

Software Verification

System Test Report

1st

Date

2018-05-28

Team 4

201411259 고수창

201411314 전소영

201412005 이세라

201511304 하지윤

1.Step1. Specification Review	3
1.1 Stage 1000 Planning	3
1.2 Stage 2030 Analysis	5
1.3 Stage 2040 Design	9
1.4 Stage 2050 Implementation	15
1.5 Stage 2060 Unit test & System Test	24
2. Category-partition Testing Report	25
2.1 Test Case	25
2.1.1 Testable Units and Representative Values	25
2.1.2 Error Constraints 적용	27
2.1.3 Single Constraints 적용	27
2.1.4 Property Constraints 적용	27
2.2 Testing Result	27
3. Pairwise Testing Report	28
3.1 Testing Case	28
3.2 Testing Result	29
4. Brute Force Testing Report	30
4.1 Testing Result	30
4.2 Failed Case Report	31
5. Overall	32
5.1 System Test Result	32
5.1.1 Category-Partition-Test	32
5.1.2 Pairwise Test	32
5.1.3 Brute Force Test	32
5.1.4 Summary	32

1.Step1. Specification Review

1.1 Stage 1000 Planning

- Activity 1001. Define draft plan

2. Project Objectives :

- 안전한 거래를 할 수 있는 시스템을 개발한다.
 - 어떤 기능을 이용해 안전한 거래를 할 수 있는지에 대한 구체적인 정의가 없다.

3. Functional Requirements

- 4. Check Criminal Account
 - Functional Requirements에는 'Check Criminal Account'로 명시되어 있지만, 이후 문서에는 'Check Criminal History'로 명시되어 있다..

4. Non-functional Requirements

- 1. Safe Transaction
 - Safe에 대한 기준이 제시되어 있지 않다.
- 2. High Performance
 - Performance의 기준이 제시되어 있지 않다.
- 3. OS-Independent
 - OS 독립적으로 실행하는 기준이 제시되어 있지 않다.

- Activity 1003. Define Requirements

1. Functional Requirements

- 1. Deposit
 - '범죄 이력이 있는 계좌에 송금을 못한다'라는 내용이 Motivation에서는 존재하나. Functional Requirements에서 제시되어 있지 않다.
- 3. Transfer
 - '유저에게서 송금할 금액을 받는다'를 '유저에게 송금할 금액을 입력받는다' 등의 명확한 표현으로 수정이 필요하다.

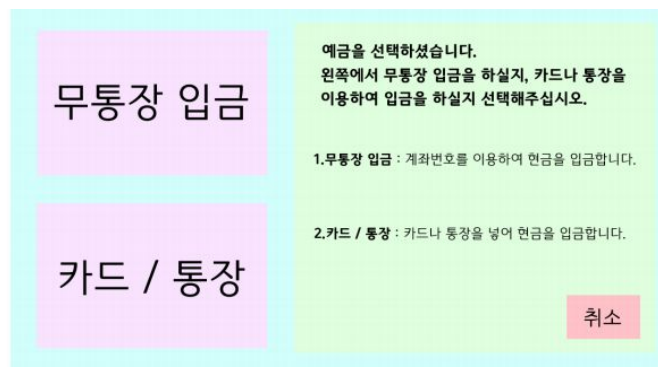
3. Operating Environments

- OS Independent : Windows 7, Windows 10, Mac OS 를 지원한다.
 - Mac OS에 대한 버전이 명시되어있지 않다.

- Java 버전이 명시되어있지 않다.

5. Other Requirements

- 1. Simple한 UI/UX구조를 가진다.
 - Simple 한 UI/UX에 대한 Design Definition 혹은 Guideline이 제시되어 있지 않다.
- Activity 1004. Record Terms in glossary
 - Card
 - 'n'에 대한 범위가 제시되지 않아 무제한으로 생성할 수 있는지 제한적으로 생성할 수 있는지에 대한 정보를 알 수 없다.
 - Bank DB
 - Activity 1003에서 Bank DB를 이용한 유효성 검증을 한다고 나와 있으나, Bank DB가 구체적으로 명시되어있지 않다.
 - Money Limit
 - 'ATM 거래 최대한도'라는 설명이 명확하게 기술되어있지 않다. ATM 전체 거래의 하루 최대한도, 개인당 거래의 최대한도 등의 명확한 기준이 명시되어있지 않다.
- Activity 1005. Implement Prototype



- 프로토타입 이미지
 - 그림 번호 및 상황에 대한 정보가 제시되어 있지 않다.
- Activity 1003 - Deposit에 대해 무통장 입금에 대한 Requirement, Description이 존재하지 않는다.
- Evident로 정의되어 있는 Requirement 'Check Criminal History'에 대한 Prototype Page가 없다.

- Activity 1006. Define Business Use Case

- 3. Actor - Goal List

- 입금, 출금, 송금, 거래내역 조회에 대한 Actor Goal은 제시되어 있으나, Check Criminal History에 대한 Goal이 제시되어 있지 않다.
 - ‘보다 완벽한 프로그램 개발’이라고 명시되어 있으나, 완벽한 프로그램에 대한 기준이 제시되어 있지 않다.

