

2018 SOFTWARE VERIFICATION

# 1st Testing

System Test

2018.05.28 (일) SV T1

201311263 김민환

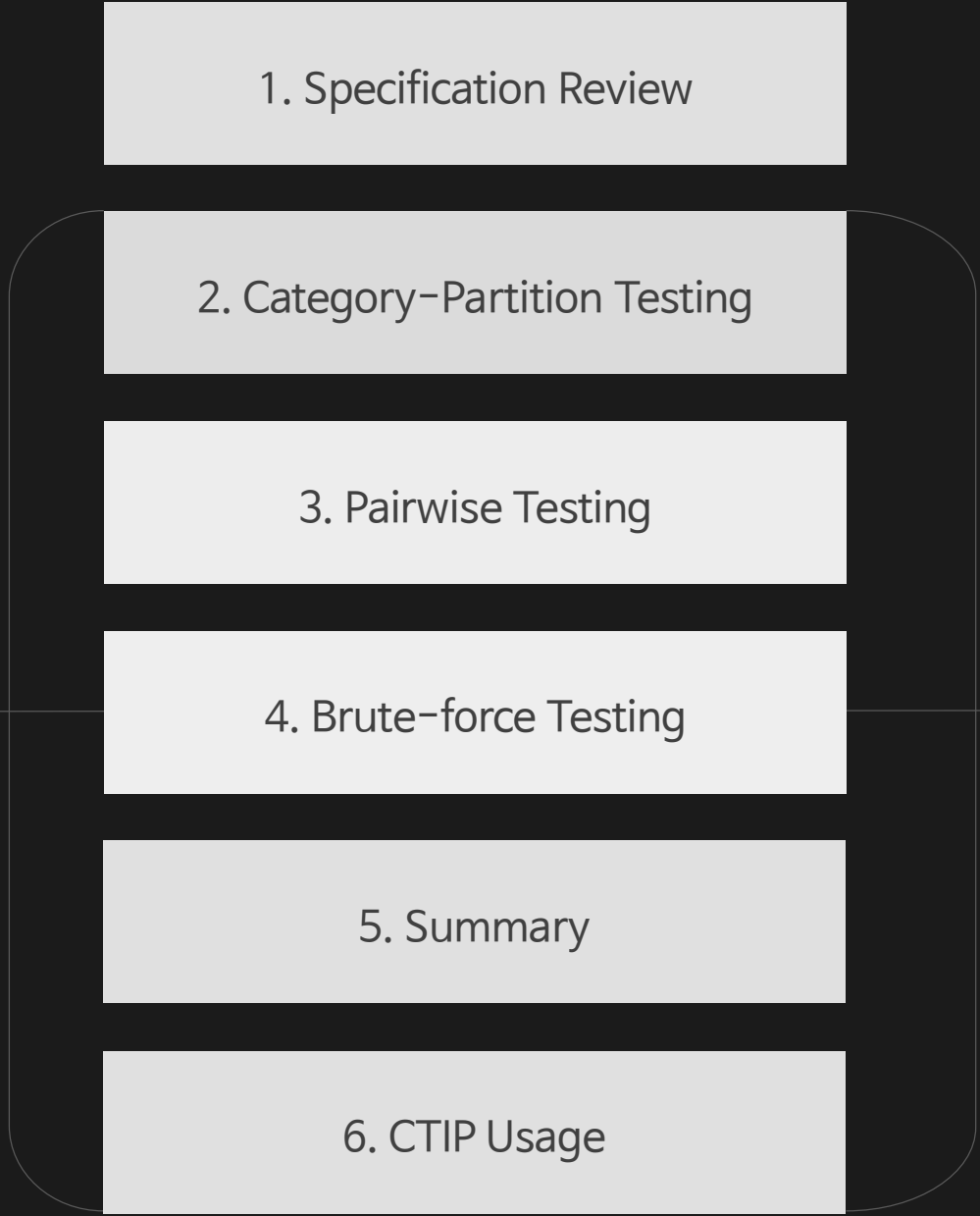
201311308 전세진

201411278 서희진

201411317 조민규

# 1st Testing

2018 SOFTWARE VERIFICATION



# 01. Specification Review

## 1001. Define Draft Plan

### 3. Functional Requirements

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

→ **Verify Sufficient Fund**가 생략되어 있다.

## 1001. Define Draft Plan

### 4. Non-Functional Requirements

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

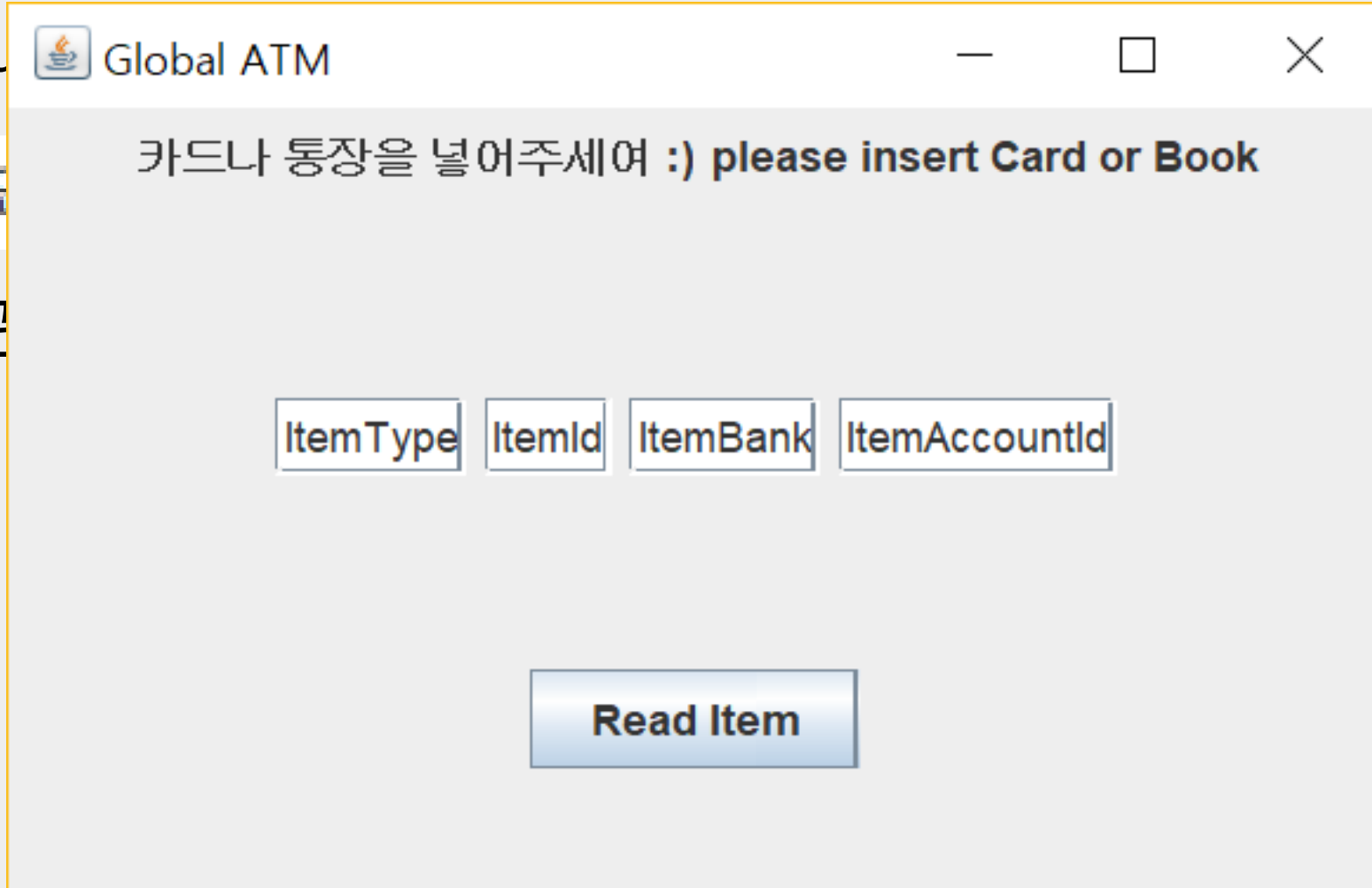
→ 직관적이지 않은 인터페이스

1001. Define Draft Plan

4. Non-Fu

- 외국인

→ 직권



1001. Define Draft Plan

6. Other Information

-현재 환율정보를 고려하여 환전을 돕는다.

→ 현재의 환율정보를 고려하지 않음.

## 1001. Define Draft Plan

### 6. Other Information

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

→ 관리자에게 문자 알림이 가지 않음.



## 1002. Create Preliminary Investigation Report

## 2. Project Justification

1) Cost : 노동력과 서버비용 이외에 지출이 없어 저렴하다

→ 서버를 사용하지 않음.

## 1002. Create Preliminary Investigation Report

## 6. Other Managerial Issues

2) 추후 **졸작으로 활용 가능**

→ 졸작으로 활용가능해 보이지 않는 Quality

## 1003. Define Requirements

## 1. Functional Requirements

No.	Function	Description
1	Check	계좌 잔액 확인
2	Deposit	현금을 받아, 확인하고, 계좌에 증감
3	Withdraw	User 가 원하는 현금만큼 계좌에서 차감하여 현금으로 교환
4	Transfer	User 가 원하는 금액만큼 입력한 타계좌로 이동
5	Update	거래 내역 업데이트
6	Status Alarm	ATM 현금/교통카드/명세표 소지량 경고 알림
7	IssueTrafficcard	외국계좌와 연동된 교통카드 발급
8	Management	관리자가 ATM 안 현금을 관리한다.

**Verify Sufficient Fund가 생략되어 있다.**

## 1003. Define Requirements

## 2. Performance Requirement

최대한 간소화하여 User 가 거래에 집중할 수 있도록 한다.  
적절한 본인 인증을 통하여 타의/악의적인 접근을 막는다.  
기기는 충분한 현금을 소지하고 있다.

