System Test Report for Team 7

2nd

Project Team

Team 7

Team Information

컴퓨터공학과 201111341 김성민

컴퓨터공학과 201111345 김종우

컴퓨터공학과 201211356 송원종

1 Specification Review

1.1 Stage 1000 Planning

1.2 Stage 2030 Analysis

1.3 Stage 2040 Design

2 Category-partition Testing Report

2.1 Test Case

2.2 Testing Result

3 Pairwise Testing Report

3.1 Testing Result

4 Brute Force Testing Report

4.1 Testing Result

4.2 Failed Case Report

5 Overall

5.1 System Test Result

1 Specification Review

1.1 Stage 1000 Planning

1006. Define Business Use Case

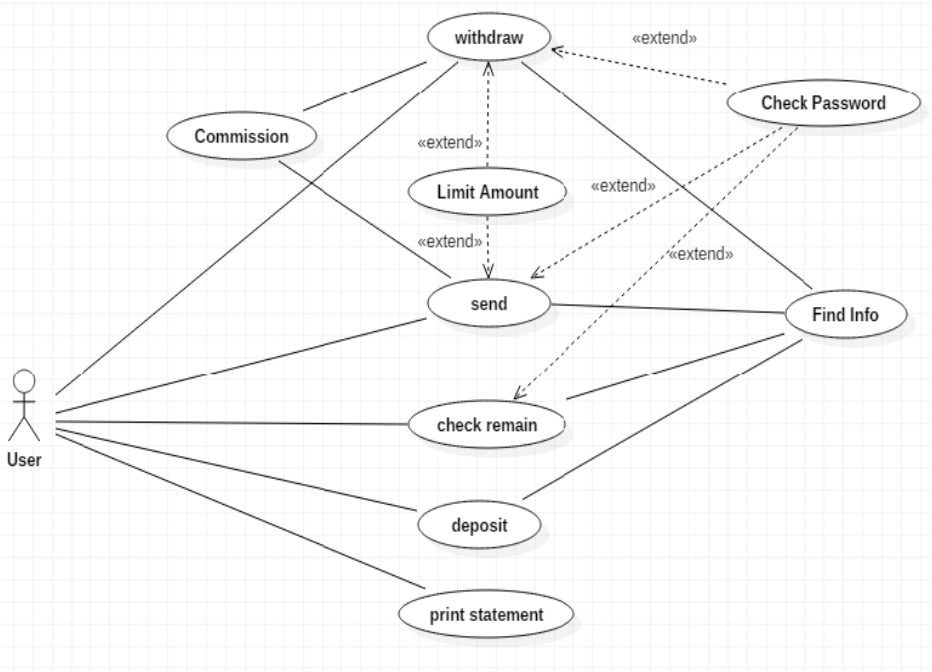
2. Identify and Describe Actors

-Customer: 은행의 일반 업무(입출금, 송금, 잔액확인 등)를 수행할 수 있다.

-Admin: ATM을 관리할 수 있는 권한(ATM 총액확인)을 가지고 있다.

- User에 아직 Admin이 명시되어있다.

6. Draw a Use-Case Diagram



- Event-based Use-Case인 print statement가 User와 직접 연결되어 있음

1.2 Stage 2030 Analysis

2036. Define Operation Contracts

```
1
2   public class Send {
3           Commission commission=new Commission();
4           public static int Amount;
5           public static int Receiver_Account;
6
7           int get_Amount(int amount){
8                   int result;
9                   this.Amount=amount;
10                  result=send_Amount();
11                  return result;
12          }
13          int send_Amount(){
14                  commission.get_Commission(Amount);
15                  this.Amount=commission.get_TotalAmount();
16                  return this.Amount;
17          }
18  }
```

- 실제 Code와 Operation Name이 일치하지 않음
- Operation Name은 실제 사용할 함수의 이름과 같게 작성

2037. Define State Diagram

- 전체적으로 입력된 Account가 실제로 존재하는 것인지 판단하는 부분 추가
- 전체적으로 Password가 Not Match로 판단이 되었을 때 'Wait Password Check'로 돌아가는 것은 새로운 Password를 입력 받는 것이 아닌 같은 Password로 Check하는 것이기 때문에 'Wait Password'로 돌아가는 것으로 수정
- 'Send'의 'Wait Receiver Info'에서 존재하지 않는 계좌에 송금할 경우에 대한 추가 필요
- 'Deposit'은 '2031. Define Essential Use Cases'에 따르면 Password에 대한 부분을 체크할 필요가 없기 때문에 삭제

