

## Fall 2013

### CS589 Project Report

Xiaoyang Lu

A20315603

#### 1. Introduction

This project is based on an account class which exhibited states behavior by the EFSM model. To test this class, we will use model-based testing, default transition testing, and multiple-condition. In this account class it has simple operation of login, pin, deposit, withdraw, balance and so on. We will design test cases to test every transition-pair, default transition and every multiple-condition in the statement. Finally, we will execute all the test cases. After get the results from test cases, we will find the defects of the program and show the reason of it.

#### 2. Model-base testing of the *account* class

There are five states in total:

##### Transition pairs for **idle**

Input: T1, T5, T6, T7, T9, T10

Output: T2, T7

(T1, T2)	(T1, T7)	(T5, T2)	(T5, T7)	(T6, T2)	(T6, T7)
(T7, T2)	(T7, T7)	(T9, T2)	(T9, T7)	(T10, T2)	(T10, T7)

##### Transition pairs for **check pin**

Input: T2, T3

Output: T3, T4, T5, T6, T8

(T2, T3)	(T2, T4)	(T2, T5)	<del>(T2, T6)</del>	(T2, T8)
<del>(T3, T3)</del>	(T3, T4)	(T3, T5)	(T3, T6)	(T3, T8)

##### Transition pairs for **ready**

Input: T4, T11, T12, T13, T15, T17

Output: T10, T11, T12, T13, T14, T16

(T4, T10)	(T4, T11)	(T4, T12)	(T4, T13)	(T4, T14)	(T4, T16)
(T11, T10)	(T11, T11)	(T11, T12)	(T11, T13)	(T11, T14)	(T11, T16)
(T12, T10)	(T12, T11)	(T12, T12)	(T12, T13)	(T12, T14)	(T12, T16)
(T13, T10)	(T13, T11)	(T13, T12)	(T13, T13)	(T13, T14)	(T13, T16)
(T15, T10)	(T15, T11)	(T15, T12)	(T15, T13)	(T15, T14)	(T15, T16)
(T17, T10)	(T17, T11)	(T17, T12)	(T17, T13)	(T17, T14)	(T17, T16)

##### Transition pairs for **locked**

Input: T16, T18, T20

Output: T17, T18, T19

(T16, T17)	(T16, T18)	<del>(T16, T19)</del>	(T18, T17)	(T18, T18)	(T18, T19)
<del>(T20, T17)</del>	(T20, T18)	(T20, T19)			

**Transition pairs for overdraw**

Input: T8, T14, T19, T21, T22

Output: T9, T15, T20, T21, T22

(T8, T9)	(T8, T15)	(T8, T20)	(T8, T21)	(T8, T22)
(T14, T9)	(T14, T15)	(T14, T20)	(T14, T21)	(T14, T22)
(T19, T9)	(T19, T15)	(T19, T20)	(T19, T21)	(T19, T22)
(T21, T9)	(T21, T15)	(T21, T20)	(T21, T21)	(T21, T22)
(T22, T9)	(T22, T15)	(T22, T20)	(T22, T21)	(T22, T22)

### Test case

**Test#1:** open 1500 123 111 login 455 login 111 pin 123 deposit 50 balance lock 123  
unlock 123 logout

**Coverage of transitions:** T1, T7, T2, T4, T12, T13, T16, T17, T10

**Transition pairs:** (T1, T7), (T7, T2), (T2, T4), (T4, T12), (T12, T13), (T13, T16),  
(T16, T17), (T17, T10) are executed.

**Test#2:** open 500 111 112 login 112 pin 113 pin 111 deposit 300 balance lock 111  
unlock 111 deposit 300 balance logout

**Coverage of transitions:** T1, T2, T3, T8, T21, T22, T20, T19, T15, T13, T10

**Transition pairs:** (T1, T2), (T2, T3), (T3, T8), (T8, T21), (T21, T22), (T22, T20),  
(T20, T19), (T19, T15), (T13, T10) are executed

**Test#3:** open 1200 234 113 login 113 pin 235 logout login 113 pin 236 pin 234  
balance withdraw 100 balance lock 234 unlock 234 withdraw 200 balance  
logout

**Coverage of transitions:** T1, T2, T3, T5, T2, T4, T3, T4, T13, T11, T13, T16, T17,  
T14, T22, T9

**Transition pairs:** (T1, T2), (T2, T3), (T3, T5), (T5, T2), (T2, T4), (T4, T3), (T3, T4),  
(T4, T13), (T13, T16), (T16, T17), (T17, T14), (T14, T22),  
(T22, T9) are executed

**Test#4:** open 300 222 114 login 113 login 115 login 114 logout login 115 login 114  
pin 223 pin 221 login 114 pin 222 logout

**Coverage of transitions:** T1, T7, T7, T2, T5, T7, T2, T3, T6, T2, T8, T9

**Transition pairs:** (T1, T7), (T7, T7), (T7, T2), (T2, T5), (T5, T7), (T7, T2), (T2, T3), (T3, T6), (T6, T2), (T2, T8), (T8, T9) are executed

**Test#5:** open 500 333 115 login 115 pin 333 balance deposit 300 balance balance  
deposit 300 withdraw 200 logout

**Coverage of transitions:** T1, T2, T8, T22, T21, T22, T22, T15, T14, T9

**Transition pairs:** (T1, T2), (T2, T8), (T8, T22), (T22, T21), (T21, T22), (T22, T22), (T22, T15), (T15, T14), (T14, 9) are executed

**Test#6:** open 2000 444 116 login 116 pin 444 logout login 116 pin 444 withdraw 500  
withdraw 300 lock 444 balance unlock 444 deposit 1000 logout

**Coverage of transitions:** T1, T2, T4, T10, T2, T4, T11, T11, T16, T18, T17, T12,  
T10

**Transition pairs:** (T1, T2), (T2, T4), (T4, T10), (T10, T2), (T2, T4), (T4, T11), (T11, T16), (T16, T18), (T18, T17), (T17, T12), (T12, T10) are executed

**Test#7:** open 900 789 117 login 117 pin 789 deposit 300 deposit 200 lock 789 unlock  
789 balance balance logout login 113

**Coverage of transitions:** T1, T2, T8, T15, T12, T16, T17, T13, T13, T10, T7

**Transition pairs:** (T1, T2), (T2, T8), (T8, T15), (T15, T12), (T12, T16), (T16, T17), (T17, T13), (T13, T13), (T13, T10), (T10, T7) are executed.

**Test#8:** open 1100 987 118 login 118 pin 123 pin 234 login 119 login 118 pin 987  
withdraw 300 lock 987 balance unlock 987 deposit 50 deposit 50 deposit 500  
logout

**Coverage of transitions:** T1, T2, T3, T6, T7, T2, T4, T14, T20, T18, T19, T21, T21,  
T15, T10

**Transition pairs:** (T1, T2), (T2, T3), (T3, T6), (T6, T7), (T7, T2), (T2, T4), (T4, T14), (T14, T20), (T20, T18), (T18, T19), (T19, T21), (T21, 21), (T21, T15), (T15, T10) are executed

**Test#9:** open 500 456 119 login 119 pin 456 logout login 118 login 119 pin 456 lock  
456 balance balance unlock 456 logout

**Coverage of transitions:** T1, T2, T8, T9, T7, T2, T8, T20, T18, T18, T19, T9

**Transition pairs:** (T1, T2), (T2, T8), (T8, T9), (T9, T7), (T7, T2), (T2, T8), (T8, T20), (T20, T18), (T18, T18), (T18, T19), (T19, T9) are executed.

**Test#10:** open 1100 321 120 login 120 pin 321 lock 321 unlock 321 lock 321 unlock 321 withdraw 50 deposit 100 deposit 100 withdraw 500 deposit 100 logout

**Coverage of transitions:** T1, T2, T4, T16, T17, T16, T17, T11, T12, T12, T14, T21, T9

**Transition pairs:** (T1, T2), (T2, T4), (T4, 16), (T16, T17), (T17, T11), (T11, T12), (T12, T12), (T12, T14), (T14, T21), (T21, T9) are executed.

**Test#11:** open 800 876 121 login 121 pin 876 logout login 121 pin 876 deposit 100 lock 876 unlock 876 balance lock 876 unlock 876 logout

**Coverage of transitions:** T1, T2, T8, T9, T2, T8, T21, T20, T19, T22, T20, T19, T9

**Transition pairs:** (T1, T2), (T2, T8), (T8, T9), (T9, T2), (T2, T8), (T8, T21), (T21, T20), (T20, T19), (T19, T22), (T22, T20), (T20, T19), (T19, T9) are executed.

**Test#12:** open 2200 101 122 login 122 pin 101 withdraw 500 withdraw 800 deposit 700 balance deposit 300 withdraw 100 logout

**Coverage of transitions:** T1, T2, T4, T11, T11, T14, T15, T13, T12, T11, T10

**Transition pairs:** (T1, T2), (T2, T4), (T4, T11), (T11, T11), (T11, T14), (T14, T15), (T15, T13), (T13, T12), (T12, T11), (T11, T10) are executed.

