list[4];
                System.out.println("State changed to S1");
                break;
            default:
                System.out.println("Operation not permitted in this
state");
                break;
        }
    }

    public void belowMinBalance() {

        as.belowMinBalance();

        switch (as.get_id()) {
            case 4:
                as = list[5];
                System.out.println("State changed to Overdrawn");
                break;

            default:
                System.out.println("Operation not permitted in this
state");
                break;
        }
    }
}
```



```
public void aboveMinBalance() {

    as.aboveMinBalance();
    switch (as.get_id()) {
    case 4:
        as = list[3];
        System.out.println("State changed to Ready");
        break;

    default:
        System.out.println("Operation not permitted in this
state");
        break;
    }
}

public void logout() {

    as.logout();
    switch (as.get_id()) {

    case 2:
        as = list[1];
        System.out.println("State changed to Idle");
        break;
    case 3:
        as = list[1];
        System.out.println("State changed to Idle");
        break;
    case 5:
        as = list[1];
        System.out.println("State changed to Idle");
        break;
    default:
        System.out.println("Operation not permitted in this
state");
        break;
    }
}

public void balance() {

    as.balance();

    switch (as.get_id()) {
    case 3:
        as = list[3];
        System.out.println("State remains in Ready");
        break;
    case 5:
        as = list[5];
        System.out.println("State remains in Overdrawn");
        break;
    case 7:
```

```

        as = list[7];
        System.out.println("State remains in Suspended");
        break;
    default:
        System.out.println("Operation not permitted in this
state");
        break;
    }
}

public void withdraw() {
    as.withdraw();
    System.out.println(as.get_id());
    switch (as.get_id()) {
    case 3:
        as = list[4];
        System.out.println("State changed to S1");
        break;
    case 5:
        as = list[5];
        System.out.println("State remains in Overdrawn");
        break;
    // case 4:
    //     as = list[5];
    //     System.out.println("State changed to Ready");
    //     break;
    default:
        System.out.println("Operation not permitted in this
state");
        break;
    }
}

public void withdrawBlowMinBalance() {
    as.withdrawBelowMinBalance();
    switch (as.get_id()) {
    case 4:
        as = list[5];
        System.out.println("State changed to Overdrawn");
        break;
    default:
        System.out.println("Operation not permitted in this
state");
        break;
    }
}

public void noFunds() {
    as.noFunds();
    switch (as.get_id()) {
    case 3:

```

```

        as = list[3];
        System.out.println("State remains in READY");
        break;

    default:
        System.out.println("Operation not permitted in this
state");
        break;
    }

}

public void lock() {

    as.lock();
    switch (as.get_id()) {
    case 5:
        as = list[6];
        System.out.println("State changed to Locked");
        break;
    case 3:
        as = list[6];
        System.out.println("State changed to Locked");
        break;
    default:
        System.out.println("Operation not permitted in this
state");
        break;
    }

}

public void incorrectLock() {

    as.incorrectLock();
    switch (as.get_id()) {
    case 3:
        as = list[3];
        System.out.println("State remains in the Ready state
only");
        break;

    default:
        System.out.println("Operation not permitted in this
state");
        break;
    }

}

public void unLock() {

    as.unLock();
    switch (as.get_id()) {
    case 6:
        as = list[4];
        System.out.println("State changed to S1");
        break;

```

```
        default:
            System.out.println("Operation not permitted in this
state");
            break;
        }
    }

    public void incorrectUnLock() {

        as.incorrectUnLock();

        switch (as.get_id()) {
        case 6:
            as = list[6];
            System.out.println("Remains in the same Locked state");
            break;

        default:
            System.out.println("Operation not permitted in this
state");
            break;
        }
    }

    public void suspend() {

        as.suspend();
        switch (as.get_id()) {
        case 3:
            as = list[7];
            System.out.println("State changed to Suspended");
            break;

        default:
            System.out.println("Operation not permitted in this
state");
            break;
        }
    }

    public void activate() {

        as.activitate();
        switch (as.get_id()) {
        case 7:
            as = list[3];
            System.out.println("State changed to Ready");
            break;

        default:
            System.out.println("Operation not permitted in this
state");
            break;
        }
    }
}
```

```

    }

    public void close() {

        as.close();

        switch (as.get_id()) {
            case 7:
                as = list[8];
                System.out.println("State changed to Stop state");
                break;

            default:
                System.out.println("Operation not permitted in this
state");
                break;
        }

        System.out.println("No change in state, End of the System");
    }

    public void set_count(int x) {
        count = x;
    }

    public int get_count() {
        return (count);
    }
}

```

//Abstract Factory pattern java classes

AbstractFactory.java

```

abstract class Abstract_Factory {

    abstract DataStore get_datastore();

    /*
    Functions listed below are abstract. The implementation of these
    functions are given in the corresponding concrete implementations.
    */

    abstract StorePin storePin();
    abstract IncorrectMsg incorrectMsg();
    abstract IncorrectPinMsg incorrectPinMsg();
    abstract TooManyAttemptsMsg tooManyAttemptsMsg();
    abstract DisplayMenu displayMenu();
    abstract MakeDeposit makeDeposit();
    abstract DisplayBalance displayBalance();
    abstract PromptForPin promptForPin();
    abstract Penalty penalty();
    abstract MakeWithdraw makeWithdraw();
    abstract IncorrectLockMsg incorrectLockMsg();
    abstract IncorrectUnLockMsg incorrectUnLockMsg();
    abstract NoFundsMsg noFundsMsg();
}

```

```
        abstract void initialize(Abstract_Factory con);
    }

    //Concrete factory Account1

    public class Concrete_Account1 extends Abstract_Factory{

        @Override
        DataStore get_datastore() {

            // TODO Auto-generated method stub
            return(new ACCOUNT1_DataStore());
        }

        @Override
        StorePin storePin() {

            //return (new ACCOUNT1_DataStore());
            // TODO Auto-generated method stub

            return(new StorePin_Account1());
        }

        @Override
        IncorrectMsg incorrectMsg() {

            // TODO Auto-generated method stub
            return(new IncorrectMsg_Account1());
        }

