

SOFTWARE VERIFICATION

1st Test - System Testing
- T3 -



송지연
윤상혁
장서연



PART 1

Specification Review

Specification Review

Stage 1000

Date

2017-04-17

2.6 Other Managerial Issues

-2017년 6월까지 개발이 완료되어야 한다

3.2 Performance Requirements

-2017년 6월까지 개발이 완료되어야 한다

다수의 오타자

5. Activity 1005. Implement Prototype

6. Activity 1006. Define System Architecture

7. Activity 1007. Define Business Use Case

8. Activity 1008 Define Business Concept Model

9. Activity 1009 Define System Test Case

10. Activity 1010 Refine Plan

6. Activity 1006. Define Business Use Case

Specification에 빈번한 오타가 발생 및 인덱스 오류가 존재한다.

Specification Review

Stage 2000-2010

- Activity 1003의 Functional Requirements ↵
- version1에서 잘못 표시된 use case들을 지우고 Actor의 기준에서 생각할 수 있는 case 추가 ↵

문서의 일관성 문제

Stage 1000에 반영되지 않은 내용이 2030에 수정사항으로 등장한다.

Specification Review

Stage 2030-2031

Use Case	1. Deposit
Actor	User
Purpose	사용자가 원하는 계좌에 입금을 한다.
Overview	User가 입금버튼을 누른다.
Type	Primary and Essential
Cross Reference	Functional Requirements : R 1,10,11,12
Pre-Requisites	N/A
Typical courses of events	(A) : Actor (S) : System 1. (A) User가 프로그램(UI)에서 Deposit버튼을 누른다. 2. (S) User에게 Deposit UI를 띄워준다. 3. (A) 입금할 계좌(or 카드번호)와 비밀번호를 프로그램에 입력한다. 4. (S) 시스템에서 해당 계좌의 비밀번호를 확인한다. 5. (S) 금액 입력창으로 UI 변경. 6. (A) User가 입금할 금액을 입력한다. 7. (S) User가 입력한 금액을 해당 계좌나 카드에 추가한다. 8. (S) 변경 내용을 저장한다. 9. (S) 변경 내용을 User에게 보여준다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	1. (S) 잘못된 계좌정보를 입력하거나 입금금액이 일정 금액을 초과하면 알림을 띄운다.

Use Case	2. Withdraw
Actor	User
Purpose	사용자가 원하는 계좌에서 출금을 한다.
Overview	User가 출금버튼을 누른다.
Type	Primary and Essential
Cross Reference	Functional Requirements : R 2,10,11,12
Pre-Requisites	N/A
Typical courses of events	(A) : Actor (S) : System 1. (A) User가 프로그램(UI)에서 Withdraw버튼을 누른다. 2. (S) User에게 Withdraw UI를 띄워준다. 3. (A) 출금할 계좌(or 카드번호)와 비밀번호를 프로그램에 입력한다. 4. (S) 시스템에서 해당 계좌의 비밀번호를 확인한다. 5. (S) 금액 입력창으로 UI 변경. 6. (A) User가 출금할 금액을 입력한다. 7. (S) User가 입력한 금액을 해당 계좌나 카드에서 빼준다. 8. (S) 변경 내용을 저장한다. 9. (S) 변경 내용을 User에게 보여준다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	1. (S) 잘못된 계좌정보 또는 비밀번호를 입력하거나 출금금액이 한계 금액이나 계좌 잔액을 초과하면 알림을 띄운다.

추상적 표현

Exceptional Courses of Events에서 한계 금액 -> 구체적 표현의 사용 필요

Specification Review

Stage 2030-2031

Use Case	3. Transfer
Actor	User
Purpose	사용자가 원하는 계좌로 자신의 계좌에서 돈을 이체한다.
Overview	User가 계좌이체 버튼을 누른다.
Type	Primary and Essential
Cross Reference	Functional Requirements : R 3,10,11,12
Pre-Requisites	N/A
Typical courses of events	<p>(A) : Actor (S) : System</p> <ol style="list-style-type: none"> (A) User가 프로그램(UI)에서 Transfer버튼을 누른다. (S) User에게 Transfer UI를 띄워준다. (A) 출금할 계좌(or 카드번호)와 비밀번호를 프로그램에 입력한다. (S) 시스템에서 해당 계좌의 비밀번호를 확인한다. (S) 이체 계좌정보 및 금액 입력창으로 UI 변경. (A) User가 이체할 계좌정보와 이체할 금액을 입력한다. (S) User가 입력한 금액을 이체 계좌나 카드에 더해주고 해당 계좌에서는 그 금액만큼 빼준다. (S) 변경 내용을 저장한다. (S) 변경 내용을 User에게 보여준다.
Alternative Courses of Events	
Exceptional Courses of Events	1. (S) 잘못된 계좌정보 또는 비밀번호를 입력하거나 이체금액이 한계 금액이나 계좌 잔액을 초과하면 알람을 띄운다.

