

System Test Report for Team 1

2nd

Project Team

Team 1

Latest update on:

2018-06-02

Team Information

컴퓨터공학과 201311263 김민환

컴퓨터공학과 201311308 전세진

컴퓨터공학과 201411278 서희진

컴퓨터공학과 201411317 조민규

Table of Contents

1	Specification Review	3
1.1	Stage 1000 Planning	3
1.2	Stage 2030 Analysis.....	4
1.3	Stage 2040 Design	10
1.4	Stage 2050 Implementation	17
1.5	Stage 2060 Unit Test & System Test.....	21
2	Category-partition Testing Report.....	22
2.1	Categorize	22
2.2	Description	23
2.3	Property & Constraints.....	24
2.4	Textfile for TSL-Generator.....	25
2.5	Test Result.....	26
3	Pairwise Testing Report	29
3.1	Categorize	29
3.2	Description	30
3.3	Textfile for PICT	31
3.4	Test Result.....	32
4	Brute Force Testing Report	33
4.1	Testing Result.....	33
4.2	Failed Case Report	34
5	Code Scroll.....	35
5.1	Code Analysis	35
5.2	Summary.....	39
6	Overall.....	40
6.1	System Test Result	40

1 Specification Review

1.1 Stage 1000 Planning

1003. Define Requirements

1. Functional Requirements

- Deposit은 계좌에 증감되지 않고, 증가하므로 증가로 수정.

1004. Record Terms in Glossary

- Code상에서 ATM객체가 하나만 생성되며 TrafficCard가 지급되는 구조가 아님.

1008. Define Business Concept Models

- 2030 단계에만 존재하는 심볼이 정의되어 있다. (History, Sensor)



- 2030단계 이후로는 등장하지 않는 Attribute가 정의되어 있다.

<u>chargeFee</u>	Attribute	수수료
------------------	-----------	-----

1.2 Stage 2030 Analysis

2031. Describe Use Case => 2030과 2040 문서가 모두 불일치

1.1 Check

2030	
Alternative Courses of Events	E0. 한국 계좌의 경우 한국어 모드, 해외 계좌일 경우 외국어 모드 E7. 명세표 출력시, 계좌 잔액 출력 / 미출력시 다음 단계.
Exceptional Courses of Events	E1. 인증에 실패한 경우, 종료한다 E8. 명세표 출력 용지가 부족할 경우 알림을 발생시킨다.
2040	
Alternative Courses of Events	E0. 한국 계좌의 경우 한국어 모드, 해외 계좌일 경우 외국어 모드 E1. 다른 거래 선택 E7. 명세표 출력시, 계좌 잔액 출력 / 미출력시 다음 단계.

1.2 Deposit

2030	
Use Case	E0. ATM 보유 현금이 초과되었을 경우 종료한다. E2. 사용자가 폐쇄요청을 하지 않을 경우, 일정 시간 후에 안내 후 자동 폐쇄한다. E9. 현금 보유량이 많을 경우 알림을 발생시킨다. E10. 명세표 출력 용지가 부족할 경우 알림을 발생시킨다.
2040	
Use Case	E0. ATM 보유 현금이 초과되었을 경우 종료한다. E2. 사용자가 폐쇄요청을 하지 않을 경우, 일정 시간 후에 안내 후 자동 폐쇄한다. E4. 유효한 현금이 아닐 경우, 반환 후 종료한다. E9. 현금 보유량이 많을 경우 알림을 발생시킨다. E10. 명세표 출력 용지가 부족할 경우 알림을 발생시킨다.

1.3 Withdraw

2030	
Use Case	E0. 한국 계좌의 경우 한국어 모드, 해외 계좌일 경우 외국어 모드 E11. 명세표 출력 시, 거래 내역 출력 / 미출력시 다음 단계.
2040	
Use Case	E0. 한국 계좌의 경우 한국어 모드, 해외 계좌일 경우 외국어 모드 E0. 다른 거래 선택 E11. 명세표 출력 시, 거래 내역 출력 / 미출력시 다음 단계.

1.4 Issue TrafficCard

2030	
Actors	E1. 본인확인 실패 시, 종료한다. E5. 계좌 잔액 부족 시, 종료한다. E11. 현금 인출이 일정 시간동안 이루어지지 않는 경우 투입구를 폐쇄하고 다시 입금한다. E14. 현금 보유량이 부족할 경우 알림을 발생시킨다. E15. 명세표 출력 용지가 부족할 경우 알림을 발생시킨다.

2040	
Actors	E1. 본인확인 실패 시, 종료한다. E5. 계좌 잔액 부족 시, 종료한다. E14. 현금 보유량이 부족할 경우 알림을 발생시킨다. E15. 명세표 출력 용지가 부족할 경우 알림을 발생시킨다.

1.6 Update

2030	
Alternative Courses of Events	E0. 한국 계좌의 경우 한국어 모드, 해외 계좌일 경우 외국어 모드 E12. 명세표 출력 시, 거래 내역 출력 / 미출력시 다음 단계.

2040	
Alternative Courses of Events	E0. 한국 계좌의 경우 한국어 모드, 해외 계좌일 경우 외국어 모드 E0. 다른 거래 선택 E12. 명세표 출력 시, 거래 내역 출력 / 미출력시 다음 단계.

다른 경우에 대해서도 문서 간의 불일치 문제를 수정하지 않았습니다.

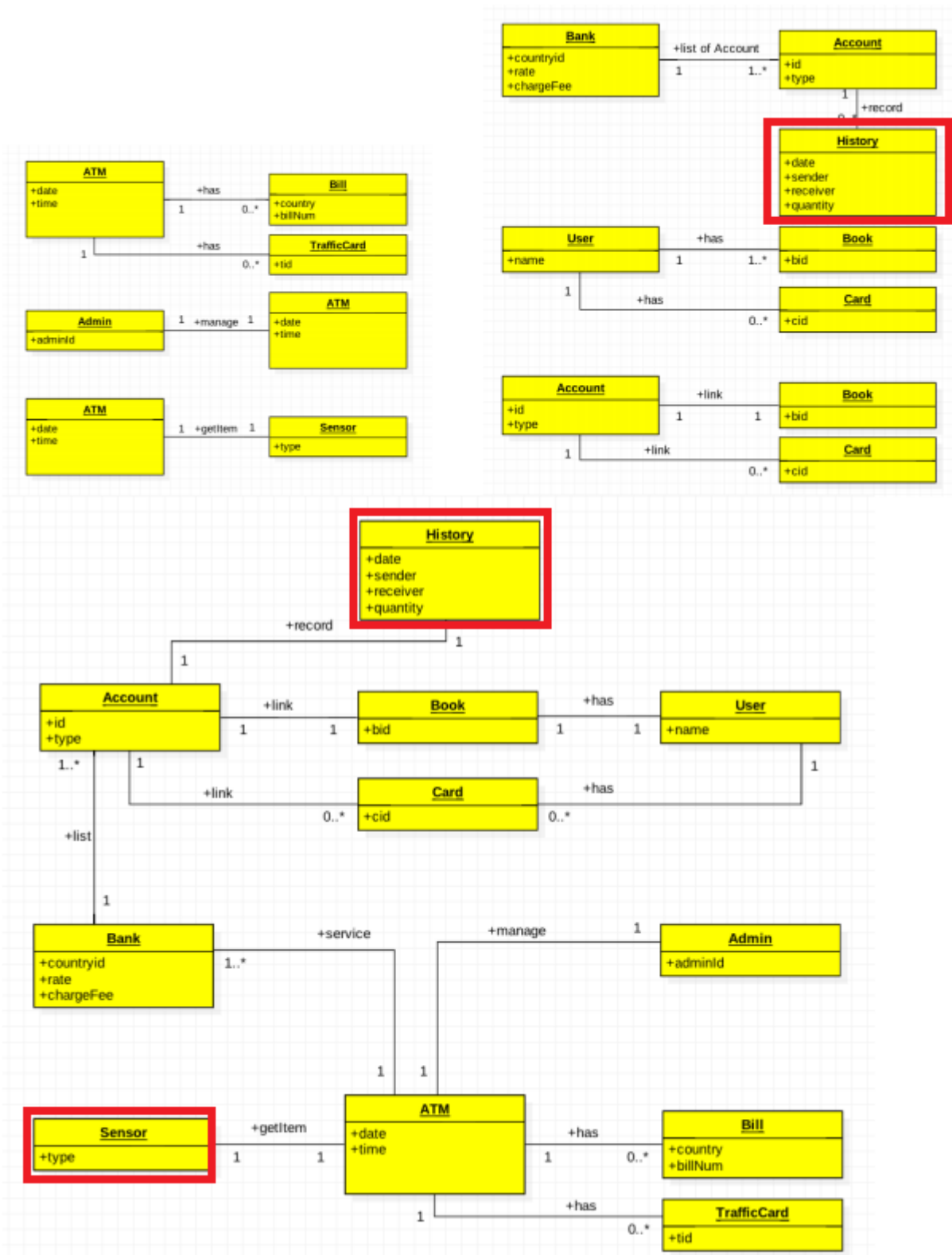
1.8 Status Alarm

Purpose ATM 보유 현금량이 충분하지 않을 경우 관리자에게 알림을 보내준다.

- 명세표의 출력용지가 충분하지 않을 경우에도 관리자에게 알림을 보내지 않는다.

2033. Define Domain Model

- 사용되지 않는 Sensor, Model, History를 사용한다.



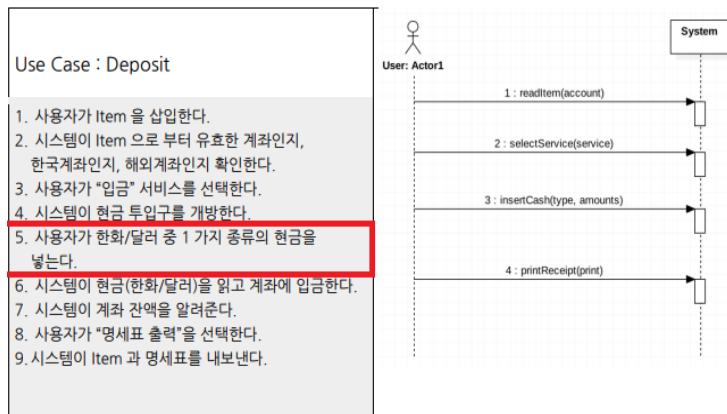
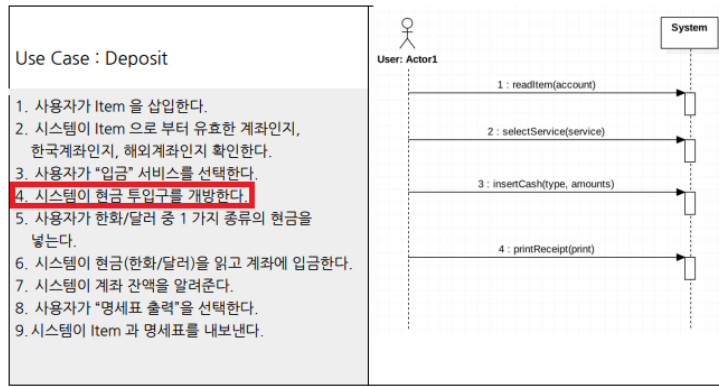
2034. Refine Clossary

- 사용하지 않는 변수가 그대로 존재한다.