        @Override
        IncorrectPinMsg incorrectPinMsg() {
            // TODO Auto-generated method stub
            return (new IncorrectPinMsg_Account1());
        }

        @Override
        TooManyAttemptsMsg tooManyAttemptsMsg() {
            // TODO Auto-generated method stub

            return (new TooManyAttemptsMsg_Account1());
        }

        @Override
        DisplayMenu displayMenu() {
            // TODO Auto-generated method stub

            return (new DisplayMenu_Account1());
        }

        @Override
        MakeDeposit makeDeposit() {
            // TODO Auto-generated method stub
```

```
        return (new MakeDeposit_Account1());
    }

    @Override
    DisplayBalance displayBalance() {
        // TODO Auto-generated method stub

        return (new DisplayBalance_Account1());
    }

    @Override
    PromptForPin promptForPin() {
        // TODO Auto-generated method stub

        return (new PromptForPin_Account1());
    }

    @Override
    Penalty penalty() {
        // TODO Auto-generated method stub

        return (new Penalty_Account1());
    }

    @Override
    MakeWithdraw makeWithdraw() {
        // TODO Auto-generated method stub

        return (new MakeWithdraw_Account1());
    }

    @Override
    IncorrectLockMsg incorrectLockMsg() {
        // TODO Auto-generated method stub

        return (new IncorrectLockMsg_Account1());
    }

    @Override
    IncorrectUnLockMsg incorrectUnLockMsg() {
        // TODO Auto-generated method stub

        return (new IncorrectUnLockMsg_Account1());
    }

    @Override
    NoFundsMsg noFundsMsg() {
        // TODO Auto-generated method stub

        return (new NoFundsMsg_Account1());
    }
}
```

```
    }

    @Override
    void initialize(Abstract_Factory con) {
        // TODO Auto-generated method stub

        Output_Processor op1 = new Output_Processor();

        op1.initialize(this);
    }
}

//Concrete factory for Account2

public class Concrete_Account2 extends Abstract_Factory{

    @Override
    DataStore get_datastore() {

        // TODO Auto-generated method stub
        return(new ACCOUNT2_DataStore());
    }

    @Override
    StorePin storePin() {
        // TODO Auto-generated method stub

        return(new StorePin_Account2());
    }

    @Override
    IncorrectMsg incorrectMsg() {
        // TODO Auto-generated method stub

        return(new IncorrectMsg_Account2());
    }

    @Override
    IncorrectPinMsg incorrectPinMsg() {
        // TODO Auto-generated method stub
        return (new IncorrectPinMsg_Account2());
    }

    @Override
    TooManyAttemptsMsg tooManyAttemptsMsg() {
        // TODO Auto-generated method stub
        return (new TooManyAttemptsMsg_Account2());
    }

    @Override
    DisplayMenu displayMenu() {
```



```
        // TODO Auto-generated method stub
        return (new DisplayMenu_Account2());
    }

    @Override
    MakeDeposit makeDeposit() {
        // TODO Auto-generated method stub
        return (new MakeDeposit_Account2());
    }

    @Override
    DisplayBalance displayBalance() {
        // TODO Auto-generated method stub
        return (new DisplayBalance_Account2());
    }

    @Override
    PromptForPin promptForPin() {
        // TODO Auto-generated method stub
        return (new PromptForPin_Account2());
    }

    @Override
    Penalty penalty() {
        // TODO Auto-generated method stub
        return (new Penalty_Account2());
    }

    @Override
    MakeWithdraw makeWithdraw() {
        // TODO Auto-generated method stub
        return (new MakeWithdraw_Account2());
    }

    @Override
    IncorrectLockMsg incorrectLockMsg() {
        // TODO Auto-generated method stub
        return (new IncorrectLockMsg_Account2());
    }

    @Override
    IncorrectUnLockMsg incorrectUnLockMsg() {
        // TODO Auto-generated method stub
        return (new IncorrectUnLockMsg_Account2());
    }

    @Override
    NoFundsMsg noFundsMsg() {
        // TODO Auto-generated method stub
        return (new NoFundsMsg_Account2());
    }

    @Override
    void initialize(Abstract_Factory con) {
        // TODO Auto-generated method stub
        Output_Processor op2 = new Output_Processor();
        op2.initialize(this);
    }
}
```

```
    }  
}  
  
//Data Store classes for storing temporary and permanent data:  
  
abstract class DataStore {  
  
    //Account1 Main  
    public void set_p1(String p){}  
    public String get_p1()  
    {  
        return(null);  
    }  
  
    public void set_y1(String y){}  
    public String get_y1()  
    {  
        return(null);  
    }  
  
    public void set_a1(float a){}  
    public float get_a1()  
    {  
        return(0);  
    }  
  
    public void set_d1(float d){}  
    public float get_d1()  
    {  
        return(0);  
    }  
  
    public void set_w1(float w){}  
    public float get_w1()  
    {  
        return(0);  
    }  
  
    public void set_x1(String x){}  
    public String get_x1()  
    {  
        return(null);  
    }  
  
    public void set_uid1(String uid){}  
    public String get_uid1()  
    {  
        return(null);  
    }  
  
    public void setPin1(String pin){}  
    public String getPin1()  
    {  
        return(null);  
    }  
}
```

```
public void setBalance1(float b){}
public float getBalance1()
{
    return(0);
}

//Account1 Temp

public void setTemp_p1(String p){}
public String getTemp_p1()
{
    return(null);
}

public void setTemp_y1(String y){}
public String getTemp_y1()
{
    return(null);
}

public void setTemp_a1(float a){}
public float getTemp_a1()
{
    return(0);
}

public void setTemp_d1(float d){}
public float getTemp_d1()
{
    return(0);
}

public void setTemp_w1(float w){}
public float getTemp_w1()
{
    return(0);
}

public void setTemp_x1(String x){}
public String getTemp_x1()
{
    return(null);
}

public void setTemp_uid1(String uid){}
public String getTemp_uid1()
{
    return(null);
}

public void setTemp_Pin1(String pin){}
public String getTemp_Pin1()
{
    return(null);
}
```

