

# System Test Report for Team

**1 1st**

**Project Team**

**Team 1**

**Latest update on:**

**2018-05-26**

---

**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.....	6
1.3	Stage 2040 Design.....	15
1.4	Stage 2050 Implementation.....	21
1.5	Stage 2060 Unit Test & System Test.....	22
2	Category-partition Testing Report.....	23
2.1	Categorize.....	23
2.2	Description.....	24
2.3	Property & Constraints.....	25
2.4	Text File for TSL-Generator.....	26
2.5	Test Result.....	27
3	Pairwise Testing Report.....	27
3.1	Categorize.....	27
3.2	Description.....	30
3.3	Text File 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	Overall.....	36
5.1	System Test Result.....	36
6	CTIP Usage.....	37

6.1	Static Analysis Result.....	37
6.2	Issue Tracker .....	38

# 1 Specification Review

## 1.1 Stage 1000 Planning

### 1001. Define Draft Plan

#### 3. Functional Requirements

- Check
- Deposit
- Withdraw
- Transfer
- Update
- Status Alarm
- Management
- Issue TrafficCard

⇒ Verify Sufficient Fund가 생략되어 있다.

#### 4. Non-Functional Requirements

- 외국인, 내국인이 쉽게 이해할 수 있는 직관적인 인터페이스여야 한다.

⇒ 직관적인 인터페이스이지 않음.

#### 5. 현재 환율정보를 고려하지 않음

- 현재 환율정보를 고려하여 환전을 돕는다.

⇒ 현재의 환율정보를 고려하지 않음.

#### 6. 현재 환율정보를 고려하지 않음.

- 현금 인출기의 오류, 현금부족, 현금포화 상태의 경우 관리자에게 문자 알림을 한다.

⇒ 관리자에게 문자 알림이 가지 않음.

### 1002. Create Preliminary Investigation Report

#### 2. Project Justificatoin

- 1) Cost : 노동력과 서버비용 이외에 지출이 없어 저렴하다

⇒ 서버를 사용하지 않음

#### 6. Other Managerial Issues

- 2) 추후 줄작으로 활용 가능

⇒ 줄작으로 활용가능해 보이지 않음.

### 1003. Define Requirements

#### 1. Functional Requirements

- Verify Sufficient Fund가 생략되어 있다.
- Management : Evident → Hidden (파일 상에서 처리하므로 Hidden)
- Deposit은 계좌에 증감되지 않고, 증가만 한다.

## 2. Performance Requirement

- 최대한, 적절한, 충분한 같은 불명확한 표현 수정 필요

## 1004. Record Terms in Glossary

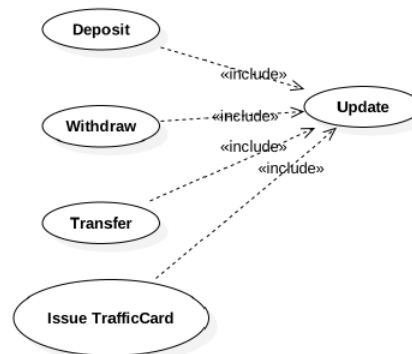
<b>TrafficCard</b>	ATM에서 지급하는 TrafficCard(이용할 때마다 계좌에서 금액이 빠져 나간다.)
--------------------	--

- TrafficCard를 이용하는 기능은 설계되지 않음

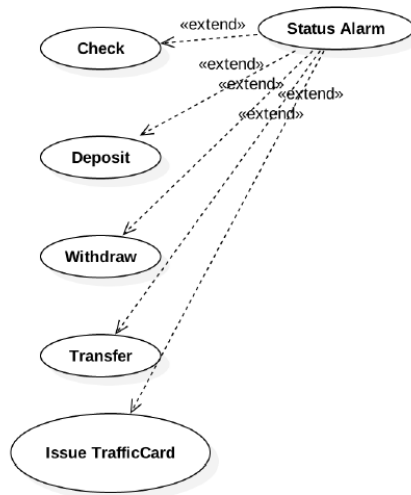
## 1007. Define Business Use Cases

### 6. Identify the relationships between Use-Case

- Event-based Use-case인 Update와의 관계가 잘못되어 있다. (방향이 반대가 되어야 한다.)



- Check, Deposit, Withdraw, Transfer, Issue TrafficCard 와 Status Alarm과의 관계가 잘못되어 있다. (extends가 아닌 include가 되어야 한다.)



## 1008. Define Business Concept Models

- 2030 단계 이후로는 등장하지 않는 심볼이 정의되어 있다.(History, Sensor)



- 2030단계 이후로는 등장하지 않는 Attribute가 정의되어 있다.

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

## 1010. Refine Plan

### 3. Functional Requirements

- Verify Sufficient Fund가 빠져있다.

### 4. Performance Requirements

- 외국인과 내국인 모두 손쉽게 접근할 수 있는 직관적인 인터페이스가 아니다.



## 1.2 Stage 2030 Analysis

### 2031. Describe Use Case

#### 1.1 Check

Typical Courses of Events	(A) : Actor / (S) : System 1. (S) 본인인증 절차를 진행한다. 2. (S) 인증에 성공한 경우, 은행으로부터 해당 계좌의 잔액을 가져온다. 3. (S) Item 을 반환한다. 4. (S) 사용자에게 잔액을 보여준다. 5. (A) 잔액을 확인한다. 6. (S) 명세표를 출력할지 물어본다. 7. (A) 명세표를 출력할지 선택한다. 8. (S) 명세표 출력 용지가 충분한지 선택한다.
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.
Exceptional Courses of Events	E1. 인증에 실패한 경우, 종료한다 E8. 명세표 출력 용지가 부족할 경우 알림을 발생시킨다.

- 8. (S) 명세표 출력 용지가 충분한지 선택한다 → 연산한다 로 수정 필요
- E1. 다른거래 선택 → 다른 거래를 동시에 선택할 수 없다.

#### 1.2 Deposit

4.(S) 유효한 현금인지 검사한다.

- 4.(S) '유효한' 현금이 무엇인지, 어떻게 검사하는 지 명시되어 있지 않으며 검사하지 않음

#### 1.3 Withdraw

E11. 현금 인출이 일정 시간동안 이루어지지 않는 경우 투입구를 폐쇄하고 다시 입금한다.

- '일정시간' 이 어느 정도인지 명시되어 있지 않으며 투입구를 폐쇄하지 않음

User 가 Item 을 통해서 자신의 계좌에 출금한다

- User가 Item을 통해 자신의 계좌에서 출금하지 않는다.

#### 1.4 Issue TrafficCard

Exceptional Courses of Events	E1. 본인확인 실패 시 종료한다.
	E5. 안내사항 숙지 확인을 거절할 경우 종료한다.
	E6. 계좌 잔액 부족 시, 종료한다
	E11. 교통 카드가 부족할 경우 알림을 발생시킨다.
	E12. 명세표 출력 용지가 부족할 경우 알림을 발생시킨다.

E11. 교통카드가 부족한지 검사하지 않으며 알림이 없음

E12. 명세표 출력 용지가 부족한 경우를 검사하지 않으며 알림이 없음

### 1.6 Update

Pre-Requisites	User 의 입금 / 출금 / 송금 / 교통 카드 발급 거래 상태 Verify Sufficient fund 처리 완료
----------------	--

- Verify Sufficient fund 처리 완료 상태가 Pre-Requisites 이 아니다..

1.(S) Update 에 필요한 정보를 받는다.

- 필요한 정보가 어떤 것인지 명시되어 있지 않다.

### 1.8 Status Alarm

---

Purpose	ATM 보유 현금량이 충분하지 않을 경우 관리자에게 알림을 보내준다.
---------	--

- ATM보유 현금량이 최대값에 도달했을 경우에도 관리자에게 알림을 보내지 않는다.

- 명세표의 출력용지가 충분하지 않을 경우에도 관리자에게 알림을 보내지 않는다.