**Test#13:** open 1000 114 123 login 123 pin 114 balance withdraw 500 deposit 600 lock 114 unlock 114 withdraw 100 deposit 300 withdraw 200 logout

**Coverage of transitions:** T1, T2, T4, T13, T14, T15, T16, T17, T14, T15, T11, T10

**Transition pairs:** (T1, T2), (T2, T4), (T4, T13), (T13, T14), (T14, T15), (T15, T16), (T16, T17), (T17, T14), (T14, T15), (T15, T11), (T11, T10) are executed

**Non-executable transition pairs:**

(T2, T6): For this transition-pair, T2 is a request to login, and T6 is happened when the pin is wrong twice. So we should at least input wrong pin once to reach T6, which means the previous transition of T6 must be T3.

(T3, T3): For this transition-pair, T3 is happened when we input a wrong pin. In this EFSM, it only allow us to input wrong pin once. So when it reached T3 once, it cannot reach it twice.

(T16, T19): For this transition-pair, T16 is happened when the state transfer from ready to locked. If the unlock action is performed, the state must be back to ready state.

(T20, T17): For this transition-pair, T20 is happened when the state transfer from overdrawn to locked. If the unlock action is performed, the state must be back to overdrawn state.

### 3. Default test of the *account* class in the EFSM

For state **idle**, there are 8 default transitions:

open, logout, deposit, pin, balance, withdraw, lock, unlock

**Test#14:** open 1100 134 124 open 1100 134 124 logout

**Open, logout** default transitions for state idle are executed.

**Test#15:** open 1200 145 125 pin 145 deposit 100 balance logout

**Pin, deposit, balance, logout** default transitions for state idle are executed.

**Test#16:** open 1150 765 126 withdraw 50 lock 765 unlock 765 logout

**Withdraw, lock, unlock, logout** default transitions for state idle are executed.

For state **check pin**, there are 7 default transitions:

Open, login, withdraw, deposit, balance, lock, unlock.

**Test#17:** open 1234 999 127 login 127 open 1234 999 127 login 127

**Open, login** default transitions are executed.

**Test#18:** open 3456 555 128 login 128 withdraw 500 deposit 100

**Withdraw, deposit** default transition are executed.

**Test#19:** open 1200 990 129 login 129 balance lock 990 unlock 990

**balance, lock, unlock** default transitions are executed.

For state **ready**, there are 4 default transitions:

Open, unlock, login, pin.

**Test#20:** open 1500 666 130 login 130 pin 666 open 1500 666 130 unlock 666

**Open, unlock** default transitions are executed.

**Test#21:** open 2000 555 131 login 131 pin 555 login 131 pin 555

**Login, pin** default transitions are executed.

For state **locked**, there are 7 default transitions

Deposit, lock, login, pin, logout, open, withdraw.

**Test#22:** open 1300 444 132 login 132 pin 444 lock 444 deposit 100 lock 444

**Deposit, lock** default transitions are executed.

**Test#23:** open 1400 333 133 login 133 pin 333 lock 333 login 133 pin 333 logout

**Login, pin, logout** default transitions are executed.

**Test#24:** open 1500 222 134 login 134 pin 222 lock 222 open 1500 222 134

withdraw 500

**Open, withdraw** default transitions are executed.

For state **overdrawn**, there are 5 default transitions:

Open, login, pin, withdraw, unlock

**Test#25:** open 900 777 135 login 135 pin 777 open 900 777 135 login 135 pin 777

**Open, login, pin** default transitions are executed.

**Test#26:** open 800 111 136 login 136 pin 111 withdraw 50 unlock 111

**Withdraw, unlock** default transitions are executed.

#### 4. Multiple Condition Testing

For account:: open

x>0	x4==-1	Test Case
T	T	Test #12
T	F	Test#27: open 1200 123 139 open 1200 123 138 login 555 lock 123 unlock 123 deposit -200 logout
F	T	Test#28: open -100 123 138 login 137 logout balance login 137 pin 23 balance logout
F	F	Test#29: open -200 123 139 open -200 123 139 login 139 pin 111

For account::login

$x4 \neq 0$	Test
T	Test#26
F	Test#27

$x5 == x$	Test
T	Test#12
F	Test#27

For account::pin

$x4 \neq 1$	Test
T	Test#22
F	Test#8

$x == x3$	Test
T	Test#12
F	Test#29

$k \geq \text{num}$	Test
T	Test#8
F	Test#12

For account::balance

$x4 \neq 2$	Test
T	Test#34
F	Test#12

For account::logout

$x4 == 0$	$x2 == 1$	Test
T	T	Non-executable
T	F	Test#26
F	T	Test#11
F	F	Test#5

When  $x4 == 0$  means the program is the idle state,  $x2 == 1$  means the account is locked. Looking from the EFSM model, there is no lock operation in the idle state. So it is non-executable, when  $x4 == 0$  and  $x2 == 1$  are both true.

For account::lock

$x4 \neq 2$	Test
T	Test#8
F	Test#5

$x \neq x3$	Test
T	Test#30: open 1200 123 140 login 140 pin 123 lock 567 lock 123 lock 123
F	Test#5

$x2 == 0$	Test
T	Test#2
F	Test#27

For account::unlock

$x4 \neq 2$	Test
T	Test#8
F	Test#5

$x2$	$x == x3$	Test
T	T	Test#13
T	F	Test#31: open 1200 222 141 login 141 pin 222 lock 221, unlock 223, logout
F	T	Test#27
F	F	Test#31

For account::deposit

$x4 \neq 2$	Test
T	Test#8
F	Test#5

$x2$	Test
T	Test#13
F	Test#27

$x1 < x7$	$d > 0$	Test
T	T	Test#2
T	F	Test#32: open 800 123 222 login 222 pin 123 deposit -100 logout
F	T	Test#12
F	F	Test#33: open 1500 123 222 login 222 pin 123 deposit -200 logout

$d > 0$	Test
T	Test#2
F	Test#32

For account:: withdraw

$x4 \neq 2$	Test
T	Test#8
F	Test#5

$x2$	Test
T	Test#13
F	Test#27

$x1 > w$	$w > 0$	Test
T	T	Test#12
T	F	Test#34: open 1500 123 222 login 222 pin 123 withdraw -200 logout
F	T	Test#35: open 800 123 222 login 222 pin 123 withdraw 1200 logout
F	F	Non-executable



$w > 0$  false means that the withdraw is less than 0,  $x1 < w$  false means  $x1 < w < 0$ . Looking from the EFSM model, when  $x1(\text{balance}) < 0$ , it is in state overdrawn, but there is no operation withdraw in this state, so it is non-executable when  $x1 > w$  and  $w > 0$  are both false.

$x1 < x7$	Test
T	Test#2
F	Test#12

  

$(x1 = x1 - w) < x7$	Test
T	Test#8
F	Test#6

**Test#27:** open 1200 123 139 open 1200 123 138 login 555 lock 123 unlock 123  
deposit -200 logout

**Test#28:** open -100 123 138 login 137 logout balance login 137  
pin 23 balance logout

**Test#29:** open -200 123 139 open -200 123 139 login 139 pin  
111

**Test#30:** open 1200 123 140 login 140 pin 123 lock 567 lock 123 lock 123

**Test#31:** open 1200 222 141 login 141 pin 222 lock 221, unlock 223, logout

**Test#32:** open 800 123 222 login 222 pin 123 deposit -100 logout

**Test#33:** open 1500 123 222 login 222 pin 123 deposit -200 logout

**Test#34:** open 1500 123 222 login 222 pin 123

withdraw -200 logout

**Test#35:** open 800 123 222 login 222 pin 123

withdraw 1200 logout

## 5. Execution Test Cases

**Test#1:** open 1500 123 111 login 455 login 111 pin 123 deposit 50 balance lock 123  
unlock 123 logout

- Open 1500 123 111

Expected result: balance =1500, state=idle, return=0.

Actual result: balance =1500, state=idle, return=0.