```
public void setTemp_Balance1(float b){}
public float getTemp_Balance1()
{
    return(0);
}

//Account2 Main

public void set_p2(int p){}
public int get_p2()
{
    return(0);
}

public void set_y2(int y){}
public int get_y2()
{
    return(0);
}

public void set_a2(int a){}
public int get_a2()
{
    return(0);
}

public void set_d2(int d){}
public int get_d2()
{
    return(0);
}

public void set_w2(int w){}
public int get_w2()
{
    return(0);
}

public void set_x2(int x){}
public int get_x2()
{
    return(0);
}

public void set_uid2(int uid){}
public int get_uid2()
{
    return(0);
}

public void setPin2(int pin){}
public int getPin2()
{
    return(0);
}
```

```
        public void setBalance2(int b) {}
        public int getBalance2()
        {
            return(0);
        }

//Account2 Temp

public void setTemp_p2(int p) {}
public int getTemp_p2()
{
    return(0);
}

public void setTemp_y2(int y) {}
public int getTemp_y2()
{
    return(0);
}

public void setTemp_a2(int a) {}
public int getTemp_a2()
{
    return(0);
}

public void setTemp_d2(int d) {}
public int getTemp_d2()
{
    return(0);
}

public void setTemp_w2(int w) {}
public int getTemp_w2()
{
    return(0);
}

public void setTemp_x2(int x) {}
public int getTemp_x2()
{
    return(0);
}

public void setTemp_uid2(int uid) {}
public int getTemp_uid2()
{
    return(0);
}

public void setTemp_Pin2(int pin) {}
public int getTemp_Pin2()
{
    return(0);
}

public void setTemp_Balance2(int b) {}
```

```

    public int getTemp_Balance2()
    {
        return (0);
    }
}

//datastore for account1

public class ACCOUNT1_DataStore extends DataStore {

    public static String p, y;//permanent
    public static float deposit, withdraw, a;//balance;//permanent

    public static String temp_p, temp_y, temp_x;//temp
    public static float temp_deposit, temp_withdraw, temp_a;//temp

    @Override
    public void setTemp_p1(String p) {
        // TODO Auto-generated method stub
        ACCOUNT1_DataStore.temp_p = p;
    }

    @Override
    public String getTemp_p1() {
        // TODO Auto-generated method stub

        return (temp_p);
    }

    @Override
    public void set_p1(String p) {
        // TODO Auto-generated method stub
        ACCOUNT1_DataStore.p = p;
    }

    @Override
    public String get_p1() {
        // TODO Auto-generated method stub
        return (p);
    }

    @Override
    public void set_y1(String y) {
        // TODO Auto-generated method stub
        ACCOUNT1_DataStore.y = y;
    }

    @Override
    public String get_y1() {
        // TODO Auto-generated method stub
        return y;
    }

    @Override
    public void set_a1(float a) {

```

```
        // TODO Auto-generated method stub
        ACCOUNT1_DataStore.a = a;
    }

    @Override
    public float get_a1() {
        // TODO Auto-generated method stub
        return a;
    }

    @Override
    public void set_d1(float d) {
        // TODO Auto-generated method stub
        deposit = d;
    }

    @Override
    public float get_d1() {
        // TODO Auto-generated method stub
        return deposit;
    }

    @Override
    public void set_w1(float w) {
        // TODO Auto-generated method stub
        withdraw = w;
    }

    @Override
    public float get_w1() {
        // TODO Auto-generated method stub
        return withdraw;
    }

    @Override
    public void setTemp_y1(String y) {
        // TODO Auto-generated method stub
        temp_y = y;
    }

    @Override
    public String getTemp_y1() {
        // TODO Auto-generated method stub
        return temp_y;
    }

    @Override
    public void setTemp_a1(float a) {
        // TODO Auto-generated method stub
        temp_a = a;
    }

    @Override
```

```
    public float getTemp_a1() {
        // TODO Auto-generated method stub
        return temp_a;
    }

    @Override
    public void setTemp_d1(float d) {
        // TODO Auto-generated method stub
        temp_deposit = d;
    }

    @Override
    public float getTemp_d1() {
        // TODO Auto-generated method stub
        return temp_deposit;
    }

    @Override
    public void setTemp_w1(float w) {
        // TODO Auto-generated method stub
        temp_withdraw = w;
    }

    @Override
    public float getTemp_w1() {
        // TODO Auto-generated method stub
        return temp_withdraw;
    }

    @Override
    public void setTemp_x1(String x) {
        // TODO Auto-generated method stub
        temp_x = x;
    }

    @Override
    public String getTemp_x1() {
        // TODO Auto-generated method stub
        return temp_x;
    }
}

//datastore for account2

public class ACCOUNT2_DataStore extends DataStore {
```



```
public static int p, y;//permanent
public static int deposit, withdraw, a;//balance;//permanent

public static int temp_p, temp_y, temp_x;//temp
public static int temp_deposit, temp_withdraw, temp_a;//temp

@Override
public void setTemp_p2(int p) {
    // TODO Auto-generated method stub
    ACCOUNT2_DataStore.temp_p = p;
}

@Override
public int getTemp_p2() {
    // TODO Auto-generated method stub
    return (temp_p);
}

@Override
public void set_p2(int p) {
    // TODO Auto-generated method stub
    ACCOUNT2_DataStore.p = p;
}

@Override
public int get_p2() {
    // TODO Auto-generated method stub
    return (p);
}

@Override
public void set_y2(int y) {
    // TODO Auto-generated method stub
    ACCOUNT2_DataStore.y = y;
}

@Override
public int get_y2() {
    // TODO Auto-generated method stub
    return y;
}

@Override
public void set_a2(int a) {
    // TODO Auto-generated method stub
    ACCOUNT2_DataStore.a = a;
}

@Override
public int get_a2() {
    // TODO Auto-generated method stub
    return a;
}
```

```
}

@Override
public void set_d2(int d) {
    // TODO Auto-generated method stub
    deposit = d;
}