Cross Reference	Functions : R.1.0 R.1.1, R.1.2, R.1.3 R.1.4 Use-cases : "Deposit", "Withdraw","Transger" , "Issue TrafficCard"
-----------------	---

- Transger → Transfer로 수정 필요.

- Use-Case "Check" 도 포함되어야 한다

### 1.9 Management

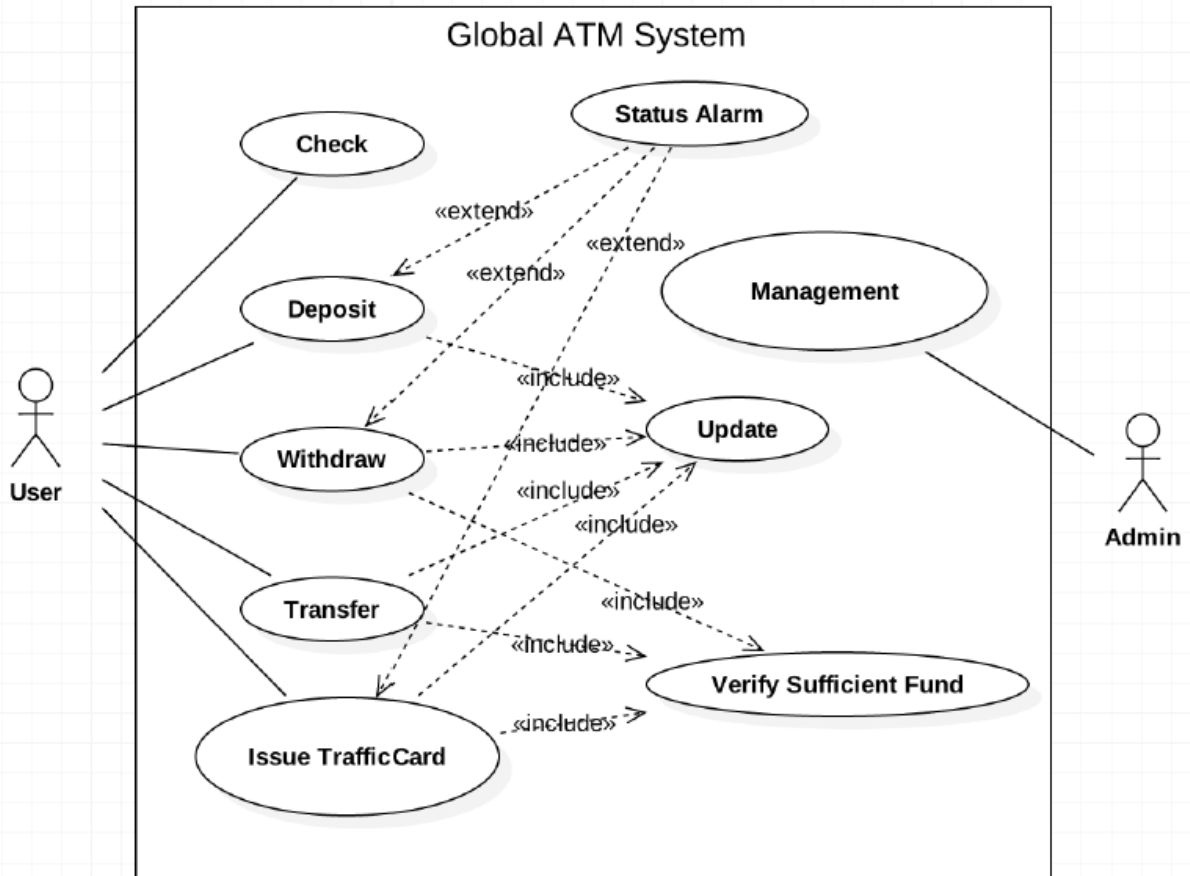
Typical Courses of Events	(A): Actor (S):System 1.(A) 관리자임을 인증한다. 2.(S) ATM 의 현금/교통카드/명세표출력용지 저장소를 개방한다.
---------------------------	--

- 3.(A)로 ATM을 관리하는 관리자의 행동이 추가되어야 한다.



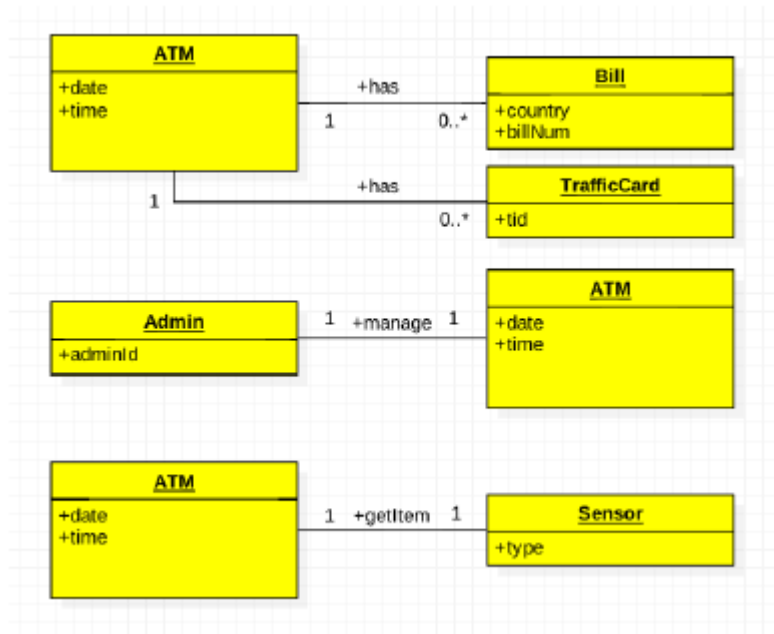
2032. Refine Use Case Diagrams

- 수정한 1000단계의 내용과 같게 수정해야 한다.



### 2033. Define Domain Model

- 사용되지 않는 Sensor, Model을 사용한다.



2034. Refine Clossary

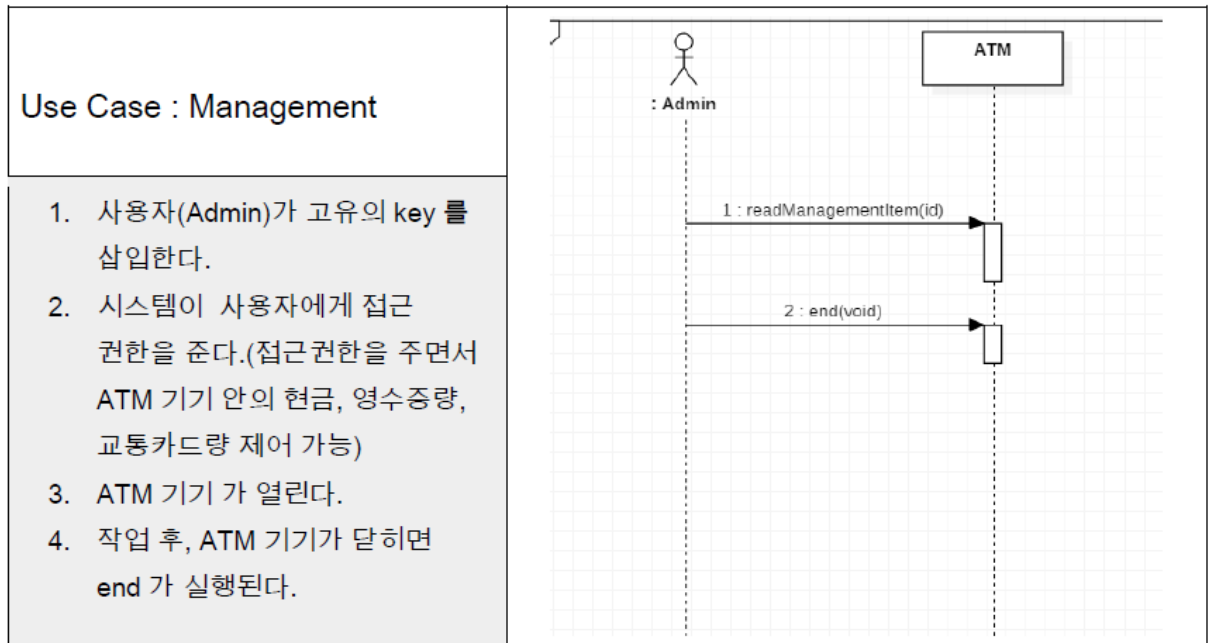
- 사용하지 않는 변수 3개 존재 (Sensor, History, chargeFee)

Sensor	Concept	ATM 센서
History	Concept	거래 내역
chargeFee	Attribute	수수료

2035. Define System Sequence Diagrams

모든 Sequence Diagram의 생명선이 잘못 되어 있다.