Check	Use cases	계좌 조회
Deposit	Use cases	예금 입금
Withdraw	Use cases	예금 인출
Transfer	Use cases	예금 송금
Issue TrafficCard	Use cases	교통카드 발급
Update	Use cases	거래 내역 연동
Verify Sufficient Fund	Use cases	충분한 잔액 확인
Status Alarm	Use cases	ATM 현금/교통카드 보유량 부적절 경고 알림
Management	Use cases	ATM 관리자 현금/교통카드 보유량 관리
Bank	Concept	ATM System 에 등록 되어있는 은행
countryid	Attribute	은행 국가(한/미)
Rate	Attribute	환율
chargeFee	Attribute	수수료
Account	Concept	계좌
id	Attribute	계좌 식별번호
type	Attribute	계좌 종류
History	Concept	거래 내역
date	Attribute	거래 날짜/시간
sender	Attribute	송신 계좌

2035. Define System Sequence Diagrams

- Sequence Diagram이 수정되어 있지 않다.



2036. Define Operation Contracts

6.2 selectService(service)

Post-conditions	비밀번호를 확인한다.
-----------------	-------------

- 모든 Service가 비밀번호를 필요로 하지 않으므로 수정 필요 ex) Deposit

6.5 insertCash(type, amount)

Name	insertCash(type, amount)
Responsibilities	입금한 지폐의 종류와 양을 확인한다.
Type	System
Cross References	Functions : R.1.1 Use cases : Deposit
Notes	
Exceptions	-유효한 지폐(원화/달러)가 아닐경우
Output	입금한 지폐의 종류와 양을 User 에게 다시 확인
Pre-conditions	입금 서비스가 선택 되어있어야 한다
Post-conditions	명세표 확인 및 출력을 한다.

- Exceptions에 '원화와 달러를 섞어서 입금할 경우' 추가 필요

6.13 end(void)

- 1000단계에서 정의되어 있지 않으므로 1000단계에서 추가 필요.

2039. Traceability Analysis

- 앞단계의 변경 내용을 반영하여 전체적으로 수정 필요.

1.3 Stage 2040 Design

2141. Design Real Use Cases

- 2030단계와 차이가 존재하므로 전체적인 비교 필요

2144. Define Interaction Diagrams

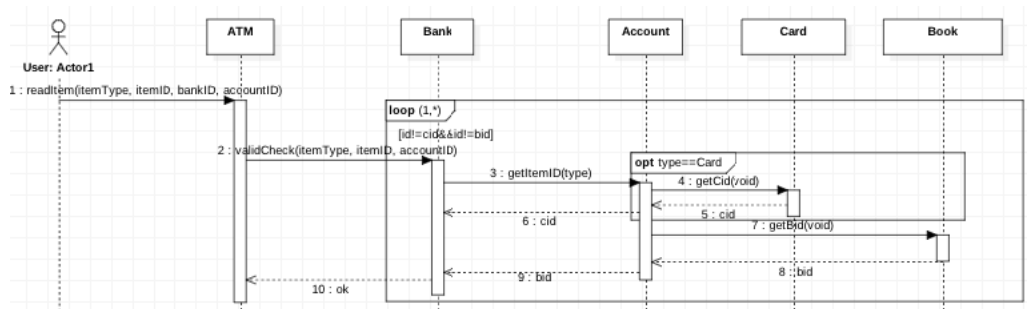
- 모든 내용이 수정 되어 있지 않다.



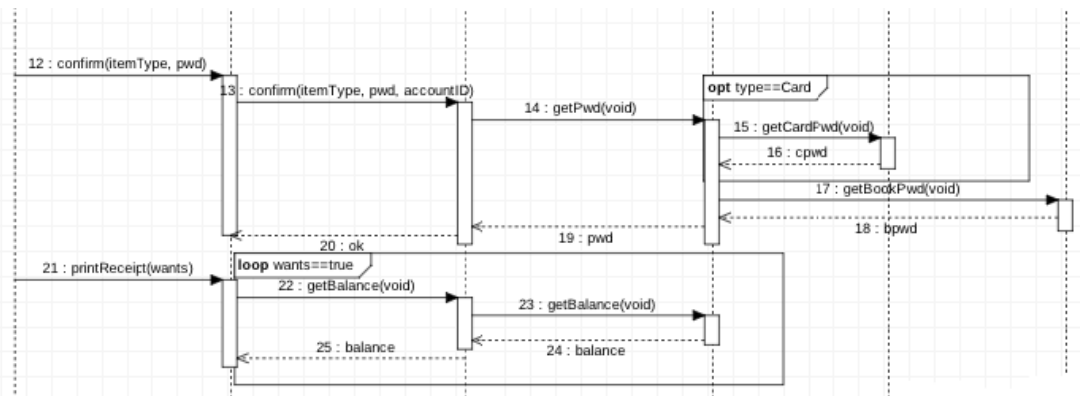
- 4.1 Check의 예시) 존재하지 않는 조건문과 존재하지 않는 return 값인 알람 존재
- readItem의 생명선이 printReceipt가 끝날때까지 이어져야 한다.

이와 관련된 부분은 아래에서 생략한다.

4.1 Check



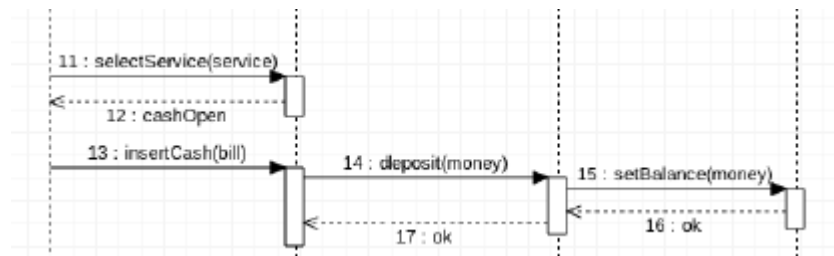
- Sequence Diagram과 흐름이 다르다.
- readItem의 생명선이 끝까지 이어져야 한다
- loop문의 조건이 잘 못 되어 있다. 명세에 의하면 모든 item이 호출될 때까지 반복한다.
- 7,8 번은 type==Book 인 경우로 분기되어야 한다.



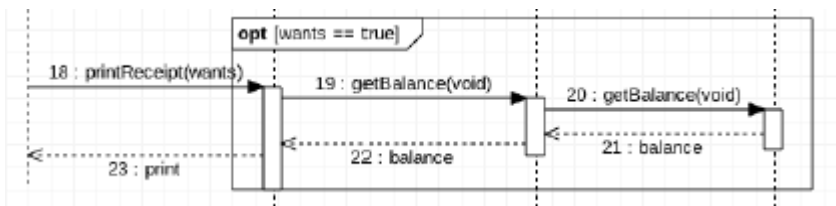
- 실제 프로그램과 동작 구조가 다르다.
객체들의 정보가 수정되고 반영되는 부분이 존재하지 않는다.
- Confirm의 생명선이 printReceipt가 끝날 때 까지 이어져야 한다.

< 이와 동일한 부분은 아래에서 생략한다. >

4.2 Deposit

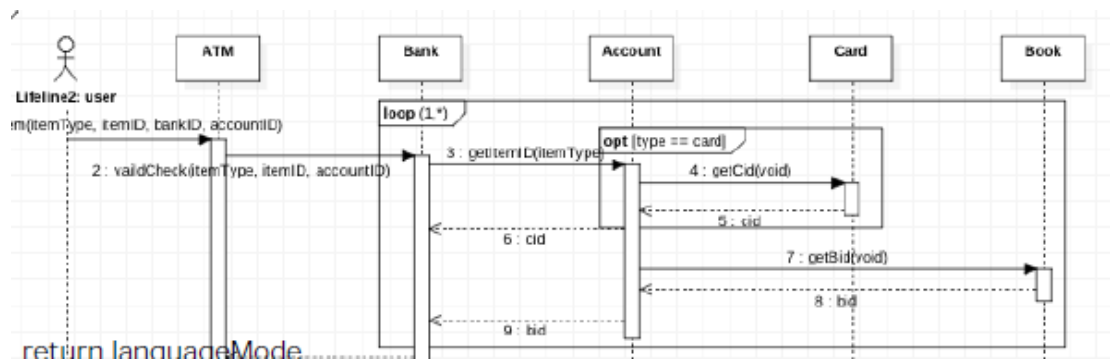


- 요구사항 명세에서는 한화와 외화를 나누는 분기가 있지만 다이어그램 상에서는 존재하지 않는다.
- deposit(money)의 반환값인 money가 명시되어 있지 않다.
- 유효한 현금인지 판별하는 분기가 나타나 있지 않다.

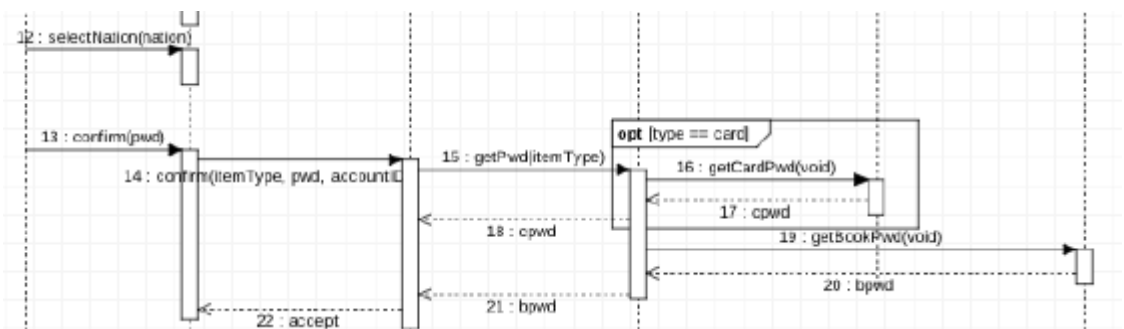


- printReceipt의 pre-condition인 confirm이 나타나 있지 않다.
- getBalance의 분기가 나타나 있지 않다.

