

JackPot ATM

OOPT Stage 2050/2060

<Implementation & Unit test>

Team 5

Date

2018-05-18

201312259 백만일

201112052 방민석

201211383 조영래

INDEX

Activity 2051. Implement Class & Method Definitions

Activity 2052. Implement Windows

Activity 2055. Write Unit Test Code

Activity 2061. Unit Testing

Activity 2063. System Testing

Activity 2051. Implement Class & Method Definitions

Type	Class
Name	System_Control
Purpose	사용자(User 혹은 Manager)로부터 데이터를 입력받아 저장하고 실제 데이터의 처리가 동작하는 Bank, Account 등 다양한 클래스와 데이터를 주고받는 클래스
Overview	N/A
Cross Reference	jackPot, withdraw, deposit,remittance, view_account_detail, manage_ATM
Exceptional Course of Event	N/A

Type	Method
Name	input_menu
Purpose	사용자로부터 진입하고자하는 ATM 의 기능을 입력받아 inputMenu 어트리뷰트에 저장하는 메소드
Overview	N/A
Cross Reference	withdraw, deposit,remittance, view_account_detail, manage_ATM
Input	void
Output	void
Abstract Operation	사용자가 입력한 기능을 inputMenu 어트리뷰트에 저장한다
Exceptional Course of Event	N/A

Type	Method
Name	input_ID

Purpose	사용자로부터 ID를 입력받아 inputID 어트리뷰트에 저장하는 메소드
Overview	N/A
Cross Reference	withdraw, deposit, remittance, view_account_detail
Input	void
Output	void
Abstract Operation	
Exceptional Course of Event	N/A

Type	Method
Name	input_amount
Purpose	사용자로부터 금액을 입력받아 inputAmount 어트리뷰트에 저장하는 메소드
Overview	N/A
Cross Reference	withdraw, deposit, remittance
Input	void
Output	void
Abstract Operation	사용자로부터 입력받은 값을 inputAmount 에 저장한다.
Exceptional Course of Event	N/A

Type	Method
Name	input_PW
Purpose	사용자로부터 비밀번호를 입력받아 inputPW 어트리뷰트에 저장하는 메소드
Overview	N/A

Cross Reference	withdraw, deposit,remittance, view_account_detail
Input	void
Output	void
Abstract Operation	사용자로부터 입력받은 값을 inputPW에 저장한다.
Exceptional Course of Event	N/A

Type	Method
Name	updateBalance
Purpose	사용자의 거래에 따른 계좌의 잔고를 변화시켜주는 메소드
Overview	N/A
Cross Reference	withdraw, deposit,remittance
Input	void
Output	void
Abstract Operation	사용자의 거래에 따라 Account.Balance를 변화시켜준다.
Exceptional Course of Event	N/A

Type	Method
Name	input_MID
Purpose	관리자의 ID 를 입력받아 inputID 어트리뷰트에 저장해주는 메소드
Overview	N/A
Cross Reference	manage_ATM
Input	void

Output	void
Abstract Operation	사용자의 입력을 inputID 에 저장한다.
Exceptional Course of Event	N/A

Type	Method
Name	input_POM
Purpose	관리자가 ATM 의 금액을 빼줄것인지 더해줄것인지를 선택받아 inputPOM 에 저장해 주는 메소드
Overview	N/A
Cross Reference	manage_ATM
Input	void
Output	void
Abstract Operation	사용자의 입력을 inputPOM 에 저장한다.
Exceptional Course of Event	N/A

Type	Method
Name	input_MPW
Purpose	관리자의 PW 를 입력받아 inputPW 어트리뷰트에 저장해주는 메소드
Overview	N/A
Cross Reference	manage_ATM
Input	void
Output	void

Abstract Operation	관리자의 입력력을 inputPW 에 저장해준다.
Exceptional Course of Event	N/A

Type	Class
Name	Manager
Purpose	ATM 의 관리자가 ATM 기기의 접근할 때 실제 관리자인지 확인할 수 있도록 관리자의 정보(ID 와 PW) 를 가지고 있는 클래스
Overview	N/A
Cross Reference	manage_ATM
Exceptional Course of Event	N/A

Type	Method
Name	Manager.checkID
Purpose	입력된 관리자의 ID 와 실제 관리자 ID 를 비교하는 메소드
Overview	N/A
Cross Reference	manage_ATM
Input	inputID:int
Output	Boolean
Abstract Operation	1.사용자로부터 입력된 inputMID를 실제 ManagerIID 와 비교 2.참인지 거짓인지를 반환
Exceptional Course of Event	N/A

Type	Method
------	--------

Name	Manager.CheckPW
Purpose	입력된 관리자의 PW 와 실제 관리자 PW를 비교하는 메소드
Overview	N/A
Cross Reference	manage_ATM
Input	inputPW:int
Output	Boolean
Abstract Operation	1.사용자로부터 입력된 inputMPW 를 실제 관리자의 ManagerPW와 비교 2.참인지 거짓인지를 반환
Exceptional Course of Event	N/A

Type	Class
Name	Timer
Purpose	ATM 의 시간의 실시간으로 체크하는 클래스
Overview	N/A
Cross Reference	Jackpot
Exceptional Course of Event	

Type	Method
Name	start
Purpose	ATM 타이머가 실시간으로 돌아갈 수 있도록 하는 쓰레드 메소드
Overview	N/A
Cross Reference	Jackpot
Input	void

Output	void
Abstract Operation	Timer 메소드가 상속받은 쓰레드를 실행
Exceptional Course of Event	N/A

Type	Method
Name	checktime
Purpose	ATM의 타이머가 Jackpot 넘버를 갱신하기 위한 12시(자정)인지를 확인하는 메소드
Overview	N/A
Cross Reference	Jackpot
Input	void
Output	Boolean
Abstract Operation	1. ATM 타이머의 시간이 12시(자정)인지 확인 2. 참 혹은 거짓을 반환
Exceptional Course of Event	N/A

Type	Method
Name	updateJackpotNum
Purpose	ATM 타이머가 12시일 때마다 JackpotNum 을 갱신해주는 메소드
Overview	N/A
Cross Reference	Jackpot
Input	void
Output	void