- Login 455

Expected result: balance =1500, state=idle, return=-1

Actual result: balance =1500, state=idle, return=-1

- Login 111

Expected result: balance = 1500, state = check pin, return = 0

Actual result: balance = 1500, state = check pin, return = 0

- Pin 123

Expected result: balance = 1500, state = ready, return = 0

Actual result: balance = 1500, state = ready, return = 0

- Deposit 50

Expected result: balance = 1550, state=ready, return=0

Actual result: balance = 1550, state=ready, return=0

- Balance

Expected result: balance= 1550 state=ready, return= 1550

Actual result: balance= 1550 state=ready, return= 1550

- Lock 123

Expected result: balance=1550, state=locked, return=0

Actual result: balance=1550, state=locked, return=0

- Unlock 123

Expected result: balance 1550, state=ready, return=0

Actual result: balance 1550, state=ready, return=0

- Logout

Expected result: balance=1550, state=idle, return=0

Actual result: balance=1550, state=idle, return=0

- Test successful

**Test#2:** open 500 111 112 login 112 pin 113 pin 111 deposit 300 balance lock 111  
unlock 111 deposit 300 balance logout

- Open 500 111 112

Expected result: balance=500, state= idle, return=0

Actual result: balance=500, state= idle, return=0

- Login 112

Expected result: balance=500, state=check pin, return=0

Actual result: balance=500, state=check pin, return=0

- Pin 113

Expected result: balance=500, state=check pin, return=-1

Actual result: balance=500, state=check pin, return=-1

- Pin 111

Expected result: balance=500, state=overdrawn, return=0

Actual result: balance=500, state=overdrawn, return=0

- Deposit 300

Expected result: balance=790, state= overdrawn, return=0

Actual result: balance=790, state= overdrawn, return=0

- Balance

Expected result: balance=790, state= overdrawn, return=790

Actual result: balance=790, state= overdrawn, return=790

- Lock 111

Expected result: balance=790, state= locked, return=0

Actual result: balance=790, state= locked, return=0

- Unlock 111

Expected result: balance=790, state= overdrawn, return=0

Actual result: balance=790, state= overdrawn, return=0

- Deposit 300

Expected result: balance=1080, state= ready, return=0

Actual result: balance=1080, state= ready, return=0

- Balance

Expected result: balance=1080, state= ready, return=1080

Actual result: balance=1080, state= ready, return=1080

- Logout

Expected result: balance=1080, state= idle, return=0

Actual result: balance=1080, state= idle, return=0

- Test successful

**Test#3:** open 1200 234 113 login 113 pin 235 logout login 113 pin 236 pin 234

balance withdraw 100 balance lock 234 unlock 234 withdraw 200 balance

logout

- Open 1200 234 113

Expected result: balance=1200, state=idle, return=0

Actual result: balance=1200, state=idle, return=0

- Login 113

Expected result: balance=1200, state=check pin, return=0

Actual result: balance=1200, state=check pin, return=0

- Pin 235

Expected result: balance=1200, state=check pin, return=-1

Actual result: balance=1200, state=check pin, return=-1

- Logout

Expected result: balance=1200 state=idle, return=0

Actual result: balance=1200 state=idle, return=0

- Login 113

Expected result: balance=1200, state=check pin, return=0

Actual result: balance=1200, state=check pin, return=0

- Pin 236

Expected result: balance=1200, state=check pin, return=-1

Actual result: balance=1200, state=check pin, return=-1

- Pin 234

Expected result: balance=1200, state=ready, return=0

Actual result: balance=1200, state=ready, return=0

- Balance

Expected result: balance=1200, state=ready, return=1200

Actual result: balance=1200, state=ready, return=1200

- Withdraw 100

Expected result: balance=1100, state=ready, return=0

Actual result: balance=1100, state=ready, return=0

- Balance

Expected result: balance=1100, state=ready, return=1100

Actual result: balance=1100, state=ready, return=1100

- Lock 234

Expected result: balance=1100, state=locked, return=0

Actual result: balance=1100, state=locked, return=0

- Unlock 234

Expected result: balance=1100, state=ready, return=0

Actual result: balance=1100, state=ready, return=0

- Withdraw 200

Expected result: balance=890, state=overdrawn, return=0

Actual result: balance=890, state=overdrawn, return=0

- Balance

Expected result: balance=890, state=overdrawn, return=890

Actual result: balance=890, state=overdrawn, return=890

- Logout

Expected result: balance=890, state=idle, return=0

Actual result: balance=890, state=idle, return=0

- Test successful

**Test#4:** open 300 222 114 login 113 login 115 login 114 logout login 115 login 114  
pin 223 pin 221 login 114 pin 222 logout

- Open 300 222 114

- Expected result: balance=890, state=overdrawn, return=890

- Actual result: balance=890, state=overdrawn, return=890

- Login 113

Expected result: balance=300, state=idle, return=-1

Actual result: balance=300, state=idle, return=-1

- Login 115

Expected result: balance=300, state=idle, return=-1

Actual result: balance=300, state=idle, return=-1

- Login 114

Expected result: balance=300, state=check pin, return=0

Actual result: balance=300, state=check pin, return=0

- Logout

Expected result: balance=300, state=idle pin, return=0

Actual result: balance=300, state=idle pin, return=0

- Login 115

Expected result: balance=300, state=idle, return=-1

Actual result: balance=300, state=idle, return=-1

- Login 114

Expected result: balance=300, state=check pin, return=0

Actual result: balance=300, state=check pin, return=0

- Pin 223

Expected result: balance=300, state=check pin, return=0

Actual result: balance=300, state=check pin, return=0

- Pin 221

Expected result: balance=300, state=idle, return=0

Actual result: balance=300, state=idle, return=0

- Login 114  
Expected result: balance=300, state=check pin, return=0  
Actual result: balance=300, state=check pin, return=0
- Pin 222  
Expected result: balance=300, state=overdrawn, return=0  
Actual result: balance=300, state= overdrawn, return=0
- Logout  
Expected result: balance=300, state=idle, return=0  
Actual result: balance=300, state=idle, return=0
- Test successful

**Test#5:** open 500 333 115 login 115 pin 333 balance deposit 300 balance balance  
deposit 300 withdraw 200 logout

- Open 500 333 115  
Expected result: balance=500, state=idle, return=0  
Actual result: balance=500, state=idle, return=0
- Login 115  
Expected result: balance=500, state=check pin, return=0  
Actual result: balance=500, state=check pin, return=0
- Pin 333  
Expected result: balance=500, state=overdrawn, return=0  
Actual result: balance=500, state=overdrawn, return=0
- Balance  
Expected result: balance=500, state=overdrawn, return=500  
Actual result: balance=500, state=overdrawn, return=500
- Deposit 300  
Expected result: balance=790, state=overdrawn, return=0  
Actual result: balance=790, state=overdrawn, return=0
- Balance  
Expected result: balance=790, state=overdrawn, return=790  
Actual result: balance=790, state=overdrawn, return=790
- Balance  
Expected result: balance=790, state=overdrawn, return=790  
Actual result: balance=790, state=overdrawn, return=790
- Deposit 300

Expected result: balance=1080, state=ready, return=0

Actual result: balance=1080, state=ready, return=0

- Withdraw 200

Expected result: balance=870, state=overdrawn, return=0

Actual result: balance=870, state=overdrawn, return=0

- Logout

Expected result: balance=870, state=idle, return=0

Actual result: balance=870, state=idle, return=0

- Test successful

**Test#6:** open 2000 444 116 login 116 pin 444 logout login 116 pin 444 withdraw 500  
withdraw 300 lock 444 balance unlock 444 deposit 1000 logout

- Open 2000 444 116

Expected result: balance=2000, state=idle, return=0

Actual result: balance=2000, state=idle, return=0

- Login 116

Expected result: balance=2000, state=check pin, return=0

Actual result: balance=2000, state=check pin, return=0

- Pin 444

Expected result: balance=2000, state=ready, return=0

Actual result: balance=2000, state=ready, return=0

- Logout

Expected result: balance=2000, state=idle, return=0

Actual result: balance=2000, state=idle, return=0

- Login 116

Expected result: balance=2000, state=check pin, return=0

Actual result: balance=2000, state=check pin, return=0

- Pin 444

Expected result: balance=2000, state=ready, return=0

Actual result: balance=2000, state=ready, return=0

- Withdraw 500

Expected result: balance=1500, state=ready, return=0

Actual result: balance=1500, state=ready, return=0

- Withdraw 300

Expected result: balance=1200, state=ready, return=0

Actual result: balance=1200, state=ready, return=0

- Lock 444

Expected result: balance=1200, state=locked, return=0

Actual result: balance=1200, state=locked, return=0

- Balance

Expected result: balance=1200, state=locked, return=1200

Actual result: balance=1200, state=locked, return=1200

- Unlock 444

Expected result: balance=1200, state=ready, return=0

Actual result: balance=1200, state=ready, return=0

- Deposit 1000

Expected result: balance=2200, state=ready, return=0

Actual result: balance=2200, state=ready, return=0

- Logout

Expected result: balance=2200, state=idle, return=0

Actual result: balance=2200, state=idle, return=0

- Test successful

**Test#7:** open 900 789 117 login 117 pin 789 deposit 300 deposit 200 lock 789 unlock  
789 balance balance logout login 113