Use Case	4. Check Balance
Actor	User
Purpose	사용자가 원하는 계좌의 잔액을 보여준다.
Overview	User가 잔액확인 버튼을 누른다.
Type	Primary and Essential
Cross Reference	Functional Requirements : R 5,10,11,12
Pre-Requisites	N/A
Typical courses of events	<p>(A) : Actor (S) : System</p> <ol style="list-style-type: none"> (A) User가 프로그램(UI)에서 Check Balance버튼을 누른다. (S) User에게 Check Balance UI를 띄워준다. (A) 확인할 계좌(or 카드번호)와 비밀번호를 프로그램에 입력한다. (S) 시스템에서 해당 계좌의 비밀번호를 확인한다. (S) 해당계좌나 카드의 잔액을 화면에 띄운다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	1. (S) 잘못된 계좌정보 또는 비밀번호를 입력하면 알람을 띄운다.

추상적 표현

Exceptional Courses of Events에서 한계 금액 -> 구체적 표현의 사용 필요
2036의 Withdraw에서도 같은 상황 발생한다.

Specification Review

Stage 2030-2031

R.1	Deposit	Evident
R.2	Withdraw	Evident
R.3	Check Balance	Evident
R.4	Transfer	Evident
R.5	Language	Evident
R.6	Robbery Report	Evident
R.7	Size Up	Evident
R.8	Add Cash	Evident
R.9	Check Cash	Evident
R.10	Load	Hidden
R.11	Store	Hidden
R.12	Check	Hidden

요구사항의 불일치

Use Case	3. Transfer
Actor	User
Purpose	사용자가 원하는 계좌로 자신의 계좌에서 돈을 이체한다.
Overview	User가 계좌이체 버튼을 누른다.
Type	Primary and Essential
Cross Reference	Functional Requirements : R 3,10,11,12

Use Case	4. Check Balance
Actor	User
Purpose	사용자가 원하는 계좌의 잔액을 보여준다.
Overview	User가 잔액확인 버튼을 누른다.
Type	Primary and Essential
Cross Reference	Functional Requirements : R 5,10,11,12

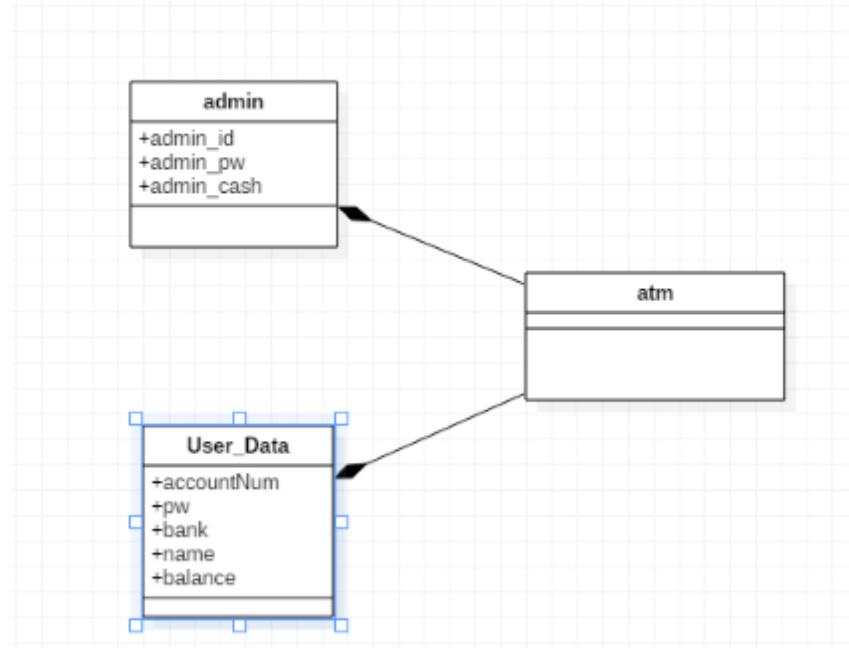
Use Case	5. Robbery Report
Actor	User
Purpose	사용자가 도난 신고를 한다.
Overview	User가 도난 신고 버튼을 누른다.
Type	Primary and Essential
Cross Reference	Functional Requirements : R 7,10,11,12