Abstract Operation	1. JackpotNum을 새로운 랜덤숫자로 갱신
Exceptional Course of Event	N/A

Type	Method
Name	checkJackpot
Purpose	출금 사용자의 순번과 JackpotNum을 비교하여 당첨여부를 결정하는 메소드
Overview	N/A
Cross Reference	Jackpot
Input	UserCount:int
Output	Boolean
Abstract Operation	1.UserCount 와 JackpotNum 을 비교한다. 2. 참 혹은 거짓을 반환
Exceptional Course of Event	N/A

Type	Class
Name	MoneyCount
Purpose	ATM 기기 및 사용자의 현금 처리를 위하여 각 화폐별 수량을 가지고 있는 클래스
Overview	N/A
Cross Reference	jackPot, withdraw, deposit,remittance, manage_ATM
Exceptional Course of Event	N/A

Type	Method
------	--------

Name	ATMBalanceUpdate
Purpose	ATM 자체가 가지고 있는 각 화폐별 잔고를 업데이트 해주는 메소드
Overview	N/A
Cross Reference	jackPot, withdraw, deposit, manage_ATM
Input	inputMoney: int, inputcount: int, inputPOM: int
Output	void
Abstract Operation	1.사용자로부터 입력받은 화폐의 input_count 를 Count화폐 에 반영
Exceptional Course of Event	N/A

Type	Class
Name	Bank
Purpose	시스템 내부에서 사용 가능한 은행의 정보를 가지고 있는 클래스
Overview	N/A
Cross Reference	withdraw, deposit,remittance
Exceptional Course of Event	N/A

Type	Method
Name	searchID
Purpose	Bank 내부에 개설된 통장 혹은 카드의 정보를 찾는 메소드
Overview	N/A
Cross Reference	withdraw, deposit,remittance, view_account_detail
Input	inputID:int

Output	int
Abstract Operation	Account 에 있는 checkID 를 호출
Exceptional Course of Event	N/A

Type	Method
Name	Bank.checkPW
Purpose	사용자로부터 입력받은 PW 를 계좌 혹은 카드의 비밀번호와 비교하는 메소드
Overview	N/A
Cross Reference	withdraw, deposit,remittance, view_account_detail
Input	inputPW: int
Output	int
Abstract Operation	Account에 있는 checkPW를 호출
Exceptional Course of Event	N/A

Type	Method
Name	balanceCount
Purpose	사용자로부터 입력받은 금액과 통장 잔고를 확인하는 메소드
Overview	N/A
Cross Reference	withdraw, deposit,remittance,
Input	inputMoney: int
Output	int
Abstract	Account 에 있는 balanceAccount 호출

Operation	
Exceptional Course of Event	N/A

Type	Method
Name	getAccountBalance
Purpose	사용자의 계좌 잔고를 받아오는 메소드
Overview	N/A
Cross Reference	view_account_detail
Input	void
Output	void
Abstract Operation	Account에 있는 Balance를 호출
Exceptional Course of Event	N/A

Type	Class
Name	Account
Purpose	실제 고객의 계좌에 대한 정보를 저장하는 클래스
Overview	N/A
Cross Reference	withdraw, deposit, remittance, view_account_detail
Exceptional Course of Event	N/A

Type	Method
Name	Account.checkID

Purpose	사용자로부터 입력된 ID 계좌번호를 찾는 메소드
Overview	N/A
Cross Reference	withdraw, deposit, remittance, view_account_detail
Input	inputID: int
Output	int
Abstract Operation	inputID와 일치하는 계좌번호를 찾음
Exceptional Course of Event	N/A

Type	Method
Name	searchCardID
Purpose	사용자가 입력한 ID가 계좌번호가 아닌 경우 일치하는 카드번호가 있는지 찾기 위한 메소드
Overview	N/A
Cross Reference	withdraw, remittance, deposit, view_account_detail
Input	inputID: int
Output	int
Abstract Operation	inputID 와 일치하는 Card의 ID를 찾음
Exceptional Course of Event	N/A

Type	Method
Name	Account.checkPW
Purpose	사용자로부터 입력받은 password와 카드의 비밀번호를 비교하는 메소드

Overview	N/A
Cross Reference	withdraw, remittance, view_account_detail
Input	inputPW: int
Output	int
Abstract Operation	inputPW와 Card 클래스에 저장된 비밀번호가 같은지 확인한다
Exceptional Course of Event	N/A

Type	Method
Name	searchCard
Purpose	사용자가 입력한 비밀번호를 카드의 비밀번호와 비교하도록 인자를 넘겨주는 메소드
Overview	N/A
Input	inputPW: int
Output	int
Abstract Operation	inputPW를 Bank 클래스에서 Account 클래스로 전달한다
Cross Reference	withdraw, remittance, view_account_detail
Exceptional Course of Event	N/A

Type	Method
Name	balanceAccount
Purpose	Account에 저장되어 있는 잔고를 변화시키기 위한 메소드
Overview	N/A

Cross Reference	withdraw,deposit,remittance
Input	inputMoney: int
Output	int
Abstract Operation	Account의 balance를 inputMoney 만큼 변화시킨다
Exceptional Course of Event	N/A

Type	Method
Name	checkBalance
Purpose	사용자로부터 입력받은 금액이 잔고보다 적은지 체크하는 메소드
Overview	N/A
Cross Reference	withdraw,remittance
Input	inputMoney: int
Output	int
Abstract Operation	inputMoney가 계좌의 잔액보다 적은지 비교한다
Exceptional Course of Event	N/A

Type	Method
Name	getBalance
Purpose	Bank 클래스에서 Account의 balance를 받아오기 위한 메소드
Overview	N/A
Cross Reference	view_account_detail
Input	void

Output	int
Abstract Operation	Account의 balance 값을 받아온다
Exceptional Course of Event	N/A