- Open 900 789 117

Expected result: balance=900, state=idle, return=0

Actual result: balance=900, state=idle, return=0

- Login 117

Expected result: balance=900, state=check pin, return=0

Actual result: balance=900, state=check pin, return=0

- Pin 789

Expected result: balance=900, state=overdrawn, return=0

Actual result: balance=900, state=overdrawn, return=0

- Deposit 300

Expected result: balance=1190, state=ready, return=0

Actual result: balance=1190, state=ready, return=0

- Deposit 200

Expected result: balance=1390, state=ready, return=0

Actual result: balance=1390, state=ready, return=0



- Lock 789  
Expected result: balance=1390, state=locked, return=0  
Actual result: balance=1390, state=locked, return=0
- Unlock 789  
Expected result: balance=1390, state=ready, return=0  
Actual result: balance=1390, state=ready, return=0
- Balance  
Expected result: balance=1390, state=ready, return=1390  
Actual result: balance=1390, state=ready, return=1390
- Balance  
Expected result: balance=1390, state=ready, return=1390  
Actual result: balance=1390, state=ready, return=1390
- Logout  
Expected result: balance=1390, state=idle, return=0  
Actual result: balance=1390, state=idle, return=0
- Login 113  
Expected result: balance=1390, state=idle, return=-1  
Actual result: balance=1390, state=idle, return=0
- Test successful

**Test#8:** open 1100 987 118 login 118 pin 123 pin 234 login 119 login 118 pin 987  
withdraw 300 lock 987 balance unlock 987 deposit 50 deposit 50 deposit 500  
logout

- Open 1100 987 118  
Expected result: balance=1100, state=idle, return=0  
Actual result: balance=1100, state=idle, return=0
- Login 118  
Expected result: balance=1100, state=check pin, return=0
- Actual result: balance=1100, state=check pin, return=0
- Pin 123  
Expected result: balance=1100, state=check pin, return=-1
- Actual result: balance=1100, state=check pin, return=-1
- Pin 234  
Expected result: balance=1100, state=idle, return=-1
- Actual result: balance=1100, state=idle, return=-1

- Login 119  
Expected result: balance=1100, state=idle, return=-1
- Actual result: balance=1100, state=idle, return=-1
- Login 118  
Expected result: balance=1100, state=check pin, return=0
- Actual result: balance=1100, state=check pin, return=0
- Pin 987  
Expected result: balance=1100, state=ready, return=0
- Actual result: balance=1100, state=ready, return=0
- Withdraw 300  
Expected result: balance=790, state=overdrawn, return=0
- Actual result: balance=790, state=overdrawn, return=0
- Lock 987  
Expected result: balance=790, state=locked, return=0
- Actual result: balance=790, state=locked, return=0
- Balance  
Expected result: balance=790, state=locked, return=790
- Actual result: balance=790, state=locked, return=790
- Unlock 987  
Expected result: balance=790, state=overdrawn, return=0
- Actual result: balance=790, state=overdrawn, return=0
- Deposit 50  
Expected result: balance=830, state=overdrawn, return=0
- Actual result: balance=790, state=overdrawn, return=0
- Deposit 50  
Expected result: balance=870, state=overdrawn, return=0
- Actual result: balance=870, state=overdrawn, return=0
- Deposit 500  
Expected result: balance=1360, state=ready, return=0
- Actual result: balance=1360, state=ready, return=0
- Logout  
Expected result: balance=1360, state=idle, return=0
- Actual result: balance=1360, state=ready, return=0
- Test successful

**Test#9:** open 500 456 119 login 119 pin 456 logout login 118 login 119 pin 456 lock  
456 balance balance unlock 456 logout

- Open 500 456 119  
Expected result: balance=500, state=idle, return=0  
Actual result: balance=500, state=idle, return=0
- Login 119  
Expected result: balance=500, state=check pin, return=0  
Actual result: balance=500, state=check pin, return=0
- Pin 456  
Expected result: balance=500, state=overdrawn, return=0  
Actual result: balance=500, state=overdrawn, return=0
- Logout  
Expected result: balance=500, state=idle, return=0  
Actual result: balance=500, state=idle, return=0
- Login 118  
Expected result: balance=500, state=idle, return=-1  
Actual result: balance=500, state=idle, return=-1
- Login 119  
Expected result: balance=500, state=check pin, return=0  
Actual result: balance=500, state=check pin, return=0
- Pin 456  
Expected result: balance=500, state=overdrawn, return=0  
Actual result: balance=500, state=overdrawn, return=0
- Lock 456  
Expected result: balance=500, state=locked, return=0  
Actual result: balance=500, state=locked, return=0
- Balance  
Expected result: balance=500, state=locked, return=500  
Actual result: balance=500, state=locked, return=500
- Balance  
Expected result: balance=500, state=locked, return=500  
Actual result: balance=500, state=locked, return=500
- Unlock 456  
Expected result: balance=500, state=overdrawn, return=0

Actual result: balance=500, state=overdrawn, return=0

- Logout

Expected result: balance=500, state=idle, return=0

Actual result: balance=500, state=idle, return=0

- Test successful

**Test#10:** open 1100 321 120 login 120 pin 321 lock 321 unlock 321 lock 321 unlock  
321 withdraw 50 deposit 100 deposit 100 withdraw 500 deposit 100 logout

- Open 1100 321 120

Expected result: balance=1100. State=idle, return=0

Actual result: balance=1100. State=idle, return=0

- Login 120

Expected result: balance=1100. State=check pin, return=0

Actual result: balance=1100. State=check pin, return=0

- Pin 321

Expected result: balance=1100. State=ready, return=0

Actual result: balance=1100. State=ready, return=0

- Lock 321

Expected result: balance=1100. State=locked, return=0

Actual result: balance=1100. State=locked, return=0

- Unlock 321

Expected result: balance=1100. State=ready, return=0

Actual result: balance=1100. State=ready, return=0

- Lock 321

Expected result: balance=1100. State=locked, return=0

Actual result: balance=1100. State=locked, return=0

- Unlock 321

Expected result: balance=1100. State=ready, return=0

Actual result: balance=1100. State=ready, return=0

- Withdraw 50

Expected result: balance=1050. State=ready, return=0

Actual result: balance=1050. State=ready, return=0

- Deposit 100

Expected result: balance=1150. State=ready, return=0

Actual result: balance=1250. State=ready, return=0

- Deposit 100  
Expected result: balance=1250. State=ready, return=0  
Actual result: balance=1250. State=ready, return=0
- Withdraw 500  
Expected result: balance=740. State=overdrawn, return=0  
Actual result: balance=740. State=overdrawn, return=0
- Deposit 100  
Expected result: balance=830. State=overdrawn, return=0  
Actual result: balance=830. State=overdrawn, return=0
- Logout  
Expected result: balance=830. State=idle, return=0  
Actual result: balance=830. State=idle, return=0
- Test successful

**Test#11:** open 800 876 121 login 121 pin 876 logout login 121 pin 876 deposit 100  
lock 876 unlock 876 balance lock 876 unlock 876 logout

- Open 800 876 121  
Expected result: balance=800, state=idle, return=0  
Actual result: balance=800, state=idle, return=0
- Login 121  
Expected result: balance=800, state=check pin, return=0  
Actual result: balance=800, state=check pin, return=0
- Pin 876  
Expected result: balance=800, state=overdrawn, return=0  
Actual result: balance=800, state= overdrawn, return=0
- Logout  
Expected result: balance=800, state=idle, return=0  
Actual result: balance=800, state=idle, return=0
- Login 121  
Expected result: balance=800, state=check pin, return=0  
Actual result: balance=800, state=check pin, return=0
- Pin 876  
Expected result: balance=800, state= overdrawn, return=0  
Actual result: balance=800, state= overdrawn, return=0
- Deposit 100

Expected result: balance=890, state= overdrawn, return=0

Actual result: balance=890, state= overdrawn, return=0

- Lock 876

Expected result: balance=890, state= locked, return=0

Actual result: balance=890, state= locked, return=0

- Unlock 876

Expected result: balance=890, state= overdrawn, return=0

Actual result: balance=890, state= overdrawn, return=0

- Balance

Expected result: balance=890, state= overdrawn, return=890

Actual result: balance=890, state= overdrawn, return=890

- Lock 876

Expected result: balance=890, state= locked, return=0

Actual result: balance=890, state= locked, return=0

- Unlock 876

Expected result: balance=890, state= overdrawn, return=0

Actual result: balance=890, state= overdrawn, return=0

- Logout

Expected result: balance=890, state= idle, return=0

Actual result: balance=890, state= overdrawn, return=0

- Test successful

**Test#12:** open 2200 101 122 login 122 pin 101 withdraw 500 withdraw 800 deposit  
700 balance deposit 300 withdraw 100 logout