→ 불명확한 표현

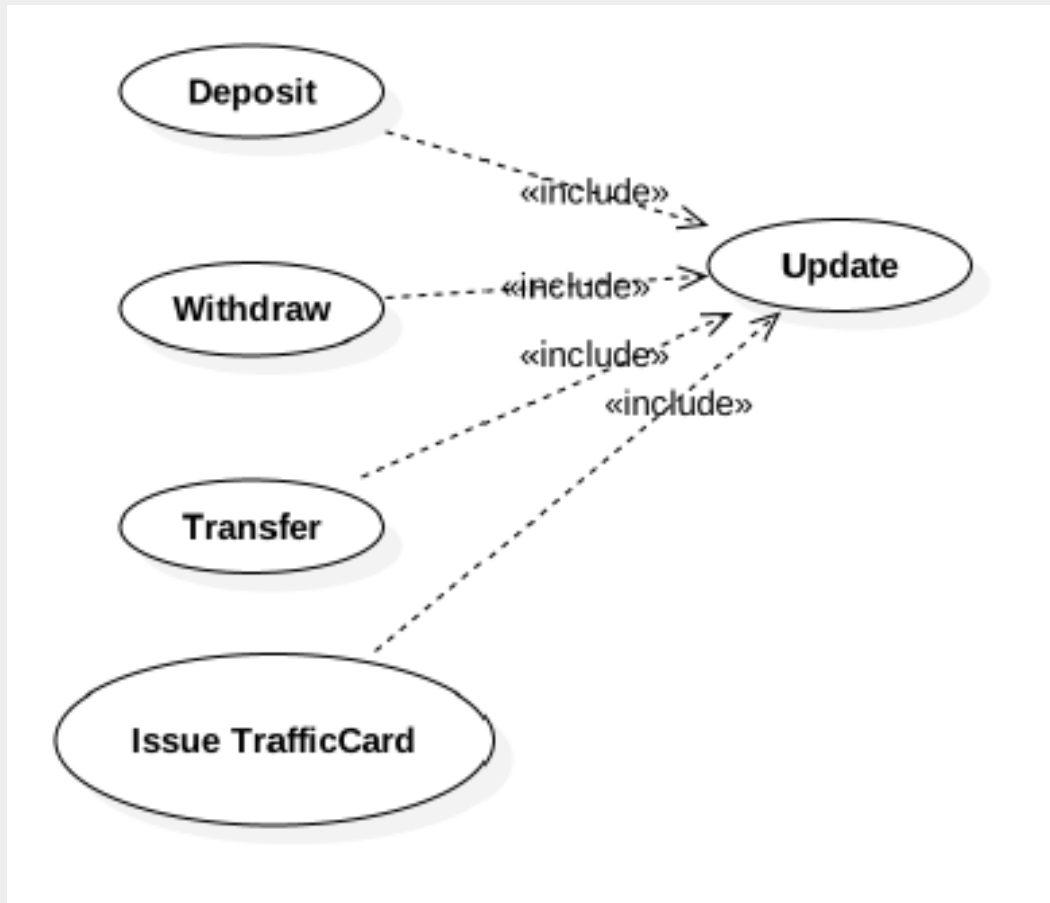
## 1004. Record Terms in Glossary

TrafficCard	ATM 에서 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

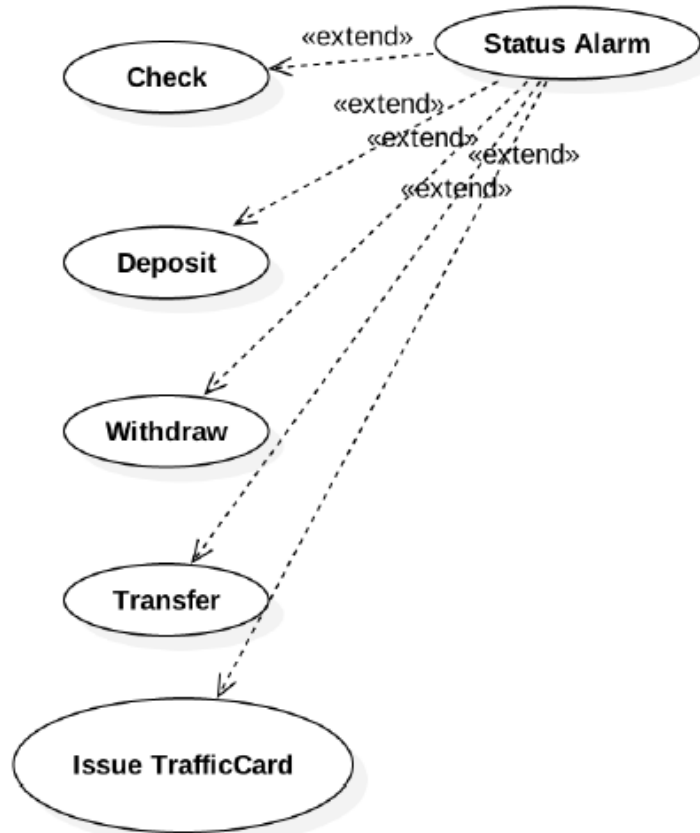


Event-based Use-case인 Update와의  
관계 오류

- 화살표 방향이 반대

## 1007. Define Business Use Cases

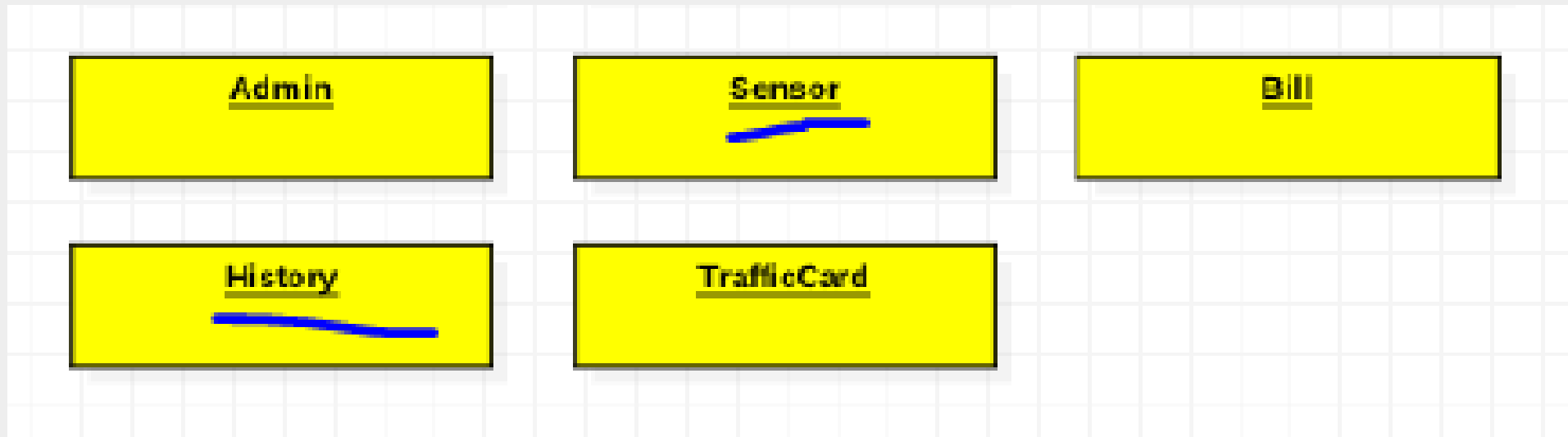
## 6. Identify the Relationships between Use-Case



Status Alarm과 Check, Deposit, Transfer, Issue TrafficCard 간의 관계 오류

- extends → include

## 1008. Define Business Concept Models



→ 2030 단계 이후로는 등장하지 않는 심볼 (Sensor, History)



## 1008. Define Business Concept Models

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

→ 2030단계 이후로는 등장하는 Attribute

## 1010. Refine Plan

## 3. Functional Requirements

## 3 Functional Requirements

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

→ Verify Sufficient Fund가 빠져있다.

## 1010. Refine Plan

## 4. Performance Requirements

- 외국인과 내국인 모두 손쉽게 접근할 수 있는 직관적인 인터페이스를 지니고 있어야 한다.

→ 직관적이지 않은 인터페이스

1010. Refine Plan

4. Per

- 외  
한다.



있어야

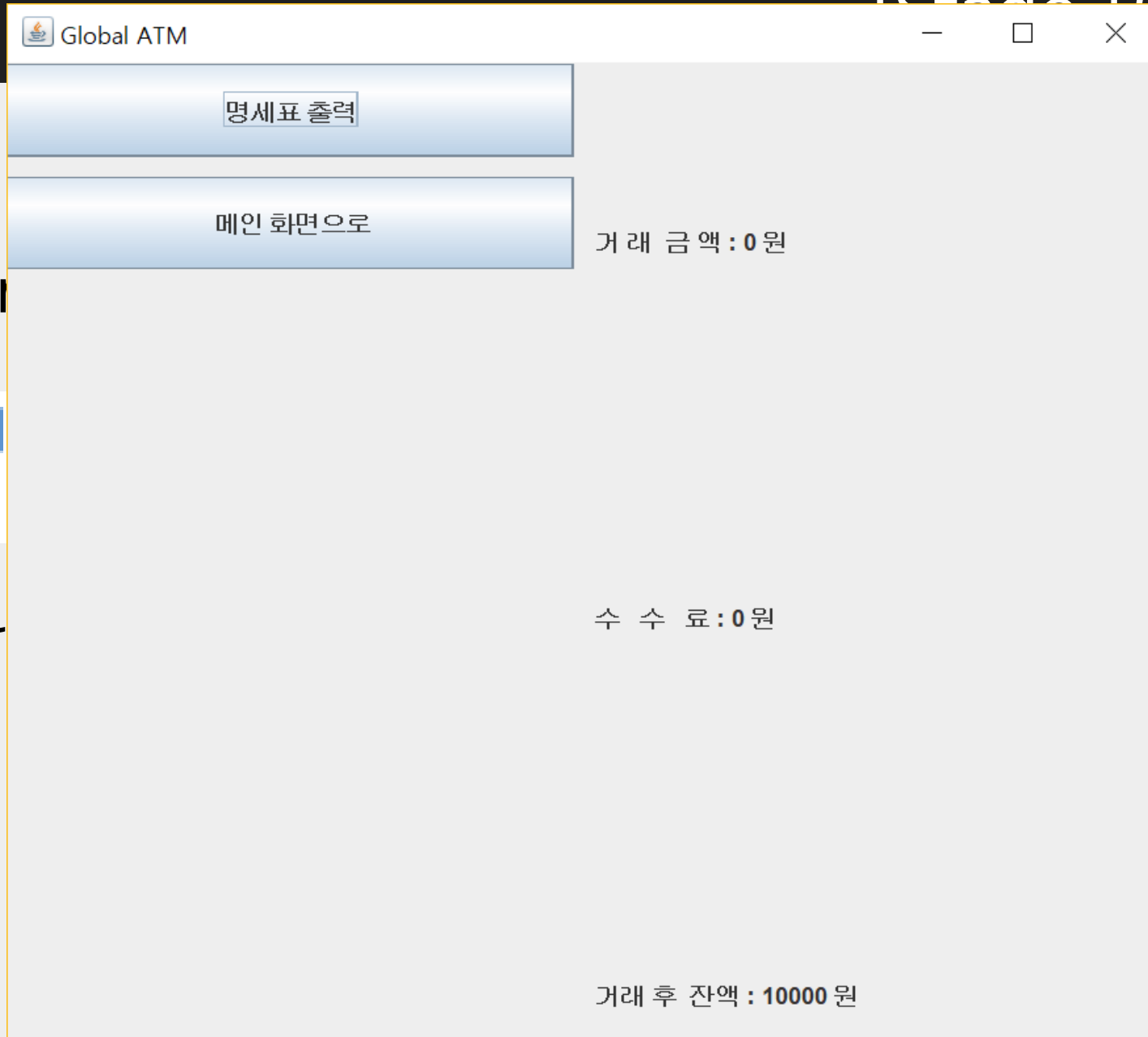


# 0. Refine Plan

## 4. Performance

- 외국인과 내  
한다.

→ 아니다



지니고 있어야

## 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.

→ ‘선택한다’를 ‘연산하다’로 수정 필요

## 2031. Describe Use Case

## 1.1 Check

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.
----------------------------------	---

→ 다른 거래를 동시에 선택할 수 없다.

## 2031. Describe Use Case

## 1.2 Deposit

4.(S) 유효한 현금인지 검사한다.

- ‘유효한’ 현금이 무엇인지 불분명.
- 어떻게 검사하는지 명시되어 있지 않음.
- 검사하지 않음.



## 2031. Describe Use Case

## 1.3 Withdraw

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

- ‘일정 시간’이 어느 정도인지 명시되어 있지 않음.
- 투입구를 폐쇄하지 않음.

## 2031. Describe Use Case

## 1.4 Issue TrafficCard

Exceptional Courses of Events

- E1. 본인확인 실패 시 종료한다.
- E5. 안내사항 속지 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.

## 2031. Describe Use Case

## 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-Requisites가 아니다.

## 2031. Describe Use Case

### 1.6 Update

1.(S) Update 에 필요한 정보를 받는다.

→ 필요한 정보가 어떤 것인지 명시되어 있지 않다.

## 2031. Describe Use Case

## 1.8 Status Alarm

## Purpose

ATM 보유 현금량이 충분하지 않을 경우 관리자에게 알림을 보내준다.

- ATM 보유 현금량이 최대값에 도달했을 때에도 관리자에게 알림을 보내지 않는다.
- 명세표의 출력용지가 충분하지 않을 경우에도 관리자에게 알림을 보내지 않는다.

## 2031. Describe Use Case

## 1.8 Status Alarm

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"

- 오타: Transger
- Use-Case에 "Check" 누락

## 2031. Describe Use Case

## 1.9 Management

Typical Courses of (A): Actor (S):System

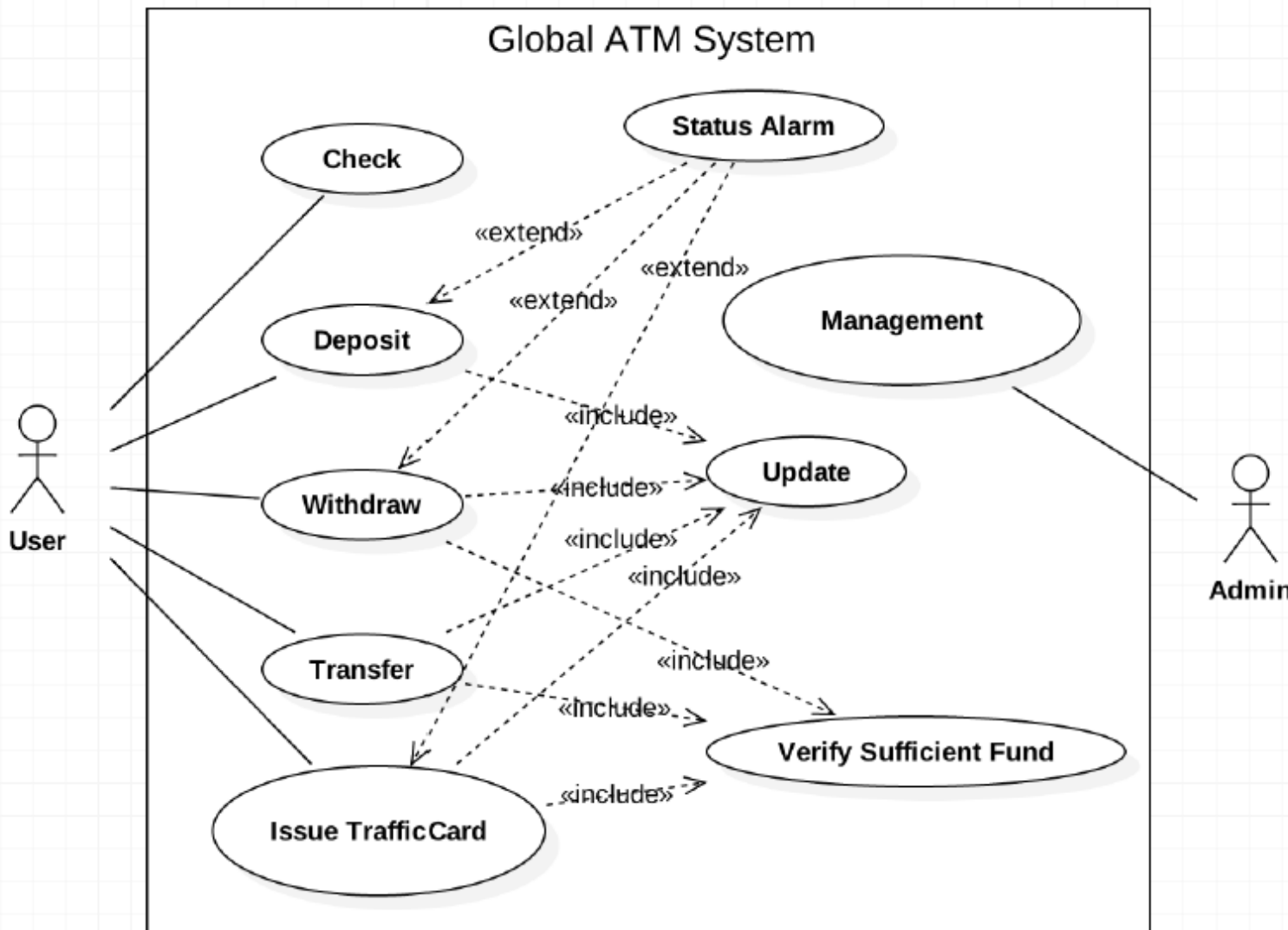
Events

1.(A) 관리자임을 인증한다.