- 앞서 문서에서 한번도 정의되지 않은 end기능이 추가되었다.



2036. Define Operation Contracts

6.2 selectService(service)

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

- Post-conditions에 '원하는 서비스의 화면으로 접근한다.' 추가 필요
- 모든 Service가 비밀번호를 필요로 하지 않으므로 수정 필요 ex) Deposit

### 6.3 printReceipt(print)

Name	printReceipt(print)
Responsibilities	명세표를 보여주고 출력 여부를 묻는다.
Type	System
Cross References	Functions : R.1.0, R.1.1, R.1.2, R.1.3, R.1.4 Use cases : Check, Deposit, Withdraw, Transfer, Issue TrafficCard
Notes	
Exceptions	
Output	명세표를 출력한다.
Pre-conditions	서비스(입금/계좌조회/송금/출금)가 성공적으로 완료되어야 한다.
Post-conditions	초기 화면으로 돌아간다.

- Functions 'R.2.1, R.2.2' 추가 필요
- Use cases에 'Verify Sufficient Fund, Status Alarm' 추가 필요
- Exceptions에 '명세표 용지가 ATM에 존재하지 않는 경우' 추가 필요
- Output : 'User가 원할 경우' 추가 필요

### 6.4 confirm(password)

Name	confirm(password)
Responsibilities	계좌 비밀번호를 확인한다.
Type	System
Cross References	Functions : R.1.0, R.1.2, R.1.3, R.1.4 Use cases : Check, Withdraw, Transfer, Issue TrafficCard
Notes	
Exceptions	-비밀번호가 틀린 경우
Output	UI output 은 없음, 비밀번호 확인
Pre-conditions	원하는 서비스 선택이 되어있어야 한다.
Post-conditions	서비스(계좌조회/송금/출금/교통카드발급)를 진행한다.

- Exceptions에 '사용자가 입력한 비밀번호가 계좌의 비밀번호와 다를 경우' 추가 필요

### 6.5 insertCash(type,amounts)

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.

- Responsibilities에 지폐를 투입구에 넣고 투입한 지폐의 종류와 양을 확인하지 않음.
- Cross Reference에 'Functions R.2.1, R.2.2' 추가 필요
- Use cases에 Verify Sufficient Fund , Status Alarm추가 필요
- Exceptions에 'ATM의 지폐 보유량이 최대값에 도달한 경우 알림을 보낸다.' 추가 필요
- Exceptions에 '원화와 달러를 섞어서 입금할 경우' 추가 필요

### 6.7 enterAmounts(amounts)

Responsibilities	출금 금액량을 확인한다.
Type	System
Cross References	Functions : R.1.2, R.1.3 Use cases : Withdraw, Transfer
Notes	
Exceptions	
Output	UI output 은 없음, 출금 금액량 확인
Pre-conditions	출금 지폐종류가 확인 되어있어야 한다.
Post-conditions	출금 후 명세표 확인 및 출력을 한다.

- Cross Reference에 'Functions R.2.1, R.2.2' 추가 필요
- Use cases에 'Verify Sufficient Fund , Status Alarm' 추가 필요
- Exceptions에 'ATM의 지폐 보유량이 부족할 경우' 추가 필요



### 6.8 destAccount(bank, account)

Name	destAccount(bank, account)
Responsibilities	송금 대상의 은행과 계좌를 확인한다.
Type	System
Cross References	Functions : R.1.3 Use cases : Transfer
Notes	
Exceptions	-송금 대상이 유효하지 않은 경우
Output	UI output 은 없음, 송금 대상 확인
Pre-conditions	비밀번호 확인이 완료되어 있어야 한다.
Post-conditions	출금 금액량을 확인한다.

- Post-conditions에 '출금 할 지폐의 종류를 확인한다.' 추가 필요

### 6.9 setDateRange(date\_range)

Type	System
Cross References	Functions : R.1.4 Use cases : Issue TrafficCard
Notes	
Exceptions	
Output	UI output 은 없음, 교통카드 사용기간 확인
Pre-conditions	비밀번호 확인이 완료되어 있어야 한다.
Post-conditions	교통카드 계좌연동 및 교통카드 비용 결제를 확인한다.

- Post-conditions에 '교통카드 계좌연동 및 비용 결제를 안내한다.' 추가 필요

### 6.10 agreement (approval)

Name	agreement(approval)
Responsibilities	교통카드 계좌연동 및 교통카드 비용 결제를 확인한다.
Type	System
Cross References	Functions : R.1.4 Use cases : Issue TrafficCard
Notes	
Exceptions	
Output	UI output 은 없음, 교통카드 계좌연동 및 교통카드 비용 결제 확인
Pre-conditions	교통카드 사용기간이 확인되어 있어야 한다.
Post-conditions	명세표 확인 및 출력을 한다.

- Responsibilities에 '교통카드 계좌연동 및 비용 결제를 안내하고 동의를 받는다.' 수정 필요
- Cross Reference에 'Functions : R.2.1, R.2.2' 추가 필요
- Use cases에 'Verify Sufficient Fund , Status Alarm' 추가 필요
- Exceptions에 'ATM의 교통카드 보유량이 부족할 경우' 추가 필요

### 6.11 printReceipt(print)

- 6.3의 printReceipt와 중복되므로 삭제 필요

### 6.12 readManagementItem(id)

- 관리자를 사용하지 않으므로 삭제 or 수정 필요

### 6.13 end(void)

- 앞에서 end에 대한 기능이 정의되어 있지 않으므로 수정 필요.

### 2039. Traceability Analysis

- 앞단계의 변경 내용을 반영하여 전체적으로 수정 필요.

### 1.3 Stage 2040 Design

#### 2141. Design Real Use Cases

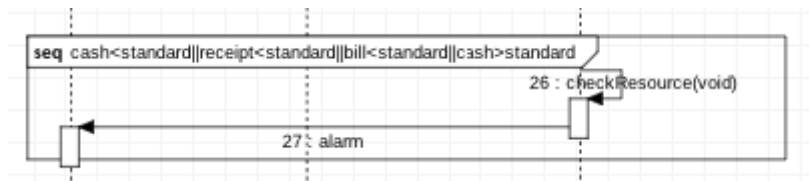
- 2030단계에서 변경된 내용을 반영해 수정 요망
- 구현단계에서 Admin과 관련된 부분이 아예 사라졌기 때문에 R.2.2 Status Alarm과 R.3.0 Management가 아예 사용되지 않는다.

#### 2142 Define Reports, UI, and Storyboards

- UI프로토 타입과 화면 개수, 구성, 디자인, 흐름이 완전히 다르다.