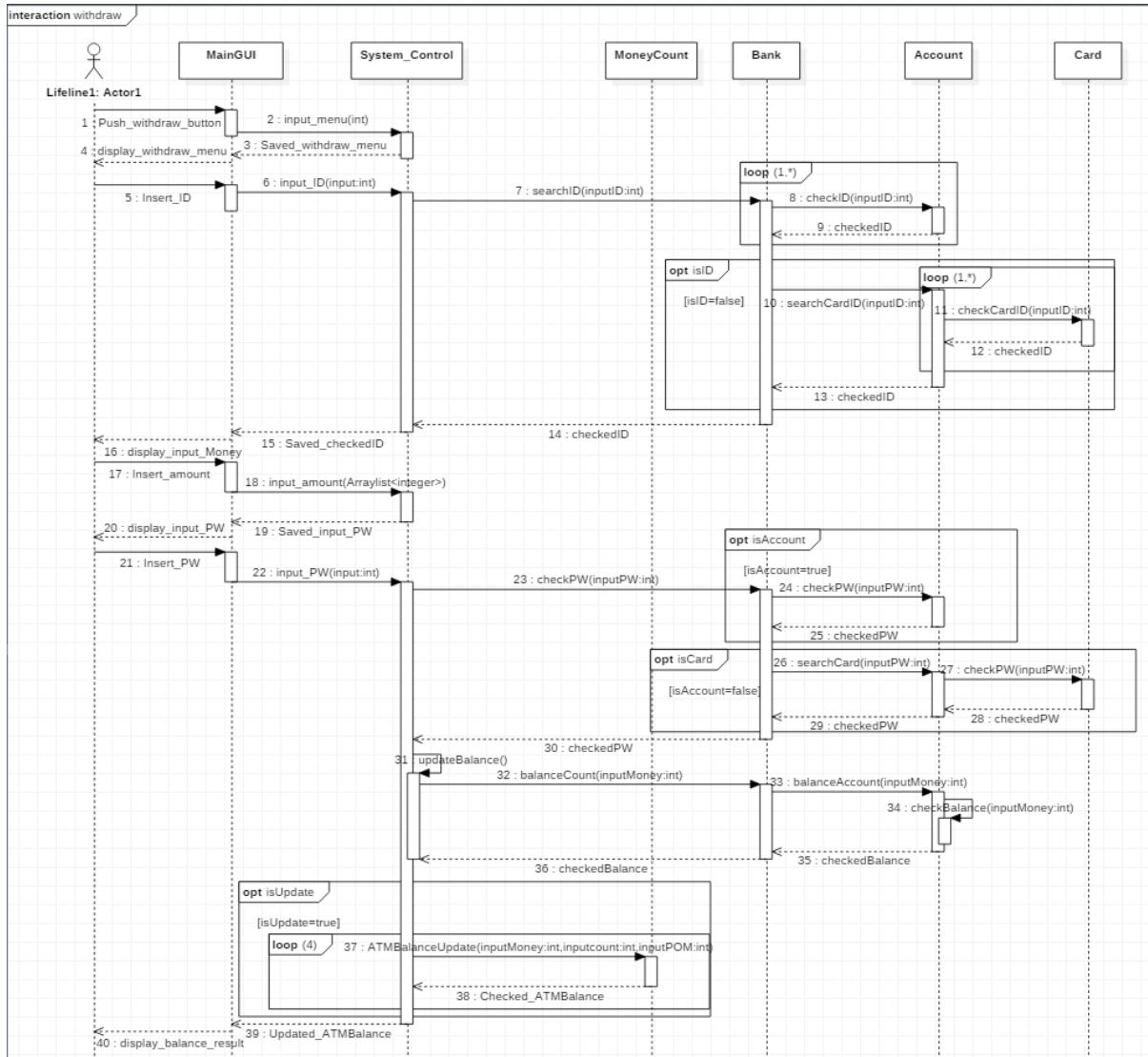
Type	Class
Name	Card
Purpose	카드번호와 비밀번호를 가지고 사용자에게서 입력받은 ID, Password와 비교하기 위한 클래스
Overview	N/A
Cross Reference	withdraw, deposit, remittance, view_account_detail
Exceptional Course of Event	N/A

Type	Method
Name	checkCardID
Purpose	사용자로부터 입력받은 ID가 카드의 ID와 일치하는지 확인하기 위한 method
Overview	N/A
Cross Reference	withdraw, deposit, remittance, view_account_detail
Input	inputID: int
Output	int
Abstract Operation	inputID가 Card의 ID와 일치하는지 비교한다
Exceptional Course of Event	N/A

Type	Method
Name	Card.checkPW
Purpose	사용자로부터 입력받은 password가 Card의 password와 일치하는지 확인하는 method
Overview	N/A
Cross Reference	withdraw,remittance, view_account_detail
Input	inputPW: int
Output	int
Abstract Operation	inputPW와 Card에 저장된 password가 같은지 비교한다
Exceptional Course of Event	N/A

Activity 2052. Implement Windows

1. Withdraw



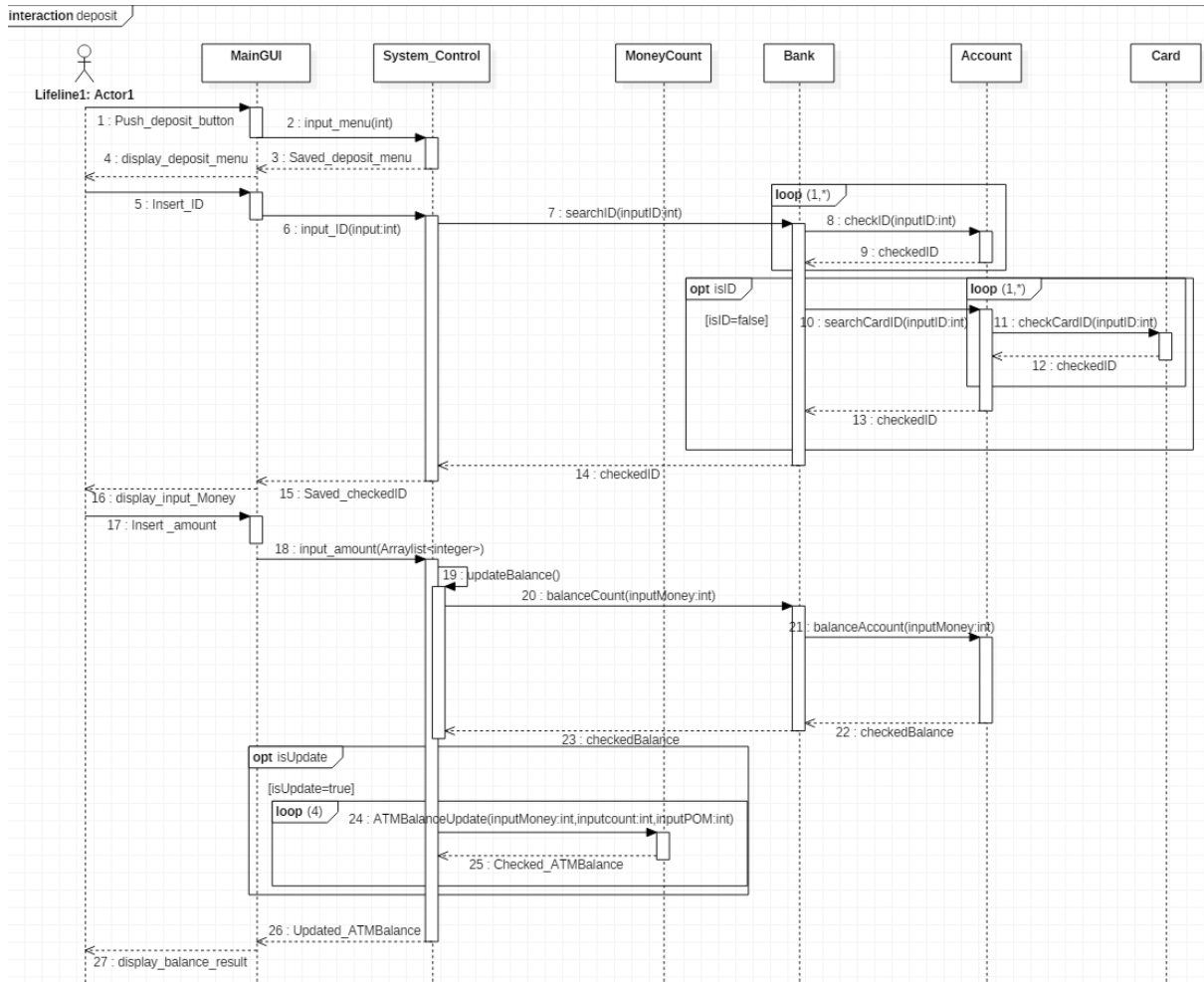
Name	Push_withdraw_button
Responsibilities	MainGUI의 withdraw 버튼을 누른다
Type	MainGUI
Cross Reference	R1.2
Notes	MainGUI의 withdraw 버튼을 누른다