2.(S) ATM 의 현금/교통카드/명세표출력용지 저장소를 개방한다.

→ ATM을 관리하는 관리자의 행동이 추가되어야 한다.

## 2032. Refine Use Case Diagrams

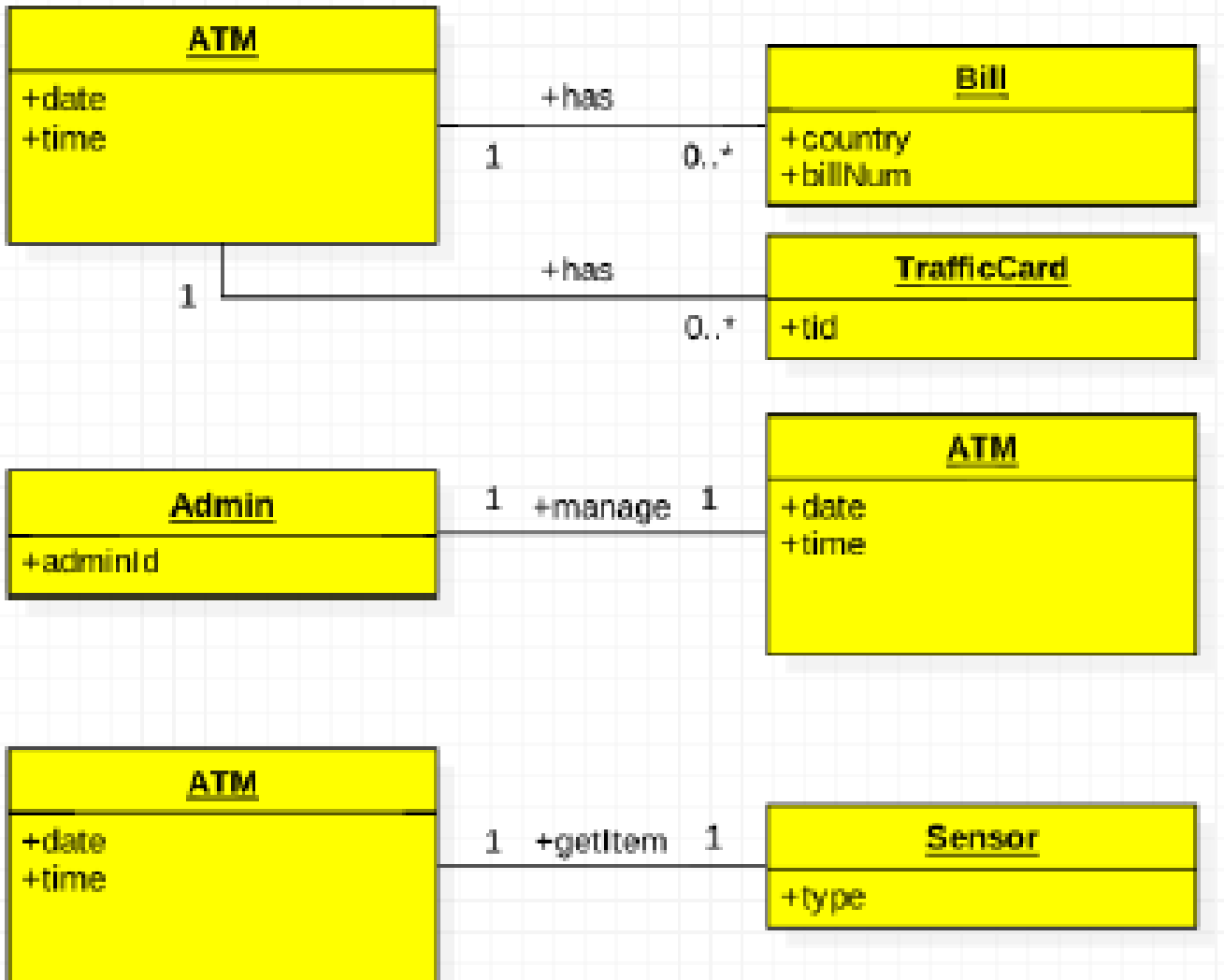


수정한 1000단계의  
내용과 다름.



## 2033. Define Domain Models

정의된 적 없는 Sensor, Module



## 2034. Refine Clossary

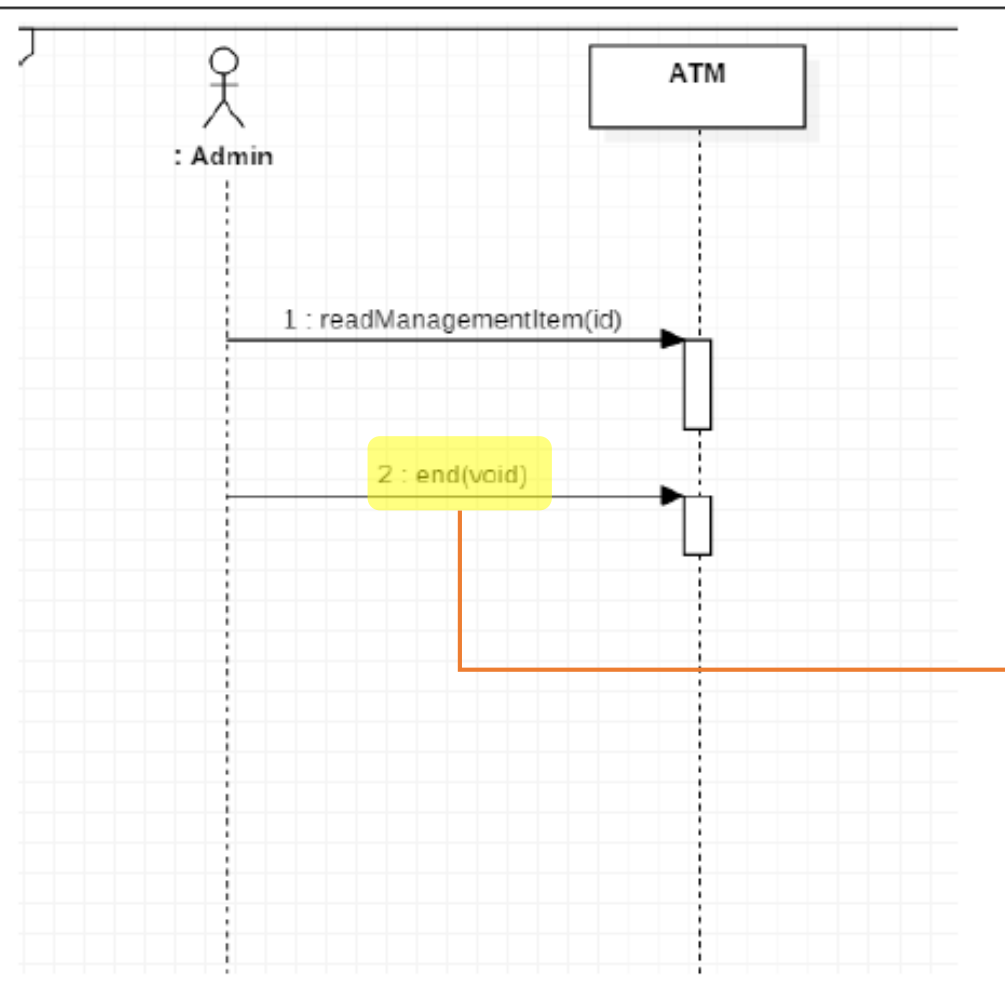
<b>chargeFee</b>	Attribute	수수료
<b>sender</b>	Attribute	송신 계좌
<b>History</b>	Concept	거래 내역

→ 사용하지 않는 변수들

## 2035. Define System Sequence Diagrams

## Use Case : Management

1. 사용자(Admin)가 고유의 key 를 삽입한다.
2. 시스템이 사용자에게 접근 권한을 준다.(접근권한을 주면서 ATM 기기 안의 현금, 영수증량, 교통카드량 제어 가능)
3. ATM 기기가 열린다.
4. 작업 후, ATM 기기가 닫히면 end 가 실행된다.



정의되지 않은  
end 기능 추가

## 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

## 2036. Define Operation Contracts

## 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.

## 2036. Define Operation Contracts

### 6.3 printReceipt(print)

- 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.

## 2036. Define Operation Contracts

## 6.4 confirm (password)

Exceptions	-비밀번호가 틀린 경우
------------	--------------

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

## 2036. Define Operation Contracts

## 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.



## 2036. Define Operation Contracts

### 6.5 insertCash(type, amounts)

#### Responsibilities

- 지폐를 투입구에 넣고 투입한 지폐의 종류와 양을 확인하지 않음.

#### Cross Reference

- ‘Functions R.2.1, R.2.2’ 추가 필요
- Use cases에 ‘Verify Sufficient Fund , Status Alarm’ 추가 필요

#### Exceptions

- ‘ATM의 지폐 보유량이 최대값에 도달한 경우 알림을 보낸다.’ 추가 필요
- Exceptions에 ‘원화와 달러를 섞어서 입금할 경우’ 추가 필요

## 2036. Define Operation Contracts

## 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.

## 2036. Define Operation Contracts

### 6.7 enterAmounts(amounts)

#### Cross Reference

- 'Functions R.2.1, R.2.2' 추가 필요
- Use cases에 'Verify Sufficient Fund, Status Alarm' 추가 필요

#### Exceptions

- 'ATM의 지폐 보유량이 부족할 경우' 추가 필요

## 2036. Define Operation Contracts

## 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.

## 2036. Define Operation Contracts

## 6.8 destAccount(bank, account)

Post-conditions

→ ‘출금할 지폐의 종류를 확인한다.’ 추가 필요

## 2036. Define Operation Contracts

## 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.

## 2036. Define Operation Contracts

## 6.9 setDateRange(date, range)

Post-conditions

→ ‘교통카드 계좌연동 및 비용 결제를 안내한다.’ 추가 필요

## 2036. Define Operation Contracts

## 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.



## 2036. Define Operation Contracts

## 6.10 agreement (approval)

## Responsibilities

- ‘교통카드 계좌연동 및 비용 결제를 안내하고 동의를 받는다.’ 수정 필요

## Cross Reference

- ‘Functions R.2.1, R.2.2’ 추가 필요
- Use cases에 ‘Verify Sufficient Fund , Status Alarm’ 추가 필요

## Exceptions

- ‘ATM의 교통카드 보유량이 부족할 경우’ 추가 필요

## 2036. Define Operation Contracts

## 6.11 printReceipt(print)

→ 6.3의 printReceipt와 중복되므로 삭제 필요

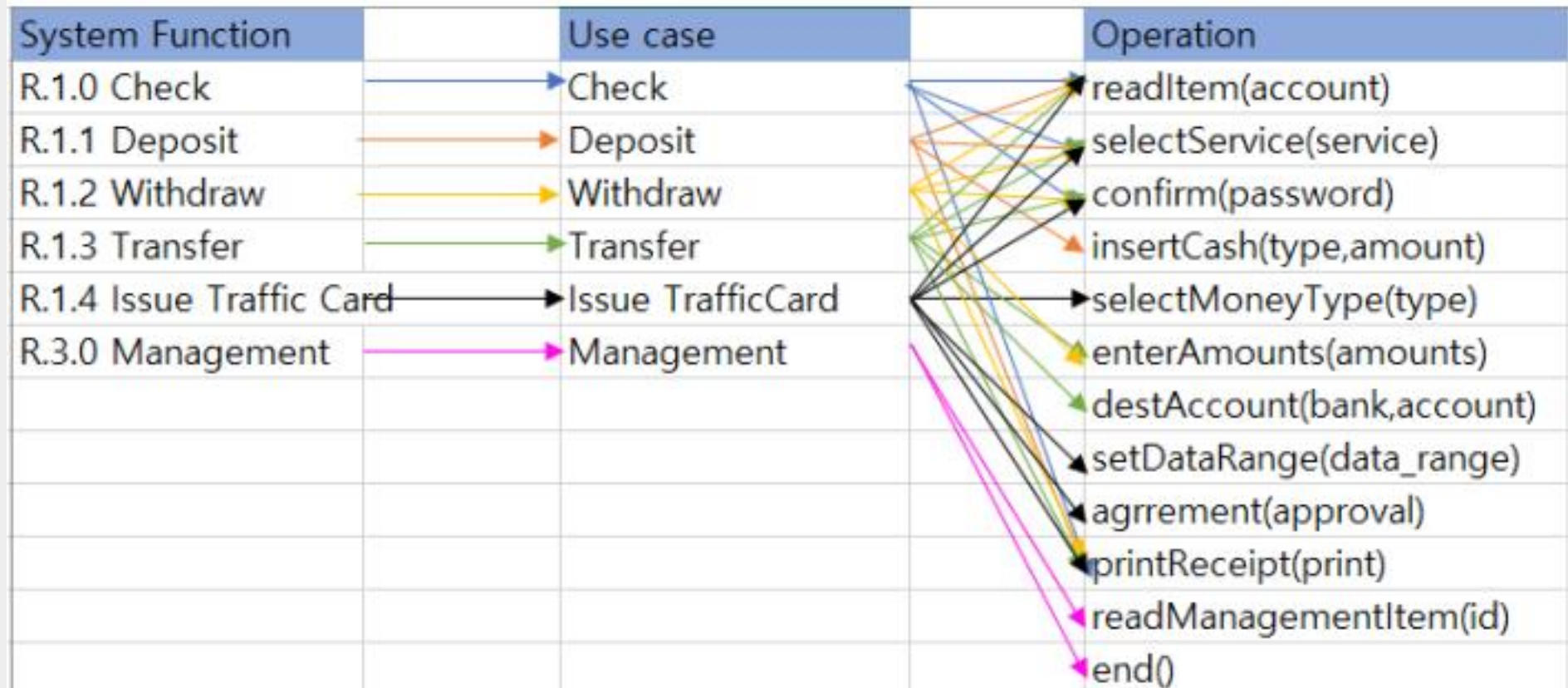
## 6.12 readManagementItem(id)

→ 관리자를 사용하지 않으므로 삭제 또는 수정 필요

## 6.13 end(void)

→ 앞서 end에 대한 기능이 정의되어 있지 않으므로 수정 필요

## 2039. Traceability Analysis



→ 앞 단계의 변경 내용을 반영하여 전체적인 수정 필요.