#### 2144. Define Interaction Diagrams

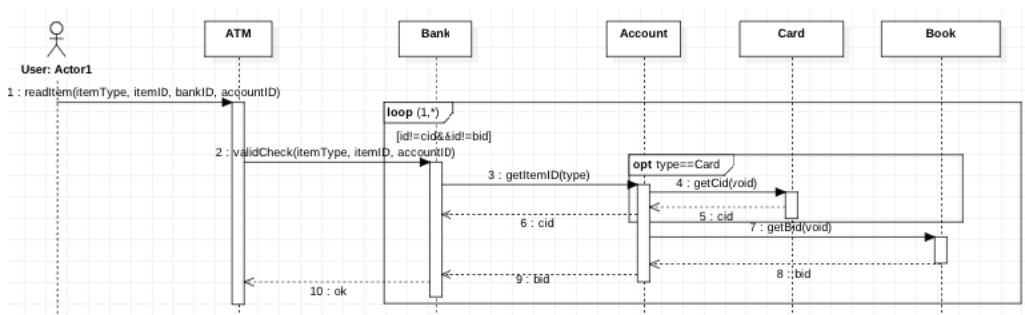
- 실제 프로그램에서 존재하지 않는 Admin이 그려져 있다.
- 따라서 Admin과 관련된 부분은 전부 빠져야 한다.



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

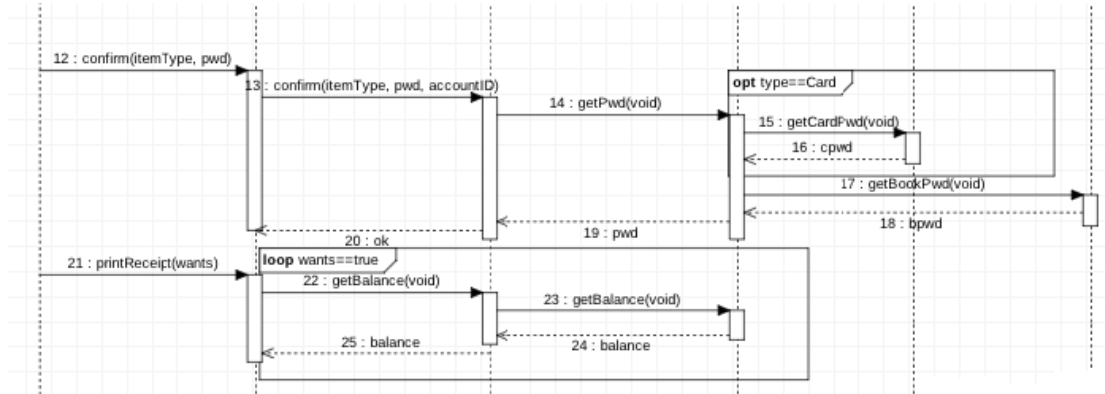
이와 관련된 부분은 아래에서 생략한다.

#### 4.1 Check



- Sequence Diagram과 흐름이 다르다.
- readItem의 생명선이 끝까지 이어져야 한다

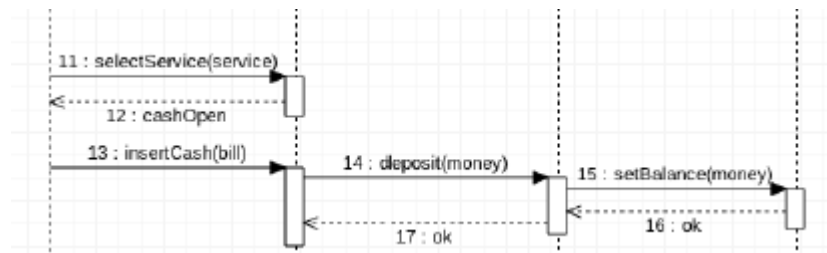
- loop문의 조건이 잘 못 되어 있다. 명세에 의하면 모든 item이 호출될 때까지 반복한다.
- 7,8 번은 type==Book 인 경우로 분기되어야 한다.



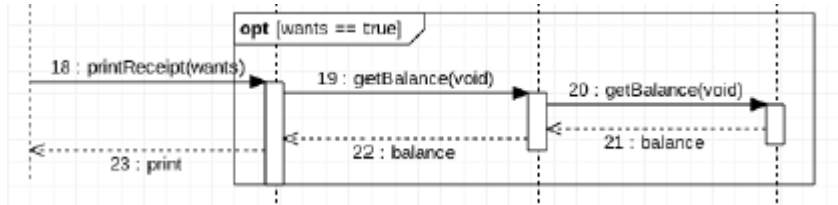
- 실제 프로그램과 동작 구조가 다르다.  
객체들의 정보가 수정되고 반영되는 부분이 존재하지 않는다.
- Confirm의 생명선이 printReceipt가 끝날 때 까지 이어져야 한다.

< 이와 동일한 부분은 아래에서 생략한다. >

#### 4.2 Deposit

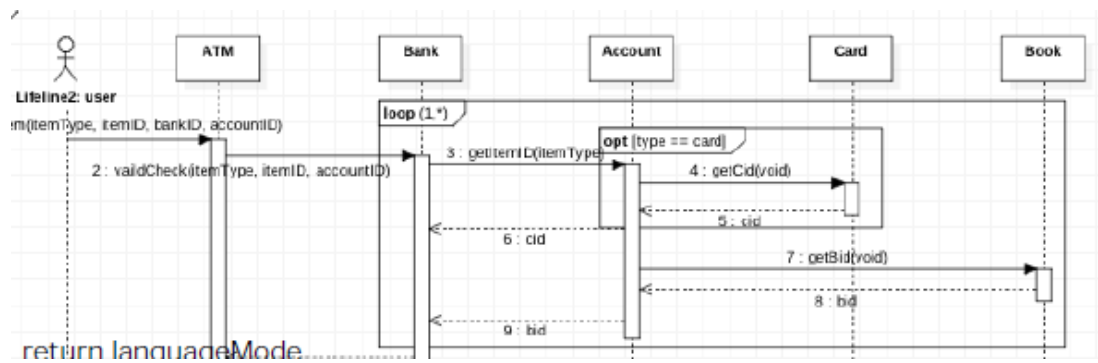


- 요구사항 명세에서는 한화와 외화를 나누는 분기가 있지만 다이어그램 상에서는 존재하지 않는다.
- deposit(money)의 반환값인 money가 명시되어 있지 않다.
- 유효한 현금인지 판별하는 분기가 나타나 있지 않다.

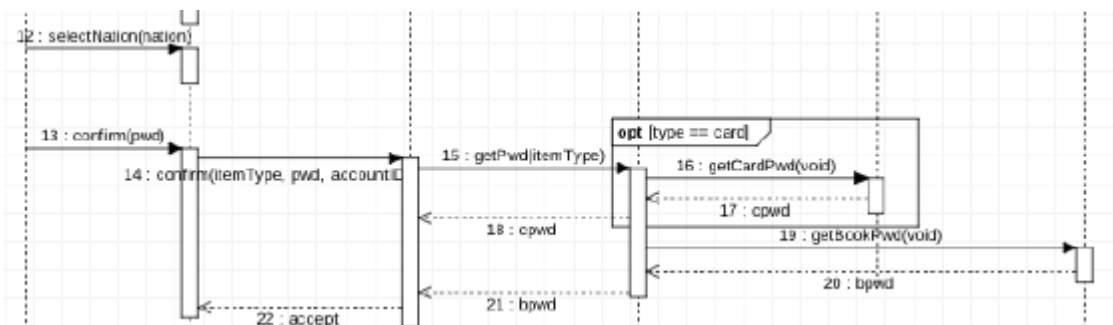


- printReceipt의 pre-condition인 confirm이 나타나 있지 않다.
- getBalance의 분기가 나타나 있지 않다.

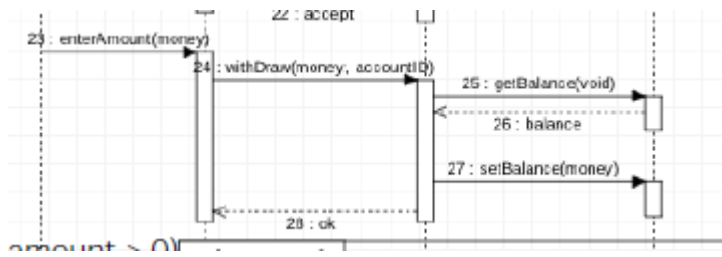
### 4.3 withDraw



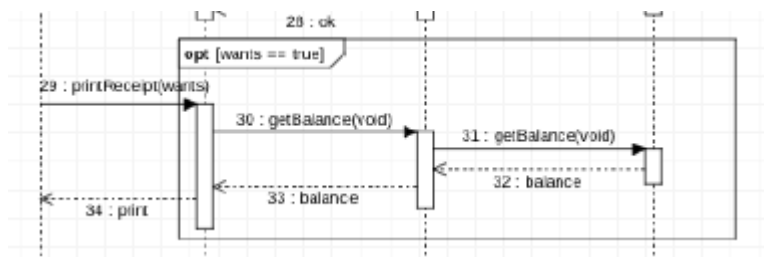
- loop문의 조건이 누락되어 있다.
- readItem의 반환값인 languageMode가 누락되었다.
- selectService는 result의 값에 따라 바뀌는데 누락되었다.