1.3 Stage 2040 Design

아직 수정된 다이어그램이 올라오지 않아서 검증 불가능

2041. Define Real Use Case

- Actor가 User가 아닌 Customer, Admin으로 세부적인 명세 필요

2044. Define Interaction Diagrams

- 전체적으로 Event, Operation에 대한 표현은 실선인 것과 Return Value는 점선으로 맞추어 작성이 필요
- Actor가 User가 아닌 Customer, Admin으로 세부적인 명세 필요
- 생명선에 대한 더 명확한 명세가 필요, Return이 되기 전에 끝나거나 생명선이 없는 곳에서 시작하는 Event가 존재
- Actor에게 보여지는 부분은 Return이 Actor까지 반환되어야 함
- Amount나 Commission을 Check하는 부분에 대한 Success와 Fail에 대한 경우를 나눠서 표현해줘야 함

1.4 Stage 2050 Implementation

2051. Implement Class & Methods Definitions

Type	Class
Name	Send
Purpose	송금기능을 수행하는 클래스
Overview(Class)	사용자와 송금대상의 계좌정보를 입력 받아 송금을 수행한다.
Cross Reference	R2.1, 1.3
Exceptional Courses of Events	송금/수수료액이 한도를 넘어가거나 잔액보다 많으면 송금기능을 수행하지 않는다.

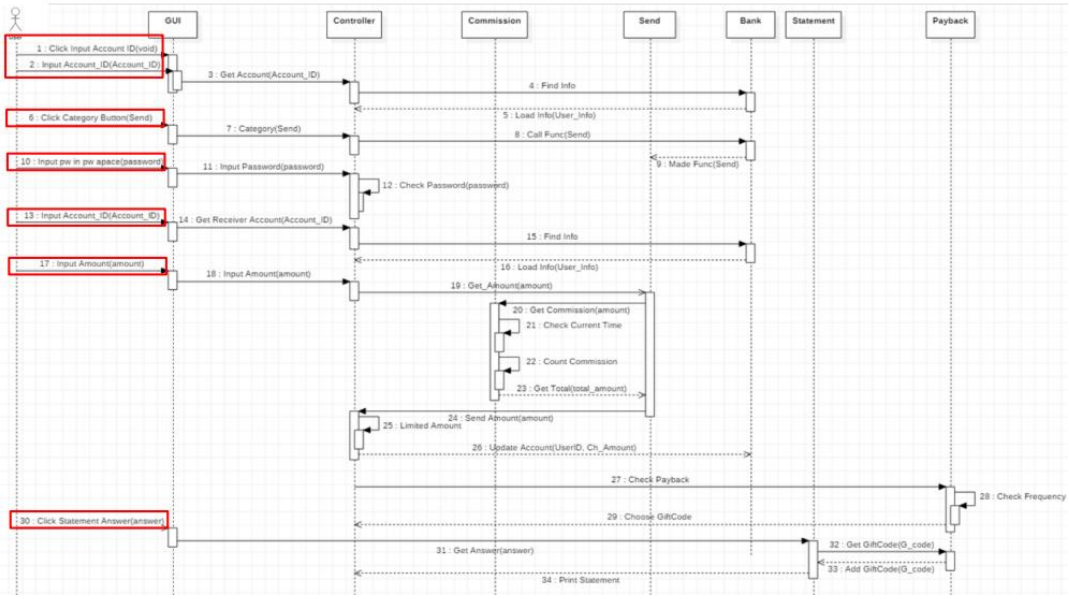
Type	Class
Name	Withdraw
Purpose	출금기능을 수행하는 클래스
Overview(Class)	사용자가 원하는 액수만큼 출금을 수행한다.
Cross Reference	R2.2, 1.3
Exceptional Courses of Events	출금/수수료액이 한도를 넘어가거나 잔액보다 많으면 송금기능을 수행하지 않는다.