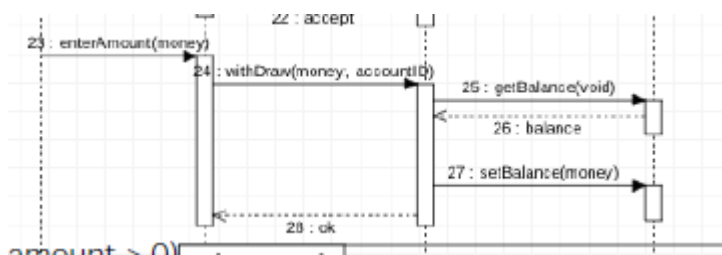
4.3 withDraw



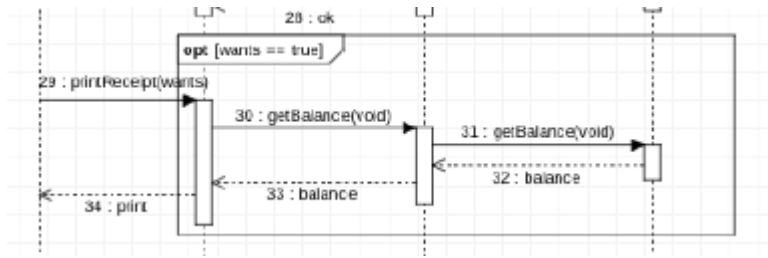
- loop문의 조건이 누락되어 있다.
- readItem의 반환값인 languageMode가 누락되었다.
- selectService는 result의 값에 따라 바뀌는데 누락되었다.



- 13번과 12번의 순서와 상관관계가 바뀌어 있다.
- Confirm의 return값이 true여야 selectNation으로 진입한다.

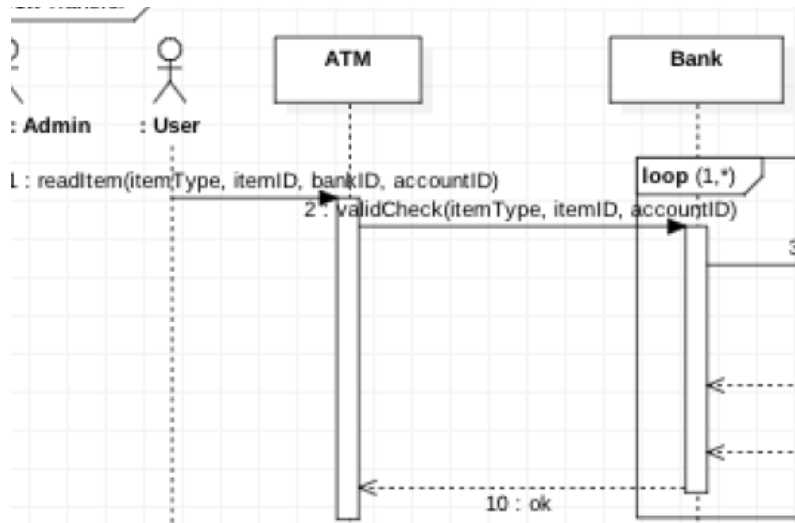


- 문서에 명시 되어있는 잔고의 부족을 검사하는 부분이 누락되어 있다.



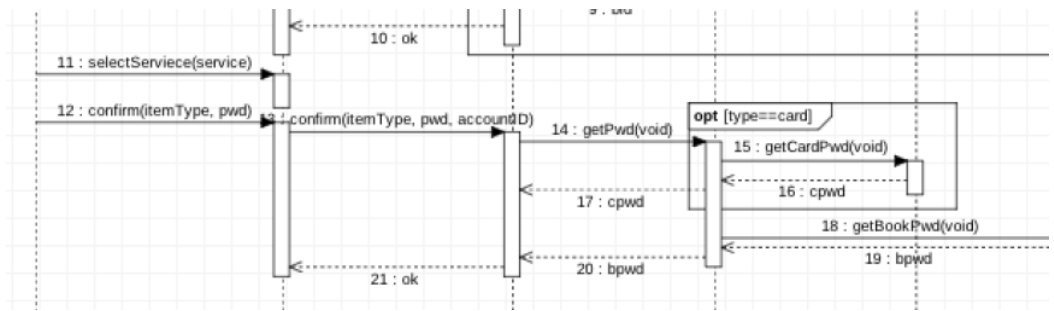
- 조건문이 잘못되어있다. ex) wants == true → amount > 0
- getBalance의 인자값이 다르다.
- ATM의 receiptAmount가 변화하는 부분이 명시되어 있지 않다.

4.4 Transfer



- readItem의 return값에 따라 selectService로의 진입 여부가 판별되는데 명시되어 있지 않다.
- selectService의 종료 이후 실행되는 inputPassword가 누락되어 있다.
- inputPassword의 결과로 반환되는 값인 mode도 누락되어 있다.
- InputTransfer이 빠져 있어 User에게 입력을 요청하는 부분이 누락되어 있다.
- printReceipt가 실행 된 후 사용자가 명세표 출력 요청을 했을 때에 명세표를 출력해 주어야 하는데, 명세표를 출력하는 부분이 누락되어 있다.

4.5 Issue TrafficCard

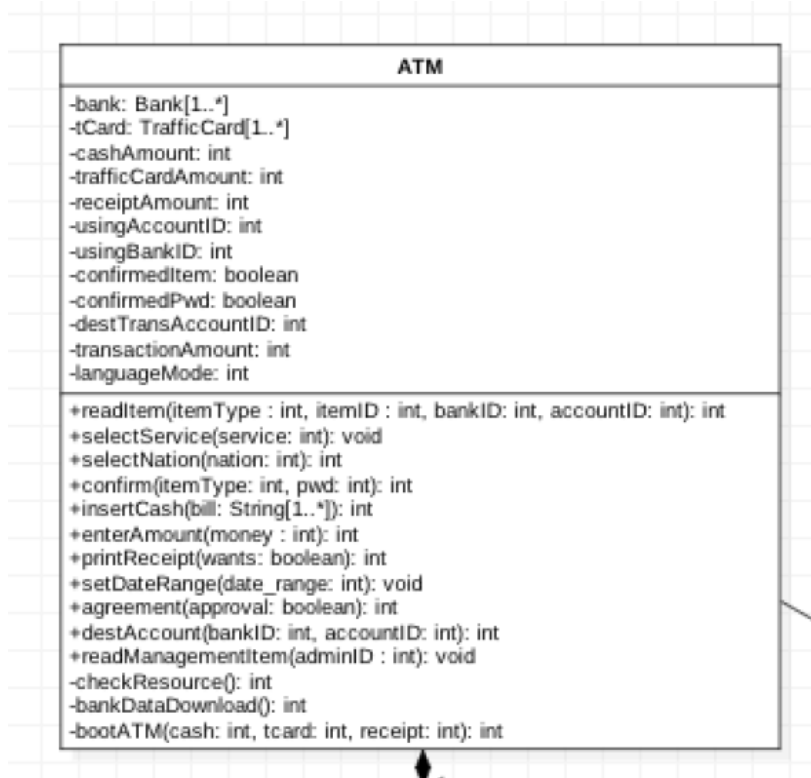


- 11번 이후 User가 Password를 입력하는 부분이 누락되어 있다.
- User가 Password를 입력하면 inputPassword를 통해 mode를 설정하는 부분이 누락되어 있다.
- Mode를 통해 나뉘는 분기문이 모두 생략되어 있다.
- dataRange를 정하는 부분이 누락되어 있다.
- 사용자가 명세표 출력을 선택하는 부분이 누락되어 있다.
- setDateRange 함수의 생명선은 agreement의 종료시까지 존재해야 한다.

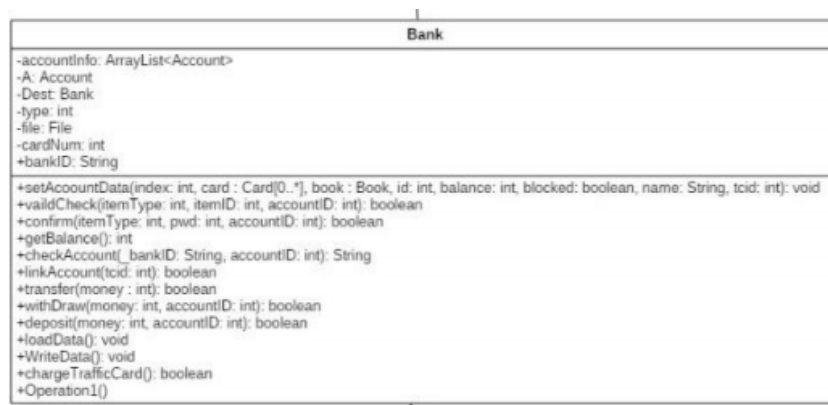
4.6 Management

- SequenceDiagram과 흐름이 다르다.
- end되기 전 관리자가 제어하는 부분이 존재하지 않는다.
- 명세에 따르면 작업 후 ATM 기기가 닫히면 end가 실행되는데, 관리자가 작업을 마치고 기기를 닫는 부분이 누락되어 있다.
- 프로그램 상에서 존재하지 않는다.

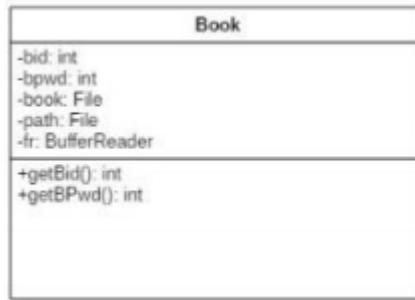
2045. Define Design Class Diagrams



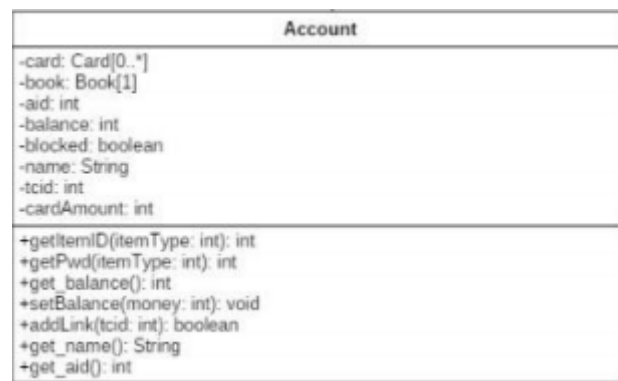
- br, bootAtm 등 중복된 변수가 여러 개 존재한다.
- Insert Cash, confirm, enterAmount, destAccount 등 메소드의 반환형이 다름.



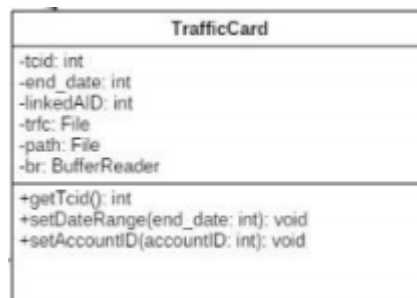
- confirm, charge, Traffic Card, withdraw, deposit의 매개변수가 코드와 문서 간 불일치
- 존재하지 않는 setAccountData가 존재한다.



- 코드에되는 File Reader가 빠져있음.