Use Case	6. Add Cash
Actor	Admin
Purpose	관리자가 ATM에 현금을 추가한다.
Overview	관리자가 비밀번호를 입력해서 현금을 추가한다.
Type	Primary and Essential
Cross Reference	Functional Requirements : R 8,10,11,12

Cross Reference-Stage1000 항목과 2031의 내용이 일치하지 않음
2036에서도 같은 사항 발생

Specification Review

Stage 2030-2034

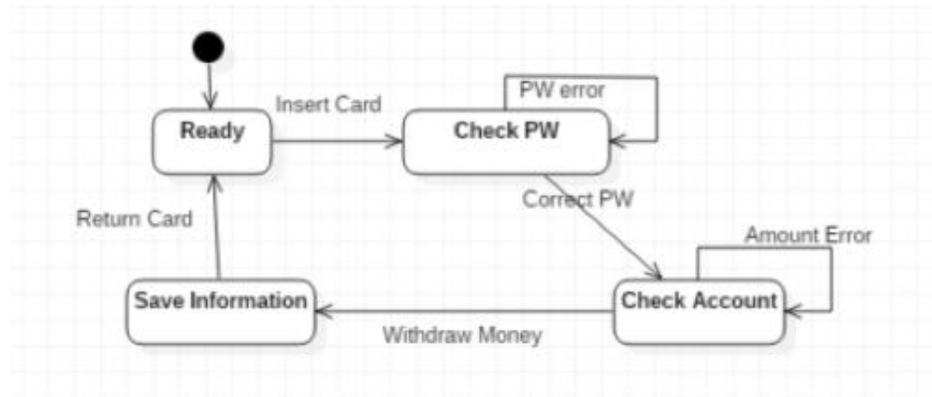


다이어그램

이전 문서와 2030에 나타난 card의 존재가 다이어그램에서 삭제 및 다이어그램의 미비

Specification Review

Stage 2030-2037



다이어그램

실제 구현과 맞지 않은 다이어그램 및 withdraw 기능밖에 다이어그램이 존재하지 않음

Specification Review

Stage 2040-2041

Use Case	1. Deposit
Actor	User
Purpose	사용자가 원하는 계좌에 입금을 한다.
Overview	User가 입금버튼을 누른다.
Type	Primary and Essential
Cross Reference	Functional Requirements : R 1,10,11,12
Pre- Requisites	N/A
Typical courses of events	(A) : Actor (S) : System 1. (A) User가 프로그램(UI)에서 Deposit버튼을 누른다. 2. (S) User에게 Deposit UI를 띄워준다. 3. (A) 입금할 계좌(or 카드번호)와 비밀번호를 프로그램에 입력한다. 4. (S) 시스템에서 해당 계좌의 비밀번호를 확인한다. 5. (S) 금액 입력창으로 UI 변경. 6. (A) User가 입금할 금액을 입력한다. 7. (S) User가 입력한 금액을 해당 계좌나 카드에 추가한다. 8. (S) 변경 내용을 저장한다. 9. (S) 변경 내용을 User에게 보여준다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	1. (S) 잘못된 계좌정보를 입력하거나 입금금액이 일정 금액을 초과하면 알림을 띄운다.

Use Case	1. Deposit
Actor	User
Purpose	사용자가 원하는 계좌에 입금을 한다.
Overview	User가 입금버튼을 누른다.
Type	Primary and Real
Cross Reference	Functional Requirements : R 1,10,11,12
Pre- Requisites	N/A
Typical courses of events	(A) : Actor (S) : System 1. (A) User가 프로그램(UI)에서 Deposit버튼을 누른다. 2. (S) User에게 Deposit UI를 띄워준다. 3. (A) 입금할 계좌(or 카드번호)와 비밀번호를 프로그램에 입력한다. 4. (S) 시스템에서 해당 계좌의 비밀번호를 확인한다. 5. (S) 금액 입력창으로 UI 변경. 6. (A) User가 입금할 금액을 입력한다. 7. (S) User가 입력한 금액을 해당 계좌나 카드에 추가한다. 8. (S) 변경 내용을 저장한다. 9. (S) 변경 내용을 User에게 보여준다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	1. (S) 잘못된 계좌정보를 입력하거나 입금금액이 일정 금액을 초과하면 알림을 띄운다.