- 8. Describe Use-Cases

- 0. Activity 1003에서 Deposit, Withdraw, Transfer, Check Transaction History, Check Criminal History 모두 Bank DB를 이용하여 유효한지 확인한다고 나와 있으나, Activity 1006에서 명시되어있지 않다.
 - 1. Activity 1003에서 Deposit, Withdraw, Transfer에 대해서 ‘작업을 진행하고 있는 ATM과 계좌의 은행이 다를 경우 수수료를 계산한다’는 Requirement가 있으나 Activity 1006에서 당행/타행 체크 및 수수료 계산에 관련된 Requirement가 명시되어있지 않다.

- Activity 1009. Define System Test Case

- Activity 1003에 ‘계좌의 유효성 여부 체크’에 대한 명시가 되어 있으나 ATM_STC_001, ATM_STC_002, ATM_STC_003, ATM_STC_004, ATM_STC_005에서 명시되어있지 않다.

1.2 Stage 2030 Analysis

- 2031. Define Essential Use Case

- 1. Deposit

- Typical Course of Event
 - 3. 계좌가 유효한지 확인하는 과정에서 Activity 1003의 Requirements에서 BankDB를 이용한다고 나와 있지만, Activity 2031에서 명시되어있지 않다.
 - 8. ‘타행으로 이체하는 경우’ 가 아닌, ‘타행으로 입금하는 경우’로 바뀌어야 한다.

- 2. Withdraw

- Typical Course of Event

- 0. Event의 번호가 누락되었다
- 계좌가 유효한지 확인하는 과정에서 Activity 1003의 Requirements에서 BankDB를 이용한다고 나와 있지만, Activity 2031에서 명시되어있지 않다.

3. Transfer

- Typical Course of Event
 - 3. 계좌가 유효한지 확인하는 과정에서 Activity 1003의 Requirements에서 BankDB를 이용한다고 나와 있지만, Activity 2031에서 명시되어있지 않다.

4. Check Transaction History

- Typical Course of Event
 - 0. Event의 번호가 누락되었다

5. Check Criminal History

- Typical Course of Event
 - 0. Event의 번호가 누락되었다
 - 4. 계좌가 유효한지 확인하는 과정에서 Activity 1003의 Requirements에서 BankDB를 이용한다고 나와 있지만, Activity 2031에서 명시되어있지 않다.

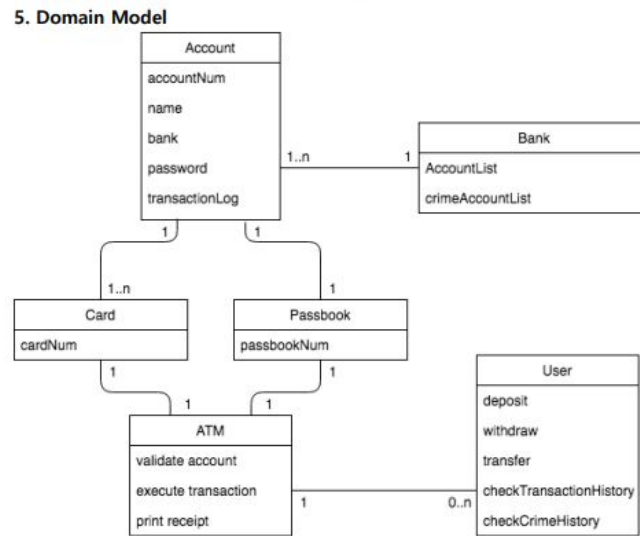
- 2033. Define Domain Model

4. Add Attributes

Account	Bank	User
accountNum name bank password transactionLog	AccountList crimeAccountList	deposit withdraw transfer checkTransactionHistory checkCrimeHistory
Card	Passbook	ATM
cardNum	passbookNum	validate account execute transaction print receipt

- Attribute의 자료형이 명확하게 명시되어 있지 않다.

5. Domain Model



- 변수와 함수들이 명확하게 명시되어 있지 않다.

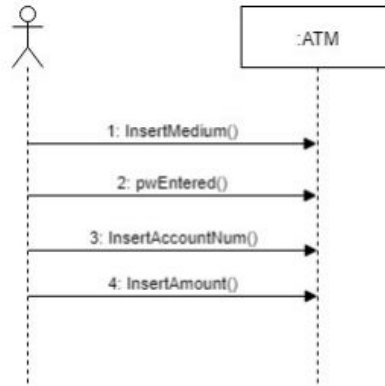
- 2034. Refine Glossary

User.deposit	Attribute	예금 실행
User.withdraw	Attribute	출금 실행
User.transfer	Attribute	거래 실행
User.checkTransactionHistory	Attribute	거래 내역 조회 실행
User.checkCrimeHistory	Attribute	범죄 이력 내역 조회 실행

- Attribute가 아닌, Operation으로 정의되어야 한다.