Pre-Conditions	N/A
Post-Conditions	출금 창을 화면에 출력한다
Name	Insert_ID
Responsibilities	MainGUI에 ID를 입력한다
Type	MainGUI
Cross Reference	R1.2
Notes	MainGUI에 ID를 입력한다
Pre-Conditions	input_menu에 withdraw값이 들어있어야 한다
Post-Conditions	금액 입력 창을 화면에 출력한다
Name	Insert_amount
Responsibilities	MainGUI에 금액을 입력한다
Type	MainGUI
Cross Reference	R1.2
Notes	MainGUI에 금액을 입력한다
Pre-Conditions	ID가 확인되어 있어야 한다
Post-Conditions	비밀번호 입력 창을 화면에 출력한다

Name	Insert_PW
Responsibilities	MainGUI에 비밀번호를 입력한다
Type	MainGUI
Cross Reference	R1.2

Notes	MainGUI에 비밀번호를 입력한다
Pre-Conditions	금액이 입력되어 있어야 한다
Post-Conditions	거래 내역을 출력한다

2. Deposit

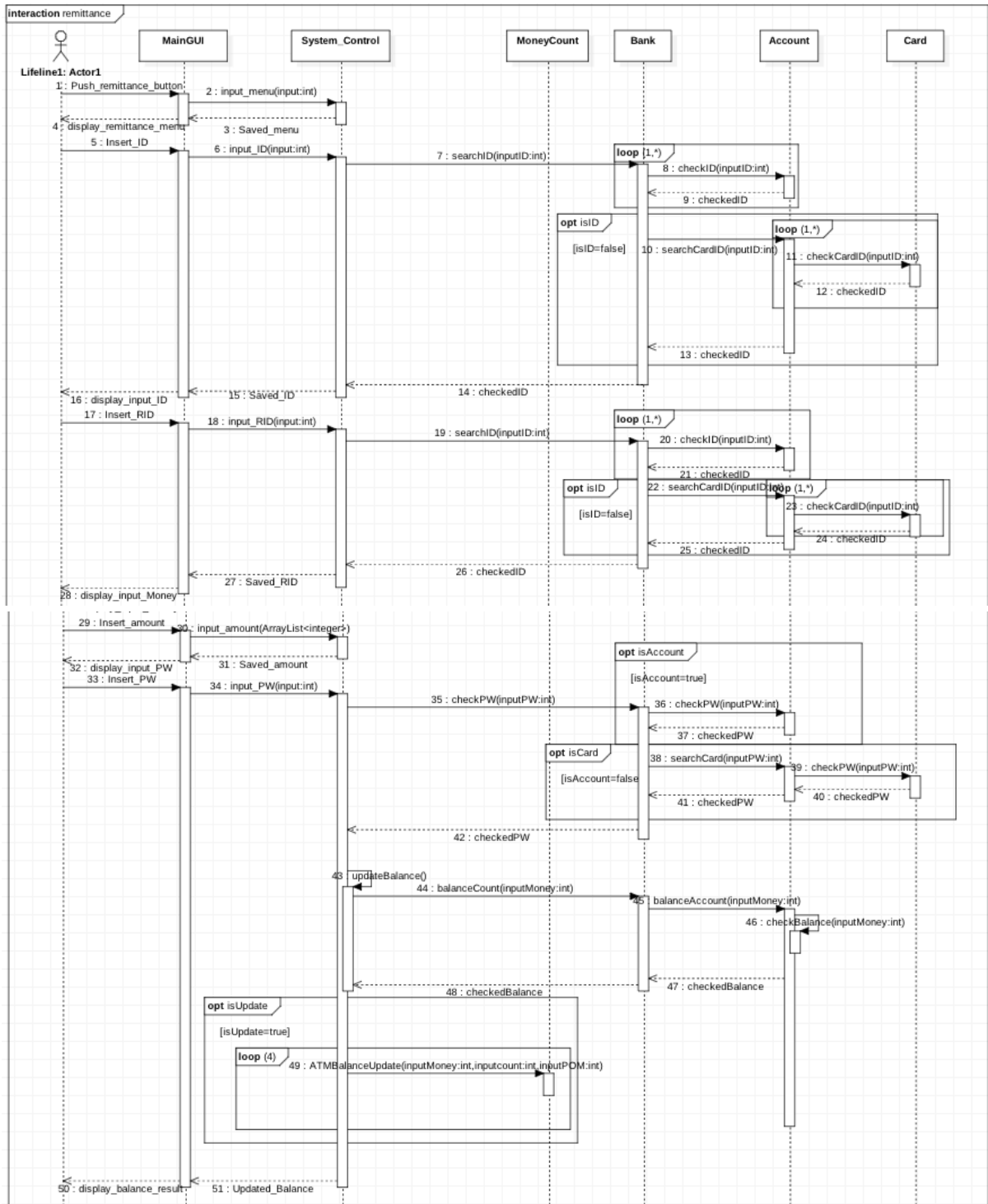


Name	Push_deposit_button
Responsibilities	MainGUI의 deposit 버튼을 누른다
Type	MainGUI
Cross Reference	R.2
Notes	MainGUI의 deposit 버튼을 누른다
Pre-Conditions	N/A
Post-Conditions	입금 창을 화면에 출력한다