- Stage 1000, 2030, 2040까지의 문서와는 다르게 'Exceptional Courses of Events'에 잔액을 검사하는 부분에 대한 내용이 있음

Type	Class
Name	Check Remain
Purpose	잔액조회기능을 수행하는 클래스
Overview(Class)	사용자의 계좌의 잔액을 확인해주는 기능을 수행한다.
Cross Reference	R2.4
Exceptional Courses of Events	N/A

- Name을 실제 코드와 일치하게 'CheckRemain'으로 수정 필요
- Method Name을 실제 코드와 일치하게 수정 필요

2052. Implements Windows



- Name을 실제 코드에 구현된 함수 Name과 일치하게 작성
- 전체적으로 Event, Operation에 대한 표현은 실선인 것과 Return Value는 점선으로 맞추어 작성이 필요
- 생명선에 대한 더 명확한 명세가 필요, Return이 되기 전에 끝나거나 생명선이 없는 곳에서 시작하는 Event가 존재
- Actor에게 보여지는 부분은 Return이 Actor까지 반환되어야 함

1.5 New Use-case

Ver1과 다르게 '사용횟수 조회'에 대한 부분이 추가되어 Stage 1000부터 수정이 요구된다.

2 Category-partition Testing Report

2.1 Test Case

2.1.1 Testable Units

Mode	key	Category
이체	1	이체 금액
		송금할 계좌
출금	2	출금 금액
입금	3	입금 금액
잔액조회	4	-
사용횟수	5	-

2.1.2 Representative Values

Category	Values	Key
로그인 계좌	범위 이내에 있고, DB에 있는 값	1
	범위 이내에 있고, DB에 없는 값	2
	오버플로우/언더플로우 값	3
	음수값	4
	기타 String	5
비밀번호	범위 이내에 있고, DB에 있는 값	1
	범위 이내에 있고, DB에 없는 값	2
	오버플로우/언더플로우 값	3
	음수값	4
	기타 String	5
송금할 계좌	범위 이내에 있고, DB에 있는 값	1
	범위 이내에 있고, DB에 없는 값	2
	오버플로우/언더플로우 값	3
	음수값	4
	기타 String	5
이체 금액	이체범위 안의 값	1
	음수값	2
	오버플로우/언더플로우 값	3
	기타 String	4

출금 금액	출금범위 안의 값	1
	음수값	2
	오버플로우/언더플로우 값	3
	기타 String	4
입금 금액	입금범위 안의 값	1
	음수값	2
	오버플로우/언더플로우 값	3
	기타 String	4
영수증 여부	허용된 대문자, 소문자	1
	허용되지 않은 대문자, 소문자	2
	"true"/"false"	3
	기타 String	4
페이백 여부	허용된 대문자, 소문자	1
	허용되지 않은 대문자, 소문자	2
	"true"/"false"	3
	기타 String	4

2.1.3 Error,Single Constraints 적용

Mode	Values	Option
이체	-	-
출금	-	-
입금	-	-
잔액조회	-	Single
사용횟수	-	Single
Category	**Values**	**Option**
로그인 계좌	범위 이내에 있고, DB에 있는 값	-
	범위 이내에 있고, DB에 없는 값	-
	오버플로우/언더플로우 값	-
	음수값	-
	기타 String	-
송금할 계좌	범위 이내에 있고, DB에 있는 값	-
	범위 이내에 있고, DB에 없는 값	-
	오버플로우/언더플로우 값	-
	음수값	-
	기타 String	-
송금/인출 금액	인출범위 안의 값	-
	음수값	-

	오버플로우/언더플로우 값	-
	기타 String	-
입금 금액	입금범위 안의 값	-
	음수값	-
	오버플로우/언더플로우 값	-
	기타 String	-
Password	범위 이내에 있고, DB에 있는 값	-
	범위 이내에 있고, DB에 없는 값	-
	오버플로우/언더플로우 값	-
	음수값	-
	기타 String	-
PrintInput	허용된 대문자, 소문자	-
	허용되지 않은 대문자, 소문자	-
	"true"/"false"	-
	기타 String	-

2.1.4 Property Constraints 적용

Set	Mode	Property
Enviroments	이체	[property 이체]
	출금	[property 출금]
	입금	[property 입금]
	잔액조회	[property 잔액조회]
	사용횟수	[property 사용횟수]

Category	Values	Property
로그인 계좌	범위 이내에 있고, DB에 있는 값	[if 이체 || 출금 || 입금 || 잔액조회 || 사용횟수] [property 로그인성공]
	범위 이내에 있고, DB에 없는 값	[if 이체 || 출금 || 입금 || 잔액조회 || 사용횟수]
	오버플로우/언더플로우 값	[if 이체 || 출금 || 입금 || 잔액조회 || 사용횟수]
	음수값	[if 이체 || 출금 || 입금 || 잔액조회 || 사용횟수]
	기타 String	[if 이체 || 출금 || 입금 || 잔액조회 ||