- 2035. Define System Sequence Diagrams

3. Transfer



- InsertMedium(), pwEntered(), InsertAccountNum(), InsertAmount()
Activity 2036과 명시된 Name of Actor-Activated Event가 다르다.

- 2036. Define Operation Contracts

- InsertMedium()
 - Exceptions
 - 계좌의 유효성 체크 방법이 명시적으로 표시되어 있지 않다.
 - Post-Conditions
 - 계좌 정보에 범죄 내역이 포함되어 있다. 이에 대한 명시가 Activity 1000에서 기술되어 있지 않고, Bank DB에 관한 내용이 명시되어있지 않다.
- EnteredPW()
 - Exceptions
 - 3회 이상 일치하지 않을 경우라는 기준이 새로 생겼다.
 - Post-conditions
 - 'Transfer일 경우 보낼 돈을 입금받는다' -> 'Transfer일 경우 보낼 돈을 입력받는다'
- InsertAccountNum()
 - Pre-Conditions에 '메뉴 선택이 Transfer여야 한다' 라고 이미 명시되어있으나, Post-Conditions 에서 'Transfer 일 경우' 라고 나와있다. 즉, 생략 가능한 부분이 중복되어 언급되어 있다.

- 2037. Define State Diagrams

- System Sequence Diagram에서 명시된 Operation이 존재하지 않고, Activity 2035에서 언급된 Sequence와 일치하지 않는다.

- 2038. Refine System Test Case

- 2031, 2035의 '4. Check Transaction'에서는 비밀번호를 입력받아야 한다고 정의되어있으나, 2038의 'ATM_STC_004'에서는 비밀번호 입력 및 검증에 관한 내용이 빠져있다.
- Activity 1003, 2031, 2036에 '계좌의 유효성 여부 체크'가 명시되어 있으나 ATM_STC_001, ATM_STC_002, ATM_STC_003, ATM_STC_004, ATM_STC_005에서 명시되어있지 않다.

1.3 Stage 2040 Design

- 2041. Design Real Use Cases
 - 2030까지는 통장이나 카드를 넣는 방식이었으나, 2041에서는 통장이나 카드번호를 입력하는 방식으로 변경되었다.
 - GUI에 대한 User interface에 대한 정보가 나와 있지 않다.
 - 1. Deposit
 - 은행 선택하는 과정이 논리적으로 빠져있다.
 - 타행거래에 대한 정보가 1000, 2030에 명시되어있지 않다.
 - 2. Withdraw
 - 은행 선택하는 과정이 논리적으로 빠져있다.
 - 1003에는 존재했던 'ATM 충분한 돈이 있는지 확인하는 단계'가 빠져있다.
 - E2. 잔액이 부족한 경우의 Exception은 정의되어 있으나 확인하는 Typical courses of event 단계에는 명시되어있지 않다.
 - 3. Transfer
 - 은행 선택하는 과정이 빠져있다. 즉, 타행계좌 확인 불가능해 보이는 논리구조를 갖고있다.
 - 2030에는 없었던 송금 후 잔고 확인 과정이 추가되었다.
 - 4. Check Transaction History
 - '18. 세션을 유지한채 메인 메뉴로 돌아간다.'에서 세션이란 용어가 처음으로 등장했다.
- 2042. Define Reports, UI and Storyboards
 - Storyboards 이미지 제목 및 설명이 누락되어있다.
- 2043. Refine System Architecture

- 결과로 Package diagram, Deployment diagram이 나와야 하지만, 패키지만 나와있고 함수나 Public 변수에 대한 정보가 누락되어있다.

