MDA-EFSM Events:
create()
card()
IncorectPin(int max)
CorrectPinBelowMin()
CorrectPinAboveMin()
deposit()
BelowMinBalance()
AboveMinBalance()
exit()
balance()
withdraw()
lock()
unlock

MDA-EFSM Actions:

store_pin	// stores pin from temporary data store to *pin* in data store
store_balance	// stores pin from temporary data store to *b* in data store
prompt_for_PIN	// prompts to enter pin
display_menu	// display a menu with a list of transactions
incorrect_pin_msg	// displays incorrect pin message
too_many_attempts_msg	// display too many attempts message
MakeDeposit	// makes deposit (increases balance by a value stored in temp. data store)
MakeWithdraw	// makes withdraw (decreases balance by a value stored in temp. data store)
Penalty	// applies penalty (decreases balance by the amount of penalty)
eject_card	// ejects the card
DisplayBalance	// displays the current value of the balance

Operations of the Input Processor
(ATM-1)

```
create() {m->create();}

card (int x, string y) {
        d->temp_x=x;
        d->temp_y=y;
        m->card();
}

deposit (int d) {
        d->temp_d=d;
        m->deposit();
        if (d->b < 1000)
              m->BelowMinBalance();
        else m->AboveMinBalance();
}

withdraw (int w) {
        d->temp_w=w;
        if ((d->b-w) > 0) m->withdraw();
        if (d->b<1000)
              m->BelowMinBalance();
        else m->AboveMinBalance();
}

pin (string x) {
        if (x==d->pn) {
              if (d->b<1000)
                    m->CorrectPinBelowMin ();
              else m->CorrectPinAboveMin();
        }
        else m->IncorrectPin(3)
}
```

```
exit() {m->exit();}

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

lock (string x) {
        if (d->pn==x) m->lock();
}

unlock (string x) {
        if (x==d->pn) {
              m->unlock();
              if (d->b<1000)
                    m->BelowMinBalance ();
              else m->AboveMinBalance();
        }
}
```

Notice:
m: pointer to the MDA-EFSM
d: pointer to the data store
In the data store:
b: contains the current balance
pn: contains the correct pin #

Operations of the Input Processor (ATM-2)

```
create() {m->create();}

CARD (float x, int y) {
        d->temp_x=x;
        d->temp_y=y;
        m->card();
}

DEPOSIT (float d) {
        d->temp_d=d;
        m->deposit();
        if (d->b<500)
              m->BelowMinBalance();
        else m->AboveMinBalance();
}

WITHDRAW (float w) {
        d->temp_w=w;
        if ((d->b-w) > 0) m->withdraw();
        if (d->b<500)
              m->BelowMinBalance();
        else m->AboveMinBalance();
}

PIN (int x) {
        if (x==d->pn) {
              if (d->b<500)
                    m->CorrectPinBelowMin ();
              else m->CorrectPinAboveMin();
        }
        else m->IncorrectPin(2)
}

EXIT() {m->exit();}

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

Notice:
m: pointer to the MDA-EFSM
d: pointer to the data store
In the data store:
b: contains the current balance
pn: contains the correct pin #