- Open 2200 101 122

Expected result: balance=2200, state=idle, return=0

Actual result: balance=2200, state=idle, return=0

- Login 122

Expected result: balance=2200, state=check pin, return=0

Actual result: balance=2200, state=check pin, return=0

- Pin 101

Expected result: balance=2200, state=ready, return=0

Actual result: balance=2200, state=ready, return=0

- Withdraw 500

Expected result: balance=1700, state=ready, return=0

Actual result: balance=1700, state=ready, return=0

- Withdraw 800

Expected result: balance=890, state=overdrawn, return=0

Actual result: balance=890, state=overdrawn, return=0

- Deposit 700

Expected result: balance=1580, state=ready, return=0

Actual result: balance=1580, state=ready, return=0

- Balance

Expected result: balance=1580, state=ready, return=1580

Actual result: balance=1580, state=ready, return=1580

- Deposit 300

Expected result: balance=1880, state=ready, return=0

Actual result: balance=1880, state=ready, return=0

- Withdraw 100

Expected result: balance=1780, state=ready, return=0

Actual result: balance=1780, state=ready, return=0

- Logout

Expected result: balance=1580, state=idle, return=0

Actual result: balance=1580, state=idle, return=0

- Test successful

**Test#13:** open 1000 114 123 login 123 pin 114 balance withdraw 500 deposit 600

lock 114 unlock 114 withdraw 100 deposit 300 withdraw 200 logout

- Open 1000 114 123

Expected result: balance=1000, state= idle, return=0

Actual result: balance=1000, state= idle, return=0

- Login 123

Expected result: balance=1000, state= check pin, return=0

Actual result: balance=1000, state= check pin, return=0

- Pin 114

Expected result: balance=1000, state= ready, return=0

Actual result: balance=1000, state= ready, return=0

- Balance

Expected result: balance=1000, state= ready, return=1000

Actual result: balance=1000, state= ready, return=1000

- Withdraw 500  
Expected result: balance=490, state= overdrawn, return=0  
Actual result: balance=490, state= overdrawn, return=0
- Deposit 600  
Expected result: balance=1080, state= ready, return=0  
Actual result: balance=1080, state= ready, return=0
- Lock 114  
Expected result: balance=1080, state= locked, return=0  
Actual result: balance=1080, state= locked, return=0
- Unlock 114  
Expected result: balance=1080, state= ready, return=0  
Actual result: balance=1080, state= ready, return=0
- Withdraw 100  
Expected result: balance=970, state= overdrawn, return=0  
Actual result: balance=970, state= overdrawn, return=0
- Deposit 300  
Expected result: balance=1260, state= ready, return=0  
Actual result: balance=1260, state= ready, return=0
- Withdraw 200  
Expected result: balance=1060, state= ready, return=0  
Actual result: balance=1060, state= ready, return=0
- Logout  
Expected result: balance=1260, state= idle, return=0  
Actual result: balance=1260, state= idle, return=0
- Test successful

**Test#14:** open 1100 134 124 open 1100 134 124 logout

- Open 1100 134 124  
Expected result: balance=1100, state=idle. Return=0  
Actual result: balance=1100, state=idle. Return=0
- Open 1100 134 124  
Expected result: balance=1100, state=idle. Return=-1  
Actual result: balance=1100, state=idle. Return=-1
- Logout  
Expected result: balance=1100, state=idle. Return=0



Actual result: balance=1100, state=idle. Return=0

- Test successful

**Test#15:** open 1200 145 125 pin 145 deposit 100 balance logout

- Open 1200 145 125

Expected result: balance=1200, state=idle, return=0

Actual result: balance=1200, state=idle, return=0

- Pin 145

Expected result: balance=1200, state=idle, return=-1

Actual result: balance=1200, state=idle, return=-1

- Deposit 100

Expected result: balance=1200, state=idle, return=-1

Actual result: balance=1200, state=idle, return=-1

- Balance

Expected result: balance=1200, state=idle, return=-1

Actual result: balance=1200, state=idle, return=-1

- Logout

Expected result: balance=1200, state=idle, return=-1

Actual result: balance=1200, state=idle, return=-1

- Test successful

**Test#16:** open 1150 765 126 withdraw 50 lock 765 unlock 765 logout

- Open 1150 765 126

Expected result: balance=1150, state=idle, return=0

Actual result: balance=1150, state=idle, return=0

- Withdraw 50

Expected result: balance=1150, state=idle, return=-1

Actual result: balance=1150, state=idle, return=-1

- Lock 765

Expected result: balance=1150, state=idle, return=-1

Actual result: balance=1150, state=idle, return=-1

- Unlock 765

Expected result: balance=1150, state=idle, return=-1

Actual result: balance=1150, state=idle, return=-1

- Logout

Expected result: balance=1150, state=idle, return=-1

Actual result: balance=1150, state=idle, return=-1

- Test successful

**Test#17:** open 1234 999 127 login 127 open 1234 999 127 login 127

- Open 1234 999 127

Expected result: balance=1234, state=idle, return=0

Actual result: balance=1234, state=idle, return=0

- Login 127

Expected result: balance=1234, state=check pin, return=0

Actual result: balance=1234, state=check pin, return=0

- Open 1234 999 127

Expected result: balance=1234, state=check pin, return=-1

Actual result: balance=1234, state=check pin, return=-1

- Login 127

Expected result: balance=1234, state=check pin, return=-1

Actual result: balance=1234, state=check pin, return=-1

- Test successful

**Test#18:** open 3456 555 128 login 128 withdraw 500 deposit 100

- Open 3456 555 128

Expected result: balance=3456, state=idle, return=0

Actual result: balance=3456, state=idle, return=0

- Login 128

Expected result: balance=3456, state=check pin, return=0

Actual result: balance=3456, state=check pin, return=0

- Withdraw 500

Expected result: balance=3456, state=check pin, return=-1

Actual result: balance=3456, state=check pin, return=-1

- Deposit 100

Expected result: balance=3456, state=check pin, return=-1

Actual result: balance=3456, state=check pin, return=-1

- Test successful

**Test#19:** open 1200 990 129 login 129 balance lock 990 unlock 990

- Open 1200 990 129

Expected result: balance=1200, state=idle, return=0

Actual result: balance=1200, state=idle, return=0

- Login 129  
Expected result: balance=1200, state=check pin, return=0  
Actual result: balance=1200, state=check pin, return=0
- Balance  
Expected result: balance=1200, state=check pin, return=-1  
Actual result: balance=1200, state=check pin, return=-1
- Lock 990  
Expected result: balance=1200, state=check pin, return=-1  
Actual result: balance=1200, state=check pin, return=-1
- Unlock 990  
Expected result: balance=1200, state=check pin, return=-1  
Actual result: balance=1200, state=check pin, return=-1
- Test successful

**Test#20:** open 1500 666 130 login 130 pin 666 open 1500 666 130 unlock 666

- Open 1500 666 130  
Expected result: balance=1500, state=idle, return=0  
Actual result: balance=1500, state=idle, return=0
- Login 130  
Expected result: balance=1500, state=check pin, return=0  
Actual result: balance=1500, state=check pin, return=0
- Pin 666  
Expected result: balance=1500, state=ready, return=0  
Actual result: balance=1500, state=ready, return=0
- Open 1500 666 130  
Expected result: balance=1500, state=ready, return=-1  
Actual result: balance=1500, state=ready, return=-1
- Unlock 666  
Expected result: balance=1500, state=ready, return=-1  
Actual result: balance=1500, state=ready, return=-1
- Test successful

**Test#21:** open 2000 555 131 login 131 pin 555 login 131 pin 555

- Open 2000 555 131  
Expected result: balance=2000, state=idle, return=0  
Actual result: balance=2000, state=idle, return=0

- Login 131  
Expected result: balance=2000, state=check pin, return=0  
Actual result: balance=2000, state=check pin, return=0
- Pin 555  
Expected result: balance=2000, state=ready, return=0  
Actual result: balance=2000, state=ready, return=0
- Login 131  
Expected result: balance=2000, state=ready, return=-1  
Actual result: balance=2000, state=ready, return=-1
- Pin 555  
Expected result: balance=2000, state=ready, return=-1  
Actual result: balance=2000, state=ready, return=-1
- Test successful

**Test#22:** open 1300 444 132 login 132 pin 444 lock 444 deposit 100 lock 444

- Open 1300 444 132  
Expected result: balance=1300, state=idle, return=0  
Actual result: balance=1300, state=idle, return=0
- Login 132  
Expected result: balance=1300, state=check pin, return=0  
Actual result: balance=1300, state=check pin, return=0
- Pin 444  
Expected result: balance=1300, state=ready, return=0  
Actual result: balance=1300, state=ready, return=0
- Lock 444  
Expected result: balance=1300, state=locked, return=0  
Actual result: balance=1300, state=locked, return=0
- Deposit 100  
Expected result: balance=1300, state=locked, return=-1  
Actual result: balance=1300, state=locked, return=-1
- Lock 444  
Expected result: balance=1300, state=locked, return=-1  
Actual result: balance=1300, state=locked, return=-1
- Test successful

**Test#23:** open 1400 333 133 login 133 pin 333 lock 333 login 133 pin 333 logout

- Open 1400 333 133  
Expected result: balance=1400, state=idle, return=0  
Actual result: balance=1400, state=idle, return=0
- Login 133  
Expected result: balance=1400, state=check pin, return=0  
Actual result: balance=1400, state=check pin, return=0
- Pin 133  
Expected result: balance=1400, state=ready, return=0  
Actual result: balance=1400, state=ready, return=0
- Lock 333  
Expected result: balance=1400, state=locked, return=0  
Actual result: balance=1400, state=locked, return=0
- Login 133  
Expected result: balance=1400, state=locked, return=-1  
Actual result: balance=1400, state=locked, return=-1
- Pin 133  
Expected result: balance=1400, state=locked, return=-1  
Actual result: balance=1400, state=locked, return=-1
- Logout  
Expected result: balance=1400, state=locked, return=-1  
Actual result: balance=1400, state=locked, return=-1
- Test successful