표현의 변화

2031-1~7과 달리 2041-1~7에서 type의 변화가 발생

Specification Review

Stage 2040-2041

R.1	Deposit
R.2	Withdraw
R.3	Check Balance
R.4	Transfer
R.5	Language
R.6	Robbery Report
R.7	Size Up
R.8	Add Cash
R.9	Check Cash
R.10	Load
R.11	Store
R.12	Check

Use Case	4. Check Balance
Actor	User
Purpose	사용자가 원하는 계좌의 잔액을 보여준다.
Overview	User가 잔액확인 버튼을 누른다.
Type	Primary and Real
Cross Reference	Functional Requirements : R.5,10,11,12
Pre- Requisites	N/A
Typical courses of events	(A) : Actor; (S) : System 1. (A) User가 프로그램(UI)에서 Check Balance버튼을 누른다. 2. (S) User에게 Check Balance UI를 띄워준다. 3. (A) 확인할 계좌(or 카드번호)와 비밀번호를 프로그램에 입력한다. 4. (S) 시스템에서 해당 계좌의 비밀번호를 확인한다. 5. (S) 해당계좌나 카드의 잔액을 화면에 띄운다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	1. (S) 잘못된 계좌정보 또는 비밀번호를 입력하면 알림을 띄운다.

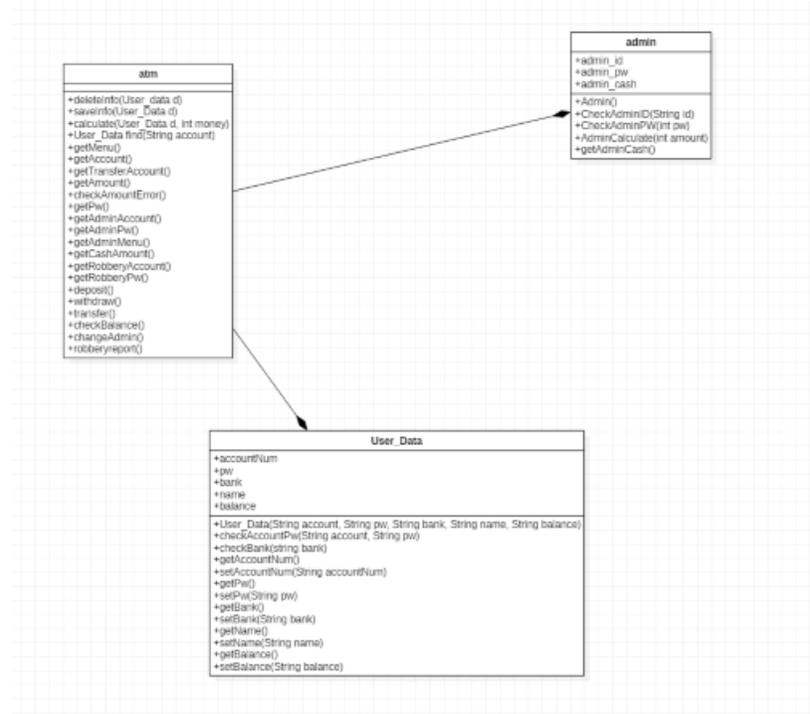
Use Case	5. Robbery Report
Actor	User
Purpose	사용자가 도난 신고를 한다.
Overview	User가 도난 신고 버튼을 누른다.
Type	Primary and Real
Cross Reference	Functional Requirements : R.7,10,11,12
Pre- Requisites	N/A
Typical courses of events	(A) : Actor; (S) : System 1. (A) User가 프로그램(UI)에서 도난 신고버튼을 누른다. 2. (S) User에게 도난신고 UI를 띄워준다. 3. (A) 도난 신고할 계좌(or 카드번호)와 비밀번호를 프로그램에 입력한다. 4. (S) 시스템에서 해당 계좌의 비밀번호를 확인한다. 5. (S) 계좌정보(카드)를 도난 상태로 변경.
Alternative Courses of Events	N/A
Exceptional Courses of Events	1. (S) 이미 있는 도난 카드의 번호를 입력했을 시에는 화면에 알림을 띄운다.