		사용횟수]
비밀번호	범위 이내에 있고, DB에 있는 값	[if (이체 || 출금) && 로그인성공] [property 올바른비밀번호]
	범위 이내에 있고, DB에 없는 값	[if (이체 || 출금) && 로그인성공]
	오버플로우/언더플로우 값	[if (이체 || 출금) && 로그인성공]
	음수값	[if (이체 || 출금) && 로그인성공]
	기타 String	[if (이체 || 출금) && 로그인성공]
송금할 계좌	범위 이내에 있고, DB에 있는 값	[if 이체 && 올바른비밀번호] [property 송금계좌인증]
	범위 이내에 있고, DB에 없는 값	[if 이체 && 올바른비밀번호]
	오버플로우/언더플로우 값	[if 이체 && 올바른비밀번호]
	음수값	[if 이체 && 올바른비밀번호]
	기타 String	[if 이체 && 올바른비밀번호]
이체 금액	인출범위 안의 값	[if 이체 && 송금계좌인증] [property 이체범위안]
	음수값	[if 이체 && 송금계좌인증]
	오버플로우/언더플로우 값	[if 이체 && 송금계좌인증]
	기타 String	[if 이체 && 송금계좌인증]
출금 금액	인출범위 안의 값	[if 출금 && 올바른비밀번호] [property 출금범위안]
	음수값	[if 출금 && 올바른비밀번호]
	오버플로우/언더플로우 값	[if 출금 && 올바른비밀번호]
	기타 String	[if 출금 && 올바른비밀번호]
입금 금액	입금범위 안의 값	[if 입금 && 로그인성공] [property 입금범위안]
	음수값	[if 입금 && 로그인성공]
	오버플로우/언더플로우 값	[if 입금 && 로그인성공]
	기타 String	[if 입금 && 로그인성공]
영수증 여부	허용된 대문자, 소문자	[if 이체범위안 || 출금범위안 || 입금범위안]
	허용되지 않은 대문자, 소문자	[if 이체범위안 || 출금범위안 || 입금범위안]
	"true"/"false"	[if 이체범위안 || 출금범위안 || 입금범위안]
	기타 String	[if 이체범위안 || 출금범위안 || 입금범위안]
페이백 여부	허용된 대문자, 소문자	[if 이체범위안 || 출금범위안 || 입금범위안]

	허용되지 않은 대문자, 소문자	[if 이체범위안 || 출금범위안 || 입금범위안]
	"true"/"false"	[if 이체범위안 || 출금범위안 || 입금범위안]
	기타 String	[if 이체범위안 || 출금범위안 || 입금범위안]

- 생성된 테스트케이스 83

2.2 Testing Result

Test Case Num.	Key	Result
1	Single	P
2	Single	P
3	(Key = 1.1.1.1.1.0.0.1.1.)	F
4	(Key = 1.1.1.1.1.0.0.1.2.)	F
5	(Key = 1.1.1.1.1.0.0.1.3.)	F
6	(Key = 1.1.1.1.1.0.0.1.4.)	F
7	(Key = 1.1.1.1.1.0.0.2.1.)	F
8	(Key = 1.1.1.1.1.0.0.2.2.)	F
9	(Key = 1.1.1.1.1.0.0.2.3.)	F
10	(Key = 1.1.1.1.1.0.0.2.4.)	F
11	(Key = 1.1.1.1.1.0.0.3.1.)	F
12	(Key = 1.1.1.1.1.0.0.3.2.)	F
13	(Key = 1.1.1.1.1.0.0.3.3.)	F
14	(Key = 1.1.1.1.1.0.0.3.4.)	F
15	(Key = 1.1.1.1.1.0.0.4.1.)	F
16	(Key = 1.1.1.1.1.0.0.4.2.)	F
17	(Key = 1.1.1.1.1.0.0.4.3.)	F
18	(Key = 1.1.1.1.1.0.0.4.4.)	F
19	(Key = 1.1.1.1.2.0.0.0.0.)	P
20	(Key = 1.1.1.1.3.0.0.0.0.)	P
21	(Key = 1.1.1.1.4.0.0.0.0.)	P
22	(Key = 1.1.1.2.0.0.0.0.0.)	P
23	(Key = 1.1.1.3.0.0.0.0.0.)	P
24	(Key = 1.1.1.4.0.0.0.0.0.)	P
25	(Key = 1.1.1.5.0.0.0.0.0.)	P
26	(Key = 1.1.2.0.0.0.0.0.0.)	P
27	(Key = 1.1.3.0.0.0.0.0.0.)	P