Name	Insert_ID
Responsibilities	MainGUI에 ID를 입력한다
Type	MainGUI
Cross Reference	R.2
Notes	MainGUI에 ID를 입력한다
Pre-Conditions	input_menu에 deposit값이 들어있어야 한다
Post-Conditions	금액 입력 창을 화면에 출력한다

Name	Insert_amount
Responsibilities	MainGUI에 금액을 입력한다
Type	MainGUI
Cross Reference	R.2
Notes	MainGUI에 금액을 입력한다
Pre-Conditions	ID가 확인되어 있어야 한다
Post-Conditions	비밀번호 입력 창을 화면에 출력한다

3. Remittance



Name	Push_remittance_button
------	------------------------

Responsibilities	MainGUI의 송금버튼을 누른다
Type	MainGUI
Cross Reference	R.5
Notes	MainGUI의 송금 버튼을 누른다
Pre-Conditions	N/A
Post-Conditions	ID 입력 화면을 출력한다.

Name	Insert_ID
Responsibilities	MainGUI를 통해 사용자의 ID를 입력한다.
Type	MainGUI
Cross Reference	R.5
Notes	MainGUI를 통해 사용자의 ID를 입력한다.
Pre-Conditions	InputMenu의 값에 remittance(3)이 입력된 상태여야 한다.
Post-Conditions	ID 입력 화면을 출력한다.

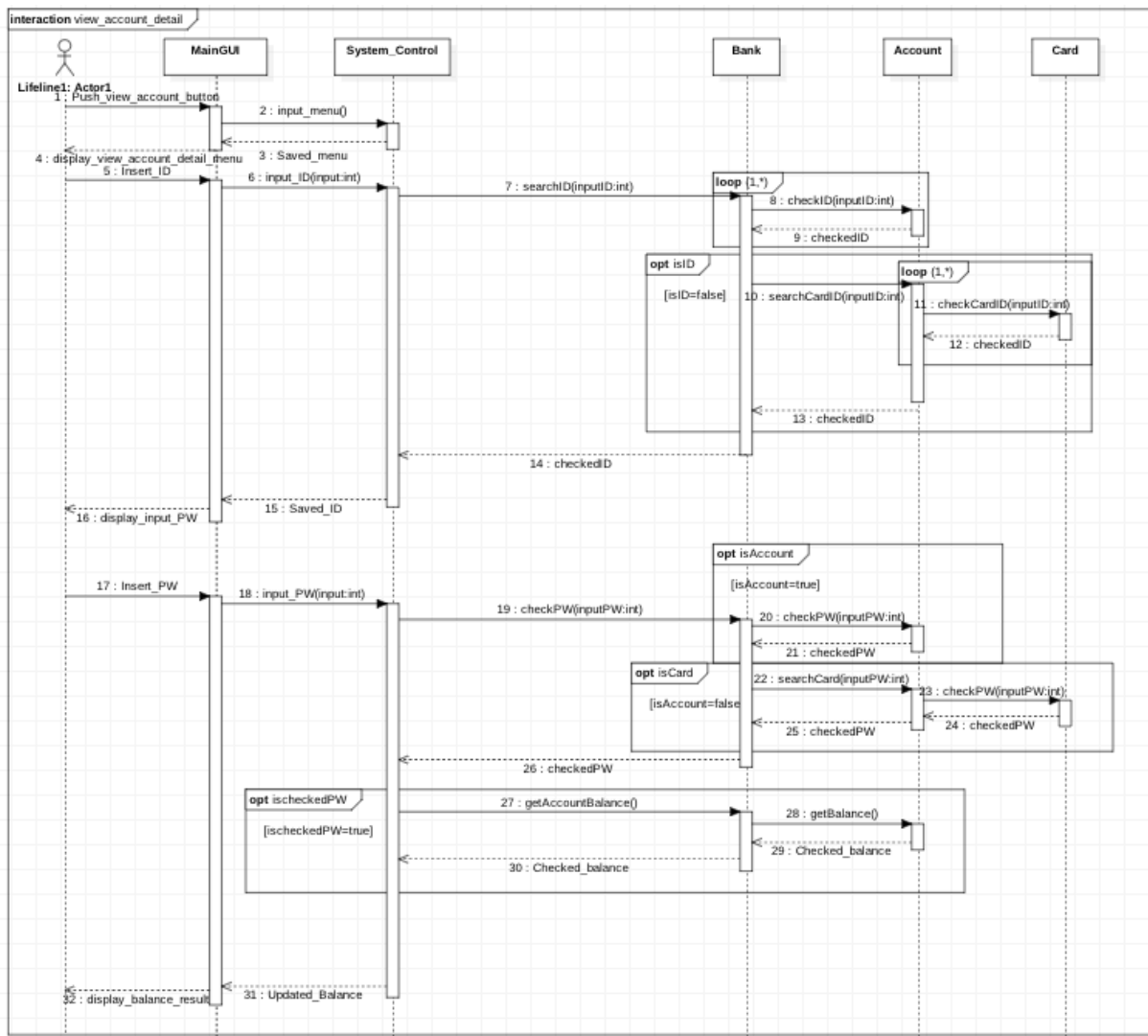
Name	Insert_RID
Responsibilities	MainGUI의 조회 버튼을 누른다
Type	MainGUI
Cross Reference	R.5
Notes	MainGUI의 조회 버튼을 누른다
Pre-Conditions	N/A

Post-Conditions	ID 입력 화면을 출력한다.
-----------------	-----------------

Name	Insert_amount
Responsibilities	MainGUI에 금액을 입력한다
Type	MainGUI
Cross Reference	R.5
Notes	MainGUI에 금액을 입력한다
Pre-Conditions	ID가 확인되어 있어야 한다
Post-Conditions	비밀번호 입력 창을 화면에 출력한다