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

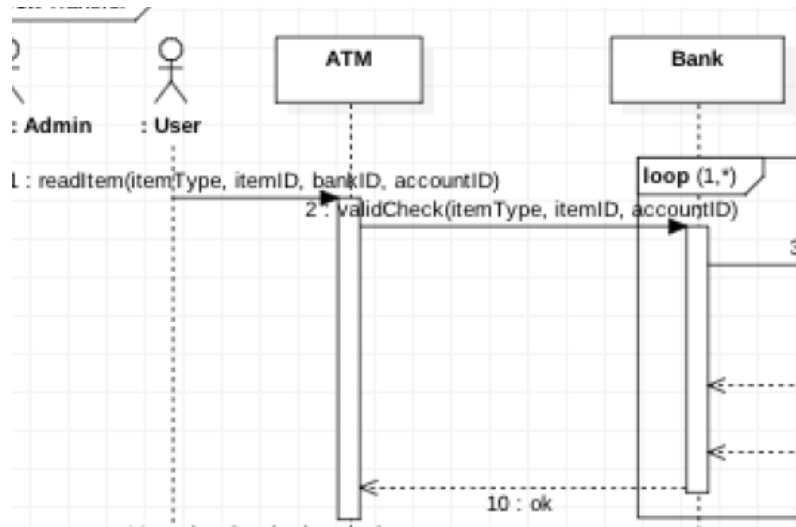


- 문서에 명시 되어있는 잔고의 부족을 검사하는 부분이 누락되어 있다.



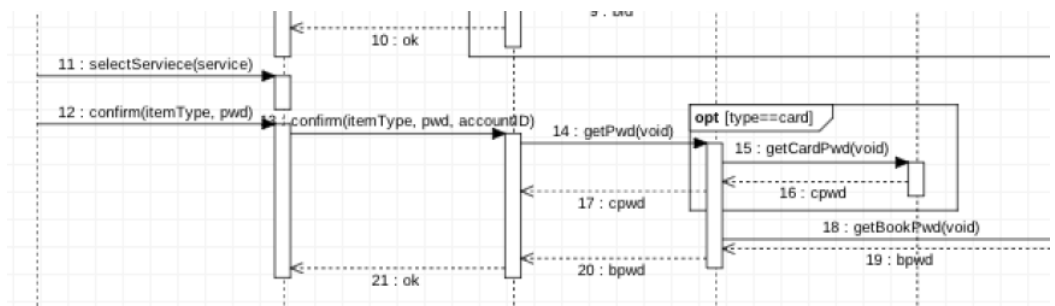
- 조건문이 잘못되어있다. ex) wants == true → amount > 0
- getBalance의 인자값이 다르다.
- ATM의 receiptAmount가 변화하는 부분이 명시되어 있지 않다.
-

#### 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

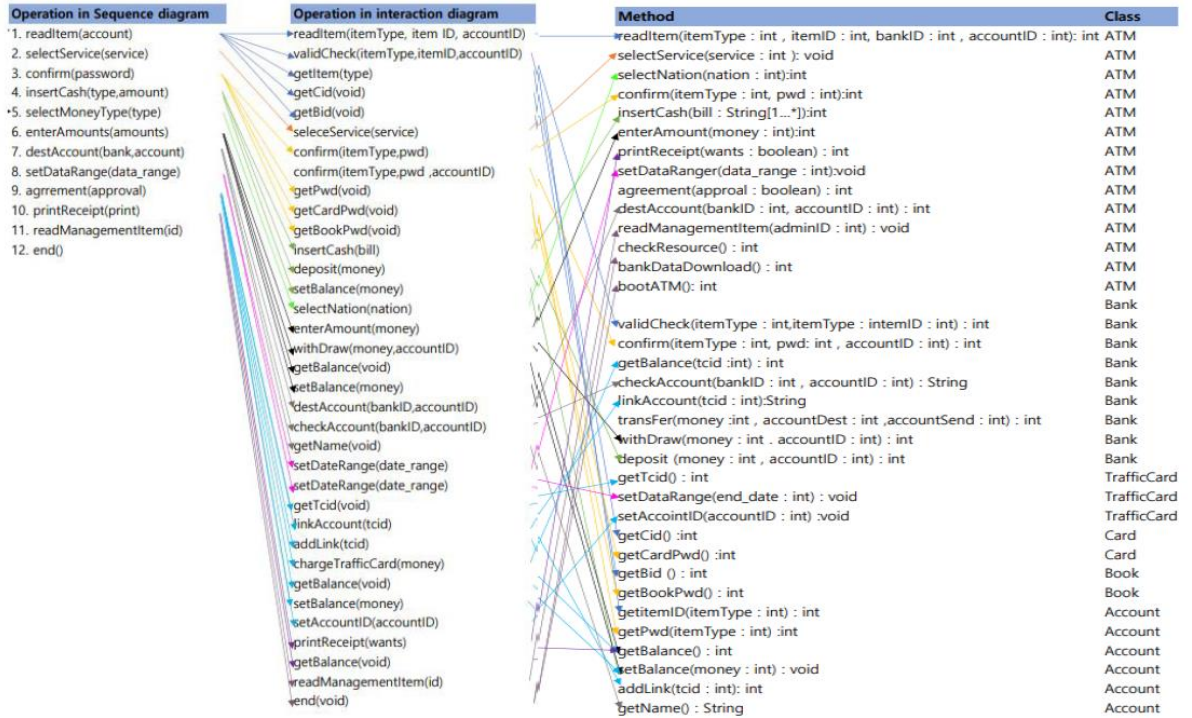


- 2044에 사용한 함수들이 명시되어 있지 않다.
- cashAmount의 데이터형이 다르다.
- ATMAdminID, ATMnation 변수가 누락되어 있다.
- confirmedItem, PWD는 프로그램상에서 존재하지 않는 변수이다.
- 함수의 이름이 다르다.  
setDateRange → setDataRange
- 함수의 반환값이 다르다.  
destAccount의 반환값은 string이다.
- checkResource의 접근제어지시자는 public이다.
- bankDataDownload와 bootATM은 존재하지 않는다.



< 다른 항목들도 모두 같은 문제가 발견되었으므로 생략한다. >

## 2046. Design Traceability Analysis



- End와 같은 Interatction diagram에서 사용되지 않은 Operation이 Method와 상관관계를 가지고 있다.
- 또한 bootATM()과 같은 존재하지 않는 Method가 정의되어 있다.
- Operation과 Method간의 상관관계가 적절하지 않다.
- 함수의 반환값이 다르다.  
Ex) linkAccount의 경우 반환값이 boolean이다.

## 1.4 Stage 2050 Implementation

### 2051. Implement Class & Methods

- 모든 Class와 Method에서 언급된 Use-case Status Alarm과 Management 는 프로그램상에서 구현되지 않았으므로 모두 빠져야 한다.
- 모든 Method의 Input과 Ouput을 작성하지 않아서 프로그램과의 비교가 불가능했다.
- 이전 단계에서의 설계와는 다르게 대부분의 기능과 로직을 GUIController라는 Class에서 구현하였는데, 이에 대한 설명이 하나도 남아있지 않다.
- 함수의 이름이 프로그램과 다르다.  
getBookPWD → getBpwd 수정 필요

getCardPwd → getCpwd 수정 필요  
getBalance → get\_balance 수정 필요  
getItemID → get\_aid 수정 필요  
setBalance → set\_balance  
setDateRange → inputDataRange

- 함수의 인자값이 프로그램과 다르다.  
confirm(itemType, pwd, accountID) → confirm(int \_pwd) 수정 필요

#### 2052. Implement Windows

- 이전 단계와는 전혀 다른 UI 구성을 보여준다.
- 이전 단계의 revision이 필요해 보인다.

