MDA-EFSM Events:
Open()
Login()
IncorrectLogin()
IncorectPin(int max)
CorrectPinBelowMin()
CorrectPinAboveMin()
Deposit()
BelowMinBalance()
AboveMinBalance()
Logout()
Balance()
Withdraw()
WithdrawBelowMinBalance()
NoFunds()
Lock()
IncorrectLock()
Unlock()
IncorrectUnlock()
Suspend()
Activate()
Close()

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

Operations of the Input Processor (ACCOUNT-1)

```
open (string p, string y, float a) {
// store p, y and a in temp data store
        ds->temp_p=p;
        ds->temp_y=y;
        ds->temp_a=a;
        m->Open();
}

pin (string x) {
        if (x==ds->pin) {
             if (d->balance > 500)
                  m->CorrectPinAboveMin ();
             else m->CorrectPinBelowMin();
        }
        else m->IncorrectPin(3)
}


deposit (float d) {
        ds->temp_d=d;
        m->Deposit();
        if (ds->balance>500)
             m->AboveMinBalance();
        else m->BelowMinBalance();
}

withdraw (float w) {
        ds->temp_w=w;
        m->withdraw();
        if ((ds->balance>500)
              m->AboveMinBalance();
        else m->WithdrawBelowMinBalance();
}
```

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

login (string y) {
        if (y==ds->uid)
             m->Login();
        else m->IncorrectLogin();
}

logout() {m->Logout();}

lock (string x) {
        if (ds->pin==x) m->Lock();
        else m->IncorrectLock();
}

unlock (string x) {
        if (x==ds->pin) {
             m->Unlock();
             if (ds->balance > 500)
                  m->AboveMinBalance ();
             else m->BelowMinBalance();
        }
        else m->IncorrectUnlock();
}
```

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* are used to store values of parameters

Operations of the Input Processor (ACCOUNT-2)

```
OPEN (int p, int y, int a) {
// store p, y and a in temp data store
        ds->temp_p=p;
        ds->temp_y=y;
        ds->temp_a=a;
        m->Open();
}

PIN (int x) {
        if (x==ds->pin)
             m->CorrectPinAboveMin ();
        else m->IncorrectPin(2)
}


DEPOSIT (int d) {
        ds->temp_d=d;
        m->Deposit();
}

WITHDRAW (int w) {
        ds->temp_w=w;
        if (ds->balance>0)
              m->Withdraw();
        else m->NoFunds();
}


BALANCE() {m->Balance();}

LOGIN (int y) {
        if (y==ds->uid)
             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* are used to store values of parameters