요구사항의 불일치

Cross Reference-Stage1000 항목과 2041-4~5의 내용이 일치하지 않음

Specification Review

Stage 2040-2041



UML의 변경

2034와 UML이 변경되었다

Specification Review

Stage 2050&2060

Add Cash	<ul style="list-style-type: none"> - 관리자 모드 인터페이스에 있는 버튼 - 관리자가 버튼을 클릭하게 되면 입금할 금액을 물어보는 화면 출력 - 입력 받은 금액을 바탕으로 데이터베이스의 잔액 정보를 수정.
Check Cash	<ul style="list-style-type: none"> - 관리자 모드 인터페이스에 있는 버튼 - 데이터베이스로부터 ATM의 잔액 정보를 불러온다. - ATM의 잔액 정보를 화면에 출력한다.

Type	Class
Name	Admin
Purpose	관리자로서 ATM기기내의 실 잔액량을 추가하고 확인 하면서 관리하는 Class
Overview	N/A
Cross Reference	AddCash, CheckCash
Exceptional Course of Event	N/A

ATM 금액 관리

1000을 비롯한 이전 문서에서는 ATM 금액 관리를 DB에서 한다고 명시되어 있지만 실제로는 클래스 변수로 관리가 되고 있다.

Specification Review

General Problems

GUI 미구현

GUI가 구현되지 않고 CUI로 구현되어 있다.

문서 불완전

2043에서 Refine System Architecture,
2046, 2067에서 traceability 부분이 작성되지 않았다.



PART 2

Brute Force Test

Brute Force Test

1 메뉴 선택 화면에서 잘못된 입력(number)	Failed
2 메뉴 선택 화면에서 잘못된 입력(malformed)	Failed
3 Database의 계좌를 12자리로 변경 후, 올바른 계좌 번호 입력	Failed
4 Database의 계좌를 malformed(6자리)로 변경 후, 올바른 계좌 번호 입력	Failed
5 Database의 계좌 보유액을 조정(1,000,000,000,000,000) 후, 900,000,000,000,000 출금	Failed
6 Database의 계좌 보유액을 조정(1,000,000,000,000,000) 후, 900,000,000,000,000 송금	Failed
7 Database의 비밀번호를 8자리로 수정 후, 올바른 비밀번호 입력	Failed
8 Database의 비밀번호를 malformed(4자리)로 수정 후, 올바른 비밀번호 입력	Failed
9 Database 고객 이름 변경(4자리 이상)	Failed
10 자기 자신에게 올바른 금액 이체	Failed
11 반복 입금으로 ATM 보유액이 long int 범위 이상	Failed
12 프로그램 종료, 재시작 후 Database 보존 여부 확인	Failed

Success 0/12
Failed 12/12

Brute Force Test

CONCLUSION

- 잘못된 입력형태가 들어올 경우 항상 에러가 발생한다.
- 잘못된 상황(자기자신에게 이체 등)에 대한 예외처리가 안되어 있다.
- Specification 상의 입력 정보가 모두 상세정보가 없다.
- 따라서 형식에 맞춰 test할 수 없었고, 프로그램이 허용하지 않는 형식에 대해서는 모두 오류가 발생했다.