- 2044. Define Interaction Diagrams

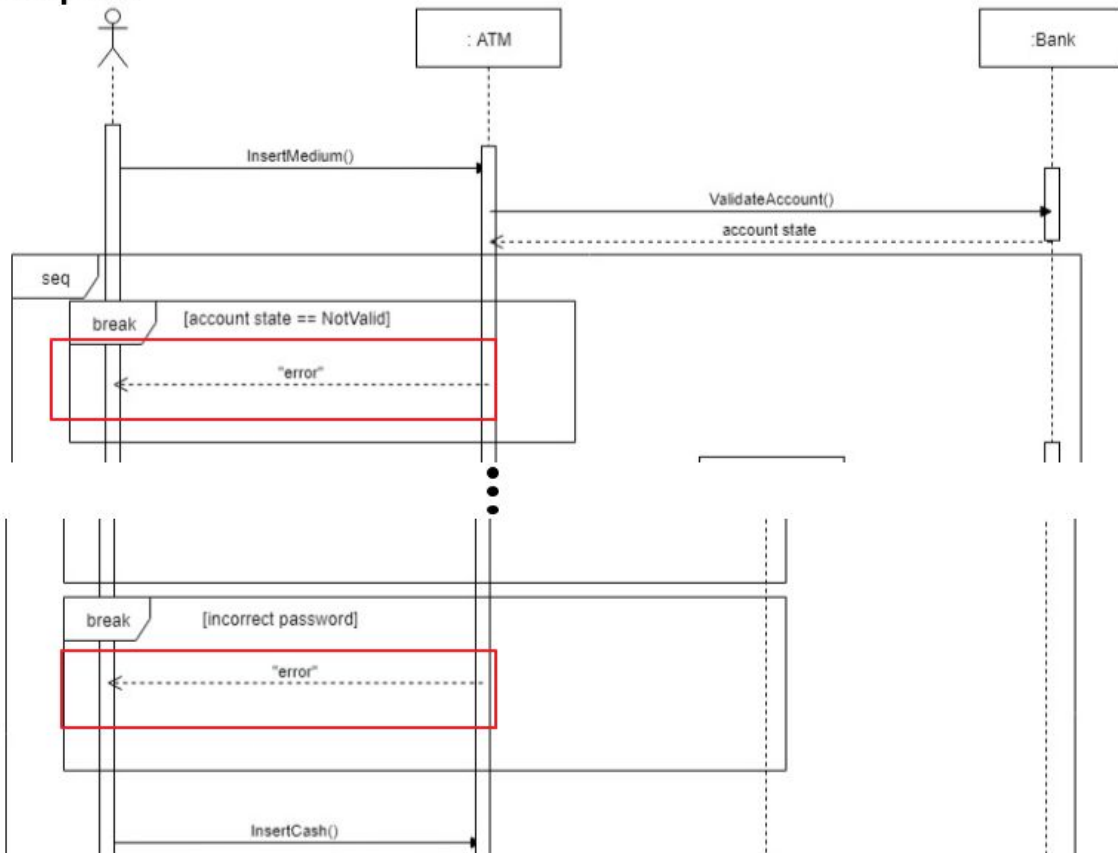
- 전체적으로 2036에 명시된 Name of Actor-Activated Event와 System Operations이 2044의 Sequence diagrams에서 사용하는 것과 상이하다.
- Sequence마다 번호를 붙이지 않아서 순서를 명확히 파악하기 어렵다.

예시 - Deposit

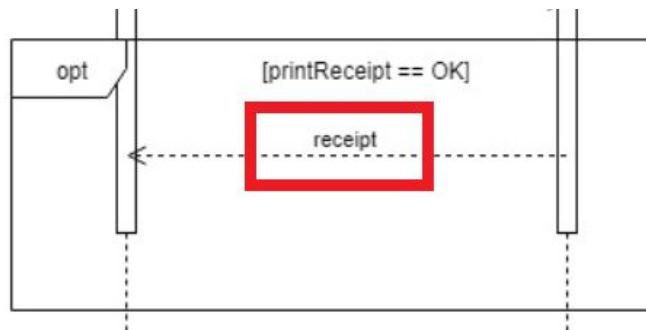
2036. Define Operation Contract	2044. Define Interaction Diagrams
Actor-Activated Event :Insert medium()	InsertMedium()
System Operations :InsertMedium()	ValidateAccount()

- 1. Deposit

1. Deposit



- 1) ValidAccount()
 - 2044에서 새로 등장했다. 2036에 따르면 InsertMedium()에서 계좌 정보를 반환해야한다.
- 2) ValidPW()
 - 2044에서 새로 등장했다. 2036에 따르면 EnteredPW()에서 계좌의 비밀번호가 일치하는지 여부까지 확인해야 한다.
- 3) CheckBank()
 - 2044에서 새로 등장했다.
- 4) CheckBalance()
 - 2044에서 새로 등장했다.
- 5) ExeDeposit()
 - 2044에서 새로 등장했다.
- 6) ShowAskReceptit()
 - 2044에서 새로 등장했다.
- 7) Account state == NotVaild, incorrect password
 - Error후의 화살표가 어디로 가는지 존재하지 않는다.



- 8) receipt를 출력하고 나서 main menu로 돌아가는 화살표가 존재하지 않는다.