Test#24: open 1500 222 134 login 134 pin 222 lock 222 open 1500 222 134 withdraw  
500

- Open 1500 222 134  
Expected result: balance=1500, state=idle, return=0  
Actual result: balance=1500, state=idle, return=0
- Login 134  
Expected result: balance=1500, state=check pin, return=0  
Actual result: balance=1500, state=check pin, return=0
- Pin 222  
Expected result: balance=1500, state=ready, return=0  
Actual result: balance=1500, state=ready, return=0
- Lock 222

Expected result: balance=1500, state=locked, return=0

Actual result: balance=1500, state=locked, return=0

- Open 1500 222 134

Expected result: balance=1500, state=locked, return=-1

Actual result: balance=1500, state=locked, return=-1

- Withdraw 500

Expected result: balance=1500, state=locked, return=-1

Actual result: balance=1500, state=locked, return=-1

- Test successful

**Test#25:** open 900 777 135 login 135 pin 777 open 900 777 135 login 135 pin 777

- Open 900 777 135

Expected result: balance=900, state=idle, return=0

Actual result: balance=900, state=idle, return=0

- Login 135

Expected result: balance=900, state=check pin, return=0

Actual result: balance=900, state=check pin, return=0

- Pin 777

Expected result: balance=900, state=overdrawn, return=0

Actual result: balance=900, state= overdrawn, return=0

- Open 900 777 135 pin 777

Expected result: balance=900, state=overdrawn, return=-1

Actual result: balance=900, state=overdrawn, return=-1

- Login 135

Expected result: balance=900, state=overdrawn, return=-1

Actual result: balance=900, state=overdrawn, return=-1

- Pin 777

Expected result: balance=900, state=overdrawn, return=-1

Actual result: balance=900, state=overdrawn, return=-1

- Test successful

**Test#26:** open 800 111 136 login 136 pin 111 withdraw 50 unlock 111

- Open 800 111 135

Expected result: balance=800, state=idle, return=0

Actual result: balance=800, state=idle, return=0

- Login 136

Expected result: balance=800, state=check pin, return=0

Actual result: balance=800, state=check pin, return=0

- Pin 111

Expected result: balance=800, state=overdrawn, return=0

Actual result: balance=800, state= overdrawn, return=0

- Withdraw 50

Expected result: balance=800, state=overdrawn, return=-1

Actual result: balance=800, state= overdrawn, return=-1

- Unlock 111

Expected result: balance=800, state=overdrawn, return=-1

Actual result: balance=800, state= overdrawn, return=-1

- Test successful

**Test#27:** open 1200 123 139 open 1200 123 138 login 555 lock 123 unlock 123  
deposit -200 logout

- Open 1200 123 139

Expected result: balance=1200, state=idle, return=0

Actual result: balance=1200, state=idle, return=0

- Open 1200 123 138

Expected result: balance=1200, state=idle, return=-1

Actual result: balance=1200, state=idle, return=-1

- Login 555

Expected result: balance=1200, state=idle, return=-1

Actual result: balance=1200, state=idle, return=-1

- Lock 123

Expected result: balance=1200, state=idle, return=-1

Actual result: balance=1200, state=idle, return=-1

- Unlock 123

Expected result: balance=1200, state=idle, return=-1

Actual result: balance=1200, state=idle, return=-1

- Deposit -100

Expected result: balance=1200, state=idle, return=-1

Actual result: balance=1200, state=idle, return=-1

- Logout

Expected result: balance=1200, state=idle, return=-1

Actual result: balance=1200, state=idle, return=-1

- Test successful

**Test#28:** open -100 123 138 login 137 logout balance login 137 pin 123 balance  
logout

- Open -100 123 138

Expected result: balance=0, state=start, return=-1

Actual result: balance=0, state=start, return=-1

- Login 137

Expected result: balance=0, state=start, return=-1

Actual result: balance=0, state=start, return=-1

- Logout

Expected result: balance=0, state=start, return=-1

Actual result: balance=0, state=idle, return=-1

- Balance

Expected result: balance=0, state=start, return=-1

Actual result: balance=0, state= idle, return=-1

- Login 137

Expected result: balance=0, state=start, return=-1

Actual result: balance=0, state= idle, return=-1

- Pin 123

Expected result: balance=0, state=start, return=-1

Actual result: balance=0, state= idle, return=-1

- Balance

Expected result: balance=0, state=start, return=-1

Actual result: balance=0, state= idle, return=-1

- Logout

Expected result: balance=0, state=start, return=-1

Actual result: balance=0, state= idle, return=-1

- Test failed

Because the open -100 123 138, the balance  $-100 < 0$ , so the program cannot get in the idle state, it is expected to stay in start state, but when logout at start state, it came to idle state, and the following operations were also in the idle state no start state, so it is a defect in the program.

**Test#29:** open -200 123 139 open -200 123 139 login 139 pin 111



- Open -200 123 139  
Expected result: balance=0, state=start, return=-1  
Actual result: balance=0, state=start, return=-1
- Open -200 123 139  
Expected result: balance=0, state=start, return=-1  
Actual result: balance=0, state=start, return=-1
- Login 139  
Expected result: balance=0, state=start, return=-1  
Actual result: balance=0, state=start, return=-1
- Pin 111  
Expected result: balance=0, state=start, return=-1  
Actual result: balance=0, state=start, return=-1
- Test successful

**Test#30:** open 1200 123 140 login 140 pin 123 lock 567 lock 123 lock 123

- Open 1200 123 140  
Expected result: balance=1200, state=idle, return=0  
Actual result: balance=1200, state=idle, return=0
- Login 140  
Expected result: balance=1200, state=check pin, return=0  
Actual result: balance=1200, state=check pin, return=0
- Pin 123  
Expected result: balance=1200, state=ready, return=0  
Actual result: balance=1200, state=ready, return=0
- Lock 567  
Expected result: balance=1200, state=ready, return=-1  
Actual result: balance=1200, state=ready, return=-1
- Lock 123  
Expected result: balance=1200, state=locked, return=0  
Actual result: balance=1200, state=locked, return=0
- Lock 123  
Expected result: balance=1200, state=locked, return=-1  
Actual result: balance=1200, state=locked, return=-1
- Test successful

**Test#31:** open 1200 222 141 login 141 pin 222 lock 221, unlock 223, logout

- Open 1200 222 141  
Expected result: balance=1200, state=idle, return=0  
Actual result: balance=1200, state=idle, return=0
- Login 141  
Expected result: balance=1200, state=check pin, return=0  
Actual result: balance=1200, state= check pin, return=0
- Pin 222  
Expected result: balance=1200, state=ready, return=0  
Actual result: balance=1200, state=ready, return=0
- Lock 221  
Expected result: balance=1200, state=ready, return=-1  
Actual result: balance=1200, state=ready, return=-1
- Unlock 223  
Expected result: balance=1200, state=ready, return=-1  
Actual result: balance=1200, state=ready, return=-1
- Logout  
Expected result: balance=1200, state=idle, return=0  
Actual result: balance=1200, state=idle, return=0
- Test successful

**Test#32:** open 800 123 222 login 222 pin 123 deposit -100 logout

- Open 800 123 222  
Expected result: balance=800, state=idle, return=0  
Actual result: balance=800, state=idle, return=0
- Login 222  
Expected result: balance=800, state=check pin, return=0  
Actual result: balance=800, state=check pin, return=0
- Pin 123  
Expected result: balance=800, state=overdrawn, return=0  
Actual result: balance=800, state= overdrawn, return=0
- Deposit -100  
Expected result: balance=690, state=overdrawn, return=0  
Actual result: balance=800, state= overdrawn, return=-1
- Logout  
Expected result: balance=690, state=idle, return=0

Actual result: balance=800, state= idle, return=0

- Test failed

Because when we deposit -100, from looking EFSM model, it should deposit -100(equals withdraw 100) and get a 10 penalty. But actually, when the value of deposit is negative the program would not run it and return to -1, besides, the balance of logout was also not like expected. So it is a bug in the program.

**Test#33:** open 1500 123 222 login 222 pin 123 deposit -200 logout

- Open 1500 123 222

Expected result: balance=1500, state=idle, return=0

Actual result: balance=1500, state=idle, return=0

- Login 222

Expected result: balance=1500, state=check pin, return=0

Actual result: balance=1500, state=check pin, return=0

- Pin 123

Expected result: balance=1500, state=ready, return=0

Actual result: balance=1500, state=ready, return=0

- Deposit -200

Expected result: balance=1300, state=ready, return=0

Actual result: balance=1500, state=ready, return=-1

- Logout

Expected result: balance=1300, state=idle, return=0

Actual result: balance=1500, state=idle, return=0

- Test failed

Because when we deposit -200, from looking EFSM model, it should deposit -200(equals withdraw 200). But actually, when the value of deposit is negative the program would not run it and return to -1, besides, the balance of logout was also not like expected. So it is a bug in the program.