- 코드에 존재하지 않는 blocked 변수 존재
- book은 배열이 아니다.



- 코드에 사용되는 File Reader와 Count가 존재하지 않는다.

1.4 Stage 2050 Implementation

2051. Implement Class & Methods

- 모든 Class와 Method의 Cross Reference가 복사 및 복붙되어 있어서 적절하지 않은 Cross Reference가 기재되어 있다. 모든 경우에 잘못되어 있으므로 이하 생략합니다.
- insertCash

Type	Method
Name	insertCash
Purpose	User 가 입금하려는 금액을 넣는다.
Cross Reference	System Function : R.1.0, R.1.1, R.1.2, R.1.3, R.1.4, R.2.0, R.2.1, R.2.2, R.3.0 Use cases : "Check", "Deposit", "Transfer", "Withdraw", "Management", "Update", "Verify Sufficient Fund", "Status Alarm"
Input	String[] bill
Output	int
Abstract operation	지폐 code 배열을 받고 그에 알맞은 돈을 bank 에 입금해준다.
Exceptional Course of Events	ATM 기기만 <u>원금이 너무 많으면</u> 돈을 더 못 넣으니 return false 를 하게 된다.

→ 존재하지 않는 Update, '너무 많다'의 기준이 명확하지 않음

- printReceipt & setDateRange

1.1.8. printReceipt	
Type	Method
Name	printReceipt
Purpose	영세표를 출력한다.
Cross Reference	System Function : R.1.0, R.2.2, R.3.0 Use cases : "Check", "Management", "Update"
Input	
Output	boolean
Abstract operation	Parameter wants == true
Exceptional Course of Events	

1.1.8. setDateRange	
Type	Method
Name	setDateRange
Purpose	교통카드 이용 날짜를 설정

→ 오타 (setDateRange) 코드상에서 사용되지 않은 메소드

→ 넘버링 오류 (번호 겹침)

- end

1.1.11. end

Type	Method
Name	end
Purpose	관리자가 작업을 끝내면 다시 파일을 읽어 atm 정보를 업데이트 한다.
Cross Reference	System Function : R.1.0, R.1.1, R.1.2, R.1.3, R.1.4, R.2.0, R.2.1, R.2.2, R.3.0 Use cases : "Check", "Deposit", "Transfer", "Withdraw", "Management", "Update", "Verify Sufficient Fund", "Status Alarm"
Input	
Output	Void
Abstract operation	관리자가 작업을 끝내면 다시 파일을 읽어 atm 정보를 업데이트 한다.
Exceptional Course of Events	-

→ 이전 단계(1000, 2030) 에서 나오지 않음.

- Confirm & WriteData

1.2.2. Confirm		1.2.2. WriteData	
Type	Method	Type	Method
Name	Confirm	Name	WriteData
Purpose	입력한 비밀번호가 맞는지 확인한다.	Purpose	계좌파일에 바뀐 정보를 쓴다 (파일 데이터 쓰기)
Cross Reference	System Function : R.1.0, R.1.1, R.1.2, R.1.3, R.1.4, R.2.0, R.2.1, R.2.2, R.3.0 Use cases : "Check", "Deposit", "Transfer", "Withdraw", "Management", "Update", "Verify Sufficient Fund", "Status Alarm"	Cross Reference	System Function : R.1.0, R.1.1, R.1.2, R.1.3, R.1.4, R.2.0, R.2.1, R.2.2, R.3.0 Use cases : "Check", "Deposit", "Transfer", "Withdraw", "Management", "Update", "Verify Sufficient Fund", "Status Alarm"
Input (Method)	id, pwd	Input (Method)	
Output (Method)	isLogin	Output (Method)	
Abstract operation (Method)	item cache 인 Type 변수를 통해 카드인지 통장인지 구분한다.	Abstract operation (Method)	
Exceptional Course of Events	틀려온 계좌와 해당 비밀번호가 일치하지 않으면 false 을 반환한다.	Exceptional Course of Events	-

→ item cache인 Type 변수..?
→ 넘버링 오류 (번호 겹침)

- getBalance & Valid Check

넘버링 오류(번호가 겹침)

- linkAccount

Type	Method
Name	linkAccount
Purpose	교통카드와 계좌를 연동시킨다.
Cross Reference	System Function : R.1.0, R.1.1, R.1.2, R.1.3, R.1.4, R.2.0, R.2.1, R.2.2, R.3.0 Use cases : "Check", "Deposit", "Transfer", "Withdraw", "Management", "Update", "Verify Sufficient Fund", "Status Alarm"
Input (Method)	int_tcid
Output (Method)	boolean
Abstract operation (Method)	
Exceptional Course of Events	-현재 계좌가 유효하지 않으면 false 를 반환한다.

→ Cross Reference 누락

- transfer

Type	Method
Name	transfer
Purpose	송금한다.
Cross Reference	System Function : R.1.0, R.1.1, R.1.2, R.1.3, R.1.4, R.2.0, R.2.1, R.2.2, R.3.0 Use cases : "Check", "Deposit", "Transfer", "Withdraw", "Management", "Update", "Verify Sufficient Fund", "Status Alarm"
Input (Method)	int_money
Output (Method)	boolean
Abstract operation (Method)	같은 은행이라면 파일에 데이터를 두 번 덮어쓰지 않도록 한 객체 데이터를 변경한다.
Exceptional Course of Events	내 계좌의 잔액이 부족하거나 유효하지 않으면 false 를 반환한다.

→ Cross Reference 누락

- set_balance

Type	Method
Name	set_balance
Purpose	해당 계좌의 잔고를 증감한다.
Cross Reference	System Function : R.1.0, R.1.1, R.1.2, R.1.3, R.1.4, R.2.0, R.2.1, R.2.2, R.3.0 Use cases : "Check", "Deposit", "Transfer", "Withdraw", "Management", "Update", "Verify Sufficient Fund", "Status Alarm"
Input (Method)	
Output (Method)	
Abstract operation (Method)	
Exceptional Course of Events	-

→ 증감하지 않는다.

- addLink

1.3.5.addLink	
Type	Method
Name	addLink
Purpose	해당 계좌에 교통카드 정보를 입력한다.

→ 역할을 유추하기 어려운 네이밍.
 주관적이라고 할까 봐 설명드립니다.
 '계좌'와 '교통카드 정보'의 '연결'을 '추가' 하는건데 '연결을 추가' 한다는 정보밖에 없음.

- waitReadItem => 기다리지 않음

Name	waitReadItem
Responsibilities	User 가 Item 을 인식시키는 것을 기다린다.
Type	GUI
Cross Reference	System Function : R.1.0 , R.1.1 , R.1.2 , R.1.3 , R.1.4 , R.2.0 , R.2.1, R.2.2, R.3.0 Use cases : "Check" , "Deposit" , "Transfer" , "Withdraw" , "Management" , "Update" , "Verify Sufficient Fund" , "Status Alarm"
Notes	-
PreConditions	ATM 실행 상태
PostConditions	User 의 Item 값 입력

- selectNation => 오타 Nation

Name	select Nation
Responsibilities	출금하고자하는 지폐 종류를 선택한다.
Type	GUI
Cross Reference	System Function : R.1.3 Use cases : "Withdraw"
Notes	-
Pre-Conditions	출금 서비스 선택
Post-Conditions	원 / 달러 선택

1.5 Stage 2060 Unit Test & System Test

2063. System Testing

- 소프트웨어 모델링 팀 측에서 테스트를 전혀 진행하지 않았다.

2 Category-partition Testing Report

2.1 Categorize

Group	Category	Values	#
Read Item	Read Item	Valid Input	100
		Invalid Item Type	101
		Invalid Item Id	102
		Invalid Item Bank	103
		Invalid Item Account ID	104
Mode	Mode	Check	200
		Deposit	201
		Withdraw	202
		Transfer	203
		Issue Traffic Card	204
Password	Password	Valid Password	300
		Invalid Password	301
Data	Select Data	Valid Money Once	400
		Valid Single Type Money	401
		Valid Multiple Type Money	402
		Invalid Money	403
		Currency	404
	Select Bank	Bank.	410
	Select Account ID	Valid Account ID	420
		Invalid Account ID	421
	Input Data	Valid Money	430
		Invalid Money	431
		Valid Duration	432
Invalid Duration		433	
Agreement	Agreement	Agree	500
		Disagree	501
Change Balance	Actor	Increase Balance	600
		Decrease Balance	601
		Send Money	602
	Admin	Increase Balance	610
Decrease Balance		611	
Print Receipt	Receipt	Print Receipt	700
		Not Print Receipt	701
Management	Manage TC	Not Enough TC	800
		Enough TC	801
	Manage Receipt	Not Enough Receipt	810
		Enough Receipt	811
	Manage Money	Not Enough Money	820
		Enough Money	821
Too Many Money		822	
Alarm	Alarm	Send Alarm	900
		Not Send Alarm	901
Cancel	Cancel	Cancel	1000
		Clear	1001
		Backspace	1002

2.2 Description

Values	Description
Valid Input	4가지 항목에 모두 올바른 값을 입력한 경우
Invalid Item Type	잘못된 Item Type을 입력한 경우
Invalid Item Id	잘못된 Item ID를 입력한 경우
Invalid Item Bank	잘못된 Item Bank를 입력한 경우
Invalid Item Account ID	잘못된 Item Account ID를 입력한 경우
Check	계좌의 잔액을 확인한다
Deposit	돈을 예금한다.
Withdraw	돈을 인출한다.
Transfer	돈을 송금한다.
Issue Traffic Card	교통카드를 발급한다.
Valid Password	유효한 비밀번호를 입력한.
Invalid Password	유효하지 않은 비밀번호를 입력한다.
Valid Money Once	금액을 한번 선택한다.
Valid Single Type Money	한 종류의 금액을 여러 번 선택한다.
Valid Multiple Type Money	여러 종류의 금액을 여러 번 선택한다.
Invalid Money	유효하지 않은 금액을 입력한다.
Currency	거래를 진행할 화폐를 선택한다.
Bank.	거래를 진행할 은행을 선택한다.
Valid Account ID	유효한 Account ID를 입력한다.
Invalid Account ID	유효하지 않은 Account ID를 입력한다.
Valid Money	유효한 금액을 입력한다.
Invalid Money	유효하지 않은 금액을 입력한다.
Valid Duration	유효한 기간을 입력한다.
Invalid Duration	유효하지 않은 기간을 입력한다.
Agree	이용 약관에 동의한다.
Disagree	이용 약관에 동의하지 않는다.
Increase Balance	행위자의 금액을 증가시킨다.
Decrease Balance	행위자의 금액을 감소시킨다
Send Money	돈을 송금한다.
Increase Balance	관리자의 금액을 증가시킨다.
Decrease Balance	관리자의 금액을 감소시킨다
Print Receipt	영수증을 출력한다.
Not Print Receipt	영수증을 출력하지 않는다.
Not Enough TC	교통카드가 부족하다.
Enough TC	교통카드가 충분하다.
Not Enough Receipt	영수증을 출력할 용지가 부족하다.
Enough Receipt	영수증을 출력할 용지가 충분하다.
Not Enough Money	ATM의 지폐가 충분하지 않다.
Enough Money	ATM의 지폐가 충분하다.
Too Many Money	ATM의 지폐가 너무 많다.
Send Alarm	관리자에게 알람을 보낸다.
Not Send Alarm	관리자에게 알람을 보내지 않는다.
Cancel	선택을 취소한다.
Clear	선택을 초기화한다.
Backspace	메인화면으로 돌아간다.