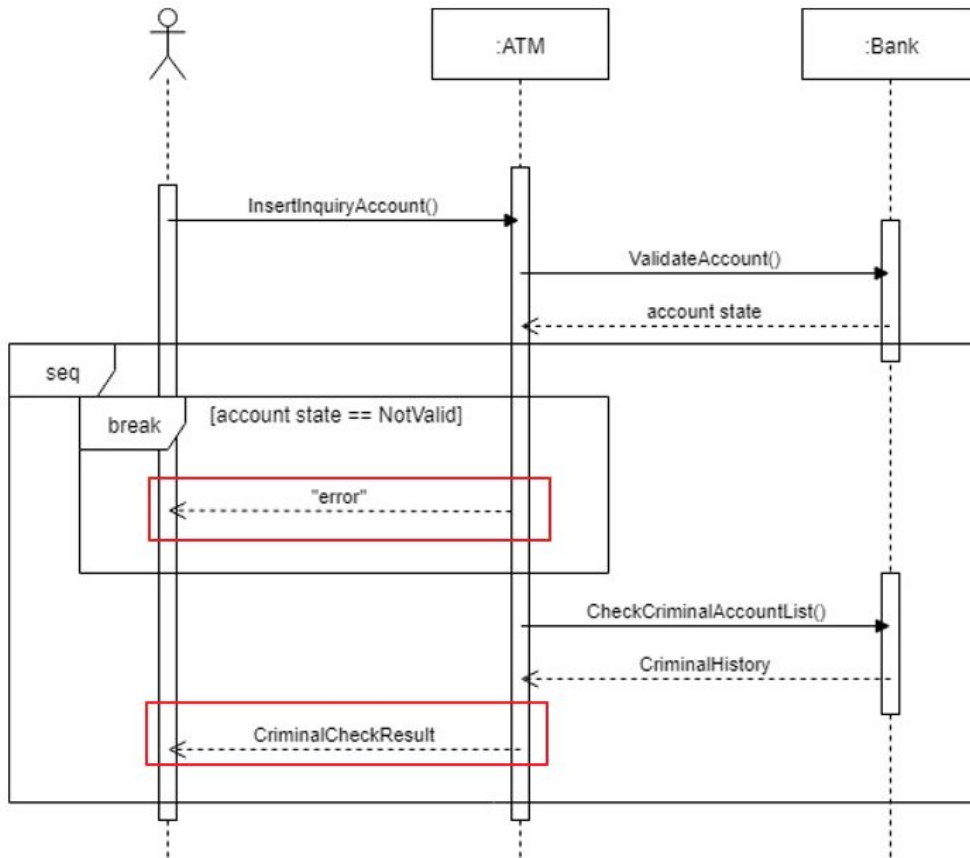
- 2. Withdraw

- 1) ValidAccount()
 - 2044에서 새로 등장했다. 2036에 따르면 InsertMedium()에서 계좌 정보를 반환해야한다.
- 2) ValidPW()
 - 2044에서 새로 등장했다. 2036에 따르면 EnteredPW()에서 계좌의 비밀번호가 일치하는지 여부까지 확인해야 한다.
- 3) CheckBank()
 - 2044에서 새로 등장했다.
- 4) CheckBalance()

- 2044에서 새로 등장했다.
- 5) ExeWithdraw()
 - 2044에서 새로 등장한다.
- 6) ShowAskReceptit()
 - 2044에서 새로 등장했다.
- 7) Account state == NotVaild, incorrect password
 - Error후의 화살표가 어디로 가는지 존재하지 않는다.
- 3. Transfer
 - 1) ValidAccount()
 - 2044에서 새로 등장했다. 2036에 따르면 InsertMedium()에서 계좌 정보를 반환해야한다.
 - 2) ValidPW()
 - 2044에서 새로 등장했다. 2036에 따르면 EnteredPW()에서 계좌의 비밀번호가 일치하는지 여부까지 확인해야 한다.
 - 3) CheckBank()
 - 2044에서 새로 등장했다.
 - 4) CheckBalance()
 - 2044에서 새로 등장했다.
 - 5) ExeWithdraw()
 - 2044에서 새로 등장했다.
 - 6) VaildateTransferAccount()
 - 2044에서 새로 등장했다. 2036에 따르면 InsertAccountNum에서 유효한 계좌임을 확인해야 한 뒤 계좌정보를 반환해야 한다.
 - 7) ExeTransfer()
 - 2044에서 새로 등장했다.
 - 8) UpdateTransferAccount()
 - 2044에서 새로 등장했다.
 - 9) ShowAskReceptit()
 - 2044에서 새로 등장했다.
 - 10) Account state == NotVaild, incorrect password
 - Error후의 화살표가 어디로 가는지 존재하지 않는다.
- 4. CheckTransactionHistory

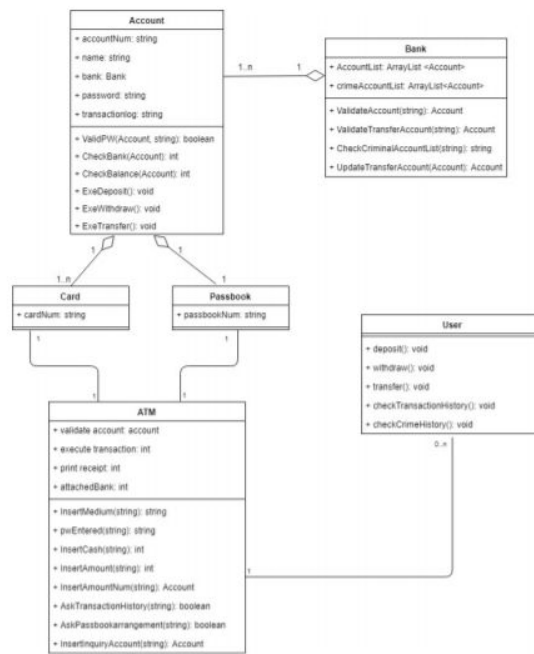
- 1) AskTransactionHistory()
 - 2044에서 새로 등장했다.
- 2) AskPassbookarrangement()
 - 2044에서 새로 등장했다.