@Override
public int get_d2() {
    // TODO Auto-generated method stub
    return deposit;
}

@Override
public void set_w2(int w) {
    // TODO Auto-generated method stub
    withdraw = w;
}

@Override
public int get_w2() {
    // TODO Auto-generated method stub
    return withdraw;
}

@Override
public void set_x2(int x) {
    // TODO Auto-generated method stub
    temp_x = x;
}

@Override
public int get_x2() {
    // TODO Auto-generated method stub
    return temp_x;
}

@Override
public void setTemp_y2(int y) {
    // TODO Auto-generated method stub
    temp_y = y;
}

@Override
public int getTemp_y2() {
    // TODO Auto-generated method stub
    return temp_y;
}

@Override
public void setTemp_a2(int a) {
```

```
        // TODO Auto-generated method stub
        temp_a = a;
    }

    @Override
    public int getTemp_a2() {
        // TODO Auto-generated method stub
        return temp_a;
    }

    @Override
    public void setTemp_d2(int d) {
        // TODO Auto-generated method stub
        temp_deposit = d;
    }

    @Override
    public int getTemp_d2() {
        // TODO Auto-generated method stub
        return temp_deposit;
    }

    @Override
    public void setTemp_w2(int w) {
        // TODO Auto-generated method stub
        temp_withdraw = w;
    }

    @Override
    public int getTemp_w2() {
        // TODO Auto-generated method stub
        return temp_withdraw;
    }

    @Override
    public void setTemp_x2(int x) {
        // TODO Auto-generated method stub
        temp_x = x;
    }

    @Override
    public int getTemp_x2() {
        // TODO Auto-generated method stub
        return temp_x;
    }

}

//State Pattern for Accounts classes:
```

```

abstract class Account_States {

    MDA_EFSM_Account create(){

        return(new MDA_EFSM_Account());

    }

    public Output_Processor output;

    Output_Processor create_output()
    {
        return(new Output_Processor());
    }

    abstract void open();
    abstract void login();
    abstract void incorrectLogin();
    abstract void incorectPin(int max);
    abstract void correctPinBelowMin();
    abstract void correctPinAboveMin();
    abstract void deposit();
    abstract void belowMinBalance();
    abstract void aboveMinBalance();
    abstract void logout();
    abstract void balance();

    abstract void withdraw();

    abstract void withdrawBelowMinBalance();
    abstract void noFunds();
    abstract void lock();
    abstract void incorrectLock();
    abstract void unLock();
    abstract void incorrectUnLock();
    abstract void suspend();
    abstract void activitate();
    abstract void close();

    abstract int get_id();

}

//States:

//start

public class Start extends Account_States{

    @Override
    void open() {
        // TODO Auto-generated method stub
        create_output().i_storePin();
    }

    @Override

```

```
void login() {  
    // TODO Auto-generated method stub  
  
}  
  
@Override  
void incorrectLogin() {  
    // TODO Auto-generated method stub  
  
}  
  
@Override  
void incorrectPin(int max) {  
    // TODO Auto-generated method stub  
  
}  
  
@Override  
void correctPinBelowMin() {  
    // TODO Auto-generated method stub  
  
}  
  
@Override  
void correctPinAboveMin() {  
    // TODO Auto-generated method stub  
  
}  
  
@Override  
void deposit() {  
    // TODO Auto-generated method stub  
  
}  
  
@Override  
void belowMinBalance() {  
    // TODO Auto-generated method stub  
  
}  
  
@Override  
void aboveMinBalance() {  
    // TODO Auto-generated method stub  
  
}  
  
@Override  
void logout() {  
    // TODO Auto-generated method stub  
  
}  
  
@Override  
void balance() {  
    // TODO Auto-generated method stub
```

```
    }

    @Override
    void withdraw() {
        // TODO Auto-generated method stub
    }

    @Override
    void withdrawBelowMinBalance() {
        // TODO Auto-generated method stub
    }

    @Override
    void noFunds() {
        // TODO Auto-generated method stub
    }

    @Override
    void lock() {
        // TODO Auto-generated method stub
    }

    @Override
    void incorrectLock() {
        // TODO Auto-generated method stub
    }

    @Override
    void unLock() {
        // TODO Auto-generated method stub
    }

    @Override
    void incorrectUnLock() {
        // TODO Auto-generated method stub
    }

    @Override
    void suspend() {
        // TODO Auto-generated method stub
    }

    @Override
    void activitate() {
        // TODO Auto-generated method stub
    }

    @Override
```

```
void close() {
    // TODO Auto-generated method stub

}

@Override
int get_id() {
    // TODO Auto-generated method stub
    return 0;
}

}

//idle

public class Idle extends Account_States{

    @Override
    void open() {
        // TODO Auto-generated method stub

    }

    @Override
    void login() {
        // TODO Auto-generated method stub
        create().set_count(0);
        create_output().i_promptForPin();

    }

    @Override
    void incorrectLogin() {
        // TODO Auto-generated method stub

        //create_output().i_incorrectPinMsg();
        create_output().i_incorrectMsg();

    }

    @Override
    void incorrectPin(int max) {
        // TODO Auto-generated method stub

    }

    @Override
    void correctPinBelowMin() {
        // TODO Auto-generated method stub

    }

    @Override
    void correctPinAboveMin() {
        // TODO Auto-generated method stub
```

```
    }

    @Override
    void deposit() {
        // TODO Auto-generated method stub
    }

    @Override
    void belowMinBalance() {
        // TODO Auto-generated method stub
    }

    @Override
    void aboveMinBalance() {
        // TODO Auto-generated method stub
    }

    @Override
    void logout() {
        // TODO Auto-generated method stub
    }

    @Override
    void balance() {
        // TODO Auto-generated method stub
    }

    @Override
    void withdraw() {
        // TODO Auto-generated method stub
    }

    @Override
    void withdrawBelowMinBalance() {
        // TODO Auto-generated method stub
    }