28	(Key = 1.1.4.0.0.0.0.0.)	P
29	(Key = 1.1.5.0.0.0.0.0.)	P
30	(Key = 1.2.0.0.0.0.0.0.)	P
31	(Key = 1.3.0.0.0.0.0.0.)	P
32	(Key = 1.4.0.0.0.0.0.0.)	P
33	(Key = 1.5.0.0.0.0.0.0.)	P
34	(Key = 2.1.1.0.0.1.0.1.)	F
35	(Key = 2.1.1.0.0.1.0.2.)	F
36	(Key = 2.1.1.0.0.1.0.3.)	F
37	(Key = 2.1.1.0.0.1.0.4.)	F
38	(Key = 2.1.1.0.0.1.0.2.1.)	F
39	(Key = 2.1.1.0.0.1.0.2.2.)	F
40	(Key = 2.1.1.0.0.1.0.2.3.)	F
41	(Key = 2.1.1.0.0.1.0.2.4.)	F
42	(Key = 2.1.1.0.0.1.0.3.1.)	F
43	(Key = 2.1.1.0.0.1.0.3.2.)	F
44	(Key = 2.1.1.0.0.1.0.3.3.)	F
45	(Key = 2.1.1.0.0.1.0.3.4.)	F
46	(Key = 2.1.1.0.0.1.0.4.1.)	F
47	(Key = 2.1.1.0.0.1.0.4.2.)	F
48	(Key = 2.1.1.0.0.1.0.4.3.)	F
49	(Key = 2.1.1.0.0.1.0.4.4.)	F
50	(Key = 2.1.1.0.0.2.0.0.0.)	P
51	(Key = 2.1.1.0.0.3.0.0.0.)	P
52	(Key = 2.1.1.0.0.4.0.0.0.)	P
53	(Key = 2.1.2.0.0.0.0.0.0.)	P
54	(Key = 2.1.3.0.0.0.0.0.0.)	P
55	(Key = 2.1.4.0.0.0.0.0.0.)	P
56	(Key = 2.1.5.0.0.0.0.0.0.)	P
57	(Key = 2.2.0.0.0.0.0.0.0.)	P
58	(Key = 2.3.0.0.0.0.0.0.0.)	P
59	(Key = 2.4.0.0.0.0.0.0.0.)	P
60	(Key = 2.5.0.0.0.0.0.0.0.)	P
61	(Key = 3.1.0.0.0.0.1.1.1.)	P
62	(Key = 3.1.0.0.0.0.1.1.2.)	F
63	(Key = 3.1.0.0.0.0.1.1.3.)	F
64	(Key = 3.1.0.0.0.0.1.1.4.)	F
65	(Key = 3.1.0.0.0.0.1.2.1.)	P

66	(Key = 3.1.0.0.0.0.1.2.2.)	P
67	(Key = 3.1.0.0.0.0.1.2.3.)	P
68	(Key = 3.1.0.0.0.0.1.2.4.)	F
69	(Key = 3.1.0.0.0.0.1.3.1.)	P
70	(Key = 3.1.0.0.0.0.1.3.2.)	P
71	(Key = 3.1.0.0.0.0.1.3.3.)	P
72	(Key = 3.1.0.0.0.0.1.3.4.)	P
73	(Key = 3.1.0.0.0.0.1.4.1.)	P
74	(Key = 3.1.0.0.0.0.1.4.2.)	P
75	(Key = 3.1.0.0.0.0.1.4.3.)	P
76	(Key = 3.1.0.0.0.0.1.4.4.)	P
77	(Key = 3.1.0.0.0.0.2.0.0.)	P
78	(Key = 3.1.0.0.0.0.3.0.0.)	P
79	(Key = 3.1.0.0.0.0.4.0.0.)	P
80	(Key = 3.2.0.0.0.0.0.0.0.)	P
81	(Key = 3.3.0.0.0.0.0.0.0.)	P
82	(Key = 3.4.0.0.0.0.0.0.0.)	P
83	(Key = 3.5.0.0.0.0.0.0.0.)	P

47/83 = 56.6% Pass

3 PairWise Testing Report

3.1 Testing Set

Category	Values	Key
로그인 계좌	범위 이내에 있고, DB에 있는 값	1
	범위 이내에 있고, DB에 없는 값	2
	오버플로우/언더플로우 값	3
	음수값	4
	기타 String	5
송금할 계좌	범위 이내에 있고, DB에 있는 값	1
	범위 이내에 있고, DB에 없는 값	2
	오버플로우/언더플로우 값	3
	음수값	4
	기타 String	5
송금/인출 금액	인출범위 안의 값	1
	음수값	2
	오버플로우/언더플로우 값	3
	기타 String	4