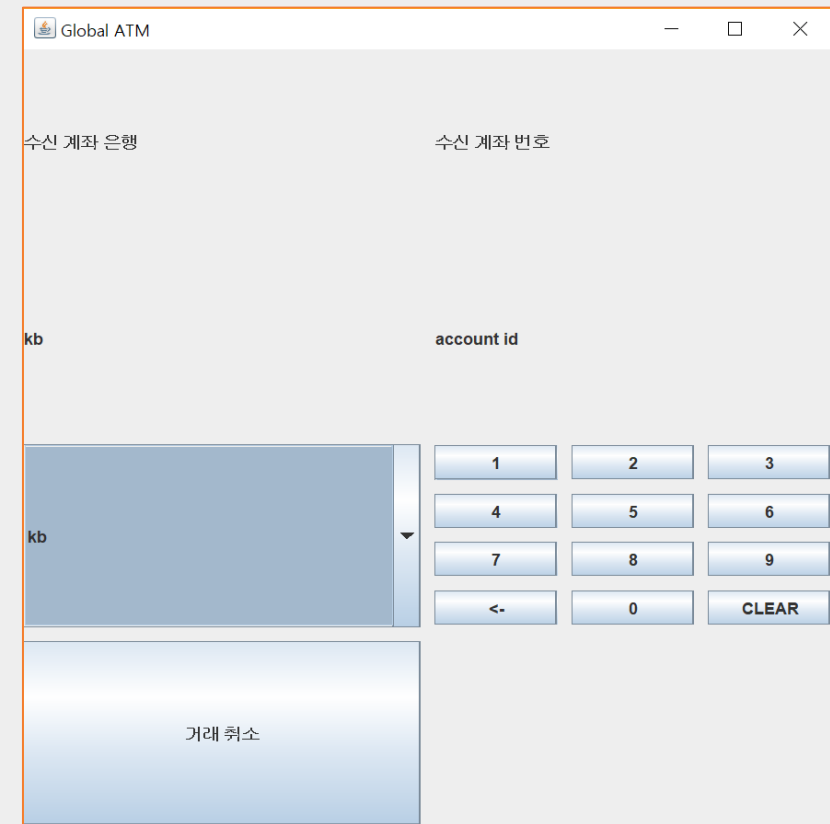
## 2141. Design Real Use Cases

- 2030 단계에서 변경된 내용을 반영해 수정해야 한다.
- 구현단계에서 Admin과 관련된 부분이 완전히 사라졌음.
  - R.2.2 Status Alarm, R.3.0 Management가 아예 사라짐.

## 2142. Define Reports, UI, and Storyboards



VS



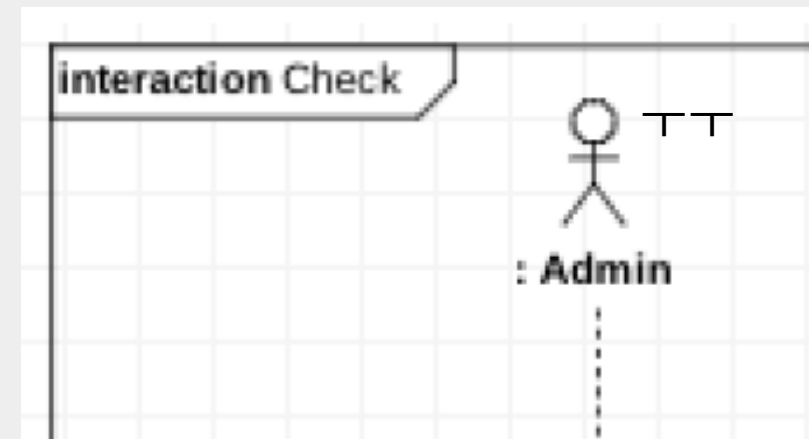
UI 프로토타입과 비교했을 때 화면 개수, 구성, 디자인, 흐름이 완전히 다름.

## 2144. Define Interaction Diagrams

- 실제 프로그램에서 존재하지 않는 Admin이 그려져 있다.
- 구현단계에서 Admin과 관련된 부분이 완전히 사라졌음.
  - R.2.2 Status Alarm(알람기능), R.3.0 Management(관리기능)이 사라짐.

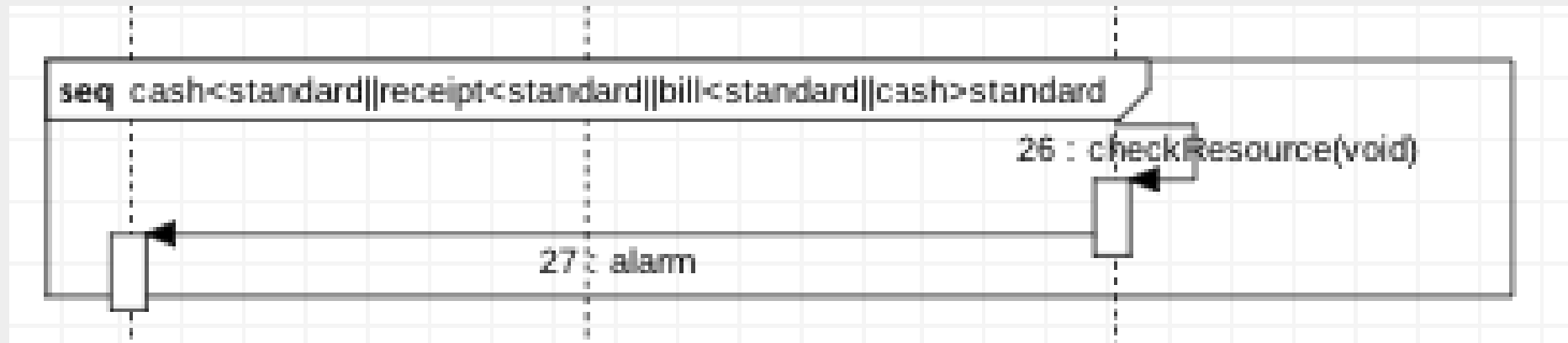
해서, 유즈케이스에도 작성하고, 인터렉션까지도 작성했는데  
소프트웨어적으로는 구현할 필요성을  
못느껴서 구현하지않았습니다!

오후 3:06



## 2144. Define Interaction Diagrams

- 모든 Interaction 다이어그램에는  
관리자가 존재하지 않기에 반환 값인 '알람'이 사라져야 한다.



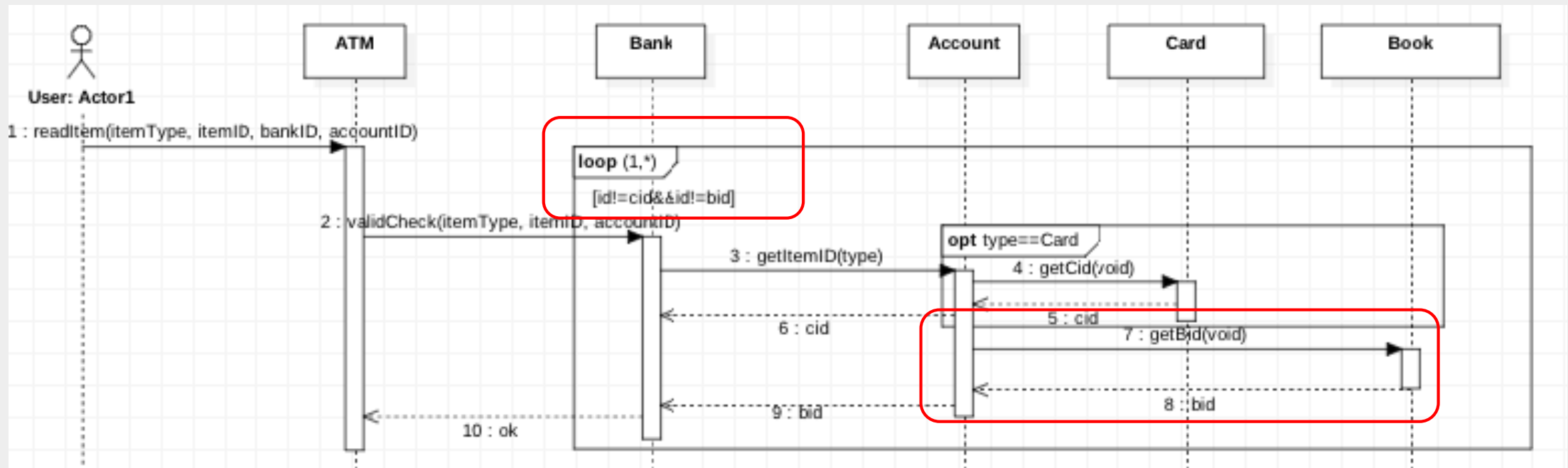
## 2144. Define Interaction Diagrams

- 또한 모든 Interaction Diagram에서 readItem의 생명선은 printReceipt 전 까지 이어져야 한다.



## 2144. Define Interaction Diagrams

## 4.1 Check



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

## 2144. Define Interaction Diagrams

### 4.1 Check

실제 프로그램과는 동작 구조가 다르다.

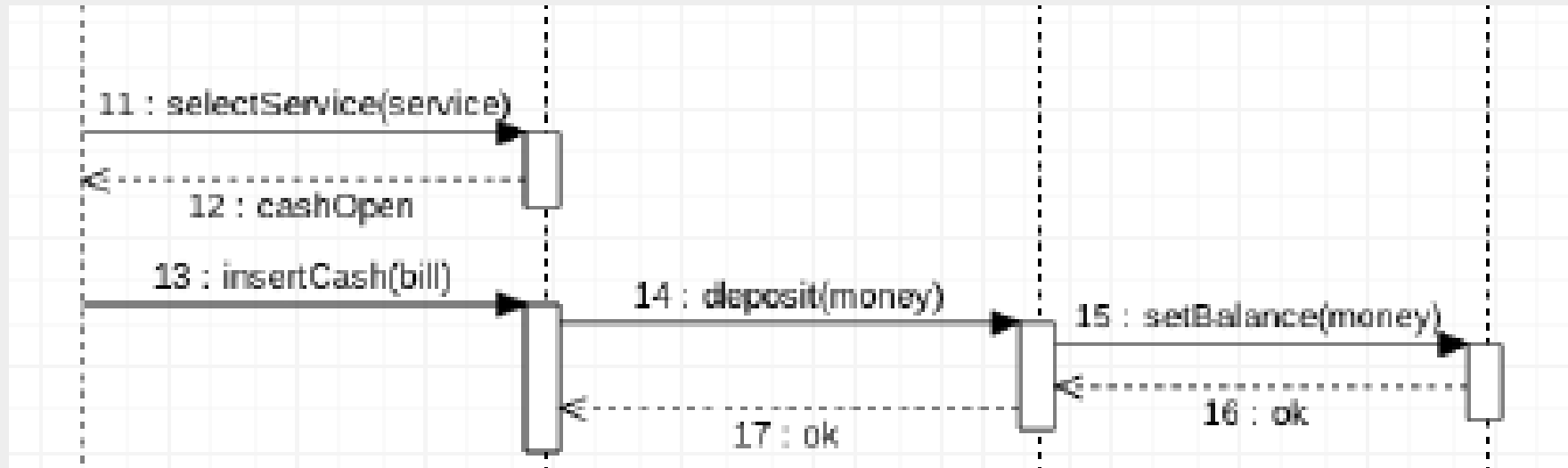
객체들의 정보가 수정되고 반영되는 (Update) 부분이 존재하지 않는다.