- 5. CheckCriminalHistory



- 1) Account state==NotVaild
 - Error이후의 화살표가 어디로 가는지 존재하지 않는다.
- 2) CheckCriminalAccountlist()
 - 2044에서 새로 등장했다.
- 3) CriminalCheckResult이후의 화살표가 어디로 가는지 존재하지 않는다.

● 2045. Define Design Class Diagrams



- User Class가 2044에 존재하지 않았다.
- User와 Account, Card, Passbook의 관계가 보이지 않는다.
- 2044와 비교하여, 2045의 Account에는 createAccount 메소드가 정의되어있지 않고, 2045에 정의된 속성과 메소드들이 2044에서 사용되지 않는 경우가 다수 존재한다. (checkBank, checkBalance, exeDeposit, exeTransfer, exeWithdraw가 2044에서 사용되지 않는다)
- card와 passbook이 각각 atm과 1:1 관계로 정의되어있다. card, passbook:atm이라고 했을 때, n:1 관계로 수정이 필요하다.
- ATM의 validate account는 2044에서 Bank의 메소드로 정의되어있으나, UML에서는 변수(속성)로 정의되어있다. 이에 대한 수정이 필요하다.
- 2045에서 ATM에 InsertCash, InsertAmountNum, InsertAmount, 메소드가 사용되는 것으로 정의되어있으나, 2044의 sequence diagram에는 나와 있지 않다.

- 2046. Design Traceability Analysis

- Operation in sequence diagram의 전체 항목이 잘못되었다. Deposit, Withdraw, Transfer, Check Transaction History, Check Criminal History로 수정이 필요하다.
- GUIMain class에 관한 내용이 이전 Activity에 존재하지 않았으나, Activity 2046에서 등장하였다.

1.4 Stage 2050 Implementation

- 2051. Implement Class & Methods Definition

1) Class definition

- class Account
 - attribute
 - 'Bank'를 'bank'로 수정이 필요하다.
 - attribute 각각의 자료형이 명시되어있지 않아 명시가 필요하다. (String accountNum, Bank bank)
- class AtmSystem
 - attribute
 - 각각의 자료형이 명시되어있지 않아 명시가 필요하다. (int amount, int fee, ArrayList<Bank> fee, ArrayList<Bank> banks, String process)
 - 몇 가지 메소드가 소스 코드 상에서 정의되어있지 않다. 즉, 문서와 소스 코드 간의 불일치가 존재한다.
 - 코드에서의 누락: transaction, getTransaction
 - 문서에서의 누락: getProcess, setProcess, process
 - 오타자: setSrv를 setSrc로 수정해야 한다.
- class Bank
 - attribute
 - attribute 각각의 자료형이 명시되어있지 않아 명시가 필요하다. (String name, String path)
 - 문서에서 getName이 누락되어있다. 즉, 소스 코드와 문서 간의 불일치가 존재한다.
- Etc
 - class Bankbook, class Card, class View, class Terminate에 대한 정의가 되어있지 않다.
 - cross reference가 별도의 테이블로 정리되어있지 않아 어떤 reference인지 참고할 수 없다.

2) Method definition

- class Account
 - Account()

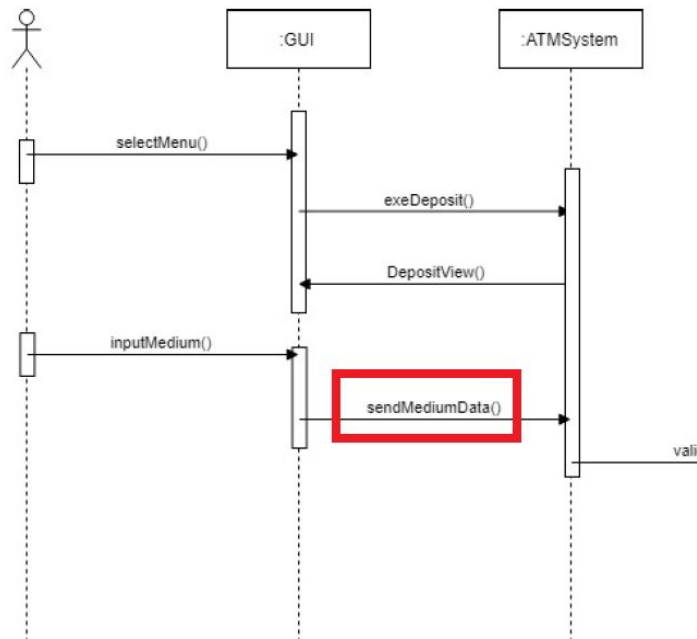
- purpose
 - ‘Account 의 객체 생성한다.’를 수정해야 한다.
Account는 생성자로서 단순히 attribute를 초기화해주는 역할만 수행하며, 별도의 객체를 생성하지 않기 때문이다.
- getBalance()
 - abstract operation
 - ‘bank DB 에서 balance 를 불러와 반환한다.’를 수정해야한다. bankDB라는 표현이 정의되어있지 않아 정확히 무엇을 의미하는지 모호하다.
- checkPassword()
 - abstract operation
 - ‘bankDB 에서 비밀번호 일치 여부를 반환한다.’를 수정해야 한다. bankDB라는 표현이 정의되어있지 않아 정확히 무엇을 의미하는지 알 수 없다.
- transaction()
 - Input
 - Sting msg에서 String msg로 수정해야한다.
- class AtmSystem
 - AtmSystem()
 - purpose
 - ‘AtmSystem 의 객체 생성한다.’를 수정해야 한다.
AtmSystem은 생성자로서 단순히 attribute를 초기화해주는 역할만 수행하며, class View의 객체를 생성하기 때문이다.
 - transaction()
 - 문서상에서 이에 대한 정의는 존재하나, 실제 소스 코드에는 transaction 메소드가 존재하지 않는다.
 - setTransaction()
 - 문서상에서 이에 정의는 존재하나, 실제 소스 코드에는 setTransaction 메소드가 존재하지 않는다.
 - getTransaction()
 - 문서상에서 이에 대한 정의는 존재하나, 실제 소스 코드에는 getTransaction 메소드가 존재하지 않는다.
 - validateSrc()