2.3 Property & Constraints

Category	Values [Property prop]	Constraints
Read Item	Valid Input	
	Invalid Item Type	[Error]
	Invalid Item Id	[Error]
	Invalid Item Bank	[Error]
	Invalid Item Account ID	[Error]
Mode	Check [Property Check]	
	Deposit [Property Deposit]	
	Withdraw [Property Withdraw]	
	Transfer [Property Transfer]	
	Issue Traffic Card [Property IssueTC]	
Password	Valid Password	[if Check Withdraw Transfer IssueTC]
	Invalid Password	[error] [if Check Withdraw Transfer IssueTC]
Select Data	Valid Money Once	[if Deposit]
	Valid Single Type Money	[if Deposit]
	Valid Multiple Type Money	[if Deposit]
	Invalid Money	[if Deposit] [Error]
	Currency	[if Withdraw]
Select Bank	Bank	[if Transfer]
Select Account ID	Valid Account ID	[if Transfer]
	Invalid Account ID	[if Transfer] [Error]
Input Data	Valid Money	[if Withdraw Transfer]
	Invalid Money	[if Withdraw Transfer] [Error]
	Valid Duration	[if IssueTC]
	Invalid Duration	[if IssueTC] [Error]
Agreement	Agree	[if Transfer]
	Disagree	[if Transfer]
Actor	Increase Balance	[if Deposit]
	Decrease Balance	[if Withdraw IssueTC]
	Send	[if Transfer]
Admin	Increase Balance [property IncreaseBalance]	[if Deposit IssueTC]
	Decrease Balance [property DecreaseBalance]	[if Transfer Withdraw]
Receipt	Print Receipt [property PrintReceipt]	
	Not Print Receipt	
Manage TC	Not Enough TC [property TCAlarm]	[if IssueTC]
	Enough TC	[if IssueTC]
Manage Receipt	Not Enough Receipt [property ReceiptAlarm]	[if PrintReceipt]
	Enough Receipt	[if PrintReceipt]
Manage Money	Not Enough Money [property NMoneyAlarm]	[if DecreaseBalance]
	Enough Money	[if IncreaseBalance DecreaseBalance]
	Too Many Money [property TMoneyAlarm]	[if IncreaseBalance]
Alarm	Send Alarm	[if TCAlarm ReceiptAlarm NMoneyAlarm TMoneyAlarm]
	Not Send Alarm	[Error]
Cancel Mode	Cancel	[Single]
	Clear	[Single]
	Backspace	[Single]

2.4 TextFile for TSL-Generator

```

# Read Item
Read Item:
    Valid Input,
    Invalid ItemType,
    Invalid ItemId,
    Invalid ItemBank,
    Invalid ItemAccountId,
    [error]
    [error]
    [error]
    [error]

# Mode
Mode:
    Check,
    Deposit,
    Withdraw,
    Transfer,
    Issue TrafficCard,
    [property Check]
    [property Deposit]
    [property Withdraw]
    [property Transfer]
    [property IssueTC]

# Password
Password:
    Valid Password,
    Invalid Password,
    [if Check || Withdraw || Transfer || IssueTC]
    [error] [if Check || Withdraw || Transfer || IssueTC]

# Data
Select Data:
    Valid Money Once,
    Valid Single Type Money,
    Valid Multiple Type Money,
    Invalid Money,
    Currency,
    [if Deposit]
    [if Deposit]
    [if Deposit]
    [error] [if Deposit] #Overflow
    [if Withdraw]

Select Bank:
    Bank,
    [if Transfer]

Select AccountID:
    Valid AccountID,
    Invalid AccountID,
    [if Transfer]
    [error] [if Transfer] #존재하지 않는 AccountID 입력

Input Data:
    Valid Money,
    Invalid Money,
    Valid Duration,
    Invalid Duration,
    [if Withdraw || Transfer]
    [error] [if Withdraw || Transfer] #가진 금액 이상 입력
    [if IssueTC]
    [error] [if IssueTC] #00일

# Agreement
Agreement:
    Agree,
    Disagree,
    [if Transfer]
    [if Transfer]

# Change Balance
Actor:
    Increase Balance,
    Decrease Balance,
    Send Money,
    [if Deposit]
    [if Withdraw || IssueTC]
    [if Transfer]

Admin:
    Increase Balance,
    Decrease Balance,
    [if Deposit || IssueTC] [property IncreaseBalance]
    [if Transfer || Withdraw] [property DecreaseBalance]

# Print Receipt
Receipt:
    Print Receipt,
    Not Print Receipt,
    [property PrintReceipt]

# Management
Manage TC:
    Not Enough TC,
    Enough TC,
    [if IssueTC] [property TCAAlarm]
    [if IssueTC]

Manage Receipt:
    Not Enough Receipt,
    Enough Receipt,
    [if PrintReceipt][property ReceiptAlarm]
    [if PrintReceipt]

Manage Money:
    Not Enough Money,
    Enough Money,
    Too Many Money,
    [if DecreaseBalance][property NMoneyAlarm]
    [if IncreaseBalance || DecreaseBalance]
    [if IncreaseBalance][property TMoneyAlarm]

# Alarm
Alarm:
    Send Alarm,
    Not Send Alarm,
    [if TCAAlarm || ReceiptAlarm || NMoneyAlarm || TMoneyAlarm]
    [error]

# Cancel
Cancel:
    Cancel,
    Clear,
    Backspace,
    [single]
    [single]
    [single]

```

2.5 Test Result

1. Error 수정 전 => 0%

TestCase	Result
Test1- 101	Failed
Test2- 102	Failed
Test3- 103	Failed
Test4- 104	Failed
Test5- 301	Failed
Test6- 403	Failed
Test7- 421	Failed
Test8- 431	Failed
Test9- 433	Failed
Test10- 901	Failed
Test11- 1000	Failed
Test12- 1001	Failed
Test13- 1002	Failed
Test14- 100.200.300.700.810.900	Failed
Test15- 100.200.300.700.811	Failed
Test16- 100.200.300.701	Failed
Test17- 100.201.400.600.610.700.810.821.900	Failed
Test18- 100.201.400.600.610.700.810.822.900	Failed
Test19- 100.201.400.600.610.700.811.821	Failed
Test20- 100.201.400.600.610.700.811.822.900	Failed
Test21- 100.201.400.600.610.701.821	Failed
Test22- 100.201.400.600.610.701.822.900	Failed
Test23- 100.201.401.600.610.700.810.821.900	Failed
Test24- 100.201.401.600.610.700.810.822.900	Failed
Test25- 100.201.401.600.610.700.811.821	Failed
Test26- 100.201.401.600.610.700.811.822.900	Failed
Test27- 100.201.401.600.610.701.821	Failed
Test28- 100.201.401.600.610.701.822.900	Failed
Test29- 100.201.402.600.610.700.810.821.900	Failed
Test30- 100.201.402.600.610.700.810.822.900	Failed
Test31- 100.201.402.600.610.700.811.821	Failed
Test32- 100.201.402.600.610.700.811.822.900	Failed
Test33- 100.201.402.600.610.701.821	Failed
Test34- 100.201.402.600.610.701.822.900	Failed
Test35- 100.202.300.404.430.601.611.700.810.820.900	Failed
Test36- 100.202.300.404.430.601.611.700.810.821.900	Failed
Test37- 100.202.300.404.430.601.611.700.811.820.900	Failed
Test38- 100.202.300.404.430.601.611.700.811.821	Failed
Test39- 100.202.300.404.430.601.611.701.820.900	Failed
Test40- 100.202.300.404.430.601.611.701.821	Failed
Test41- 100.203.300.410.420.430.500.602.611.700.810.820.900	Failed
Test42- 100.203.300.410.420.430.500.602.611.700.810.821.900	Failed
Test43- 100.203.300.410.420.430.500.602.611.700.811.820.900	Failed
Test44- 100.203.300.410.420.430.500.602.611.700.811.821	Failed
Test45- 100.203.300.410.420.430.500.602.611.701.820.900	Failed
Test46- 100.203.300.410.420.430.500.602.611.701.821	Failed
Test47- 100.203.300.410.420.430.501.602.611.700.810.820.900	Failed
Test48- 100.203.300.410.420.430.501.602.611.700.810.821.900	Failed
Test49- 100.203.300.410.420.430.501.602.611.700.811.820.900	Failed
Test50- 100.203.300.410.420.430.501.602.611.700.811.821	Failed
Test51- 100.203.300.410.420.430.501.602.611.701.820.900	Failed
Test52- 100.203.300.410.420.430.501.602.611.701.821	Failed
Test53- 100.204.300.432.601.610.700.801.810.821.900	Failed
Test54- 100.204.300.432.601.610.700.801.810.822.900	Failed
Test55- 100.204.300.432.601.610.700.801.811.821.900	Failed
Test56- 100.204.300.432.601.610.700.801.811.822.900	Failed
Test57- 100.204.300.432.601.610.700.801.810.821.900	Failed
Test58- 100.204.300.432.601.610.700.801.810.822.900	Failed
Test59- 100.204.300.432.601.610.700.801.811.821	Failed
Test60- 100.204.300.432.601.610.700.801.811.822.900	Failed
Test61- 100.204.300.432.600.610.701.801.821.900	Failed
Test62- 100.204.300.432.601.610.701.801.822.900	Failed
Test63- 100.204.300.432.601.610.701.801.821	Failed
Test64- 100.204.300.432.601.610.701.801.822.900	Failed
전체 Test Case: 64개	Pass: 0개

0/64 (0%) Pass

첫번째 테스트에서 Text file의 경로오류가 있어 수정하였습니다. 수정된 코드를 SMA 팀에 전달하였으나 이를 주석처리하고 다시 제대로 동작하지 않는 코드를 깃에 업로드했기 때문에 0%의 결과가 나왔습니다.