### 1.5 Stage 2060 Unit Test & System Test

#### 2063. System Testing

- 소프트웨어 모델링 팀 측에서 테스트를 전혀 진행하지 않았다.
- 관리자의 경우 존재하지 않기 때문에 테스트 할 수 없다.

## 2 Category-partition Testing Report

### 2.1 Categorize

Group	Category	Values	#
<b>Read Item</b>	Read Item	Valid Input	100
		Invalid Item Type	101
		Invalid Item Id	102
		Invalid Item Bank	103
		Invalid Item Account ID	104
<b>Mode</b>	Mode	Check	200
		Deposit	201
		Withdraw	202
		Transfer	203
		Issue Traffic Card	204
<b>Password</b>	Password	Valid Password	300
		Invalid Password	301
<b>Data</b>	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	
<b>Agreement</b>	Agreement	Agree	500
		Disagree	501
<b>Change Balance</b>	Actor	Increase Balance	600
		Decrease Balance	601
		Send Money	602
	Admin	Increase Balance	610
		Decrease Balance	611
<b>Print Receipt</b>	Receipt	Print Receipt	700
		Not Print Receipt	701
<b>Management</b>	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
<b>Alarm</b>	Alarm	Send Alarm	900
		Not Send Alarm	901
<b>Cancel</b>	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 입력한다.
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
<b>Read Item</b>	Valid Input	
	Invalid Item Type	[Error]
	Invalid Item Id	[Error]
	Invalid Item Bank	[Error]
	Invalid Item Account ID	[Error]
<b>Mode</b>	Check [Property Check]	
	Deposit [Property Deposit]	
	Withdraw [Property Withdraw]	
	Transfer [Property Transfer]	
	Issue Traffic Card [Property IssueTC]	
<b>Password</b>	Valid Password	[if Check    Withdraw    Transfer    IssueTC]
	Invalid Password	[error] [if Check    Withdraw    Transfer    IssueTC]
<b>Select Data</b>	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]
<b>Select Bank</b>	Bank	[if Transfer]
<b>Select Account ID</b>	Valid Account ID	[if Transfer]
	Invalid Account ID	[if Transfer] [Error]
<b>Input Data</b>	Valid Money	[if Withdraw    Transfer]
	Invalid Money	[if Withdraw    Transfer] [Error]
	Valid Duration	[if IssueTC]
	Invalid Duration	[if IssueTC] [Error]
<b>Agreement</b>	Agree	[if Transfer]
	Disagree	[if Transfer]
<b>Actor</b>	Increase Balance	[if Deposit]
	Decrease Balance	[if Withdraw    IssueTC]
	Send	[if Transfer]
<b>Admin</b>	Increase Balance [property IncreaseBalance]	[if Deposit    IssueTC]
	Decrease Balance [property DecreaseBalance]	[if Transfer    Withdraw]
<b>Receipt</b>	Print Receipt [property PrintReceipt]	
	Not Print Receipt	
<b>Manage TC</b>	Not Enough TC [property TCAlarm]	[if IssueTC]
	Enough TC	[if IssueTC]
<b>Manage Receipt</b>	Not Enough Receipt [property ReceiptAlarm]	[if PrintReceipt]
	Enough Receipt	[if PrintReceipt]
<b>Manage Money</b>	Not Enough Money [property NMoneyAlarm]	[if DecreaseBalance]
	Enough Money	[if IncreaseBalance    DecreaseBalance]
	Too Many Money [property TMoneyAlarm]	[if IncreaseBalance]
<b>Alarm</b>	Send Alarm	[if TCAlarm    ReceiptAlarm    NMoneyAlarm    TMoneyAlarm]
	Not Send Alarm	[Error]
<b>Cancel Mode</b>	Cancel	[Single]
	Clear	[Single]
	Backspace	[Single]