12.Confirm의 생명선이 21. printReceipt가 끝날때까지 이어져야 한다.

동일한 문제가 다른 모든 Interaction Diagram에서도 발견됨

## 2144. Define Interaction Diagrams

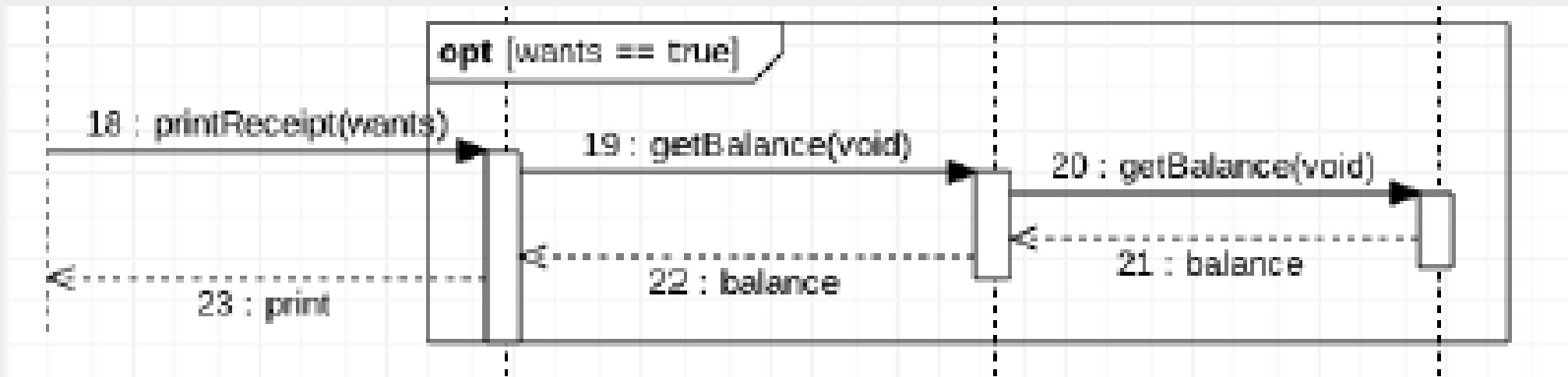
## 4.2 Deposit



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.

## 2144. Define Interaction Diagrams

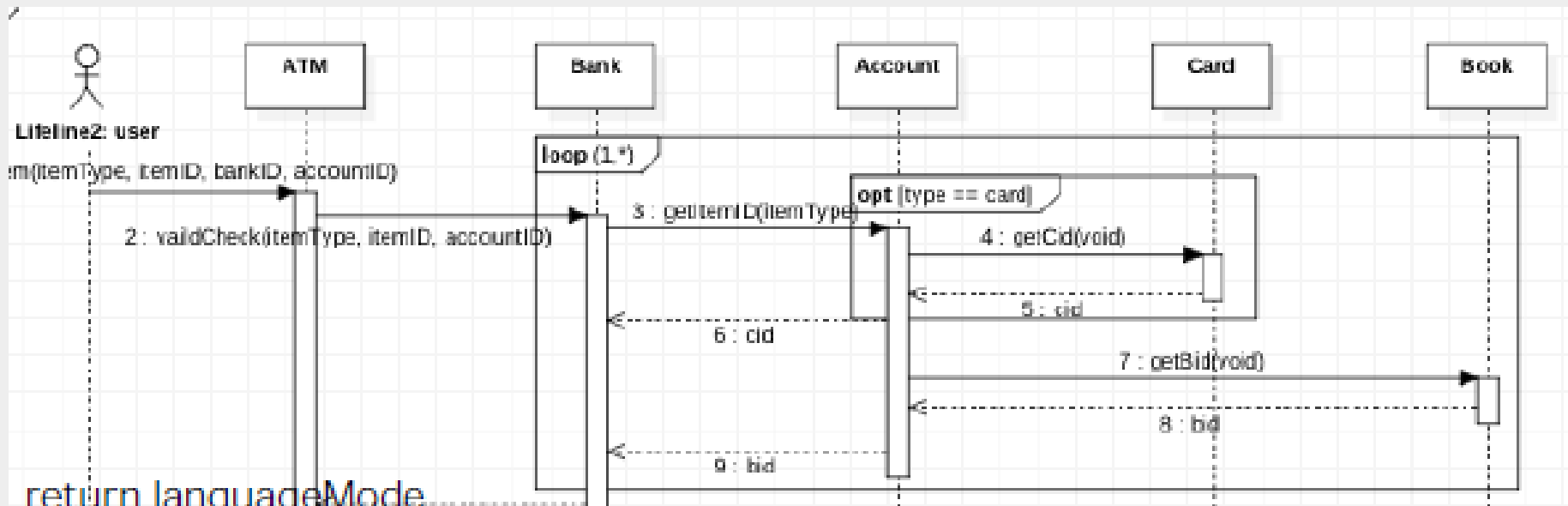
## 4.2 Deposit



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

## 2144. Define Interaction Diagrams

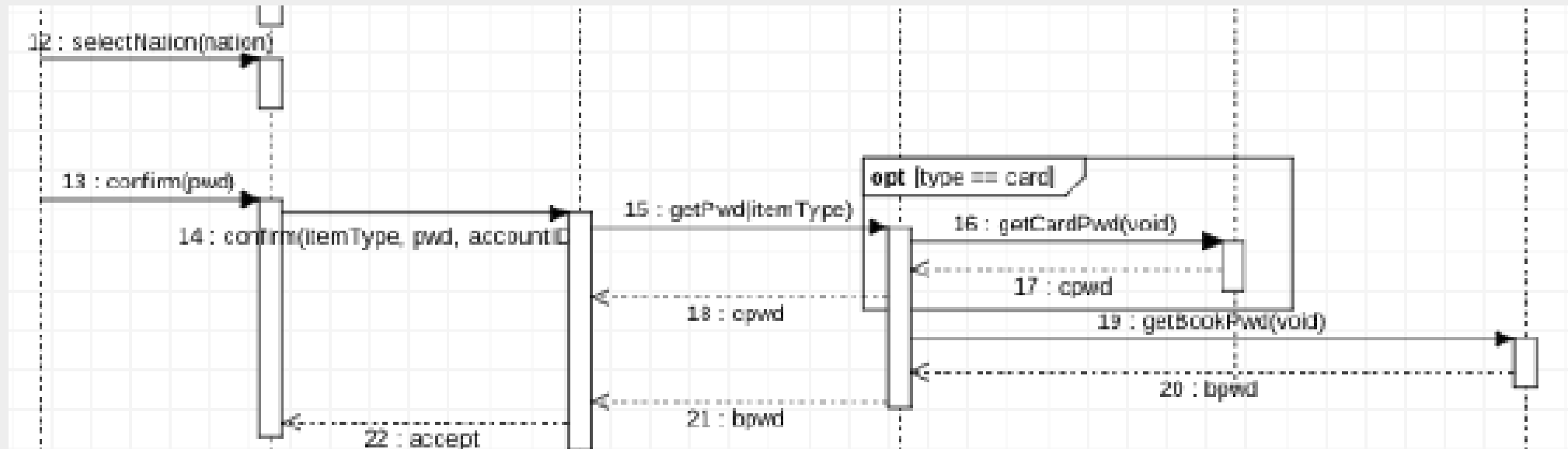
## 4.3 withDraw



Loop문의 조건이 누락되어 있다.  
 readItem의 반환값인 languageMode가 누락되어 있다.  
 selectService는 result의 값에 따라 바뀌는데 누락되어 있다.

## 2144. Define Interaction Diagrams

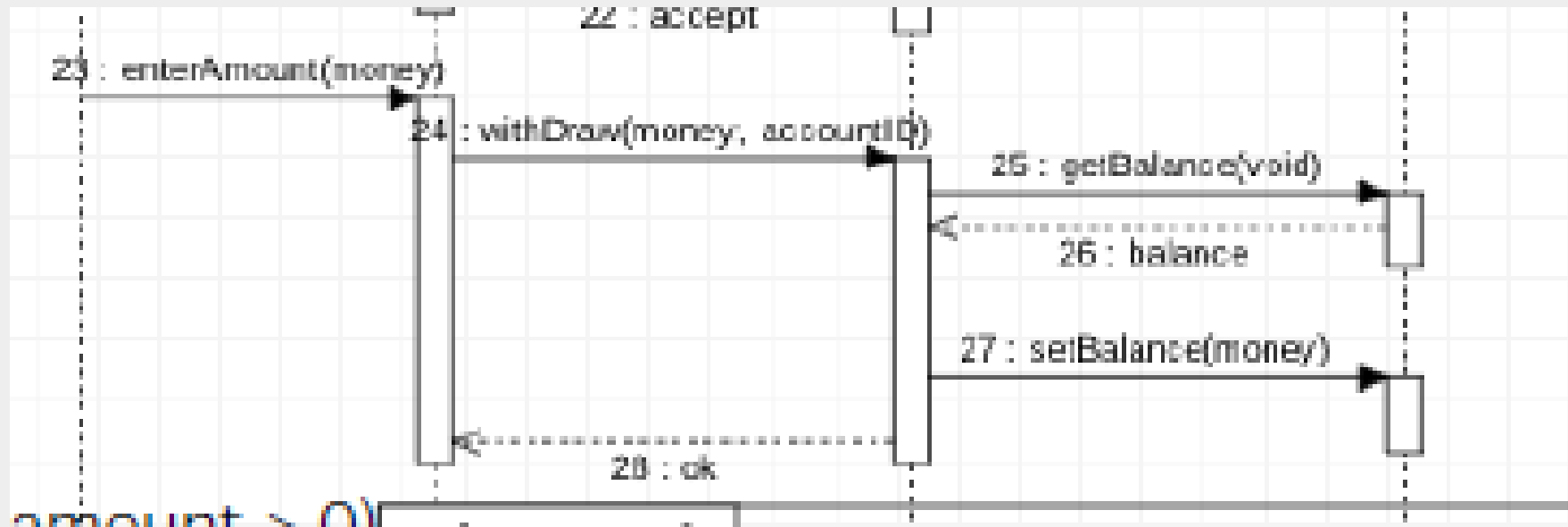
## 4.3 withDraw



14번과 12번의 순서와 상관관계가 바뀌어 있다.  
Confirm의 retur값이 true여야 selectNation으로 진입하는데, 누락되어 있다.  
또한 confirm이 끝날 때 까지 selectNation의 생명선이 이어져야 한다.

## 2144. Define Interaction Diagrams

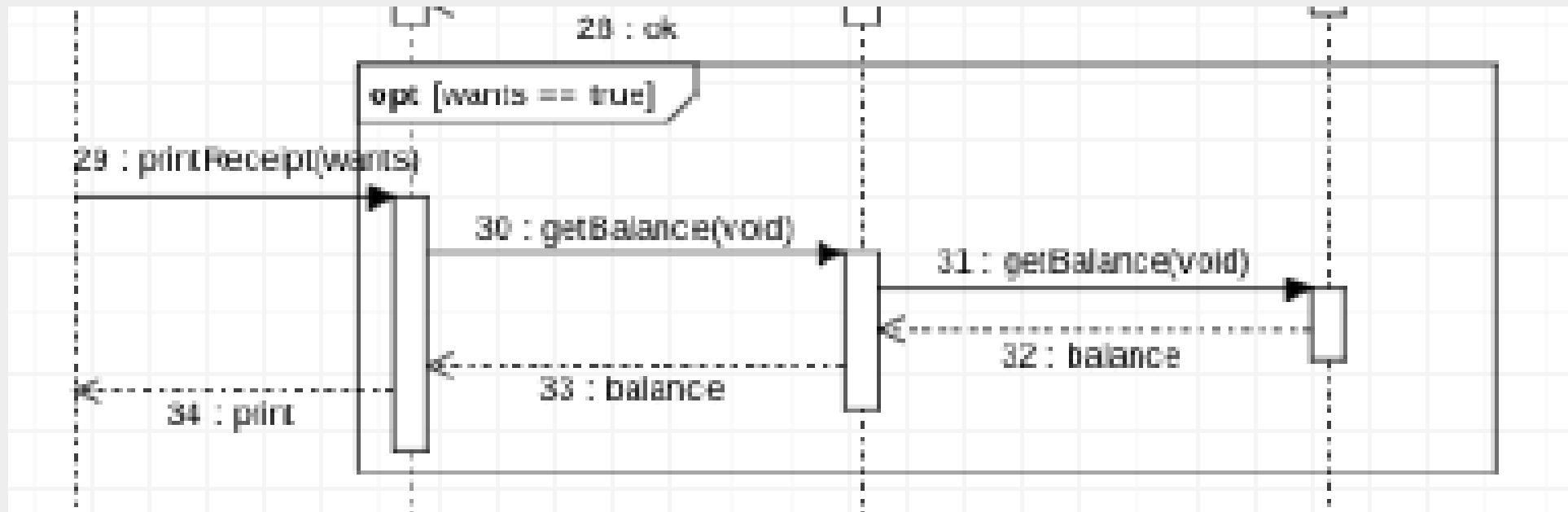
## 4.3 withDraw



문서에 명시되어 있는 잔고의 부족을 검사하는 부분이 누락되어 있다.