Name	Insert_PW
Responsibilities	MainGUI에 비밀번호를 입력한다
Type	MainGUI
Cross Reference	R.5
Notes	MainGUI에 비밀번호를 입력한다
Pre-Conditions	금액이 입력되어 있어야 한다
Post-Conditions	거래 내역을 출력한다

4. View Account Detail



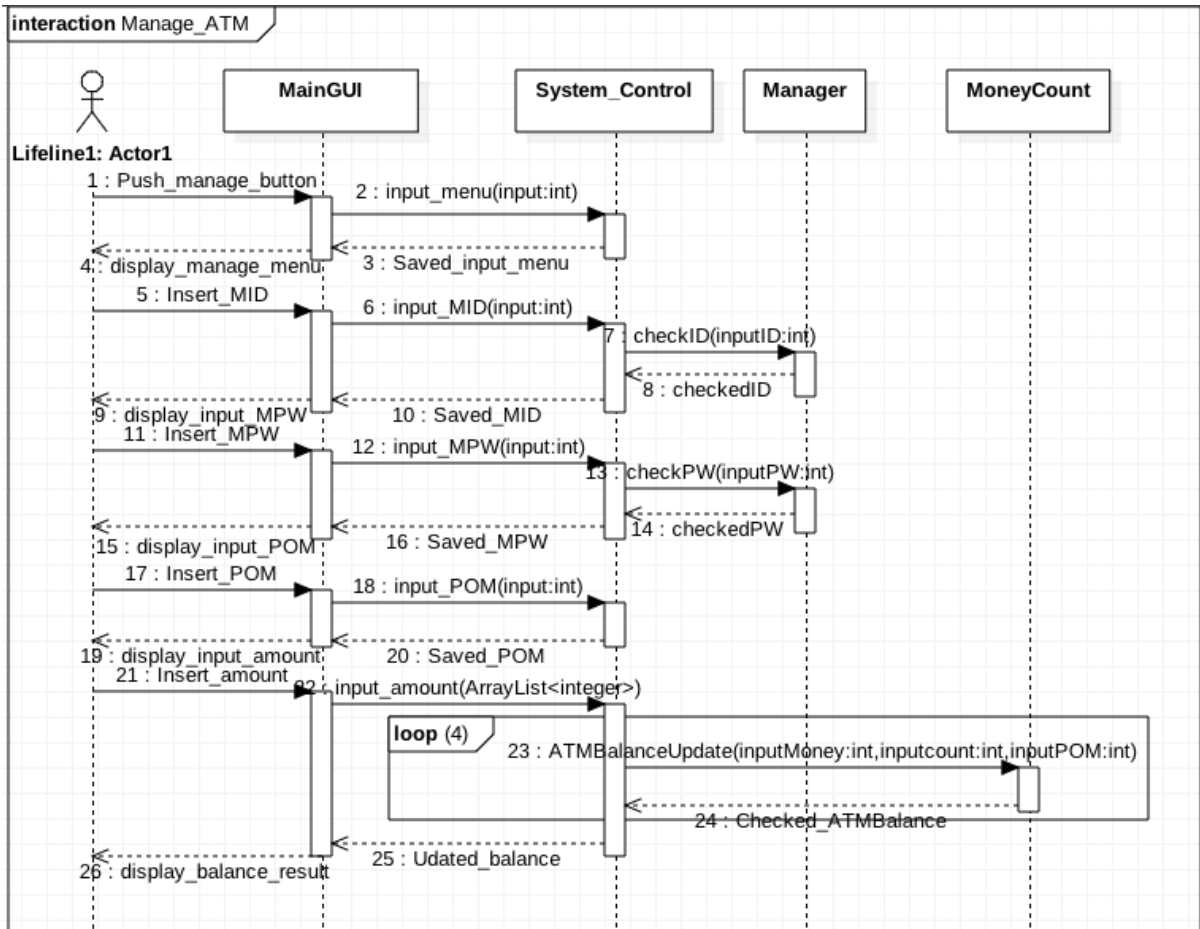
Name	Push_view_account_button
Responsibilities	MainGUI의 조회 버튼을 누른다
Type	MainGUI
Cross Reference	R.5
Notes	MainGUI의 조회 버튼을 누른다
Pre-Conditions	N/A

Post-Conditions	ID 입력 화면을 출력한다.
-----------------	-----------------

Name	Insert_ID
Responsibilities	MainGUI를 통해 사용자의 ID를 입력한다.
Type	MainGUI
Cross Reference	R.5
Notes	MainGUI를 통해 사용자의 ID를 입력한다.
Pre-Conditions	InputID에 사용자로부터 입력된 ID 가 저장된 상태여야한다.
Post-Conditions	매니저 PW 입력 화면을 출력한다

Name	Insert_PW
Responsibilities	MainGUI를 통해 사용자의 PW를 입력한다.
Type	MainGUI
Cross Reference	R.5
Notes	MainGUI를 통해 사용자의 PW를 입력한다.
Pre-Conditions	N/A
Post-Conditions	매니저 PW 입력 화면을 출력한다

5. Manage



Name	Push_manage_button
Responsibilities	MainGUI의 manage 버튼을 누른다
Type	MainGUI
Cross Reference	R.5
Notes	MainGUI의 manage 버튼을 누른다
Pre-Conditions	N/A
Post-Conditions	매니저 PW 입력 화면을 출력한다

Name	Insert_MID
------	------------

Responsibilities	관리자의 ID 를 입력한다
Type	MainGUI
Cross Reference	R.5
Notes	MainGUI를 통해 매니저의 ID를 입력한다.
Pre-Conditions	inputMenu에 Manage 가 저장된 상태여야 한다.
Post-Conditions	매니저 ID 입력 화면을 출력한다

Name	Insert_MPW
Responsibilities	관리자의 PW 를 입력한다.
Type	MainGUI
Cross Reference	R.5
Notes	MainGUI를 통해 매니저의 PW를 입력한다.
Pre-Conditions	관리자의 PW가 유효한지 확인된 상태여야한다.
Post-Conditions	입금 혹은 출력을 선택하는 화면을 출력한다

Name	Insert_POM
Responsibilities	MainGUI에서 출금 또는 입금 버튼을 누른다
Type	MainGUI
Cross Reference	R.5
Notes	MainGUI의 입금 또는 출금 버튼을

	누른다
Pre-Conditions	N/A
Post-Conditions	입금 또는 출금 화면을 출력한다.