2. Error 수정 후(SV 팀에서 수정 진행) => 25%

TestCase	Result	Fail Reason (2nd)
Test1- 101	Passed	
Test2- 102	Passed	
Test3- 103	Passed	
Test4- 104	Passed	
Test5- 301	Passed	
Test6- 403	Passed	
Test7- 421	Passed	
Test8- 431	Passed	
Test9- 433	Failed	교통카드 개수가 남아있음에도 교통카드 개수가 부족하다고 함
Test10- 901	Failed	Alarm 을 보내지 않는 경우는 Error 상황
Test11- 1000	Passed	
Test12- 1001	Passed	
Test13- 1002	Passed	
Test14- 100.200.300.700.810.900	Failed	영수증이 부족한 경우에 알람을 보내지 않음
Test15- 100.200.300.700.811	Passed	
Test16- 100.200.300.701	Passed	
Test17- 100.201.400.600.610.700.810.821.900	Failed	영수증이 부족한 경우에 알람을 보내지 않음
Test18- 100.201.400.600.610.700.810.822.900	Failed	영수증이 부족하고 돈이 너무 많은 경우 알람을 보내지 않음
Test19- 100.201.400.600.610.700.811.821	Failed	50000 원을 넣었을 때 화면에 5000 원이 출력된다.
Test20- 100.201.400.600.610.700.811.822.900	Failed	돈이 너무 많은 경우 알람을 보내지 않음
Test21- 100.201.400.600.610.701.821	Failed	50000 원을 넣었을 때 화면에 5000 원이 출력된다.
Test22- 100.201.400.600.610.701.822.900	Failed	돈이 너무 많은 경우 알람을 보내지 않음
Test23- 100.201.401.600.610.700.810.821.900	Failed	영수증이 부족한 경우 알람을 보내지 않음
Test24- 100.201.401.600.610.700.810.822.900	Failed	영수증이 부족하거나 돈이 너무 많은 경우 알람을 보내지 않음
Test25- 100.201.401.600.610.700.811.821	Failed	50000 원을 넣었을 때 화면에 5000 원이 출력된다.
Test26- 100.201.401.600.610.700.811.822.900	Failed	돈이 너무 많은 경우 알람을 보내지 않음
Test27- 100.201.401.600.610.701.821	Failed	50000 원을 넣었을 때 화면에 5000 원이 출력된다.
Test28- 100.201.401.600.610.701.822.900	Failed	돈이 너무 많은 경우 알람을 보내지 않음
Test29- 100.201.402.600.610.700.810.821.900	Failed	여러 화폐가 동시에 입금되지 않으며 영수증이 부족한 경우 알람을 보내지 않음
Test30- 100.201.402.600.610.700.810.822.900	Failed	여러 화폐가 동시에 입금되지 않으며 영수증이 부족하거나 돈이 많은 경우 알람을 보내지 않음
Test31- 100.201.402.600.610.700.811.821	Failed	50000 원을 넣었을 때 화면에 5000 원이 출력된다.
Test32- 100.201.402.600.610.700.811.822.900	Failed	여러 화폐가 동시에 입금되지 않으며 돈이 너무 많은 경우 알람을 보내지 않음
Test33- 100.201.402.600.610.701.821	Failed	50000 원을 넣었을 때 화면에 5000 원이 출력된다.
Test34- 100.201.402.600.610.701.822.900	Failed	여러 화폐가 동시에 입금되지 않으며 돈이 너무 많은 경우 알람을 보내지 않음.
Test35- 100.202.300.404.430.601.611.700.810.820.900	Failed	영수증이 부족하거나 돈이 부족한 경우 알람을 보내지 않음
Test36- 100.202.300.404.430.601.611.700.810.821.900	Failed	영수증이 부족한 경우 알람을 보내지 않음
Test37- 100.202.300.404.430.601.611.700.811.820.900	Failed	돈이 부족한 경우 알람을 보내지 않음
Test38- 100.202.300.404.430.601.611.700.811.821	Failed	돈을 0 원 또는 0 달러 출금시 수수료만 요구
Test39- 100.202.300.404.430.601.611.701.820.900	Failed	돈이 부족한 경우 알람을 보내지 않음
Test40- 100.202.300.404.430.601.611.701.821	Failed	돈을 0 원 또는 0 달러 출금시 수수료만 요구
Test41- 100.203.300.410.420.430.500.602.611.700.810.820.900	Failed	영수증이 부족하거나 돈이 부족한 경우 알람을 보내지 않음
Test42- 100.203.300.410.420.430.500.602.611.700.810.821.900	Failed	영수증이 부족한 경우 알람을 보내지 않음
Test43- 100.203.300.410.420.430.500.602.611.700.811.820.900	Failed	돈이 부족한 경우 알람을 보내지 않음
Test44- 100.203.300.410.420.430.500.602.611.700.811.821	Passed	
Test45- 100.203.300.410.420.430.500.602.611.701.820.900	Failed	돈이 부족한 경우 알람을 보내지 않음
Test46- 100.203.300.410.420.430.500.602.611.701.821	Passed	
Test47- 100.203.300.410.420.430.501.602.611.700.810.820.900	Failed	영수증이 부족하거나 돈이 부족한 경우 알람을 보내지 않음
Test48- 100.203.300.410.420.430.501.602.611.700.810.821.900	Failed	영수증이 부족한 경우 알람을 보내지 않음
Test49- 100.203.300.410.420.430.501.602.611.700.811.820.900	Failed	돈이 부족한 경우 알람을 보내지 않음
Test50- 100.203.300.410.420.430.501.602.611.700.811.821	Failed	한 자리수(예를 들어 2)를 입력하기 위해 2 Enter 를 누르면 제대로 인식되지 않고 002 를 눌러야 함
Test51- 100.203.300.410.420.430.501.602.611.701.820.900	Failed	돈이 부족한 경우 알람을 보내지 않음
Test52- 100.203.300.410.420.430.501.602.611.701.821	Failed	한 자리수(예를 들어 2)를 입력하기 위해 2 Enter 를 누르면 제대로 인식되지 않고 002 를 눌러야 함
Test53- 100.204.300.432.601.610.700.801.810.821.900	Failed	교통카드가 부족하거나 영수증이 부족한 경우 알람을 보내지 않음
Test54- 100.204.300.432.601.610.700.801.810.822.900	Failed	교통카드가 부족하거나 영수증이 부족하거나 돈이 많은 경우 알람을 보내지 않음
Test55- 100.204.300.432.601.610.700.801.811.821.900	Failed	교통카드가 부족한 경우 알람을 보내지 않음
Test56- 100.204.300.432.601.610.700.801.811.822.900	Failed	교통카드가 부족하거나 돈이 많은 경우 알람을 보내지 않음
Test57- 100.204.300.432.601.610.700.801.810.821.900	Failed	영수증이 부족한 경우 알람을 보내지 않음

Test58- 100.204.300.432.601.610.700.801.810.822.900	Failed	영수증이 부족하거나 돈이 많은 경우 알람을 보내지 않음
Test59- 100.204.300.432.601.610.700.801.811.821	Failed	문서에는 카드발급수수료가 0.3 달러라고 되어있으나 3 달러가 빠져나감
Test60- 100.204.300.432.601.610.700.801.811.822.900	Failed	돈이 너무 많은 경우 알람을 보내지 않음
Test61- 100.204.300.432.600.610.701.801.821.900	Failed	교통카드가 부족한 경우 알람을 보내지 않음
Test62- 100.204.300.432.601.610.701.801.822.900	Failed	교통카드가 부족하고 돈이 많은 경우 알람을 보내지 않음
Test63- 100.204.300.432.601.610.701.801.821	Failed	문서에는 카드발급수수료가 0.3 달러라고 되어있으나 3 달러가 빠져나감
Test64- 100.204.300.432.601.610.701.801.822.900	Failed	돈이 너무 많은 경우 알람을 보내지 않음

16/64 (25%) Pass

3 Pairwise Testing Report

3.1 Categorize

Category	No	Values
ReadItem	100	Valid
Mode	200	Deposit
	201	Withdraw
	202	Transfer
	203	IssueTC
InputPassword	300	Valid
	301	Invalid
InputBank	400	Select
InputAccountID	500	Valid
	501	Invalid
InputCurrency	600	Select
InputMoney	700	ValidInput
	701	ValidSelectOnce
	702	ValidSelectSingleType
	703	ValidSelectMultipleType
	704	Invalid
Agreement	800	Agree
	801	Disagree
TCAlarm	900	Send
	901	NotSend
ActorTransaction	1000	IncreaseBalance
	1001	DecreaseBalance
	1002	SendMoney
AdminTransaction	1100	IncreaseBalance
	1101	DecreaseBalance
TransactionAlarm	1200	Send
	1201	NotSend
Receipt	1300	Print
	1301	NotPrint
ReceiptAlarm	1400	Send
	1401	NotSend

3.2 Description

Values	Description
Valid	4 가지 항목에 모두 올바른 값을 입력한 경우
Check	계좌의 잔액을 확인한다
Deposit	돈을 예금한다.
Withdraw	돈을 인출한다.
Transfer	돈을 송금한다.
IssueTC	교통카드를 발급한다.
Valid	유효한 비밀번호를 입력한다.
Invalid	유효하지 않은 비밀번호를 입력한다.
Select	거래를 진행할 은행을 선택한다.
Valid	유효한 Account ID 를 입력한다.
Invalid	유효하지 않은 Account ID 를 입력한다.
Valid	유효한 기간을 입력한다.
Invalid	유효하지 않은 기간을 입력한다.
Select	거래를 진행할 화폐를 선택한다.
ValidInput	유효한 금액을 입력한다.
ValidSelectOnce	금액을 한번 선택한다.
ValidSelectSingleType	한 종류의 금액을 여러 번 선택한다.
ValidSelectMultipleType	여러 종류의 금액을 여러 번 선택한다.
Invalid	유효하지 않은 금액을 입력한다.
Agree	이용 약관에 동의한다.
Disagree	이용 약관에 동의하지 않는다.
Send	교통카드 수량이 부족한 경우 알람을 보낸다.
NotSend	교통카드 수량이 부족한 경우 알람을 보낸다.
IncreaseBalance	행위자의 금액을 증가시킨다.
DecreaseBalance	행위자의 금액을 감소시킨다
SendMoney	돈을 송금한다.
IncreaseBalance	관리자의 금액을 증가시킨다.
DecreaseBalance	관리자의 금액을 감소시킨다
Send	현금이 부족한 경우 알람을 보낸다.
NotSend	현금이 부족한 경우 알람을 보낸다.
Print	영수증을 출력한다.
NotPrint	영수증을 출력하지 않는다.
Send	영수증 용지가 부족한 경우 알람을 보낸다.
NotSend	영수증 용지가 부족한 경우 알람을 보내지 않는다.