## 2.4 TextFile for TSL-Generator

```

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

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

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

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

Select Bank:
    Bank,                     [if Transfer]

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

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

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

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

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

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

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

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

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

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

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

코드를 분석해 본 결과 Text file의 경로가 잘못 지정되어 있는 심각한 오류가 원인으로 밝혀졌습니다.



## 2. Error 수정 후(SV 팀에서 수정 진행) => 42%

TestCase	Result	Description
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	기간이 1일부터 시작인데 0일 입력한 경우 결재 진행
Test10- 901	Failed	Alarm을 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- 100.201.400.600.610.700.811.822.900	Failed	영수증이 부족하고 돈이 너무 많은 경우 알람을 보내지 않음
Test19- 100.201.400.600.610.700.811.821	Passed	
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	Passed	
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	Passed	
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	Passed	
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	Passed	
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	Passed	
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	Passed	
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	Passed	
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	Passed	
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	Passed	
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	Passed	
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	Passed	
Test64- 100.204.300.432.601.610.701.801.822.900	Failed	돈이 너무 많은 경우 알람을 보내지 않음
전체 Test Case: 64개	Pass: 27개	Pass Percentage: 42%

27/64 (42%) 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
TCAlarm: 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 [Password] = "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 TCAlarm
#
if [Mode] = "IssueTC" then [TCAlarm] = "Send"
else [TCAlarm] = "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	50000 원을 눌렀을 때 화면에 5000 원이 표시됨.
5	100.201.300.802.1100.1200.1300.1401.1500	Failed	50000 원을 눌렀을 때 화면에 5000 원이 표시됨.
6	100.201.300.803.1100.1200.1300.1400.1500	Failed	50000 원을 눌렀을 때 화면에 5000 원이 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	50000 원을 눌렀을 때 화면에 5000 원이 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	Passed	
14	100.202.301	Passed	
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 01 부터 시작해야 하나 00 인 경우도 발급됨.
25	100.204.301	Passed	

8 / 25 (32%) Pass

## 4 Brute Force Testing Report

### 4.1 Testing Result

Test	Test Num	Description	P/F
아이템 받아오기	1-1	존재하지 않는 계좌에 접근하려고 시도하면 중지되는지 확인한다.	Fail
	1-2	존재하지 않는 카드에 접근하려고 시도하면 중지되는지 확인한다.	Fail
	1-3	카드번호에 문자를 넣으면 중지되는지 확인한다.	Fail
	1-4	계좌번호에 문자를 넣으면 중지되는지 확인한다.	Fail
	1-5	아이템 종류에 문자를 넣으면 중지되는지 확인한다.	Fail
입금	2-1	현재 ATM기기가 현금을 가득 보유하고 있을 때 1000원의 입금을 시도하면 중지되는지 확인한다.	Fail
	2-2	현재 ATM기기가 현금을 가득 보유하고 있을 때 100\$의 입금을 시도하면 중지되는지 확인한다.	Fail
출금	3-1	현재 ATM기기가 보유하고 있는 50000원권의 수량이 0일 때 50000원을 출금하려고 시도하면 10000원짜리 다섯장으로 출금되는지 확인한다.	Fail
	3-2	현재 ATM기기가 현금을 보유하고 있지 않을 때 출금을 시도하면 중지되는지 확인한다.	Fail
	3-3	50달러를 정상적으로 출금하는 지 확인한다.	Fail
	3-4	5000만원을 정상적으로 출금하는 지 확인한다.	Fail
계좌이체	4-1	현재 ATM 기기가 현금을 보유하고 있지 않을 때 계좌이체가 가능한 지 확인한다.	Fail
	4-2	50달러를 정상적으로 이체하는 지 확인한다.	Fail
	4-3	5000만원을 정상적으로 출금하는 지 확인한다.	Fail
	4-4	등록되지 않은 계좌번호를 입력하고 이체를 시도했을 때 중지되는지 확인한다.	Fail
	4-5	자신의 계좌번호로 이체를 시도했을 때 중지되는지 확인한다.	Fail
알림	5-1	명세표 용지가 남아 있지 않을 때 명세표를 출력하면 알림이 가는지 확인한다.	Fail
	5-2	ATM기기의 현재 1000원 보유량이 최대치일 때 1000원의 입금을 시도하면 알림이 뜨는 지 확인한다.	Fail
	5-3	ATM기기의 현재 10\$ 보유량이 최대치일 때 10\$의 입금을 시도하면 알림이 뜨는 지 확인한다.	Fail
	5-4	ATM기기가 1000원을 보유하고 있지 않을 때 23000원의 출금을 시도하면 알림이 뜨는 지 확인한다.	Fail
교통카드	6-1	교통카드 생성 시 0.3달러가 출금되는 지 확인한다.	Fail
	6-2	현재 ATM기기가 교통카드 여분을 보유하고 있지 않을 때 교통카드 생성 시 중지되는지 확인한다.	Fail

0/22 (0%) Pass

코드를 분석해 본 결과 Text file의 경로가 잘못 지정되어 있는 심각한 오류가 원인으로 밝혀졌습니다.

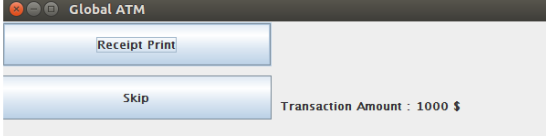
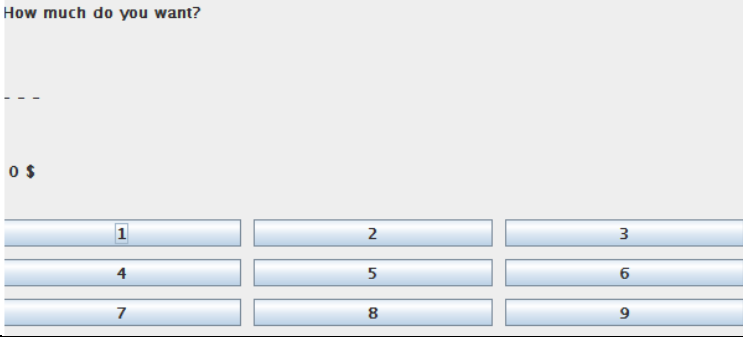
테스팅을 진행해보기 위해 이를 수정한 후 재 테스트 해보았습니다.

Test	Test Num	Description	P/F
아이템 받아오기	1-1	존재하지 않는 계좌에 접근하려고 시도하면 중지되는지 확인한다.	Pass
	1-2	존재하지 않는 카드에 접근하려고 시도하면 중지되는지 확인한다.	Pass
	1-3	카드번호에 문자를 넣으면 중지되는지 확인한다.	Fail
	1-4	계좌번호에 문자를 넣으면 중지되는지 확인한다.	Fail
	1-5	아이템 종류에 문자를 넣으면 중지되는지 확인한다.	Fail
입금	2-1	현재 ATM기기가 현금을 가득 보유하고 있을 때 1000원의 입금을 시도하면 중지되는지 확인한다.	Fail
	2-2	현재 ATM기기가 현금을 가득 보유하고 있을 때 100\$의 입금을 시도하면 중지되는지 확인한다.	Pass
출금	3-1	현재 ATM기기가 보유하고 있는 50000원권의 수량이 0일 때 50000원을 출금하려고 시도하면 10000원짜리 다섯장으로 출금되는지 확인한다.	Fail
	3-2	현재 ATM기기가 현금을 보유하고 있지 않을 때 출금을 시도하면 중지되는지 확인한다.	Pass
	3-3	50달러를 정상적으로 출금하는 지 확인한다.	Fail
	3-4	5000만원을 정상적으로 출금하는 지 확인한다.	Fail
계좌이체	4-1	현재 ATM 기기가 현금을 보유하고 있지 않을 때 계좌이체가 가능한 지 확인한다.	Pass
	4-2	50달러를 정상적으로 이체하는 지 확인한다.	Fail
	4-3	5000만원을 정상적으로 출금하는 지 확인한다.	Fail
	4-4	등록되지 않은 계좌번호를 입력하고 이체를 시도했을 때 중지되는지 확인한다.	Pass
	4-5	자신의 계좌번호로 이체를 시도했을 때 중지되는지 확인한다.	Pass
알림	5-1	명세표 용지가 남아 있지 않을 때 명세표를 출력하면 알림이 가는지 확인한다.	Fail
	5-2	ATM기기의 현재 1000원 보유량이 최대치일 때 1000원의 입금을 시도하면 알림이 뜨는 지 확인한다.	Fail
	5-3	ATM기기의 현재 10\$ 보유량이 최대치일 때 10\$의 입금을 시도하면 알림이 뜨는 지 확인한다.	Fail
	5-4	ATM기기가 1000원을 보유하고 있지 않을 때 23000원의 출금을 시도하면 알림이 뜨는 지 확인한다.	Fail
교통카드	6-1	교통카드 생성 시 0.3달러가 출금되는 지 확인한다.	Fail
	6-2	현재 ATM기기가 교통카드 여분을 보유하고 있지 않을 때 교통카드 생성 시 중지되는지 확인한다.	Fail

7/22 pass (31.8 %) Pass

## 4.2 Failed Case Report

Test	Test Num	Description	P/F
아이템 받아오기	1-3	카드번호에 문자를 넣으면 중지되는지 확인한다. → java.lang.NumberFormatException 와 함께 프로그램이 강제종료된다.	Fail
	1-4	계좌번호에 문자를 넣으면 중지되는지 확인한다.	Fail

		→ java.lang.NumberFormatException와 함께 프로그램이 강제종료된다.	
	1-5	<p><b>아이템 종류에 문자를 넣으면 중지되는지 확인한다.</b></p> <p>→ java.lang.NumberFormatException와 함께 프로그램이 강제종료된다.</p> <pre>Exception in thread "AWT-EventQueue-0" java.lang.ArrayIndexOutOfBoundsException: 3     at BankSystem.Bank.loadData(Bank.java:58)     at BankSystem.Bank.&lt;init&gt;(Bank.java:35)     at BankSystem.Bank.checkAccount(Bank.java:170)     at ATM.ATM.destAccount(ATM.java:359)</pre>	Fail
입금	2-1	<p><b>현재 ATM기기가 현금을 가득 보유하고 있을 때 1000원의 입금을 시도하면 중지되는지 확인한다.</b></p> <p>→ 중지되지 않고 입금이 완료된다.</p>	Fail
출금	3-1	<p><b>현재 ATM기기가 보유하고 있는 50000원권의 수량이 0일 때 50000원을 출금하려고 시도하면 10000원짜리 다섯장으로 출금되는지 확인한다.</b></p> <p>→ 출금이 되지 않는다.</p>	Fail
	3-3	<p>50달러를 정상적으로 출금하는 지 확인한다.</p> <p>→ 50달러로 출금된다.</p>	Fail
	3-4	<p><b>5000만원을 정상적으로 출금하는 지 확인한다.</b></p> <p>→ 세 자리 숫자만 입력 가능하다.</p>	Fail
	4-2	<p><b>50달러를 정상적으로 이체하는 지 확인한다.</b></p> <p>→ 50달러로 이체된다.</p> <p>(6) 기준 금액의 10배가 출금/송금 됨(ex. 100달러를 출금/송금할 시 1,000달러가 출금/송금됨).</p> 	Fail
	4-3	<p><b>5000만원을 정상적으로 출금하는 지 확인한다.</b></p> <p>→ 세 자리 숫자만 입력 가능하다.</p> 	Fail
알림	5-1	<p><b>명세표 용지가 남아 있지 않을 때 명세표를 출력하면 알림이 뜨는 지 확인한다.</b></p> <p>→ 알림이 뜨지 않고 출력된다.</p>	Fail
	5-2	<p><b>ATM기기의 현재 10000원 보유량이 최대치일 때 10000원의 입금을 시도하면 알림이 뜨는 지 확인한다.</b></p> <p>→ 알림이 뜨지않고 입금된다.</p>	Fail
	5-3	<p><b>ATM기기의 현재 10\$ 보유량이 최대치일 때 10\$의 입금을 시도하면 알림이 뜨는 지 확인한다.</b></p> <p>→알림이 뜨지않고 입금된다.</p>	Fail
	5-4	<p><b>ATM기기가 1000원을 보유하고 있지 않을 때 23000원의 출금을 시도하면 알림이 뜨는 지 확인한다.</b></p> <p>→알림이 뜨지않고 출금된다.</p>	Fail



교통카드	6-1	<p>교통카드 생성 시 0.3달러가 출금되는 지 확인한다. →3달러가 출금된다.</p> <p><b>Transaction Amount : 3 \$</b></p>	Fail
	6-2	<p>현재 ATM기기가 교통카드 여분을 보유하고 있지 않을 때 교통카드 생성 시 알림이 뜨고 중지되는지 확인한다. →알림이 뜨지 않고 교통카드가 생성된다.</p>	Fail

## 5 Overall

### 5.1 System Test Result

#### 5.1.1 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.

#### 5.1.2 Pairwise Test

aaaa

#### 5.1.3 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.

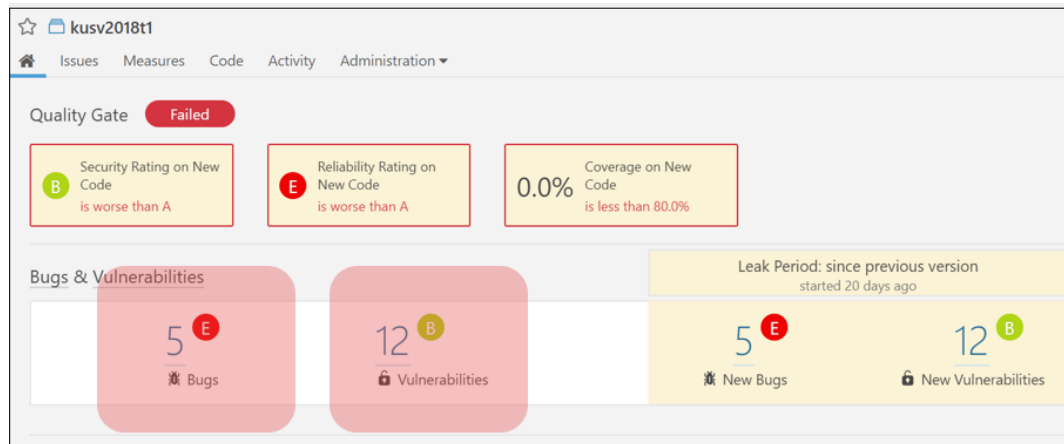
#### 5.1.4 Summary

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 닫지 않으면 계속 출력된다.
    - vi. account id를 입력할 시 기존에 있던 label이 삭제되고 그 곳에 account id가 입력된다.

## 6 CTIP Usage

### 6.1 Static Analysis Result

Build Case 별로 정적분석 결과 확인이 가능하다.



#75	2018. 5. 25 오후 1:44	
#74	2018. 5. 25 오전 9:16	
#73	2018. 5. 24 오후 1:22	





