    @Override
    void noFunds() {
        // TODO Auto-generated method stub
    }

    @Override
    void lock() {
        // TODO Auto-generated method stub
    }

    @Override
```



```
    void incorrectLock() {
        // TODO Auto-generated method stub

    }

    @Override
    void unLock() {
        // TODO Auto-generated method stub

    }

    @Override
    void incorrectUnLock() {
        // TODO Auto-generated method stub

    }

    @Override
    void suspend() {
        // TODO Auto-generated method stub

    }

    @Override
    void activitate() {
        // TODO Auto-generated method stub

    }

    @Override
    void close() {
        // TODO Auto-generated method stub

    }

    @Override
    int get_id() {
        // TODO Auto-generated method stub
        return (1);
    }

}

public class Ready extends Account_States{

    @Override
    void open() {
        // TODO Auto-generated method stub

    }

    @Override
    void login() {
        // TODO Auto-generated method stub

    }

}
```

```
@Override
void incorrectLogin() {
    // TODO Auto-generated method stub

}

@Override
void incorrectPin(int max) {
    // TODO Auto-generated method stub

}

@Override
void correctPinBelowMin() {
    // TODO Auto-generated method stub

}

@Override
void correctPinAboveMin() {
    // TODO Auto-generated method stub

}

@Override
void deposit() {
    // TODO Auto-generated method stub
    create_output().i_makeDeposit();
}

@Override
void belowMinBalance() {
    // TODO Auto-generated method stub
    create_output().i_displayBalance();
}

@Override
void aboveMinBalance() {
    // TODO Auto-generated method stub

}

@Override
void logout() {
    // TODO Auto-generated method stub

}

@Override
void balance() {
    // TODO Auto-generated method stub
    create_output().i_displayBalance();
}

@Override
```

```
void withdraw() {
    // TODO Auto-generated method stub
    create_output().i_makeWithdraw();
}

@Override
void withdrawBelowMinBalance() {
    // TODO Auto-generated method stub
}

@Override
void noFunds() {
    // TODO Auto-generated method stub
    create_output().i_noFundsMsg();
}

@Override
void lock() {
    // TODO Auto-generated method stub
}

@Override
void incorrectLock() {
    // TODO Auto-generated method stub
    create_output().i_incorrectLockMsg();
}

@Override
void unlock() {
    // TODO Auto-generated method stub
}

@Override
void incorrectUnLock() {
    // TODO Auto-generated method stub
}

@Override
void suspend() {
    // TODO Auto-generated method stub
}

@Override
void activitate() {
    // TODO Auto-generated method stub
}

@Override
void close() {
    // TODO Auto-generated method stub
}
```

```
    }

    @Override
    int get_id() {
        // TODO Auto-generated method stub
        return (3);
    }
}

public class Stop extends Account_States{

    @Override
    void open() {
        // TODO Auto-generated method stub

    }

    @Override
    void login() {
        // TODO Auto-generated method stub

    }

    @Override
    void incorrectLogin() {
        // TODO Auto-generated method stub

    }

    @Override
    void incorrectPin(int max) {
        // TODO Auto-generated method stub

    }

    @Override
    void correctPinBelowMin() {
        // TODO Auto-generated method stub

    }

    @Override
    void correctPinAboveMin() {
        // TODO Auto-generated method stub

    }

    @Override
    void deposit() {
        // TODO Auto-generated method stub

    }

    @Override
    void belowMinBalance() {
```

```
        // TODO Auto-generated method stub

    }

    @Override
    void aboveMinBalance() {
        // TODO Auto-generated method stub

    }

    @Override
    void logout() {
        // TODO Auto-generated method stub

    }

    @Override
    void balance() {
        // TODO Auto-generated method stub

    }

    @Override
    void withdraw() {
        // TODO Auto-generated method stub

    }

    @Override
    void withdrawBelowMinBalance() {
        // TODO Auto-generated method stub

    }

    @Override
    void noFunds() {
        // TODO Auto-generated method stub

    }

    @Override
    void lock() {
        // TODO Auto-generated method stub

    }

    @Override
    void incorrectLock() {
        // TODO Auto-generated method stub

    }

    @Override
    void unLock() {
        // TODO Auto-generated method stub

    }
```

```

@Override
void incorrectUnLock() {
    // TODO Auto-generated method stub

}

@Override
void suspend() {
    // TODO Auto-generated method stub

}

@Override
void activitate() {
    // TODO Auto-generated method stub

}

@Override
void close() {
    // TODO Auto-generated method stub

}

@Override
int get_id() {
    // TODO Auto-generated method stub
    return (8);
}

}

public class Suspended extends Account_States{

@Override
void open() {
    // TODO Auto-generated method stub

}

@Override
void login() {
    // TODO Auto-generated method stub

}

@Override
void incorrectLogin() {
    // TODO Auto-generated method stub

}

@Override
void incorrectPin(int max) {

```

```
        // TODO Auto-generated method stub

    }

    @Override
    void correctPinBelowMin() {
        // TODO Auto-generated method stub

    }

    @Override
    void correctPinAboveMin() {
        // TODO Auto-generated method stub

    }

    @Override
    void deposit() {
        // TODO Auto-generated method stub

    }

    @Override
    void belowMinBalance() {
        // TODO Auto-generated method stub

    }

    @Override
    void aboveMinBalance() {
        // TODO Auto-generated method stub

    }

    @Override
    void logout() {
        // TODO Auto-generated method stub

    }

    @Override
    void balance() {
        // TODO Auto-generated method stub
        create_output().i_displayBalance();
    }

    @Override
    void withdraw() {
        // TODO Auto-generated method stub

    }

    @Override
    void withdrawBelowMinBalance() {
        // TODO Auto-generated method stub

    }
```

```
@Override
void noFunds() {
    // TODO Auto-generated method stub

}

@Override
void lock() {
    // TODO Auto-generated method stub

}

@Override
void incorrectLock() {
    // TODO Auto-generated method stub

}

@Override
void unlock() {
    // TODO Auto-generated method stub

}

@Override
void incorrectUnlock() {
    // TODO Auto-generated method stub

}

@Override
void suspend() {
    // TODO Auto-generated method stub

}

@Override
void activitate() {
    // TODO Auto-generated method stub

}

@Override
void close() {
    // TODO Auto-generated method stub

}

@Override
int get_id() {
    // TODO Auto-generated method stub
    return (7);
}