PART 3

Category Partition Test

Category Partition Test

Deposit	101 잘못된 계좌번호(number) 입력	Success
	102 잘못된 계좌번호(malformed) 입력	Success
	103 올바른 계좌번호 입력했으나, 잘못된 금액(< 0) 입력	Failed
	104 올바른 계좌번호 입력했으나, 잘못된 금액(malformed) 입력	Failed
	105 올바른 계좌번호 입력 후, 올바른 금액 입력	Success
Withdraw	201 잘못된 계좌번호(number) 입력	Success
	202 잘못된 계좌번호(malformed) 입력	Success
	203 올바른 계좌번호 입력 후, 잘못된 비밀번호(number) 입력	Success
	204 올바른 계좌번호 입력 후, 잘못된 비밀번호(malformed) 입력	Success
	205 올바른 계좌번호와 비밀번호 입력 후, 잘못된 금액(< 0) 입력	Failed
	206 올바른 계좌번호와 비밀번호 입력 후, 잘못된 금액(> AccountAmount) 입력	Success
	207 올바른 계좌번호와 비밀번호 입력 후, 잘못된 금액(> ATMAmount) 입력	Failed
	208 올바른 계좌번호와 비밀번호 입력 후, 올바른 금액 입력	Success
Transfer	301 잘못된 출금 계좌번호(number) 입력	Success
	302 잘못된 출금 계좌번호(malformed) 입력	Success
	303 올바른 출금 계좌 입력 후, 잘못된 비밀번호(number) 입력	Success
	304 올바른 출금 계좌 입력 후, 잘못된 비밀번호(malformed) 입력	Success
	305 올바른 출금 계좌와 비밀번호 입력 후, 잘못된 입금 계좌(number) 입력	Success
	306 올바른 출금 계좌와 비밀번호 입력 후, 잘못된 입금 계좌(malformed) 입력	Success
	307 올바른 출금 계좌, 비밀번호, 입금 계좌 입력 후, 잘못된 금액(< 0) 입력	Failed
	308 올바른 출금 계좌, 비밀번호, 입금 계좌 입력 후, 잘못된 금액(< 출금 계좌 보유액) 입력	Success
	309 올바른 출금 계좌, 비밀번호, 입금 계좌 입력 후, 잘못된 금액(malformed) 입력	Success
	310 올바른 출금 계좌, 비밀번호, 입금 계좌 입력 후, 올바른 금액 입력	Success

Category Partition Test

Check Balance	401 잘못된 계좌번호 입력	Success
	403 올바른 계좌번호 입력 후, 잘못된 비밀번호 입력	Success
	405 올바른 계좌번호 입력 후, 올바른 비밀번호 입력	Success
Robbery Report	501 잘못된 계좌번호 입력	Success
	503 올바른 계좌번호 입력 후, 잘못된 비밀번호 입력	Success
	505 올바른 계좌번호 입력 후, 올바른 비밀번호 입력	Success
AddCash	601 잘못된 관리자 아이디 입력	Success
	602 올바른 관리자 아이디를 입력 했으나, 잘못된 관리자 비밀번호(number) 입력	Success
	603 올바른 관리자 아이디를 입력 했으나, 잘못된 관리자 비밀번호(malformed) 입력	Failed
	604 올바른 관리자 아이디 비밀번호를 입력 했으나, 잘못된 금액(< 0) 입력	Failed
	605 올바른 관리자 아이디 비밀번호를 입력 했으나, 잘못된 금액(malformed) 입력	Success
	606 올바른 관리자 아이디, 비밀번호, 금액 입력	Success
Check Cash	701 잘못된 관리자 아이디 입력	Success
	702 올바른 관리자 아이디를 입력 했으나, 잘못된 관리자 비밀번호(number) 입력	Success
	703 올바른 관리자 아이디를 입력 했으나, 잘못된 관리자 비밀번호(malformed) 입력	Failed
	704 올바른 관리자 아이디, 비밀번호 입력	Success

Success 31/39
Failed 8/39

Category Partition Test

CONCLUSION

- 이상적인 금액 외의 결과가 들어올 경우 항상 에러가 발생한다.
- 금액이 음수로 입력 되었을 경우 원하는 작동과 반대의 현상이 일어난다.
- 즉, 입금에선 출금이 출금에선 입금이 된다.
- 전체적으로 예외처리가 부족하다.



PART 4

Pairwise Combination Test

Pairwise Combination Test

105-704	입금 후 ATM 잔액 확인	Success
105-405	입금 후 계좌 잔액 확인	Success
208-704	출금 후 ATM 잔액 확인	Success
208-405	출금 후 계좌 잔액 확인	Success
310-405	송금 후 출금 계좌 잔액 확인	Success
310-405	송금 후 입금 계좌 잔액 확인	Success
505-505	도난 신고 후 도난신고	Failed
505-208	도난 신고 후 출금	Failed
505-105	도난 신고 후 입금	Failed
505-310	도난 신고 후 송금	Failed
505-405	도난 신고 후 계좌 잔액 확인	Failed

Success 6/11
Failed 5/11

Pairwise Combination Test

CONCLUSION

- 도난신고 후 계좌를 모두 지워 버림으로써 명세에 적혀있는 기능들이 모두 구현이 안되어 있다.



PART 5

Conclusion

예외 처리

구현 완성

Specification 재작성