**Test#34:** open 1500 123 222 login 222 pin 123 withdraw -200 logout

- Open 1500 123 222

Expected result: balance=1500, state=idle, return=0

Actual result: balance=1500, state=idle, return=0

- Login 222

Expected result: balance=1500, state=check pin, return=0

Actual result: balance=1500, state=check pin, return=0

- Pin 123  
Expected result: balance=1500, state=ready, return=0  
Actual result: balance=1500, state=ready, return=0
- Withdraw -200  
Expected result: balance=1700, state=ready, return=0  
Actual result: balance=1500, state=ready, return=-1
- Logout  
Expected result: balance=1700, state=idle, return=0  
Actual result: balance=1500, state=idle, return=0
- Test failed  
Because when we withdraw -200, from looking EFSM model, it should withdraw -200(equals deposit 200). But actually, when the value of withdraw is negative the program would not run it and return to -1, besides, the balance of logout was also not like expected. So it is a bug in the program.

**Test#35:** open 800 123 222 login 222 pin 123 withdraw 1200 logout

- Open 800 123 222  
Expected result: balance=800, state=idle, return=0  
Actual result: balance=800, state=idle, return=0
- Login 222  
Expected result: balance=800, state=check pin, return=0  
Actual result: balance=800, state=check pin, return=0
- Pin 123  
Expected result: balance=800, state=overdrawn, return=0  
Actual result: balance=800, state= overdrawn, return=0
- Withdraw -200  
Expected result: balance=990, state= overdrawn, return=0  
Actual result: balance=800, state= overdrawn, return=-1
- Logout  
Expected result: balance=990, state=idle, return=0  
Actual result: balance=800, state=idle, return=0
- Test failed

Because when we withdraw -200, from looking EFSM model, it should withdraw -200(equals deposit 200) and get a 10 penalty. But actually, when the value of

withdraw is negative the program would not run it and return to -1, besides, the balance of logout was also not like expected. So it is a bug in the program.

There one more bug I found in the program.

**Test#36:** open 500 123 333, login 333, pin 123, deposit 501

- Open 500 123 333

Expected result: balance=500, state=idle, return=0

Actual result: balance=500, state=idle, return=0

- Login 333

Expected result: balance=500, state=check pin, return=0

Actual result: balance=500, state=check pin, return=0

- Pin 123

Expected result: balance=500, state=overdrawn, return=0

Actual result: balance=500, state= overdrawn, return=0

- Deposit 501

Expected result: balance=991, state=ready, return=0

Actual result: balance=991, state= overdrawn, return=0

- Test failed

Because when we deposit 501 in the overdrawn state, looking from EFSM model, it would come to ready state, because  $500+501 > 1000$ , and the balance would be  $1001-10=991$ . But actually it would still stay on overdrawn state. So it is a bug of the program.

## 6. Conclusion

For the test driver, in my view, we could put more methods on it. But this time I only add reset and quit methods for it. I did not add method like show state, but I show the state in the bottom after every operation we executed. I think that is more directly and easy to find out the state after each operation. The quit method is used for quit the program, it is very easy to use. However, if I add methods those could check multiple data or store the information of each account for further uses, the program would be better.

For the test cases, in my opinion, even the three methods of testing cover almost 100%, I still could not find all the defects of the program. Then I explained each defect I found in the testing process, but there still may be some defects I did not find in the program.

Finally, for automatic activities part, I showed balance, lock state, pin, EFSM state, account, penalty, minimum balance, attempt times, maximum attempts, and current state after execute every operation automatically. In my opinion, it will help to see the details of the account after every operation so we could easily find where is the program running and get the information of that state. Furthermore, I think if we could automatically input the test cases, it would save lots of time and reduce more mistakes when inputting manually. If after inputting automatically, the program could choose which test case to execute, that would be much better and efficiently.

## Source Code

account.h

```
#pragma once
#include <string>
using namespace std;

class account
{
public:

    // =====
    // Original Account Member
    // =====

    account();
    int open(int, int, int);
    int deposit(int);
    int withdraw(int);
    int balance();
    int lock(int);
    int unlock(int);
    int login(int);
    int logout();
    int pin(int);
    int show_balance() { return x1; }; //testing oriented method

private:

    // =====
    // Original Account Member
    // =====

    int x1;           // balance
    int x2;           // is locked
    int x3;           // pin #
    int x4;           // state
    int x5;           // account #
    int x6;           // penalty
    int x7;           // minimum balance
    int k;
    int num;          //maximum # of attempts with incorrect pin
```

```

public:

    // =====
    // Testing Methods
    // =====

    int test_get_x1() { return x1; }      // show_balance
    int test_get_x2() { return x2; }      // isLocked
    int test_get_x3() { return x3; }      // show_pin
    int test_get_x4() { return x4; }      // show_state
    int test_get_x5() { return x5; }      // show_account
    int test_get_x6() { return x6; }      // show_penalty
    int test_get_x7() { return x7; }      // show_minimumBalance
    int test_get_k() { return k; }
    int test_get_num() { return num; }    // show_maximumAttempts

    void test_set_x1(const int &x1) { this->x1 = x1; }
    void test_set_x2(const int &x2) { this->x2 = x2; }
    void test_set_x3(const int &x3) { this->x3 = x3; }
    void test_set_x4(const int &x4) { this->x4 = x4; }
    void test_set_x5(const int &x5) { this->x5 = x5; }
    void test_set_x6(const int &x6) { this->x6 = x6; }
    void test_set_x7(const int &x7) { this->x7 = x7; }
    void test_set_k(const int &k) { this->k = k; }
    void test_set_num(const int &num) { this->num = num; }

    string test_show_state() {
        switch (x4)
        {
            case -1:
                return "START";
            case 0:
                return "IDLE";
            case 1:
                return "CHECK PIN";
            case 2:
                if (0 == x2 && x1 < x7) { return "OVERDRAWN"; }
                if (0 == x2 && x1 >= x7) { return "READY"; }
                if (1 == x2) { return "LOCKED"; }
            }
            return "UNKNOWN";
        }
    };

```

## account.cpp

```

#include "account.h"

account::account() {
    x2=0;
    x4=-1;
    x6=10;
    x7=1000;
    k=0;
    num=2;
}

```

```
int account::open(int x, int y, int z){
    if ((x>0)&&(x4==1)) {
        x1=x;
        x3=y;
        x5=z;
        x4=0;
        return 0;
    };
    return -1;
}

int account::pin(int x) {
    if (x4!=1) return -1;
    if (x==x3) {x4=2; return 0;}
    else k++;
    if (k>=num) x4=0;
    return -1;
}

int account::logout() {
    if ((x4==0)|| (x2==1)) return -1;
    x4=0;
    return 0;
}

int account::login(int x) {
    if (x4!=0) return -1;
    if (x5==x) {x4=1; k=0; return 0;}
    return -1;
}

int account::balance() {
    if (x4!=2) return -1;
    return x1;
}

int account::lock(int x) {
    if (x4!=2) return -1;
    if (x!=x3) return -1;
    if (x2==0) {x2=1; return 0;}
    else return -1;
}

int account::unlock(int x) {
    if (x4!=2) return -1;
    if ((x2)&&(x==x3)) {x2=0;
    return 0;}
    else return -1;
}

int account::deposit(int d) {
    if (x4!=2) return -1;
    if (x2) {return -1;};
    if ((x1<x7)&&(d>0)) {
```



```

        x1=x1+d-x6;
        return 0;}
    else { if (d>0) {
        x1=x1+d;
        return 0; }
    }
    return -1;
}

int account::withdraw(int w) {
    if (x4!=2) return -1;
    if (x2) {return -1;};
    if ((x1>w)&&(w>0)) {
        if (x1<x7) {return -1;}
        else {x1=x1-w;};
        if (x1<x7) x1=x1-x6;
        return 0;
    }
    return -1;
}

```

Main.cpp

```

#include <iostream>
#include <iomanip>
#include <vector>
#include <algorithm>
#include <conio.h>
#include "account.h"
using namespace std;

account *accObj;

int * readInt() {
    string buffer;
    cin.clear(); cin.sync();
    cin >> buffer;
    cin.clear(); cin.sync();

    int *r = new int;
    if (sscanf(buffer.c_str(), "%d", r) ==1) {
        return r;
    }
    return nullptr;
}

int readKey() {
    return _getch();
}

void resetAccObj() {
    delete accObj;
    accObj = new account();
    accObj->test_set_x1(0);
    accObj->test_set_x3(0);
}