Name	Insert_amount
Responsibilities	MainGUI를 통해 매니저의 입금 혹은 출금하고자 하는 금액을 입력한다.
Type	MainGUI
Cross Reference	R.5
Notes	MainGUI를 통해 매니저의 입금 혹은 출금하고자 하는 금액을 입력한다.
Pre-Conditions	N/A
Post-Conditions	매니저 ID 입력 화면을 출력한다

Activity 2055. Write Unit Test Code

```
1 package DataBase;
2
3 import static org.junit.Assert.*;
4
5
6
7 public class BankTest {
8
9     @Test //아이디 제대로 입력
10    public void searchIDtest1() {
11        //fail("Not yet implemented");
12        Bank test = new Bank(1);
13        assertEquals(1, test.searchID(325, 1));
14    }
15
16    @Test //아이디 다르게 입력
17    public void searchIDtest2() {
18        Bank test = new Bank(1);
19        assertEquals(0, test.searchID(324, 1));
20    }
21 }
22
23
```

```
1 package DataBase;
2
3 import static org.junit.Assert.*;
4
5
6
7 public class ManagerTest {
8
9     @Test
10    public void input_MID_TrueTest() {
11        Manager test = new Manager(123,235);
12        boolean output = test.checkID(123);
13        assertEquals(true,output);
14    }
15    @Test
16    public void input_MID_FalseTest() {
17        Manager test = new Manager(123,235);
18        boolean output = test.checkID(11111);
19        assertEquals(false,output);
20    }
21
22    @Test
23    public void input_MPW_TrueTest() {
24        Manager test = new Manager(123,235);
25        boolean output = test.checkPW(235);
26        assertEquals(true,output);
27    }
28    @Test
29    public void input_MPW_FalseTest() {
30        Manager test = new Manager(123,235);
31        boolean output = test.checkPW(111111);
32        assertEquals(false,output);
33    }
34 }
```

```
1 package System;
2
3 import static org.junit.Assert.*;
4
5
6
7 public class System_ControlTest {
8
9     @Test
10    public void input_ID_TrueTest() { //시스템에 등록되어 있는 계좌를 확인하는 테스트
11        System_Control test = new System_Control();
12        boolean output = test.input_ID(325);
13        assertEquals(true,output);
14    }
15    @Test
16    public void input_ID_FalseTest() {
17        System_Control test = new System_Control();
18        boolean output = test.input_ID(111111);
19        assertEquals(false,output);
20    }
21    }
22    @Test
23    public void input_RID_TrueTest() {
24        System_Control test = new System_Control();
25        boolean output = test.input_RID(325);
26        assertEquals(true,output);
27    }
28    }
29    @Test
30    public void input_RID_FalseTest() {
31        System_Control test = new System_Control();
32        boolean output = test.input_RID(11111);
33        assertEquals(false,output);
34    }
35    }
36 }
```

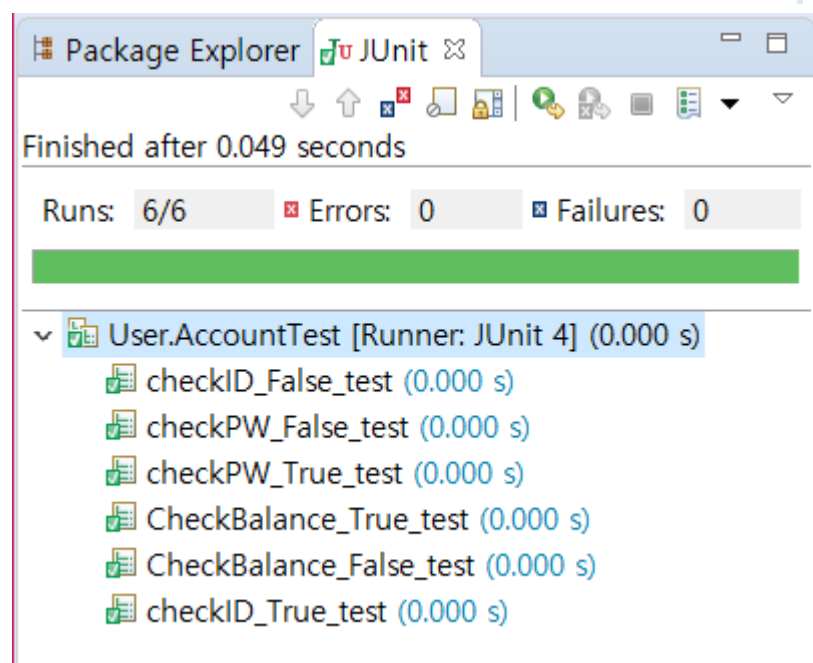
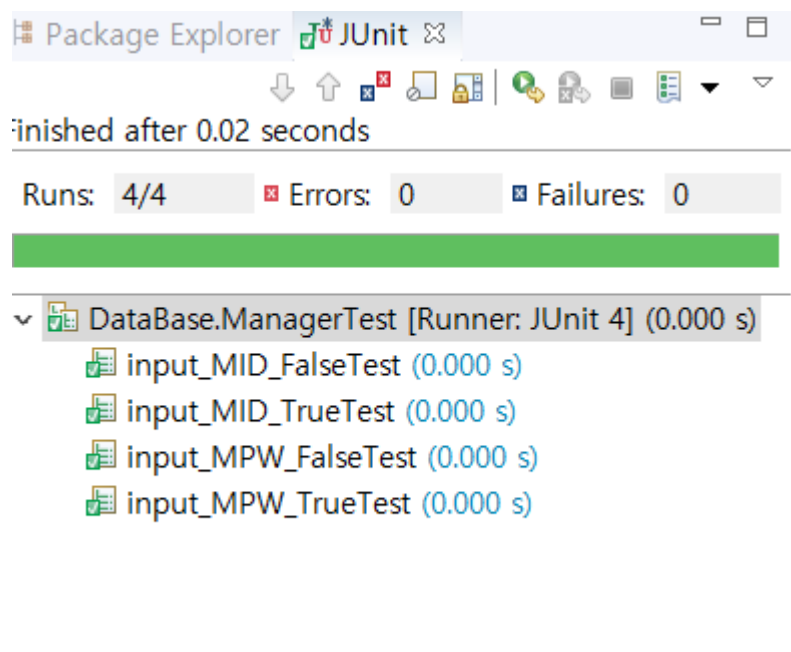
```
36 @Test
37 public void input_MID_TrueTest() {
38     System_Control test = new System_Control();
39     boolean output = test.input_MID(123);
40     assertEquals(true,output);
41 }
42 @Test
43 public void input_MID_FalseTest() {
44     System_Control test = new System_Control();
45     boolean output = test.input_MID(11111);
46     assertEquals(false,output);
47 }
48
49 @Test
50 public void input_MPW_TrueTest() {
51     System_Control test = new System_Control();
52     boolean output = test.input_MPW(235);
53     assertEquals(true,output);
54 }
55 @Test
56 public void input_MPW_FalseTest() {
57     System_Control test = new System_Control();
58     boolean output = test.input_MPW(111111);
59     assertEquals(false,output);
60 }
61
62
63 }
```

```

1 package User;
2
3 import static org.junit.Assert.*;
4
5
6
7 public class AccountTest {
8
9     @Test
10    public void checkID_True_test() {
11        Account test = new Account(123,456,"신한",10000,1);
12        int output = test.checkID(123);
13        assertEquals(1,output);
14    }
15
16    @Test
17    public void checkID_False_test() {
18        Account test = new Account(123,456,"신한",10000,1);
19        int output = test.checkID(11111);
20        assertEquals(0,output);
21    }
22
23    @Test
24    public void checkPW_True_test() {
25        Account test = new Account(123,456,"신한",10000,1);
26        int output = test.checkPW(456);
27        assertEquals(1,output);
28    }
29
30    @Test
31    public void checkPW_False_test() {
32        Account test = new Account(123,456,"신한",10000,1);
33        int output = test.checkPW(11111);
34        assertEquals(0,output);
35    }
36
37    @Test
38    public void CheckBalance_True_test() {
39        Account test = new Account(123,456,"신한",10000,1);
40        int output = test.checkBalance(5000);
41        assertEquals(5000,output);
42    }
43    @Test
44    public void CheckBalance_False_test() {
45        Account test = new Account(123,456,"신한",10000,1);
46        int output = test.checkBalance(50000);
47        assertEquals(-1,output);
48    }
49
50 }
51