```
Remove this conditional structure or edit its code blocks so that they're not all the same. 3 days ago L94
```

```
Bug Major Open Not assigned 15min effort Comment No tags
```

```
// 계좌조회
case 1:
    break;
// 입금
case 2:
    break;
// 출금
case 3:
    break;
// 송금
case 4:
    break;
// 교통카드 발급
case 5:
    break;
}
return;
```

## 6.2 Issue Tracker

테스트 결과를 이슈트래커에 등록했다. 이슈트래커는 G메일과 연동되어 있어 SMA 팀에게 notification이 전달된다.

 [Category Partitioning Test] 100.201.400.600.610.700.810.821.900 <b>bug</b> #45 opened 12 minutes ago by kusv2018t1 	
 [Category Partitioning Test] 100.201.400.600.610.701.822.900 <b>bug</b> #44 opened 12 minutes ago by kusv2018t1 	
 [Category Partitioning Test] 100.201.400.600.610.701.822.900 <b>bug</b> #43 opened 13 minutes ago by kusv2018t1 	
 [Category Partitioning Test] 100.201.400.600.610.701.821 <b>bug</b> #42 opened 13 minutes ago by kusv2018t1 	
 [Category Partitioning Test] 100.201.400.600.610.700.811.822.900 <b>bug</b> #41 opened 13 minutes ago by kusv2018t1 	
 [Category Partitioning Test] 100.201.400.600.610.700.811.821 <b>bug</b> #40 opened 13 minutes ago by kusv2018t1 	
 [Category Partitioning Test] 100.201.400.600.610.700.810.822.900 <b>bug</b> #39 opened 13 minutes ago by kusv2018t1 	
 [Category Partitioning Test] 100.201.400.600.610.700.810.821.900 <b>bug</b> #38 opened 14 minutes ago by kusv2018t1 	
 [Category Partitioning Test] 100.200.300.701 <b>bug</b> #37 opened 14 minutes ago by kusv2018t1 	
 [Category Partitioning Test] 100.200.300.700.811 <b>bug</b> #36 opened 14 minutes ago by kusv2018t1 	
 [Category Partitioning Test] 100.200.300.700.810.900 <b>bug</b> #35 opened 14 minutes ago by kusv2018t1 	
 [Category Partitioning Test] 1002 <b>bug</b> #34 opened 14 minutes ago by kusv2018t1 	

- #21 opened 22 minutes ago by kusv2018t1  System Test
- 
- ① [BruteForceTest] 교통카드 생성 시 출금 금액을 확인한다. **bug**  
#20 opened 22 minutes ago by kusv2018t1  System Test
- 
- ① [BruteForceTest] ATM기기가 1000원을 보유하고 있지 않을 때 23000원의 출금을 시도한다. **bug**  
#19 opened 22 minutes ago by kusv2018t1  System Test
- 
- ① [BruteForceTest] ATM기기가 10\$를 최대로 보유하고 있을 때 10\$의 입금을 시도한다. **bug**  
#18 opened 23 minutes ago by kusv2018t1  System Test
- 
- ① [BruteForceTest] ATM기기가 10000원을 최대로 보유하고 있을 때 10000원의 입금을 시도한다. **bug**  
#17 opened 23 minutes ago by kusv2018t1  System Test
- 
- ① [BruteForceTest] 명세표 용지가 남아있지 않을 때 명세표를 출력한다. **bug**  
#16 opened 24 minutes ago by kusv2018t1  System Test
- 
- ① [BruteForceTest] 5000만원을 이체한다. **bug**  
#15 opened 24 minutes ago by kusv2018t1  System Test
- 
- ① [BruteForceTest] 50달러를 이체한다. **bug**  
#14 opened 25 minutes ago by kusv2018t1  System Test
- 
- ① [BruteForceTest] 5000만원을 출금한다. **bug**  
#13 opened 25 minutes ago by kusv2018t1  System Test
- 
- ① [BruteForceTest] 50달러를 출금한다. **bug**  
#12 opened 26 minutes ago by kusv2018t1  System Test
- 
- ① [BruteForceTest] ATM기기가 보유하고 있는 50000원권이 없을 때 50000원을 출금한다. **bug**  
#11 opened 26 minutes ago by kusv2018t1  System Test
- 
- ① [BruteForceTest] ATM기기에 현금이 가득 찼을 때 1000원의 입금을 시도한다. **bug**  
#10 opened 26 minutes ago by kusv2018t1  System Test