```

```

        accObj->test_set_x5(0);
    }

    // =====
    // Interface
    // =====

    void outputTitle() {
        cout
            << "          -----"
        << endl
            << "          *              CS 589 Project              *"
        << endl
            << "          *              Writen by              *"
        << endl
            << "          *              Xiaoyang Lu              *"
        << endl
            << "          -----"
        << endl << endl;
    }

    void showAccountStatus() {
        cout << "          account object state by calling all test methods" << endl <<
        endl
            << setw(20) << "Balance: " << setw(6) << accObj->test_get_x1() << endl
            << setw(20) << "Locked: " << setw(6) << accObj->test_get_x2() << endl
            << setw(20) << "Pin: " << setw(6) << accObj->test_get_x3() << endl
            << setw(20) << "EFSM State: " << setw(6) << accObj->test_get_x4() <<
        endl
            << setw(20) << "Account: " << setw(6) << accObj->test_get_x5() << endl
            << setw(20) << "Penalty: " << setw(6) << accObj->test_get_x6() << endl
            << setw(20) << "Minimum Balance:: " << setw(6) << accObj->test_get_x7()
        << endl
            << setw(20) << "Attempt Times: " << setw(6) << accObj->test_get_k() <<
        endl
            << setw(20) << "Maximum Attempts: " << setw(6) << accObj->test_get_num()
        << endl
            << setw(20) << "Current State: " << setw(6) << accObj->test_show_state()
        << endl << endl;
    }

    int uiFunc_open() {
        // Show all menu elements
        system("cls");
        outputTitle();
        showAccountStatus();

        // Guide
        cout << " ** Execute account::open() ** " << endl
            << " Please input the following value:" << endl

        // Initial balance.
            << " -- Initial balance: ";
        int *initialBalance = nullptr;
        while (nullptr == (initialBalance = readInt())) {
            cout << "          Invalid integer! Please re-input Initial balance: ";

```

```

    }

    // Pin number.
    cout << " -- Pin number: ";
    int *pinNumber = nullptr;
    while (nullptr == (pinNumber = readInt())) {
        cout << "      Invalid integer! Please re-input Pin number: ";
    }

    // Account Number.
    cout << " -- Account number: ";
    int *accNumber = nullptr;
    while (nullptr == (accNumber = readInt())) {
        cout << "      Invalid integer! Please re-input Account number: ";
    }

    int returnValue = accObj->open(*initialBalance, *pinNumber, *accNumber);
    cout << endl << "      The value returned by the method is: "
        << returnValue << endl << endl;
    showAccountStatus();
    cout << endl << "Press any key to continue.";

    return readKey();
}

int uiFunc_deposit() {
    // Show all menu elements
    system("cls");
    outputTitle();
    showAccountStatus();

    // Guide
    cout << " ** Execute account::deposit() ** " << endl
        << " Please input the following value:" << endl

    // Initial balance.
        << " -- Deposit amount: ";
    int *depositAmount = nullptr;
    while ((depositAmount = readInt()) == nullptr) {
        cout << "      Invalid integer! Please re-input Deposit amount: ";
    }

    int returnValue = accObj->deposit(*depositAmount);
    cout << endl << "      The value returned by the method is: "
        << returnValue << endl << endl;
    showAccountStatus();
    cout << endl << "Press any key to continue.";

    return readKey();
}

int uiFunc_withdraw() {
    // Show all menu elements
    system("cls");
    outputTitle();
    showAccountStatus();

```

```

// Guide
cout << " ** Execute account::withdraw() ** " << endl
    << " Please input the following value:" << endl;

// Initial balance.
    cout << " -- Withdraw amount: ";
int *withdrawAmount = nullptr;
while ((withdrawAmount = readInt()) == nullptr) {
    cout << " Invalid integer! Please re-input Withdraw amount: ";
}

int returnValue = accObj->withdraw(*withdrawAmount);
cout << endl << " The value returned by the method is: "
    << returnValue << endl << endl;
showAccountStatus();
cout << endl << "Press any key to continue.";

return readKey();
}

int uiFunc_balance() {
    // Show all menu elements
    system("cls");
    outputTitle();
    showAccountStatus();

    // Guide
    cout << " ** Execute account::balance() ** " << endl << endl;

    int returnValue = accObj->balance();
    cout << "OUTPUT: " << returnValue << endl << endl;
    cout << endl << " The value returned by the method is: "
        << returnValue << endl << endl;
    showAccountStatus();
    cout << endl << "Press any key to continue.";

    return readKey();
}

int uiFunc_lock() {
    // Show all menu elements
    system("cls");
    outputTitle();
    showAccountStatus();

    // Guide
    cout << " ** Execute account::lock() ** " << endl
        << " Please input the following value:" << endl;

    // Initial balance.
    cout << " -- Pin number: ";
    int *pinNumber = nullptr;
    while ((pinNumber = readInt()) == nullptr) {
        cout << " Invalid integer! Please re-input Pin number: ";
    }
}

```

```
int returnValue = accObj->lock(*pinNumber);
cout << endl << "    The value returned by the method is: "
    << returnValue << endl << endl;
showAccountStatus();
cout << endl << "Press any key to continue.";

return readKey();
}

int uiFunc_unlock() {
    // Show all menu elements
    system("cls");
    outputTitle();
    showAccountStatus();

    // Guide
    cout << " ** Execute account::unlock() ** " << endl
        << " Please input the following value:" << endl;

    // Initial balance.
    cout << " -- Pin number: ";
    int *pinNumber = nullptr;
    while ((pinNumber = readInt()) == nullptr) {
        cout << "    Invalid integer! Please re-input Pin number: ";
    }

    int returnValue = accObj->unlock(*pinNumber);
    cout << endl << "    The value returned by the method is: "
        << returnValue << endl << endl;
    showAccountStatus();
    cout << endl << "Press any key to continue.";

    return readKey();
}

int uiFunc_login() {
    // Show all menu elements
    system("cls");
    outputTitle();
    showAccountStatus();

    // Guide
    cout << " ** Execute account::login() ** " << endl
        << " Please input the following value:" << endl;

    // Initial balance.
    cout << " -- Account number: ";
    int *accNumber = nullptr;
    while ((accNumber = readInt()) == nullptr) {
        cout << "    Invalid integer! Please re-input Account number: ";
    }

    int returnValue = accObj->login(*accNumber);
    cout << endl << "    The value returned by the method is: "
        << returnValue << endl << endl;
```

```
        showAccountStatus();
        cout << endl << "Press any key to continue.";

        return readKey();
    }

    int uiFunc_pin() {
        // Show all menu elements
        system("cls");
        outputTitle();
        showAccountStatus();

        // Guide
        cout << " ** Execute account::pin() ** " << endl
              << " Please input the following value:" << endl;

        // Initial balance.
        cout << " -- Pin number: ";
        int *pinNumber = nullptr;
        while ((pinNumber = readInt()) == nullptr) {
            cout << " Invalid integer! Please re-input Pin number: ";
        }

        int returnValue = accObj->pin(*pinNumber);
        cout << endl << " The value returned by the method is: "
              << returnValue << endl << endl;
        showAccountStatus();
        cout << endl << "Press any key to continue.";

        return readKey();
    }

    int uiFunc_logout() {
        // Show all menu elements
        system("cls");
        outputTitle();
        showAccountStatus();

        // Guide
        cout << " ** Execute account::logout() ** " << endl << endl;

        int returnValue = accObj->logout();
        cout << endl << " The value returned by the method is: "
              << returnValue << endl << endl;
        showAccountStatus();
        cout << endl << "Press any key to continue.";

        return readKey();
    }

    bool uiMainMenu() {
        // Show all menu elements
        system("cls");
        outputTitle();
```

```

showAccountStatus();

cout
    << "          DRIVER for the account" << endl << endl
    << "    0. open" << endl
    << "    1. deposit" << endl
    << "    2. withdraw" << endl
    << "    3. balance" << endl
    << "    4. lock" << endl
    << "    5. unlock" << endl
    << "    6. login" << endl
    << "    7. pin" << endl
    << "    8. logout" << endl << endl
    << "        Testing-related functions" << endl << endl
    << "    a. reset account object" << endl;

cout << endl << "        q. quit" << endl << endl << ">>>";

// Make choice
switch (readKey()) {
case '0':
    uiFunc_open();
    break;
case '1':
    uiFunc_deposit();
    break;
case '2':
    uiFunc_withdraw();
    break;
case '3':
    uiFunc_balance();
    break;
case '4':
    uiFunc_lock();
    break;
case '5':
    uiFunc_unlock();
    break;
case '6':
    uiFunc_login();
    break;
case '7':
    uiFunc_pin();
    break;
case '8':
    uiFunc_logout();
    break;

case 'a':
    // Reset account object
    resetAccObj();
    break;

case 'q':
    return false;
}

```

```
        break;
    }

    return true;
}

// =====
// Program Entrance
// =====

int main(int argc, char *argv[]) {
    accObj = new account();
    accObj->test_set_x1(0);
    accObj->test_set_x3(0);
    accObj->test_set_x5(0);

    while (uiMainMenu()) { }

    return 0;
}
```