}
```



```
public class Overdrawn extends Account_States{

    @Override
    void open() {
        // TODO Auto-generated method stub
    }

    @Override
    void login() {
        // TODO Auto-generated method stub
    }

    @Override
    void incorrectLogin() {
        // TODO Auto-generated method stub
    }

    @Override
    void incorrectPin(int max) {
        // TODO Auto-generated method stub
    }

    @Override
    void correctPinBelowMin() {
        // TODO Auto-generated method stub
    }

    @Override
    void correctPinAboveMin() {
        // TODO Auto-generated method stub
    }

    @Override
    void deposit() {
        // TODO Auto-generated method stub
        create_output().i_makeDeposit();
    }

    @Override
    void belowMinBalance() {

    }

    @Override
    void aboveMinBalance() {
        // TODO Auto-generated method stub
    }
}
```

```
@Override
void logout() {
    // TODO Auto-generated method stub

}

@Override
void balance() {
    // TODO Auto-generated method stub
    create_output().i_displayBalance();
}

@Override
void withdraw() {
    // TODO Auto-generated method stub
    create_output().i_noFundsMsg();
}

@Override
void withdrawBelowMinBalance() {

}

@Override
void noFunds() {
    // TODO Auto-generated method stub
}

@Override
void lock() {
    // TODO Auto-generated method stub
}

@Override
void incorrectLock() {
    // TODO Auto-generated method stub
}

@Override
void unLock() {
    // TODO Auto-generated method stub
}

@Override
void incorrectUnLock() {
    // TODO Auto-generated method stub
}

@Override
void suspend() {
    // TODO Auto-generated method stub
}
```

```
    }

    @Override
    void activitate() {
        // TODO Auto-generated method stub
    }

    @Override
    void close() {
        // TODO Auto-generated method stub
    }

    @Override
    int get_id() {
        // TODO Auto-generated method stub
        return (5);
    }
}

public class Locked extends Account_States{

    @Override
    void open() {
        // TODO Auto-generated method stub
    }

    @Override
    void login() {
        // TODO Auto-generated method stub
    }

    @Override
    void incorrectLogin() {
        // TODO Auto-generated method stub
    }

    @Override
    void incorrectPin(int max) {
        // TODO Auto-generated method stub
    }

    @Override
    void correctPinBelowMin() {
        // TODO Auto-generated method stub
    }
}
```

```
@Override
void correctPinAboveMin() {
    // TODO Auto-generated method stub

}

@Override
void deposit() {
    // TODO Auto-generated method stub

}

@Override
void belowMinBalance() {
    // TODO Auto-generated method stub

}

@Override
void aboveMinBalance() {
    // TODO Auto-generated method stub

}

@Override
void logout() {
    // TODO Auto-generated method stub

}

@Override
void balance() {
    // TODO Auto-generated method stub

}

@Override
void withdraw() {
    // TODO Auto-generated method stub

}

@Override
void withdrawBelowMinBalance() {
    // TODO Auto-generated method stub

}

@Override
void noFunds() {
    // TODO Auto-generated method stub

}

@Override
void lock() {
    // TODO Auto-generated method stub
```

```

    }

    @Override
    void incorrectLock() {
        // TODO Auto-generated method stub
    }

    @Override
    void unlock() {
        // TODO Auto-generated method stub
    }

    @Override
    void incorrectUnlock() {
        // TODO Auto-generated method stub
        create_output().i_incorrectUnlockMsg();
    }

    @Override
    void suspend() {
        // TODO Auto-generated method stub
    }

    @Override
    void activitate() {
        // TODO Auto-generated method stub
    }

    @Override
    void close() {
        // TODO Auto-generated method stub
    }

    @Override
    int get_id() {
        // TODO Auto-generated method stub
        return (6);
    }
}

public class S1 extends Account_States{

    @Override
    void open() {
        // TODO Auto-generated method stub
    }

    @Override

```

```
void login() {  
    // TODO Auto-generated method stub  
  
}  
  
@Override  
void incorrectLogin() {  
    // TODO Auto-generated method stub  
  
}  
  
@Override  
void incorrectPin(int max) {  
    // TODO Auto-generated method stub  
  
}  
  
@Override  
void correctPinBelowMin() {  
    // TODO Auto-generated method stub  
  
}  
  
@Override  
void correctPinAboveMin() {  
    // TODO Auto-generated method stub  
  
}  
  
@Override  
void deposit() {  
    // TODO Auto-generated method stub  
  
}  
  
@Override  
void belowMinBalance() {  
  
}  
  
@Override  
void aboveMinBalance() {  
    // TODO Auto-generated method stub  
  
}  
  
@Override  
void logout() {  
    // TODO Auto-generated method stub  
  
}  
  
@Override  
void balance() {  
    // TODO Auto-generated method stub  
    create_output().i_displayBalance();  
}
```

```
    }

    @Override
    void withdraw() {
        // TODO Auto-generated method stub
        create_output().i_makeWithdraw();
    }

    @Override
    void withdrawBelowMinBalance() {
        // TODO Auto-generated method stub
        create_output().i_penalty();
    }

    @Override
    void noFunds() {
        // TODO Auto-generated method stub
    }

    @Override
    void lock() {
        // TODO Auto-generated method stub
    }

    @Override
    void incorrectLock() {
        // TODO Auto-generated method stub
    }

    @Override
    void unlock() {
        // TODO Auto-generated method stub
    }

    @Override
    void incorrectUnlock() {
        create_output().i_incorrectUnlockMsg();
    }

    @Override
    void suspend() {
        // TODO Auto-generated method stub
    }

    @Override
    void activitate() {
        // TODO Auto-generated method stub
    }
}
```

```

@Override
void close() {
    // TODO Auto-generated method stub

}

@Override
int get_id() {
    // TODO Auto-generated method stub
    return (4);
}