## 2144. Define Interaction Diagrams

## 4.3 withDraw

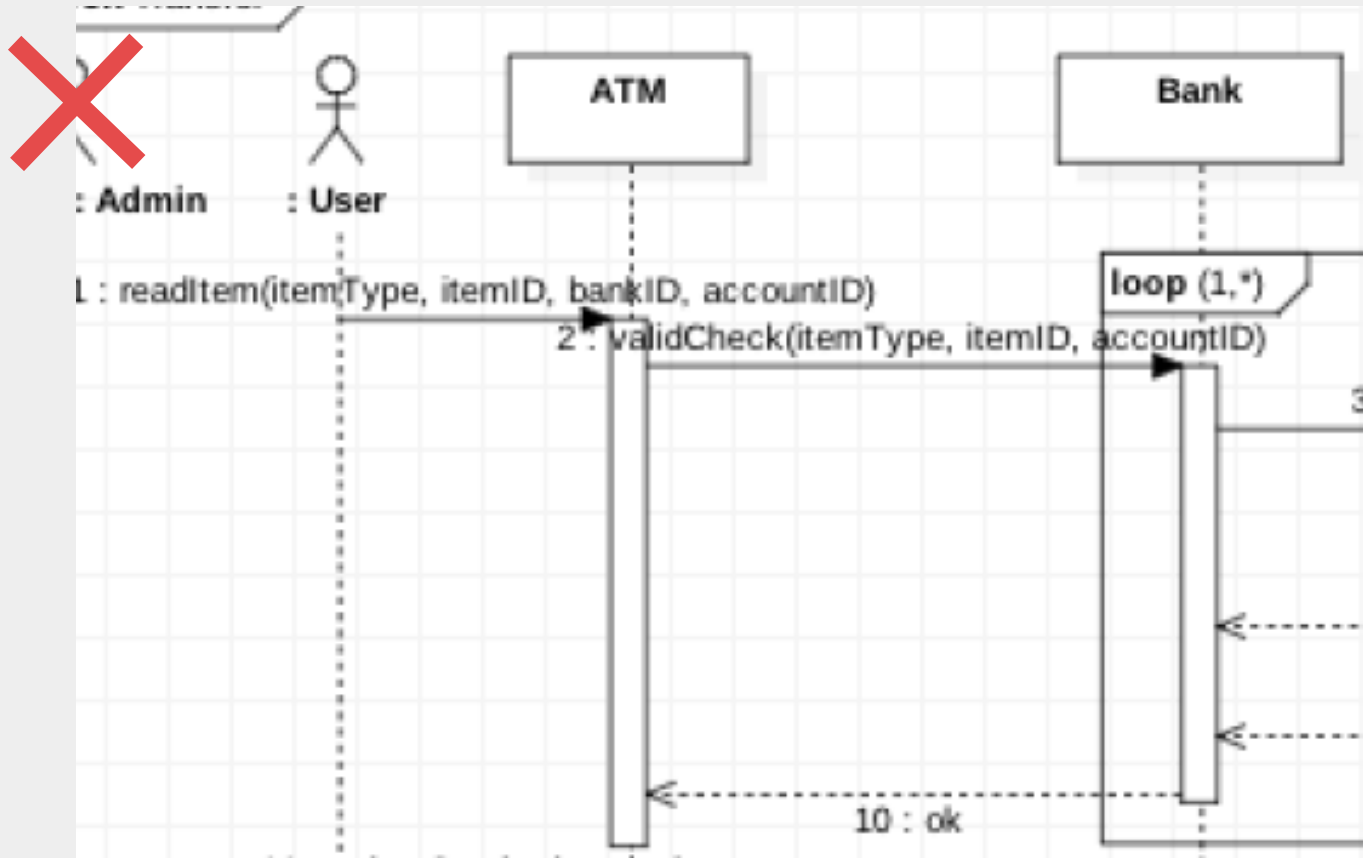


조건문이 잘못되어있다. ex) `wants == true` → `amount > 0`  
getBalance의 인자값이 다르다.  
ATM의 receiptAmount를 검사하는 부분이 명시되어 있지 않다.



## 2144. Define Interaction Diagrams

## 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.

## 2144. Define Interaction Diagrams

## 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.

setDataRange 함수의 생명선은 agreement의 종료시까지 존재해야 한다.

## 2144. Define Interaction Diagrams

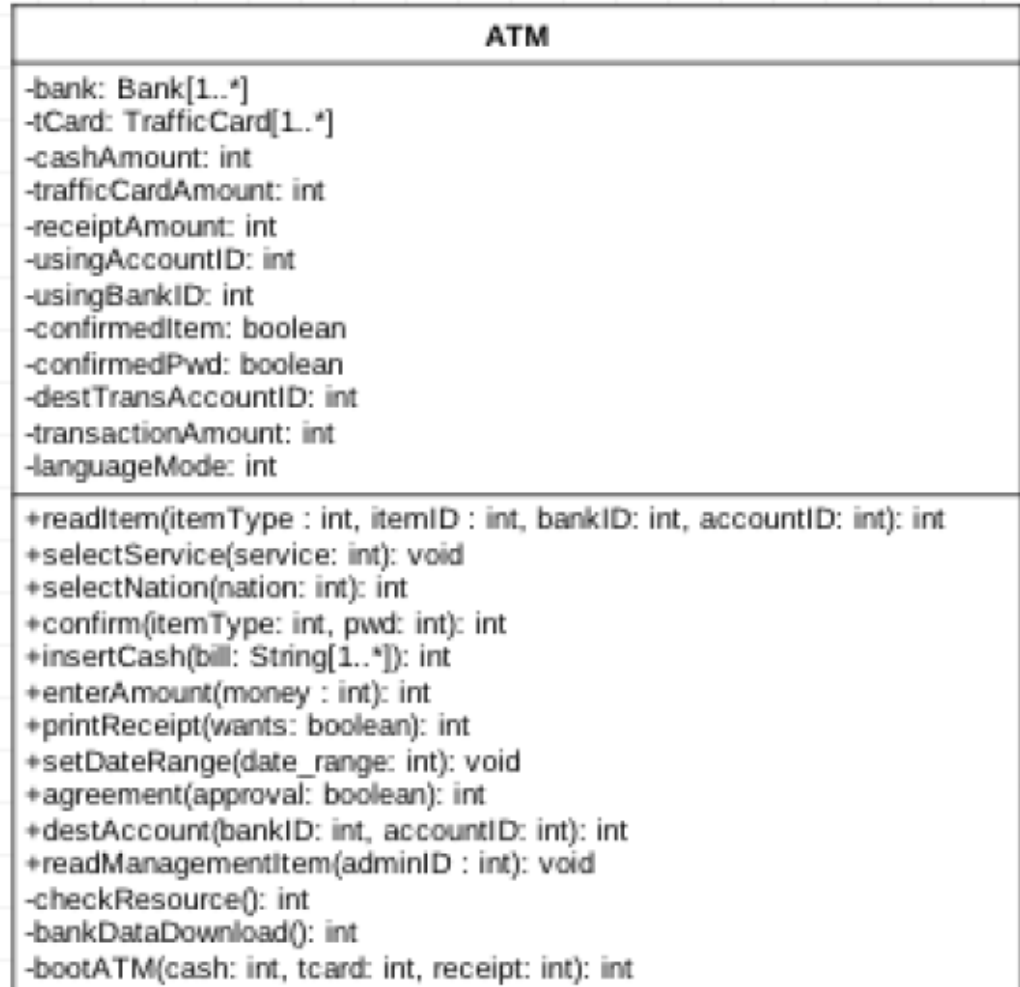
### 4.6 Management

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 Class에서도 모두 같은 문제가 발견됨.

## 2046. Design Traceability Analysis

실제로 구현되지않은 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.

## 2051. Implement Class &amp; Method

모든 Class와 Method에서 언급된 Use-case Status Alarm과 Management 는 프로그램상에서 구현되지 않았으므로 모두 빠져야 한다.

모든 Method의 Input과 Output을 작성하지 않아서 프로그램과의 비교가 불가능하다.

이전 단계에서의 설계와는 다르게 대부분의 기능과 로직을 GUIController라는 Class에서 구현하였는데, 이에 대한 설명이 하나도 언급되어있지 않다.

## 2051. Implement Class &amp; Method

함수의 이름이 프로그램과 다르다.

getBookPWD → getBpwd  
 getCardPwd → getCpwd  
 getBalance → get\_balance  
 getItemID → get\_aid  
 setBalance → set\_balance  
 setDateRange → inputDataRange  
 외 다수

함수의 인자값이 프로그램과 다르다.

confirm(itemType, pwd, accountID)  
 → confirm(int \_pwd)

외 다수

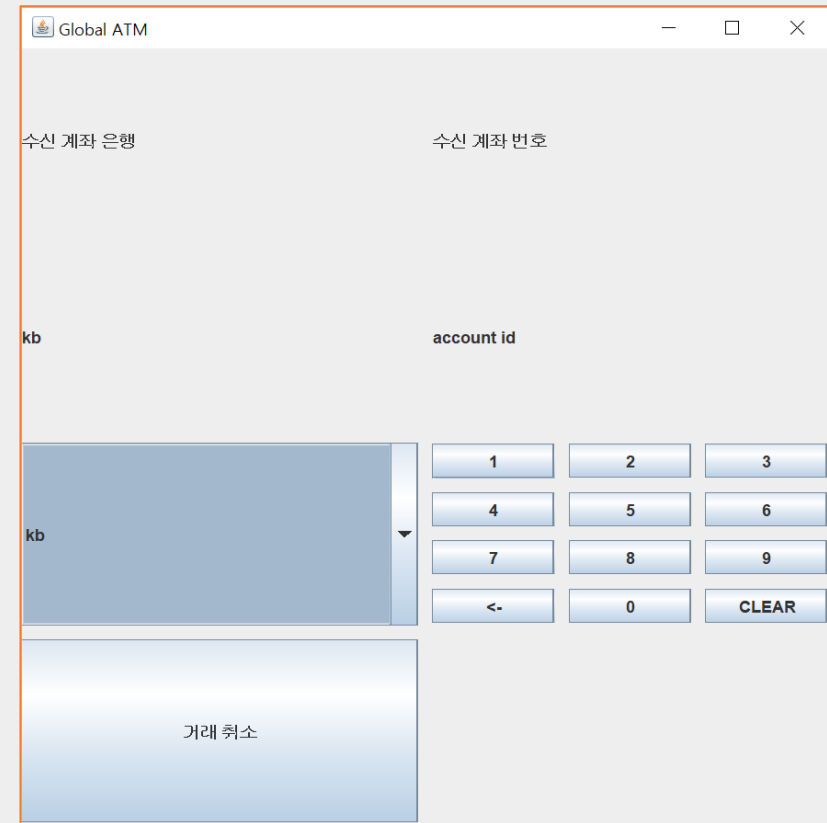
PreConditions	송금 서비스 선택
PostConditions	송금 대상 계좌 정보 이법

오타

## 2052. Implement Windows



VS



UI 프로토타입과 비교했을 때 화면 개수, 구성, 디자인, 흐름이 완전히 다름.



## 2063. System Testing

## 4. Activity 2063. System Testing

Test Number	Test 항목	Description	Use Case	System Function	P/F
1-1	Read Item Test	사용가능한 Card을 삽입 했을 때 서비스 선택창으로 전환 되는지 test	-		
1-2	Read Item Test	사용 불가능한 Card을 삽입 했을 때, 다시 item삽입하라는 창으로 나오는지 확인	-		
1-3	Read Item Test	사용가능한 Book을 삽입 했을 때 서비스 선택창으로 전환 되는지 test	-		
1-4	Read Item Test	사용 불가능한 Book을 삽입 했을 때, 다시 item삽입하라는 창으로 나오는지 확인	-		
2-1	서비스 선택 test	계좌조회 서비스 선택했을 때, 비밀번호 입력 창으로 전환되는지 test	Check	R.1.0	
2-2	서비스 선택 test	입금 서비스 선택했을 때, 금액 입력 안내와 금액을 투입하라는 메시지 창으로 전환되는지 test	Deposit	R.1.2	

소모팀에서 테스트를 전혀 진행하지 않음.

또한 관리자 기능의 경우 존재하지 않기 때문에 테스트 불가

# 02. Category-Partition Testing

## 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

## Categorize

<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

## 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.

## Description

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.