- abstract operation
 - '유효성'에 대한 정의가 명확하지 않다. (코드상에서는 src를 null과 비교한다)
- findAccount()
 - 'bankDB'라는 표현이 정의되어있지 않아서 정확히 무엇을 의미하는지 알 수 없다.
- withdraw()
 - abstract operation
 - 'amount + fee 만큼 감소'라고 명시되어있으나, 'amount - fee 만큼 감소'라 표현을 고쳐야 한다.
 - 소스 코드의 withdraw 메소드에는 'src.transaction(amount * -1,"출금")'으로 작성되어있다. 'src.transaction(-1 * (amount - fee),"출금")'으로 수정이 필요하다.
- transfer()
 - abstract operation
 - 'amount + fee 만큼 감소'라고 명시되어있으나, 'amount - fee만큼 감소'라고 표현을 고쳐야한다.
 - 소스 코드의 transfer 메소드에는 'src.transaction(amount * -1,"송금")'으로 작성되어있다. 'src.transaction(-1 * (amount-fee),"송금")'으로 수정이 필요하다.
- class Bank
 - Bank()
 - purpose
 - 'Bank 의 객체를 생성한다.'를 수정해야 한다. Bank는 생성자로서 단순히 attribute를 초기화해주는 역할만 수행하며, 별도의 객체를 생성하지 않기 때문이다.
 - validateAccount()
 - 'bankDB'라는 표현이 정의되어있지 않아서 정확히 무엇을 의미하는지 모호하다.
- Etc
 - deposit(), withdraw(), transfer()
 - 'fee'라는 것이 정확히 무엇을 뜻하는지 정의되어있지 않다. 여기서 fee는 수수료를 의미하는 것 같은데, 정확히 수수료를 몇

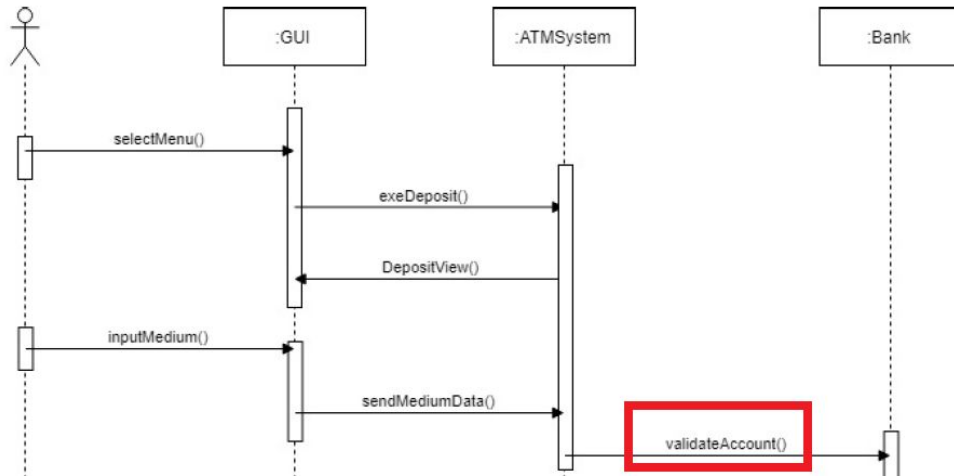
%를 차감하는 것인지 명확하지 않다. (소스 코드에서는 타행계좌인 경우 거래액의 0.01만큼 차감하는 것으로 구현되어있다)