}

public class Check_Pin extends Account_States{

@Override
void open() {
    // TODO Auto-generated method stub

}

@Override
void login() {
    // TODO Auto-generated method stub

}

@Override
void incorrectLogin() {
    // TODO Auto-generated method stub

}

@Override
void incorrectPin(int max) {
    // TODO Auto-generated method stub
    create_output().i_incorrectMsg();

    if(max > create().get_count()){

        create().set_count(create().get_count()+1);

    }
    if(max ==create().get_count()){
        create_output().i_tooManyAttemptsMsg();
    }

}

@Override
void correctPinBelowMin() {

```



```
        // TODO Auto-generated method stub
        create_output().i_displayMenu();
    }

    @Override
    void correctPinAboveMin() {
        // TODO Auto-generated method stub

        create_output().i_displayMenu();
    }

    @Override
    void deposit() {
        // TODO Auto-generated method stub
    }

    @Override
    void belowMinBalance() {
        // TODO Auto-generated method stub
    }

    @Override
    void aboveMinBalance() {
        // TODO Auto-generated method stub
    }

    @Override
    void logout() {
        // TODO Auto-generated method stub
    }

    @Override
    void balance() {
        // TODO Auto-generated method stub
    }

    @Override
    void withdraw() {
        // TODO Auto-generated method stub
    }

    @Override
    void withdrawBelowMinBalance() {
        // TODO Auto-generated method stub
    }

    @Override
    void noFunds() {
        // TODO Auto-generated method stub
    }
}
```

```
@Override
void lock() {
    // TODO Auto-generated method stub

}

@Override
void incorrectLock() {
    // TODO Auto-generated method stub

}

@Override
void unLock() {
    // TODO Auto-generated method stub

}

@Override
void incorrectUnLock() {
    // TODO Auto-generated method stub

}

@Override
void suspend() {
    // TODO Auto-generated method stub

}

@Override
void activitate() {
    // TODO Auto-generated method stub

}

@Override
void close() {
    // TODO Auto-generated method stub

}

@Override
int get_id() {
    // TODO Auto-generated method stub
    return (2);
}

}

//Strategy patterns
```

```
abstract class DisplayBalance {

    abstract void i_displayBalance(DataStore ds);

}

public class DisplayBalance_Account2 extends DisplayBalance {

    @Override
    public void i_displayBalance(DataStore ds) {
        // TODO Auto-generated method stub
        System.out.println("Balance: $" + ds.get_a2());
    }

}

public class DisplayBalance_Account1 extends DisplayBalance {

    @Override
    public void i_displayBalance(DataStore ds) {
        // TODO Auto-generated method stub
        System.out.println("Balance: $" + ds.get_a1());
    }

}

abstract class DisplayMenu {

    abstract void i_displayMenu();

}

public class DisplayMenu_Account2 extends DisplayMenu {

    @Override
    public void i_displayMenu() {
        // TODO Auto-generated method stub
        System.out.println("Message: Display menu for Account2");
    }

}

public class DisplayMenu_Account1 extends DisplayMenu {
```

```
@Override
public void i_displayMenu() {
    // TODO Auto-generated method stub
    System.out.println("Message: Display menu for Account1");
}

}

abstract class StorePin {

    abstract void i_storePin(DataStore ds);

}

public class StorePin_Account1 extends StorePin {

    @Override
    public void i_storePin(DataStore ds) {
        // TODO Auto-generated method stub
        ds.set_p1(ds.getTemp_p1());
        ds.set_y1(ds.getTemp_y1());
        ds.set_a1(ds.getTemp_a1());
    }

}

public class StorePin_Account2 extends StorePin {

    @Override
    public void i_storePin(DataStore ds) {
        // TODO Auto-generated method stub
        ds.set_p2(ds.getTemp_p2());
        ds.set_y2(ds.getTemp_y2());
        ds.set_a2(ds.getTemp_a2());
    }

}

abstract class IncorrectLockMsg {
```

```
        abstract void i_incorrectLockMsg();
    }

    public class IncorrectLockMsg_Account2 extends IncorrectLockMsg {

        @Override
        public void i_incorrectLockMsg() {
            // TODO Auto-generated method stub
            System.out.println("Message: Incorrect Lock operation for
Account2");
        }

    }

    public class IncorrectLockMsg_Account1 extends IncorrectLockMsg {

        @Override
        public void i_incorrectLockMsg() {
            // TODO Auto-generated method stub
            System.out.println("Message: Incorrect Lock operation for
Account1");
        }

    }

    abstract class IncorrectMsg {

        abstract void i_incorrectMsg();

    }

    public class IncorrectMsg_Account2 extends IncorrectMsg {

        @Override
        public void i_incorrectMsg() {
            // TODO Auto-generated method stub
            System.out.println("Message: Incorrect ID for Account2");
        }

    }

    public class IncorrectMsg_Account1 extends IncorrectMsg {

        @Override
        public void i_incorrectMsg() {
            // TODO Auto-generated method stub
            System.out.println("Message: Incorrect ID for Account1");
        }

    }
```

```
}

abstract class IncorrectPinMsg {

    abstract void i_incorrectPinMsg();

}

public class IncorrectPinMsg_Account2 extends IncorrectPinMsg {

    @Override
    public void i_incorrectPinMsg() {
        // TODO Auto-generated method stub
        System.out.println("Message: Incorrect PIN for Account2");
    }

}

public class IncorrectPinMsg_Account1 extends IncorrectPinMsg {

    @Override
    public void i_incorrectPinMsg() {
        // TODO Auto-generated method stub
        System.out.println("Message: Incorrect PIN for Account1");
    }

}

abstract class IncorrectUnLockMsg {

    abstract void i_incorrectUnLockMsg();

}

public class IncorrectUnLockMsg_Account2 extends IncorrectUnLockMsg {

    @Override
    public void i_incorrectUnLockMsg() {
        // TODO Auto-generated method stub
        System.out.println("Message: Incorrect Unlock operation for
Account2");
    }