## Property &amp; Const

```

# 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]

```

## Property &amp; Const

```

# 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 TCAlarm]
    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 TCAlarm || ReceiptAlarm || NMoneyAlarm || TMoneyAlarm]
    Not Send Alarm,          [error]

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

```



## Category Partition Testing

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

0% Passed

## Category Partition Testing

0% Passed

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

```

java.io.FileNotFoundException: TrafficCard.txt (지정된 파일을 찾을 수 없습니다)
  at java.base/java.io.FileInputStream.open0(Native Method)
  at java.base/java.io.FileInputStream.open(Unknown Source)
  at java.base/java.io.FileInputStream.<init>(Unknown Source)
  at java.base/java.io.FileReader.<init>(Unknown Source)
  at Item.TrafficCard.<init>(TrafficCard.java:43)
  at ATM.ATM.<init>(ATM.java:16)
  at GUI.GUIController.readyReadCard(GUIController.java:108)
  at GUI.GUIController.<init>(GUIController.java:101)
  at GUI.GUIController.main(GUIController.java:94)
java.io.FileNotFoundException: management.txt (지정된 파일을 찾을 수 없습니다)
  at java.base/java.io.FileInputStream.open0(Native Method)
  at java.base/java.io.FileInputStream.open(Unknown Source)
  at java.base/java.io.FileInputStream.<init>(Unknown Source)
  at java.base/java.io.FileReader.<init>(Unknown Source)
  at ATM.ATM.<init>(ATM.java:41)
  at GUI.GUIController.readyReadCard(GUIController.java:108)
  at GUI.GUIController.<init>(GUIController.java:101)
  at GUI.GUIController.main(GUIController.java:94)

```

```

Exception in thread "AWT-EventQueue-0" java.lang.ArrayIndexOutOfBoundsException: 3
  at BankSystem.Bank.loadData(Bank.java:58)
  at BankSystem.Bank.<init>(Bank.java:35)
  at ATM.ATM.readItem(ATM.java:74)
  at GUI.GUIController$1.actionPerformed(GUIController.java:240)
  at java.desktop/javax.swing.AbstractButton.fireActionPerformed(Unknown Source)
  at java.desktop/javax.swing.AbstractButton$Handler.actionPerformed(Unknown Source)
  at java.desktop/javax.swing.DefaultButtonModel.fireActionPerformed(Unknown Source)
  at java.desktop/javax.swing.DefaultButtonModel.setPressed(Unknown Source)
  at java.desktop/javax.swing.plaf.basic.BasicButtonListener.mouseReleased(Unknown Source)
  at java.desktop/java.awt.Component.processMouseEvent(Unknown Source)
  at java.desktop/javax.swing.JComponent.processMouseEvent(Unknown Source)
  at java.desktop/java.awt.Component.processEvent(Unknown Source)
  at java.desktop/java.awt.Container.processEvent(Unknown Source)
  at java.desktop/java.awt.Component.dispatchEventImpl(Unknown Source)
  at java.desktop/java.awt.Container.dispatchEventImpl(Unknown Source)
  at java.desktop/java.awt.Component.dispatchEvent(Unknown Source)
  at java.desktop/java.awt.LightweightDispatcher.retargetMouseEvent(Unknown Source)
  at java.desktop/java.awt.LightweightDispatcher.processMouseEvent(Unknown Source)
  at java.desktop/java.awt.LightweightDispatcher.dispatchEvent(Unknown Source)
  at java.desktop/java.awt.Container.dispatchEventImpl(Unknown Source)

```

SV팀에서 Error를 처리하여 Testing 진행

42% Passed

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 결제 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- 100.201.400.600.610.700.810.822.900	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	

## Category Partition Testing

42% 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%

- File 경로, Parsing 中 Array Index Out of Bounds와 같은 치명적인 Error 수정 실패.
- Admin에게 알림을 주지 않음
- 몇가지 예외처리가 잘 되어있지 않음

# 03. Pairwise Testing

- 파라미터의 모든 조합을 테스트 → TMI !!
- 대부분의 테스트케이스는 pair 쌍만 서로 다르면 구분 가능
- 약간의 정확도 희생 → 테스트케이스 수 대폭 감소 가능



## PICT

```
C:\>pict  
Pairwise Independent Combinatorial Testing
```

- pairwise testcase를 자동으로 뽑아줄 수 있는 도구

# 선택 이유

- 자세한 공식 문서 및 간단한 사용법

**PICT 3.3 User's Guide**  
Jacek Czerwonka, Test Lead, Microsoft Corporation

Updated: February 4, 2009 for the Web interface

[Overview](#)  
[Using PICT to Combine Test Case Parameters](#)  
[Running PICT](#)  
[Model Files](#)  
[Model Sections](#)  
[Simple Model](#)  
[Pairwise and Higher-Order Generation](#)  
[Sub-Models](#)  
[Model Options](#)  
[Constraints](#)  
[Conditional Constraints](#)  
[Unconditional Constraints \(Invariants\)](#)  
[Parameter Types](#)  
[Aliasing](#)  
[Negative Testing](#)  
[Weighting](#)  
[Seeding](#)  
[Output Randomization](#)  
[Minimizing the Number of Test Cases](#)  
[Case Sensitivity](#)  
[Output Format](#)  
[Warning Messages](#)  
[All or no values satisfy relation...](#)  
[Restrictive constraints. Output will not contain following values...](#)  
[Sample Models](#)  
[Complete Model File for the Volume Partitioning Example](#)  
[Model to Test Hardware Configurations](#)  
[Constraints Syntax Examples](#)

## 스크립트 파일

### 1. 카테고리를 나열한다.

```
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
```

## 스크립트 파일

### 2. 카테고리간 의존성을 표현한다.

```
#  
# 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";
```

## 결과 파일

ReadItem	Mode	InputPassword	InputBank	InputAccountID	InputDuration	InputCurrency	InputMoney	Agreement	TCArm	ActorTransaction	AdminTransaction			
Valid	Check	Valid	Skip	Skip	Skip	Skip	Skip	NotPrint	Send					
Valid	Check	Valid	Skip	Skip	Skip	Skip	Skip	Print	Send					
Valid	Deposit	Skip	Skip	Skip	Skip	Skip	Skip	IncreaseBalance	IncreaseBalance	Send	Print	Send		
Valid	Deposit	Valid	Skip	Skip	Skip	Skip	Skip	IncreaseBalance	IncreaseBalance	Send	NotPrint	Send		
Valid	Deposit	Valid	Skip	Skip	Skip	Skip	Skip	IncreaseBalance	IncreaseBalance	Send	Print	Send		
Valid	Deposit	Skip	Skip	Skip	Skip	Skip	Skip	IncreaseBalance	IncreaseBalance	Send	NotPrint	Send		
Valid	Deposit	Skip	Skip	Skip	Skip	Skip	Skip	IncreaseBalance	IncreaseBalance	Send	NotPrint	Send		
Valid	Deposit	Valid	Skip	Skip	Skip	Skip	Skip	IncreaseBalance	IncreaseBalance	Send	Print	Send		
Valid	Withdraw	Valid	Skip	Skip	Skip	Skip	Skip	DecreaseBalance	DecreaseBalance	Send	NotPrint	Send		
Valid	Withdraw	Valid	Skip	Skip	Skip	Skip	Skip	DecreaseBalance	DecreaseBalance	Send	Print	Send		
Valid	Transfer	Valid	Select	Valid	Skip	Skip	ValidInput	Agree	Skip	SendMoney	DecreaseBalance	Send	NotPrint	Send
Valid	Transfer	Valid	Select	Valid	Skip	Skip	ValidInput	Disagree	Skip	SendMoney	DecreaseBalance	Send	NotPrint	Send
Valid	Transfer	Valid	Select	Valid	Skip	Skip	ValidInput	Agree	Skip	SendMoney	DecreaseBalance	Send	Print	Send
Valid	Transfer	Valid	Select	Valid	Skip	Skip	ValidInput	Disagree	Skip	SendMoney	DecreaseBalance	Send	Print	Send
Valid	IssueTC	Valid	Skip	Skip	Valid	Skip	Skip	Skip	Send	DecreaseBalance	IncreaseBalance	Send	Print	Send
Valid	IssueTC	Valid	Skip	Skip	Valid	Skip	Skip	Skip	Send	DecreaseBalance	IncreaseBalance	Send	NotPrint	Send

03

## Test Cases & Result

CPT와 마찬가지로 프로그램 시작 부분에 에러.

→ 수정 전 테스트 결과: **All Fail**

## Test Cases &amp; Result (코드 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	Passed	
14	100.202.301	Passed	
15	100.203.300.400.500.800.900.1102.1201.1300.1401.1500	Failed	대부분의 계좌가 조회되지 않음.

## Test Cases & Result (코드 수정 후)

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	

→ 8 / 25 (32%) Pass



## Pairwise Testing Summary

- 파일 입출력 과정에서의 경로 / 자료구조 문제 심각.
- alarm 기능 작동하지 않음.
- 예외처리 미흡.
- 약간의 UI 오류.

# 04. Brute-force Testing

## 04

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

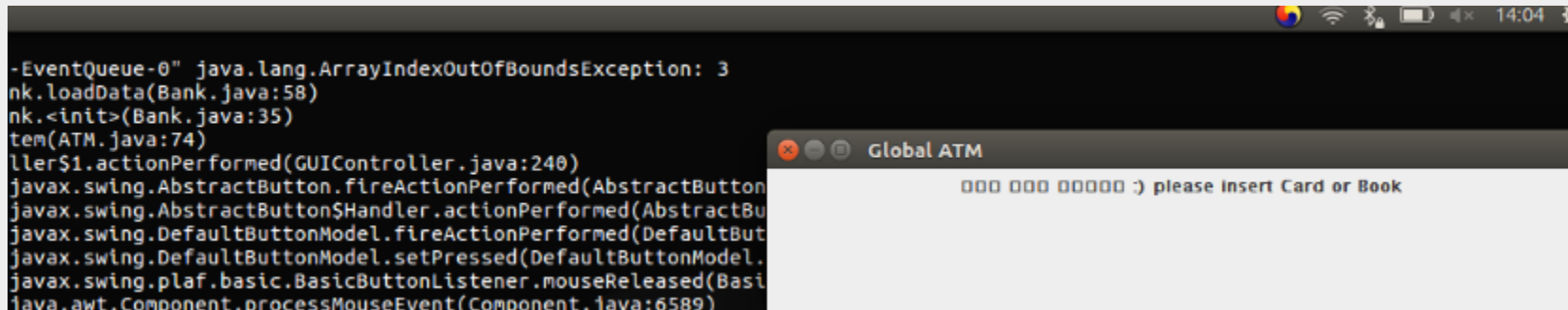
0 / 22 = 0% Passed

## 04

Why?

```
Exception in thread "AWT-EventQueue-0" java.lang.ArrayIndexOutOfBoundsException: 3
    at BankSystem.Bank.loadData(Bank.java:58)
    at BankSystem.Bank.<init>(Bank.java:35)
    at ATM.ATM.readItem(ATM.java:74)
    at GUI.GUIController$1.actionPerformed(GUIController.java:240)
    at
```

소모팀의 구현상의 문제  
: 카드와 통장을 관리하는 txt 파일의 경로오류



기능 실행 불가능!

테스팅을 해 보기 위해 경로관련 부분만 수정 후 진행

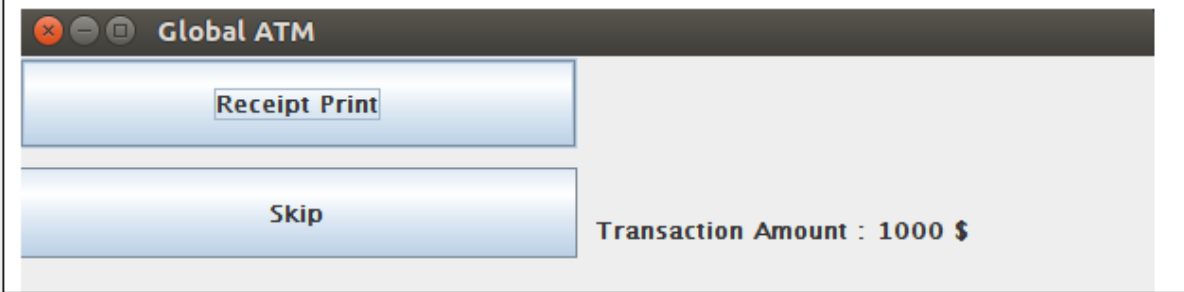
Test	Test Num	Description	P/F
아이템 받아오기	1-1	존재하지 않는 계좌에 접근하려고 시도하면 중지되는지 확인한다.	Pass
	1-2	존재하지 않는 카드에 접근하려고 시도하면 중지되는지 확인한다.	Pass
	1-3	카드번호에 문자를 넣으면 중지되는지 확인한다.	Fail
	1-4	계좌번호에 문자를 넣으면 중지되는지 확인한다.	Fail
	1-5	아이템 종류에 문자를 넣으면 중지되는지 확인한다.	Fail
입금	2-1	현재 ATM기기가 현금을 가득 보유하고 있을 때 1000원의 입금을 시도하면 중지되는지 확인한다.	Fail
	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

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. → java.lang.NumberFormatException와 함께 프로그램이 강제종료된다.	Fail
	1-5	아이템 종류에 문자를 넣으면 중지되는지 확인한다. → java.lang.NumberFormatException와 함께 프로그램이 강제종료된다.	Fail
입금	2-1	현재 ATM기기가 현금을 가득 보유하고 있을 때 1000원의 입금을 시도하면 중지되는지 확인한다. → 중지되지 않고 입금이 완료된다.	Fail

```
Exception in thread "AWT-EventQueue-0" java.lang.ArrayIndexOutOfBoundsException: 3
    at BankSystem.Bank.loadData(Bank.java:58)
    at BankSystem.Bank.<init>(Bank.java:35)
    at BankSystem.Bank.checkAccount(Bank.java:170)
    at ATM.ATM.destAccount(ATM.java:359)
```

출금	3-1	현재 ATM기기가 보유하고 있는 50000원권의 수량이 0일 때 50000원을 출금하려고 시도하면 출금되는지 확인한다. → 출금이 되지 않는다.	Fail
	3-3	50달러를 정상적으로 출금하는 지 확인한다. → 500달러로 출금된다.	Fail
	3-4	5000만원을 정상적으로 출금하는 지 확인한다. → 세 자리 숫자만 입력 가능하다.	Fail
	4-2	50달러를 정상적으로 이체하는 지 확인한다. → 500달러로 이체된다.	Fail
	4-3	5000만원을 정상적으로 출금하는 지 확인한다. → 세 자리 숫자만 입력 가능하다.	Fail
알림	5-1	명세표 용지가 남아 있지 않을 때 명세표를 출력하면 알림이 뜨는 지 확인한다. → 알림이 뜨지 않고 출력된다.	Fail
	5-2	ATM기기의 현재 1000원 보유량이 최대치일 때 1000원의 입금을 시도하면 알림이 뜨는 지 확인한다. → 알림이 뜨지 않고 입금된다.	Fail
	5-3	ATM기기의 현재 10\$ 보유량이 최대치일 때 10\$의 입금을 시도하면 알림이 뜨는 지 확인한다. → 알림이 뜨지 않고 입금된다.	Fail
	5-4	ATM기기가 1000원을 보유하고 있지 않을 때 23000원의 출금을 시도하면 알림이 뜨는 지 확인한다. → 알림이 뜨지 않고 출금된다.	Fail
교통카드	6-1	교통카드 생성 시 0.3달러가 출금되는 지 확인한다. → 3달러가 출금된다.	Fail
	6-2	현재 ATM기기가 교통카드 여분을 보유하고 있지 않을 때 교통카드 생성 시 알림이 뜨고 중지되는지 확인한다. → 알림이 뜨지 않고 교통카드가 생성된다.	Fail

(6) 기준 금액의 10배가 출금/송금 됨(ex. 100달러를 출금/송금할 시 1,000달러가 출금/송금됨).