3.3 Textfile for PICT

```
ReadItem: Valid
Mode: Check, Deposit, Withdraw, Transfer, IssueTC
InputPassword: Valid, Skip
InputBank: Select, Skip
InputAccountID: Valid, Skip
InputDuration: Valid, Skip
InputCurrency: Select, Skip
InputMoney: ValidInput, ValidSelectOnce, ValidSelectSingleType, ValidSelectMultipleType, Skip
Agreement: Agree, Disagree, Skip
TCAAlarm: Send, Skip
ActorTransaction: IncreaseBalance, DecreaseBalance, SendMoney, Skip
AdminTransaction: IncreaseBalance, DecreaseBalance, Skip
TransactionAlarm: Send, Skip
Receipt: Print, NotPrint
ReceiptAlarm: Send

#
# Mode has constraints on InputPassword
#
if [Mode] in {"Check", "Withdraw", "Transfer", "IssueTC"} then [InputPassword] = "Valid"
else [InputPassword] = "Skip";

#
# Mode has constraints on InputBank
#
if [Mode] = "Transfer" then [InputBank] = "Select"
else [InputBank] = "Skip";

#
# Mode has constraints on InputAccountID
#
if [Mode] = "Transfer" then [InputAccountID] = "Valid"
else [InputAccountID] = "Skip";

#
# Mode has constraints on InputDuration
#
if [Mode] = "IssueTC" then [InputDuration] = "Valid"
else [InputDuration] = "Skip";

#
# Mode has constraints on InputCurrency
#
if [Mode] = "Withdraw" then [InputCurrency] = "Select"
else [InputCurrency] = "Skip";

#
# Mode has constraints on InputMoney
#
if [Mode] = "Deposit" then [InputMoney] in {"ValidSelectOnce", "ValidSelectSingleType", "ValidSelectMultipleType"};
if [Mode] in {"Withdraw", "Transfer"} then [InputMoney] = "ValidInput";
if [Mode] in {"Check", "IssueTC"} then [InputMoney] = "Skip";

#
# Mode has constraints on Agreement
#
if [Mode] = "Transfer" then [Agreement] in {"Agree", "Disagree"}
else [Agreement] = "Skip";

#
# Mode has constraints on TCAAlarm
#
if [Mode] = "IssueTC" then [TCAAlarm] = "Send"
else [TCAAlarm] = "Skip";

#
# Mode has constraints on ActorTransaction
#
if [Mode] = "Deposit" then [ActorTransaction] = "IncreaseBalance";
if [Mode] in {"Withdraw", "IssueTC"} then [ActorTransaction] = "DecreaseBalance";
if [Mode] = "Transfer" then [ActorTransaction] = "SendMoney";
if [Mode] = "Check" then [ActorTransaction] = "Skip";

#
# Mode has constraints on AdminTransaction
#
if [Mode] in {"Deposit", "IssueTC"} then [AdminTransaction] = "IncreaseBalance";
if [Mode] in {"Transfer", "Withdraw"} then [AdminTransaction] = "DecreaseBalance";
if [Mode] = "Check" then [AdminTransaction] = "Skip";

#
# Mode has constraints on TransactionAlarm
#
if [Mode] in {"Deposit", "Withdraw", "Transfer", "IssueTC"} then [TransactionAlarm] = "Send"
else [TransactionAlarm] = "Skip";
```

3.4 Test Result

1. Error 수정 전

CPT와 마찬가지로 프로그램의 도입부에 치명적인 에러가 존재하여 테스트 결과가 0/25(0%)로 계산되었습니다.

2. Error 수정 후

No.	TestCase	Result	Description
1	100.200.300.1401.1500	Passed	
2	100.200.300.1400.1500	Failed	영수증 용지가 부족한 경우 알람이 오지 않음.
3	100.200.301	Passed	
4	100.201.802.1100.1200.1300.1400.1500	Failed	50000원을 눌렀을 때 화면에 5000원이 표시됨.
5	100.201.300.802.1100.1200.1300.1401.1500	Failed	50000원을 눌렀을 때 화면에 5000원이 표시됨.
6	100.201.300.803.1100.1200.1300.1400.1500	Failed	50000원을 눌렀을 때 화면에 5000원이 표시됨.
7	100.201.801.1100.1200.1300.1401.1500	Failed	현금이 부족한 경우 알람이 오지 않음.
8	100.201.803.1100.1200.1300.1401.1500	Failed	50000원을 눌렀을 때 화면에 5000원이 표시됨.
9	100.201.300.801.1100.1200.1300.1400.1500	Failed	현금이나 영수증 용지가 부족한 경우 알람이 오지 않음.
10	100.201.301	Passed	
11	100.202.300.700.800.1100.1201.1300.1401.1500	Failed	한 자리 또는 두 자리의 돈을 정상적으로 출금할 수 없음.
12	100.202.300.700.800.1100.1201.1300.1400.1500	Failed	한 자리 또는 두 자리의 돈을 정상적으로 출금할 수 없음.
13	100.202.300.700.804	Failed	돈을 0원 또는 0달러 출금시 수수료만 요구
14	100.202.301	Failed	돈을 0원 또는 0달러 출금시 수수료만 요구
15	100.203.300.400.500.800.900.1102.1201.1300.1401.1500	Failed	대부분의 계좌가 조회되지 않음.
16	100.203.300.400.500.800.901.1102.1201.1300.1401.1500	Failed	대부분의 계좌가 조회되지 않음.
17	100.203.300.400.500.800.900.1102.1201.1300.1400.1500	Failed	대부분의 계좌가 조회되지 않음.
18	100.203.300.400.500.800.901.1102.1201.1300.1400.1500	Failed	대부분의 계좌가 조회되지 않음.
19	100.203.300.400.501	Passed	
20	100.203.300.400.500.804	Failed	대부분의 계좌가 조회되지 않음.
21	100.203.301	Passed	
22	100.204.300.600.1000.1100.1200.1300.1400.1500	Failed	교통카드 잔량이나 현금이 부족한 경우 알람이 오지 않음.
23	100.204.300.600.1000.1100.1200.1300.1401.1500	Failed	교통카드 잔량이나 현금이 부족한 경우 알람이 오지 않음.
24	100.204.301.601	Failed	유효기간이 01부터 시작해야 하나 00인 경우도 발급됨.
25	100.204.301	Passed	

6 / 25 (24%) Pass

4 Brute Force Testing Report

4.1 Testing Result

테스트번호	기존 P/F	현재 P/F	이슈 번호	비고
1-1	Fail (Pass)	Fail (Pass)		- (변화 없음)
1-2	Fail (Pass)	Fail (Pass)		- (변화 없음)
1-3	Fail (Fail)	Fail (Pass)		수정 확인.
1-4	Fail (Fail)	Fail (Pass)		수정 확인.
1-5	Fail (Fail)	Fail (Pass)		수정 확인.
2-1	Fail (Fail)	Fail (Pass)		수정 확인. (GitHub issue tracker 내용 확인 요함)
2-2	Fail (Pass)	Fail (Pass)		- (변화 없음)
3-1	Fail (Fail)	Fail (Pass)		수정 확인.
3-2	Fail (Pass)	Fail (Pass)		- (변화 없음)
3-3	Fail (Fail)	Fail (Pass)		수정 확인.
3-4	Fail (Fail)	Fail (Fail)		999 만원 까지 밖에 출금을 진행할 수 없다.
4-1	Fail (Pass)	Fail (Pass)		- (변화 없음)
4-2	Fail (Fail)	Fail (Pass)		수정 확인.
4-3	Fail (Fail)	Fail (Fail)		999 만원 까지 밖에 이체를 진행할 수 없다.
4-4	Fail (Pass)	Fail (Pass)		- (변화 없음)
4-5	Fail (Pass)	Fail (Pass)		- (변화 없음)

5-1	Fail (Fail)	Fail (Fail)		메일 기능 비정상. (신규 오류에 추가)
5-2	Fail (Fail)	Fail (Fail)		메일 기능 비정상. (신규 오류에 추가)
5-3	Fail (Fail)	Fail (Fail)		메일 기능 비정상. (신규 오류에 추가)
5-4	Fail (Fail)	Fail (Fail)		메일 기능 비정상. (신규 오류에 추가)
6-1	Fail (Fail)	Fail (Fail)		3 달러가 출금됨.
6-2	Fail (Fail)	Fail (Pass)		수정 확인.

FileNotFoundException은 2주 연속으로 문제가 발생하였습니다.
따라서 Pass Percentage는 0%로 책정이 가능함(괄호 안은 위의 Exception을 제외한 결과)

0/22 (0%) Pass

4.2 Additional Bruteforce Testing

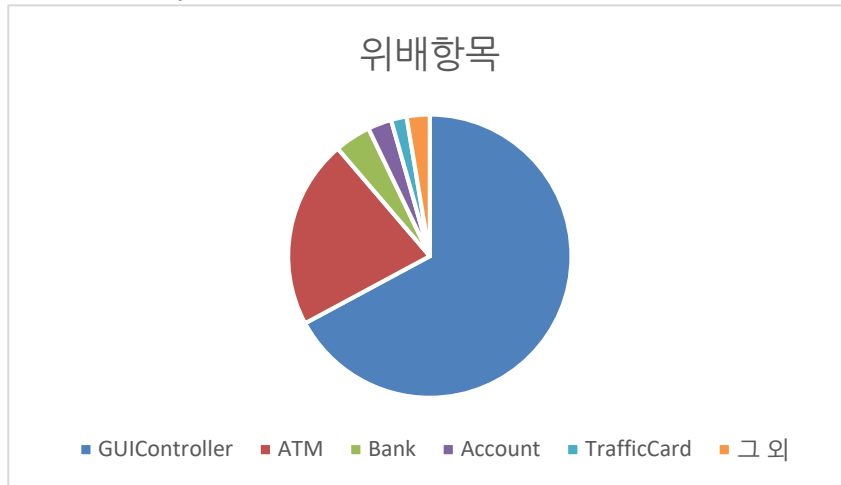
Test	Test Num	Description	P/F
명세표 출력	7-1	명세표 용지가 남아있지 않을 때 명세표를 출력한다.	Fail
입금	2-1	현재 ATM기기가 10000원을 최대로 보유하고 있을 때 10000원의 입금을 시도한다.	Fail
	2-2	현재 ATM기기가 10\$을 최대로 보유하고 있을 때 10%의 입금을 시도한다.	Fail
출금	3-1	현재 ATM기기가 1000원을 보유하고 있지 않을 때 23000원의 출금을 시도한다.	Fail

0/4 pass (0 %) Pass

5 Code Scroll

5.1 Code Analysis

5.1.1 GUIController.java에서 가장 많은 위배 항목 발견



5.1.2 ATM.java에서 규칙 위배 밀도가 가장 높은것으로 나타남Pairwise Test

소스	위배 수	무시된 위배 수	위배 규칙 수	*RCR (규칙 준수율)	**RVD (규칙 위배 밀도)
GUIController.java	715	0	19	80.81%	0.36
ATM.java	230	0	19	80.81%	0.47
Bank.java	44	0	12	87.88%	0.27
Account.java	29	0	7	92.93%	0.46
TrafficCard.java	19	0	11	88.89%	0.37
Card.java	15	0	7	92.93%	0.34
Book.java	13	0	6	93.94%	0.32