}

public class IncorrectUnLockMsg_Account1 extends IncorrectUnLockMsg {

    @Override
    public void i_incorrectUnLockMsg() {
        // TODO Auto-generated method stub
```

```
        System.out.println("Message: Incorrect Unlock operation for
Account1");
    }
}

abstract class MakeDeposit {
    abstract void i_makeDeposit(DataStore ds);
}

public class MakeDeposit_Account1 extends MakeDeposit {
    @Override
    public void i_makeDeposit(DataStore ds) {
        // TODO Auto-generated method stub
        ds.set_a1(ds.getTemp_d1() + ds.get_a1());
    }
}

public class MakeDeposit_Account2 extends MakeDeposit {
    @Override
    public void i_makeDeposit(DataStore ds) {
        // TODO Auto-generated method stub
        ds.set_a2(ds.getTemp_d2() + ds.get_a2());
    }
}

//

abstract class MakeWithdraw {
    abstract void i_makeWithdraw(DataStore ds);
}

public class MakeWithdraw_Account2 extends MakeWithdraw {
    @Override
    public void i_makeWithdraw(DataStore ds) {
        // TODO Auto-generated method stub
        ds.set_a2(ds.get_a2() - ds.getTemp_w2());
    }
}
```

```
public class MakeWithdraw_Account1 extends MakeWithdraw {

    @Override
    public void i_makeWithdraw(DataStore ds) {
        // TODO Auto-generated method stub
        ds.set_a1(ds.get_a1() - ds.getTemp_w1());
    }

}

//

abstract class NoFundsMsg {

    abstract void i_noFundsMsg();

}

public class NoFundsMsg_Account2 extends NoFundsMsg {

    @Override
    public void i_noFundsMsg() {
        // TODO Auto-generated method stub
        System.out.println("Message: No more funds left for Account2");
    }

}

public class NoFundsMsg_Account1 extends NoFundsMsg {

    @Override
    public void i_noFundsMsg() {
        // TODO Auto-generated method stub
        System.out.println("Message: No funds left for Account1");
    }

}

abstract class Penalty {

    abstract void i_penalty(DataStore ds);

}

public class Penalty_Account2 extends Penalty {

    @Override
    public void i_penalty(DataStore ds) {
        // TODO Auto-generated method stub
        System.out.println("Penalty Applied of 20 $");
    }

}
```



```

        ds.set_a2(ds.get_a2() - 20);
    }

}

public class Penalty_Account1 extends Penalty {

    @Override
    public void i_penalty(DataStore ds) {
        // TODO Auto-generated method stub
        System.out.println("Penalty Applied of 20 $");
        ds.set_a1(ds.get_a1() - 20);
    }

}

abstract class PromptForPin {

    abstract void i_promptForPin();

}

public class PromptForPin_Account2 extends PromptForPin {

    @Override
    public void i_promptForPin() {
        // TODO Auto-generated method stub
        System.out.println("Message: Enter the Pin for Account2");
    }

}

public class PromptForPin_Account1 extends PromptForPin {

    @Override
    public void i_promptForPin() {
        // TODO Auto-generated method stub
        System.out.println("Message: Enter the Pin for Account1");
    }

}

abstract class TooManyAttemptsMsg {

    abstract void i_incorrectPinMsg();

}

public class TooManyAttemptsMsg_Account2 extends TooManyAttemptsMsg {

    @Override
    public void i_incorrectPinMsg() {
        // TODO Auto-generated method stub
        System.out.println("Message: Too many attempts for Account2");
    }

}

```

```

    }

}

public class TooManyAttemptsMsg_Account1 extends TooManyAttemptsMsg {

    @Override
    public void i_incorrectPinMsg() {
        // TODO Auto-generated method stub
        System.out.println("Message: Too many attempts for Account1");
    }

}

//Output Processor

public class Output_Processor {

    public static DataStore ds;
    public static StorePin sp;
    public static IncorrectMsg iim;
    public static IncorrectPinMsg ip;
    public static TooManyAttemptsMsg tmam;
    public static DisplayMenu dm;
    public static MakeDeposit md;
    public static DisplayBalance db;
    public static PromptForPin pfp;
    public static MakeWithdraw mw;
    public static Penalty p;
    public static IncorrectLockMsg ilm;
    public static IncorrectUnLockMsg iulm;
    public static NoFundsMsg nfm;

    void initialize(Abstract_Factory af){

        ds = af.get_datastore();
        sp = af.storePin();
        iim = af.incorrectMsg();
        ip = af.incorrectPinMsg();
        tmam = af.tooManyAttemptsMsg();
        dm = af.displayMenu();
        md = af.makeDeposit();
        db = af.displayBalance();
        pfp = af.promptForPin();
        mw = af.makeWithdraw();
        p = af.penalty();
        ilm = af.incorrectLockMsg();
        iulm = af.incorrectUnLockMsg();
        nfm = af.noFundsMsg();
    }
}

```

```
    }

    DataStore get_datastore() {

        return(ds);

    }

    public void i_storePin() {

        sp.i_storePin(ds);

    }

    void i_incorrectMsg(){

        iim.i_incorrectMsg();

    }

    void i_incorrectPinMsg(){

        ip.i_incorrectPinMsg();

    }

    void i_tooManyAttemptsMsg(){

        tmam.i_incorrectPinMsg();

    }

    void i_displayMenu(){

        dm.i_displayMenu();

    }

    void i_makeDeposit(){

        md.i_makeDeposit(ds);

    }

    void i_displayBalance(){

        db.i_displayBalance(ds);

    }

    void i_promptForPin(){

        pfp.i_promptForPin();

    }
```

```
void i_makeWithdraw() {  
    mw.i_makeWithdraw(ds);  
}  
  
void i_penalty() {  
    p.i_penalty(ds);  
}  
  
void i_incorrectLockMsg() {  
    ilm.i_incorrectLockMsg();  
}  
  
void i_incorrectUnLockMsg() {  
    iulm.i_incorrectUnLockMsg();  
}  
  
void i_noFundsMsg() {  
    nfm.i_noFundsMsg();  
}  
  
}
```