▲ 4-2. 외화를 출금 시 10배의 금액이 출금된다.

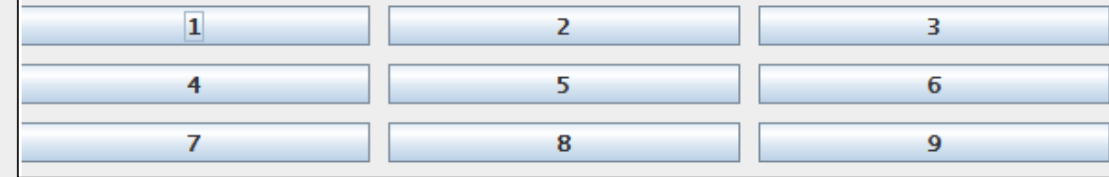
Transaction Amount : 3 \$

▲ 6-1.

교통카드의 발급비용은 0.3\$라고 명시되어 있지만 3\$가 출금된다.

How much do you want?

0 \$



▲ 4-3. 세 자리의 숫자만 출금이 가능하다.



# 05. Summary

**Category-Partition Test** →  $27 / 64 = 42\%$  Pass

**Pairwise Test** →  $8 / 25 = 32\%$  Pass

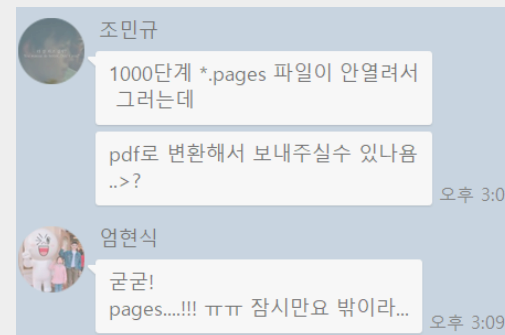
**Brute-force Test** →  $7 / 22 = 31.8\%$  Pass

- ✓ Text File의 경로가 잘못되어 실행이 되지 않는 Error 발생
- ✓ Text File을 Parsing하는 과정에서 Array Index Out of Bounds Error 발생
- ✓ 관리자에게 Alarm을 보내야 하는 모든 경우 실패
  1. ATM에 돈이 부족한 경우
  2. ATM에 돈이 너무 많은 경우
  3. 영수증이 부족한 경우
  4. 교통카드가 부족한 경우
- ✓ User Interface 오류
  1. 입금시 금액 출력 오류 (50000원 -> 5000원)
  2. 출금시 금액 입력 화면에 확인 버튼이 없음

- ✓ ATM의 핵심기능인 입금과 출금이 작동하지 않는다.
- ✓ 여러 경우에 대해 정확한 예외처리가 되어있지 않다.
- ✓ 위와 같은 상황 등 Error가 많으므로 수정 필요

- ✓ 문서들의 단계마다 일치하지 않는 부분들이 매우 많다.
- ✓ 문서와 구현된 코드 간의 상당한 차이가 존재한다.
- ✓ 용어가 통일되지 않았다.
- ✓ 객체지향적인 설계를 하지 않았다.
- ✓ 외화로 출금, 이체를 할 경우 자릿수가 바뀌어 연산되는 버그가 발생한다.
- ✓ 돈 관련 거래를 하여도 파일에 Write하지 않는다.
- ✓ Functional Requirement를 만족시키지 못하였다.
- ✓ Non Functional Requirement를 만족시키지 못하였다.
- ✓ 여러 경우에 대해 정확한 예외처리가 되어있지 않다.
- ✓ 위의 경우 외에도 다양한 문제들이 존재한다.

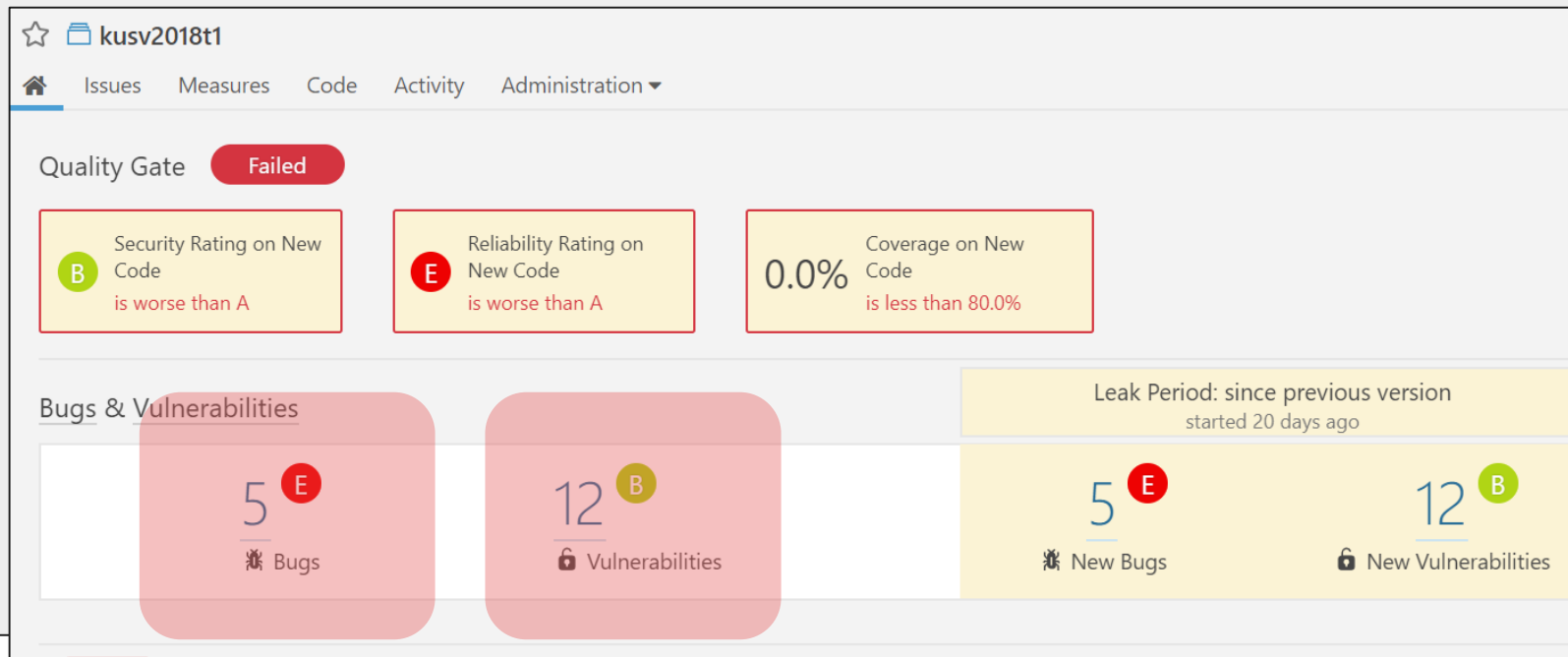
- 지속적인 소통을 위한 노력
  - 매뉴얼 제작 등 요구 즉각 반영
  - 진행상황에 대한 피드백



- 운영체제 등 HELL 과목들의 과제 폭탄에도 불구하고 성실히 프로그램 제작
  - 팀원 3명 모두 과제가 어렵기로 악명 높은 진교수님의 운영체제 과목 수강
- 비록 완성되지 않은 부분이 많지만 현실의 ATM을 반영하기 위해 노력
  - 사소한 입력도 keyboard input이 아닌 GUI로 구현
  - 지폐 잔량 검사를 위해 돈을 배열 형태로 표현

```
int cash = Integer.parseInt(getStr);
cashAmount[0] = cash;
cashAmount[1] = cash;
cashAmount[2] = cash;
cashAmount[3] = cash;
```

# 06. CTIP Usage



☆ kusv2018t1

Issues Measures Code Activity Administration

Quality Gate **Failed**

- Security Rating on New Code: **B** is worse than A
- Reliability Rating on New Code: **E** is worse than A
- Coverage on New Code: **0.0%** is less than 80.0%

Bugs & Vulnerabilities

Leak Period: since previous version started 20 days ago

Category	Count	Grade
Bugs	5	E
Vulnerabilities	12	B
New Bugs	5	E
New Vulnerabilities	12	B

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

Build Case 별 정적분석 결과를 확인가능



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;
}
```

Build Case 별 정적분석 결과를 확인가능

#21 opened 22 minutes ago by kusv2018t1 System Test	① [Category Partitioning Test] 100.201.400.600.610.700.810.821.900 bug
① [BruteForceTest] 교통카드 생성 시 출금 금액을 확인한다. bug #20 opened 22 minutes ago by kusv2018t1 System Test	① [Category Partitioning Test] 100.201.400.600.610.701.822.900 bug
① [BruteForceTest] ATM기기가 1000원을 보유하고 있지 않을 때 23000원의 출금을 시도한다. bug #19 opened 22 minutes ago by kusv2018t1 System Test	① [Category Partitioning Test] 100.201.400.600.610.701.822.900 bug
① [BruteForceTest] ATM기기가 10\$를 최대로 보유하고 있을 때 10\$의 입금을 시도한다. bug #18 opened 23 minutes ago by kusv2018t1 System Test	① [Category Partitioning Test] 100.201.400.600.610.701.821 bug
① [BruteForceTest] ATM기기가 10000원을 최대로 보유하고 있을 때 10000원의 입금을 시도한다. bug #17 opened 23 minutes ago by kusv2018t1 System Test	① [Category Partitioning Test] 100.201.400.600.610.700.811.822.900 bug
① [BruteForceTest] 명세표 용지가 남아있지 않을 때 명세표를 출력한다. bug #16 opened 24 minutes ago by kusv2018t1 System Test	① [Category Partitioning Test] 100.201.400.600.610.700.811.821 bug
① [BruteForceTest] 5000만원을 이체한다. bug #15 opened 24 minutes ago by kusv2018t1 System Test	① [Category Partitioning Test] 100.201.400.600.610.700.810.822.900 bug
① [BruteForceTest] 50달러를 이체한다. bug #14 opened 25 minutes ago by kusv2018t1 System Test	① [Category Partitioning Test] 100.201.400.600.610.700.810.821.900 bug
① [BruteForceTest] 5000만원을 출금한다. bug #13 opened 25 minutes ago by kusv2018t1 System Test	① [Category Partitioning Test] 100.200.300.701 bug
① [BruteForceTest] 50달러를 출금한다. bug #12 opened 26 minutes ago by kusv2018t1 System Test	① [Category Partitioning Test] 100.200.300.700.811 bug
① [BruteForceTest] ATM기기가 보유하고 있는 50000원권이 없을 때 50000원을 출금한다. bug #11 opened 26 minutes ago by kusv2018t1 System Test	① [Category Partitioning Test] 100.200.300.700.810.900 bug
① [BruteForceTest] ATM기기에 현금이 가득 찼을 때 1000원의 입금을 시도한다. bug #10 opened 26 minutes ago by kusv2018t1 System Test	① [Category Partitioning Test] 1002 bug
	#45 opened 12 minutes ago by kusv2018t1 System Test
	#44 opened 12 minutes ago by kusv2018t1 System Test
	#43 opened 13 minutes ago by kusv2018t1 System Test
	#42 opened 13 minutes ago by kusv2018t1 System Test
	#41 opened 13 minutes ago by kusv2018t1 System Test
	#40 opened 13 minutes ago by kusv2018t1 System Test
	#39 opened 13 minutes ago by kusv2018t1 System Test
	#38 opened 14 minutes ago by kusv2018t1 System Test
	#37 opened 14 minutes ago by kusv2018t1 System Test
	#36 opened 14 minutes ago by kusv2018t1 System Test
	#35 opened 14 minutes ago by kusv2018t1 System Test
	#34 opened 14 minutes ago by kusv2018t1 System Test

Failed된 Test Case를 Github issue Tracker에 등록해 E-mail로 소모팀에게 알림