입금 금액	입금범위 안의 값	1
	음수값	2
	오버플로우/언더플로우 값	3
	기타 String	4
비밀번호	범위 이내에 있고, DB에 있는 값	1
	범위 이내에 있고, DB에 없는 값	2
	오버플로우/언더플로우 값	3
	음수값	4
	기타 String	5
영수증옵션	허용된 대문자, 소문자	1
	허용되지 않은 대문자, 소문자	2
	"true"/"false"	3
	기타 String	4

3.2 Testing Result

Test Case Num.	Key	Result
1	1.3.3.3.2.2.	F
2	1.5.4.2.1.4.	F
3	1.4.2.1.5.4.	F
4	1.2.1.2.2.3	F
5	1.1.1.2.4.1.	F
6	1.3.3.3.3.1.	P
7	1.5.2.4.5.3..	F
8	2.3.2.2.5.3..	F
9	2.2.4.1.3.2.	F
10	2.4.1.4.2.4.	F
11	2.1.4.3.3.3.	F
12	2.2.3.1.1.1.	P
13	2.5.3.3.3.4.	F
14	2.4.2.3.4.2.	F
15	3.4.4.4.4.1.	P
16	3.1.3.3.5.1.	P
17	3.3.1.3.1.4.	F
18	3.5.3.2.4.3.	F
19	3.3.4.1.2.2.	F
20	3.2.2.4.3.2.	F
21	4.2.2.3.4.3.	F

22	4.1.2.4.1.2.	F
23	4.5.2.1.2.1.	P
24	4.4.1.2.3.4..	F
25	4.3.3.4.5.3..	F
26	4.2.4.1.2.2..	F
27	5.1.1.1.4.4..	F
28	5.4.3.2.1.3...	F
29	5.5.1.4.5.2.	F
30	5.2.2.4.3.1.	P
31	5.3.4.2.4.2.	F
32	5.1.2.1.2.3.	F
33	5.2.4.3.5.4.	F

6/33 = 18.2% Pass

4 Brute Force Testing Report

4.1 Testing Result

Test Case Num	Test Case	Result
1	페이백 종류 범위값 내 번호 입력	F
2	페이백 종류 입력하지 않음	F
3	페이백 종류 범위값 외 번호 입력	F
4	Printstatement의 입력으로 임의의 스트링	P
5	로그인 계좌와 동일한 계좌로 송금 요청	P
6	입금/출금/송금 금액에 음수값 입력	P
7	100000이상의 입금을 수행하고 DB파일 잔액 변화유무 확인	P
8	DB파일 한도를 음수값으로 설정한 후 프로그램 실행	P
9	영수증 출력(y/n) 입력하지 않음	P
10	이체시 DB에 없는 계좌번호값 입력	F
11	페이백 종류, 영수증 출력값 둘 다 입력하지 않음	P
12	입금-금액입력시 특정 값("333")입력	P
13	입력값이 잘못된 경우에 다음으로 진행하는데(이미 Fail인 상태), 그 때 영수증, 페이백 입력	F
14	수수료 발생 유무 확인	p

9/14 = 64% Pass

4.2 Failed Case Report

Num	Report
1	.한도가 여전히 0인 계좌 존재
2	존재하지 않는 계좌로 이체 가능
3	계좌 이체 시 수수료를 포함한 금액이 이체됨
4	Payback 옵션 선택 시 예외처리 되지 않음
5	의도 하지않은 에러메시지 출력

5 Overall

5.1 System Test Result

5.1.1 Category Partition Test

5.1.2 47/83 = 56.6% pass

5.1.3 Pairwise Test

6/33 = 18.2% pass

5.1.4 Brute Force Test

9/14 = 64% Pass

5.1.5 Summary

5.1.5.1 문서의 단계마다 일치하지 않는 부분이 다수 존재한다.

5.1.5.2 대응서 에서 수정했다고 작성한 부분 중 올바르게 수정되지 않은 부분이 다수 존재한다.

5.1.5.3 프로그램이 수정되었음에도 불구하고, 문서에 명시하지 않았다.

5.1.5.4 프로그램이 개선되긴 하였으나, 아직 고치지 못한 부분이 다수 존재한다.