```

Activity 2061. Unit Testing



Package Explorer JUnit

Finished after 0.554 seconds

Runs: 8/8 Errors: 1 Failures: 0

- System.System_ControlTest [Runner: JUnit 4] (0.293 s)
 - input_RID_FalseTest (0.101 s)
 - input_MID_FalseTest (0.010 s)
 - input_MID_TrueTest (0.008 s)
 - input_MPW_FalseTest (0.010 s)
 - input_RID_TrueTest (0.013 s)
 - input_MPW_TrueTest (0.010 s)
 - input_ID_TrueTest (0.131 s)
 - input_ID_FalseTest (0.010 s)

Package Explorer JUnit

Finished after 0.037 seconds

Runs: 2/2 Errors: 0 Failures: 0

- DataBase.BankTest [Runner: JUnit 4] (0.000 s)
 - searchIDtest1 (0.000 s)
 - searchIDtest2 (0.000 s)

Activity 2063. System Testing

Test Number	Test 항목	Description	Use Case	Pass / Fail
1-1	input_menu Test	사용자가 입력한 해당 메뉴 화면으로 올바르게 진입하는지 확인한다.	R1.2 ,R.2, R.3, R.4, R.5	Pass
2-1	input_ID	사용자가 입력한 계좌 혹은 카드번호가 유효하지 않으면 오류메시지를 출력하는지 확인한다.	R1.2, R.2,R.3,R.4	Pass
2-2	input_ID	사용자가 입력한 계좌 혹은 카드번호가 유효하면 다음 단계로 넘어가는지 확인한다.	R1.2, R.2,R.3,R.4	Pass
3-1	input_PW	사용자가 입력한 비밀번호가 계좌번호의 비밀번호가 아니면 오류메시지를 출력하는지 확인한다	R1.2, R.3, R.4	Pass
3-2	input_PW	사용자가 입력한 비밀번호가 계좌번호의 비밀번호가 맞으면 다음단계로 넘어가는지 확인한다.	R1.2, R.3, R.4	Pass
4-1	input_amount	출금의 경우 사용자가 입력한 금액이 계좌 잔고보다 많으면 오류메시지를 출력하는지 확인한다.	R1.2	Pass
4-2	input_amount	출금의 경우 사용자가 입력한 금액이 ATM 잔고보다 많으면 오류메시지를 출력하는지 확인한다.	R1.2	Pass
4-3	input_amount	송금의 경우 사용자가 입력한 금액이 계좌 잔고보다 많으면 오류메시지를 출력하는지 확인한다.	R.3	Pass
4-4	input_amount	사용자의 입력금액이 계좌 잔고보다 작고 ATM 잔고보다 작을때 다음 단계로 넘어가는지 확인한다.	R1.2, R.2, R.3	Pass
5-1	input_RID	입력받은 받는사람 계좌가 유효하지 않을 때 오류메시지를 출력하는지 확인한다	R.3	Pass
5-2	input_RID	입력받은 받는사람 계좌가 유효할때 다음 단계로 넘어가는지 확인한다.	R.3	Pass
6-1	input_MID	입력받은 매니저ID 가 정확하지 않을 때 오류메시지를 출력하는지 확인한다	R.5	Pass

6-2	input_MID	입력받은 매니저ID 가 정확할때 다음단계로 넘어가는지 확인한다	R.5	Pass
7-1	input_MPW	입력받은 매니저PW 가 정확하지 않을때 오류 메시지를 출력하는지 확인한다.	R.5	Pass
7-2	input_MPW	입력받은 매니저 PW 가 정확할때 다음단계로 넘어가는지 확인한다.	R.5	Pass
8-1	input_POM	plus를 입력하였을때, 해당 금액만큼을 ATM 잔고에 더해주는지 확인한다.	R.5	Pass
8-2	input_POM	minus를 입력하였을때, 해당 금액만큼을 ATM 잔고에서 빼주는지를 확인한다.	R.5	Pass
9-1	Random_Jackpot	난수로 생성되는 숫자를 임의의 숫자로 지정하고 해당 숫자번째 출금 거래에서 추가적으로 출금을 해주는지 확인한다.	R.1.1	Pass