5.1.3 주요 위배 내역 / 규칙

규칙	위배 수	규칙 설명
Sun_24	581	숫자 상수 사용 금지
Sun_08	105	블록의 시작 부분에서만 선언 검사
Sun_10	72	지역 변수 선언 시 초기화 검사
JAVA_72	63	메서드의 설명 주석이 있어야 함
JAVA_64	42	반복문 안에서 String concatenation assignment 연산자의 사용 금지
Sun_06	26	소스 라인 길이 검사
Sun_17	20	메서드 내부의 지역 변수 선언부와 실행 문장 사이를 빈 줄로 구분

5.1.4 위배내역

1. 메소드 printStackTrace가 사용됨. <매우 높음>

Book.java	50	Book.Book(String, int)	Item
ATM.java	96	ATM.ATM()	ATM

ATM.java	98	ATM.ATM()	ATM
ATM.java	465	ATM.getBalance()	ATM
ATM.java	634	ATM.checkResource()	ATM
ATM.java	679	ATM.end()	ATM
ATM.java	681	ATM.end()	ATM
Book.java	50	Book.Book(String, int)	Item

```

90 :         getStr = br.readLine();
91 :         getMailID = getStr;
92 :
93 :         getStr = br.readLine();
94 :         rate = Integer.parseInt(getStr);
95 :     } catch (FileNotFoundException e) {
96 :         e.printStackTrace();

```

- [JAVA_21] catch 절에서 exception(java.io.FileNotFoundException)이 그대로 출력됨.
- [ERR01-J] catch 절에서 (java.io.FileNotFoundException)의 메서드를 사용함
- [JAVA_44] 메서드 (printStackTrace)가 사용됨.
- [ERR00-J] catch 절에 printStackTrace 메서드만 존재

Figure 1 ATM.java line 96

시스템 내부 데이터나 디버깅 관련 정보가 공개되면 공격의 빌미가 될 수 있다.
따라서 예외 발생 시 스택 정보를 출력하지 말아야 한다.

2. 클래스의 설명 주석이 없거나 모든 태그가 포함되어 있지 않음. <높음>

Book.java	8	Item	Item
Book.java	16	Book	Item
Book.java	54	Book	Item
Book.java	58	Book	Item

GUIController.java	2167	GUIController\$37.actionPerformed(ActionEvent)	GUI
GUIController.java	2191	GUIController\$38.actionPerformed(ActionEvent)	GUI
GUIController.java	2194	GUIController\$38.actionPerformed(ActionEvent)	GUI
GUIController.java	2338	GUIController\$40.actionPerformed(ActionEvent)	GUI
GUIController.java	2341	GUIController\$40.actionPerformed(ActionEvent)	GUI
GUIController.java	2400	GUIController\$41.actionPerformed(ActionEvent)	GUI
GUIController.java	2403	GUIController\$41.actionPerformed(ActionEvent)	GUI
GUIController.java	2426	GUIController\$42.actionPerformed(ActionEvent)	GUI

GUIController.java	2429	GUIController\$42.actionPerformed(ActionEvent)	GUI
GUIController.java	2451	GUIController\$43.actionPerformed(ActionEvent)	GUI
GUIController.java	2454	GUIController\$43.actionPerformed(ActionEvent)	GUI

```

801 :                                     //print pressed amount
802 :                                     for(int j = 0; j < pw_i; j++){
803 :                                         ptn_s += "X" + " ";

```

[JAVA_64] 반복문 안에서 String concatenation 연산자를 사용함.

Figure 2 GUIController.java line 803

3. 반복문 안에서 String concatenation 연산자를 사용함 <높음>

GUIController.java	5	GUI	GUI
--------------------	---	-----	-----

```
5 : import javax.swing.*;
```

[JAVA_68] "*"를 이용한 import문을 사용함.

```
6 : import java.awt.*;
```

[JAVA_68] "*"를 이용한 import문을 사용함.

Figure 3 GUIController.java line 5

4. "*"을 이용한 import문을 사용함. <높음>

GUIController.java	5	GUI	GUI
--------------------	---	-----	-----

```
5 : import javax.swing.*;
```

[JAVA_68] "*"를 이용한 import문을 사용함.

```
6 : import java.awt.*;
```

[JAVA_68] "*"를 이용한 import문을 사용함.

Figure 4 GUIController.java line 5

5. Identifier(ptn_s)가 이미 필드로 선언되어 있어 hide가 발생함. <높음>

GUIController.java	619	GUIController\$9.actionPerformed(ActionEvent)	GUI
--------------------	-----	---	-----

GUIController.java	741	GUIController\$10.actionPerformed(ActionEvent)	GUI
GUIController.java	768	GUIController\$11.actionPerformed(ActionEvent)	GUI
GUIController.java	1402	GUIController\$28.actionPerformed(ActionEvent)	GUI
GUIController.java	1560	GUIController\$29.actionPerformed(ActionEvent)	GUI
GUIController.java	1708	GUIController\$30.actionPerformed(ActionEvent)	GUI
GUIController.java	1735	GUIController\$31.actionPerformed(ActionEvent)	GUI
GUIController.java	1792	GUIController\$33.actionPerformed(ActionEvent)	GUI
GUIController.java	2019	GUIController\$35.actionPerformed(ActionEvent)	GUI
GUIController.java	2093	GUIController\$36.actionPerformed(ActionEvent)	GUI
GUIController.java	2154	GUIController\$37.actionPerformed(ActionEvent)	GUI
GUIController.java	2181	GUIController\$38.actionPerformed(ActionEvent)	GUI
GUIController.java	2301	GUIController\$40.actionPerformed(ActionEvent)	GUI
GUIController.java	2366	GUIController\$41.actionPerformed(ActionEvent)	GUI
GUIController.java	2416	GUIController\$42.actionPerformed(ActionEvent)	GUI
GUIController.java	2443	GUIController\$43.actionPerformed(ActionEvent)	GUI

645 :

```
String ptn_s = "";
```

[Sun_09] identifier(ptn_s)가 이미 필드로 선언되어 있어 hide가 발생함

[JAVA_70] identifier(ptn_s)가 이미 필드로 선언되어 있어 hide가 발생함

Figure 5 GUIController.java line 645

5.2 Summary

1. 문서들의 단계마다 일치하지 않는 부분들이 매우 많다.
 - A. 2050단계의 경우, 모든 method의 Input과 Output칸을 비워놓는 등 최소한의 성의를 보이지 않았다.
2. 실제 구현된 코드와 문서 간에 굉장한 차이점이 존재한다.
3. 용어가 통일되지 않았다.
4. GUI class에서 상당 수의 연산을 수행해서 본래 그 연산을 수행했어야 할 class에는 빈 함수가 존재한다.
5. 객체 지향적인 설계를 하지 않았다.
6. 외화로 출금, 이체를 할 경우 자릿수가 바뀌어 연산 되는 등의 치명적인 버그들이 다수 존재한다.
7. Functional Requirement를 만족시키지 못하였다.
 - A. 관리자와 같은 기능의 경우, 문서에는 존재하지만 코드상에서는 존재를 찾아볼 수 없다.
8. Non Functional Requirement를 모두 만족시키지 못하였다.
 - A. 직관적이지 않은 UI
 - i. 한글 깨짐 현상 발생 (파일과 코드 전체가 EUC-KR로 저장되어있다.)
 - ii. Placeholder 처리가 되어있지 않다. (default text로 값이 들어가 있다.)
 - iii. Maxlength가 지정되어 있지 않다.
 - iv. Number format이 적용되어 있지 않다. (구분자가 없다.)
 - v. Return Card가 닫지 않으면 계속 출력된다.
 - vi. account id를 입력할 시 기존에 있던 label이 삭제되고 그 곳에 account id가 입력된다.

6 Overall

3.1 System Test Result

3.1.1 Category-Partition Test

데이터가 담겨있는 Text File의 경로가 잘못되어 아무것도 읽어오지 못하는 심각한 오류가 있었지만 소모 팀분들이 해결하지 못해서 임의로 해결하고 테스트를 진행했습니다. 문서에 정의된 기능 대부분은 정상적으로 작동하지만 예외처리가 되어있지 않으며, 관리자에게 알람을 보내는 부분에서 많은 Fail이 발생하였습니다.

3.1.2 Pairwise Test

데이터가 담겨있는 Text File의 경로가 잘못되어 아무것도 읽어오지 못하는 심각한 오류가 있었지만 소모 팀분들이 해결하지 못해서 임의로 해결하고 테스트를 진행했습니다. 문서에 명시된 기능 대부분이 동작은 겨우 이루어지지만 예외처리가 되어있는 경우는 찾아보기 힘들었고, 관리자에게 알람을 보내는 부분은 전혀 작동하지 않았습니다. 일부 기능은 작동은 하지만 과연 이 동작을 버그가 아닌 기능이라고 봐야 하는지 의심스러울 만큼 괴상하게 동작하기도 했습니다.

3.1.3 Brute Force Test

데이터가 담겨있는 text file의 경로가 잘못되어 아무것도 읽어오지 못하는 심각한 오류가 있었지만 소모 팀분들이 해결하지 못해서 임의로 해결하고 테스트를 진행했습니다. 그럼에도 불구하고 예외처리가 잘 되어있지 않았고, 주요 알고리즘이 제대로 작동하지 않는 모습을 보였습니다.

3.1.4 Summary

1. 문서들의 단계마다 일치하지 않는 부분들이 매우 많다.
 - A. 2050단계의 경우, 모든 method의 Input과 Output칸을 비워놓는 등 최소한의 성의를 보이지 않았다.
2. 실제 구현된 코드와 문서 간에 굉장한 차이점이 존재한다.
3. 용어가 통일되지 않았다.
4. GUI class에서 상당 수의 연산을 수행해서 본래 그 연산을 수행했어야 할 class에는 빈 함수가 존재한다.
5. 객체 지향적인 설계를 하지 않았다.
6. 외화로 출금, 이체를 할 경우 자릿수가 바뀌어 연산 되는 등의 치명적인 버그들이 다수 존재한다.
7. Functional Requirement를 만족시키지 못하였다.
 - A. 관리자와 같은 기능의 경우, 문서에는 존재하지만 코드상에서는 존재를 찾아볼 수 없다.
8. Non Functional Requirement를 모두 만족시키지 못하였다.
 - A. 직관적이지 않은 UI
 - i. 한글 깨짐 현상 발생 (파일과 코드 전체가 EUC-KR로 저장되어있다.)
 - ii. Placeholder 처리가 되어있지 않다. (default text로 값이 들어가 있다.)
 - iii. Maxlength가 지정되어 있지 않다.
 - iv. Number format이 적용되어 있지 않다. (구분자가 없다.)
 - v. Return Card가 달지 않으면 계속 출력된다.
 - vi. account id를 입력할 시 기존에 있던 label이 삭제되고 그 곳에 account id가 입력된다.