- 2052. Implements Windows

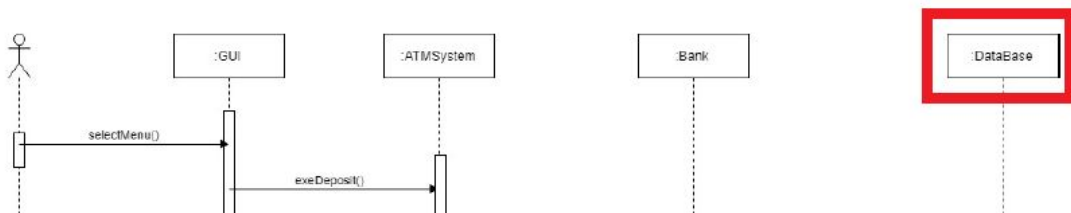
- 프로그램 종료(exit) 및 돌아가기(back)에 대한 정의가 없다. 추가가 필요하다.
- SelectMenu에서 SelectMenuView로 표현을 수정해야 한다.
- 소스 코드와 비교했을때, inputMedium에서 inputAccount로 표현을 수정해야 한다.
- inputPassword()
 - Pre-condition
 - '입금할 돈을 입력받은 상황이어야 한다.'를 '출금 혹은 입금, 이체할 돈을 입력받는 상황이어야 한다'로 변경이 필요하다.
- ResultView()
 - Post-condition
 - '끝내기 혹은 돌아가기 버튼을 누를 수 있다'라고 고쳐서 명시해야 한다.
- Sequence diagram



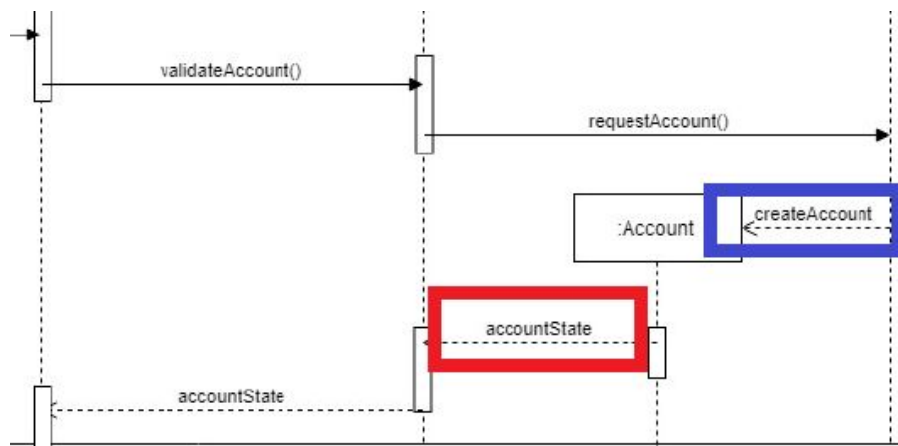
- inputMedium 다음 과정인 sendMedium에 대한 정의가 누락되어 있으며, AtmSystem 코드에 구현되어있지 않다.



- validateAccount라고 정의되어있으나, AtmSystem의 코드에서는 validateSrc 및 validateDst로 구현되어있다.

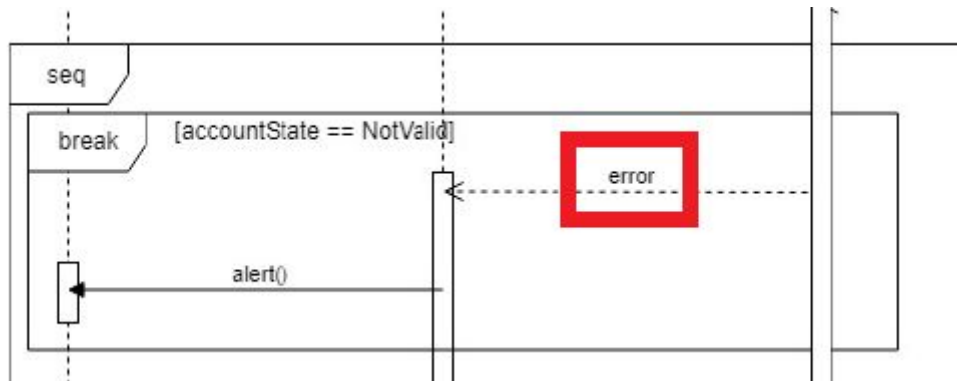


- Database가 정의되어있으나, 실제 코드에서 Database 클래스는 존재하지 않는다. 따라서 requestAccount 메소드 또한 구현되어있지 않다.

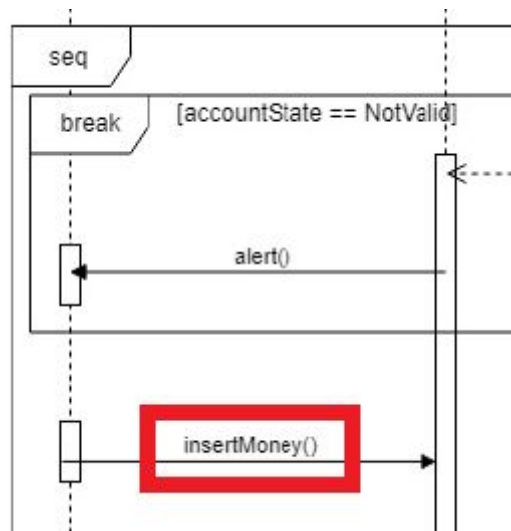


- Account 클래스에 createAccount 반환에 관한 것이 명시되어있으나, 실제 코드에는 구현되어있지 않다. 또한, 문서에서도 관련 정의가 존재하지 않는다.

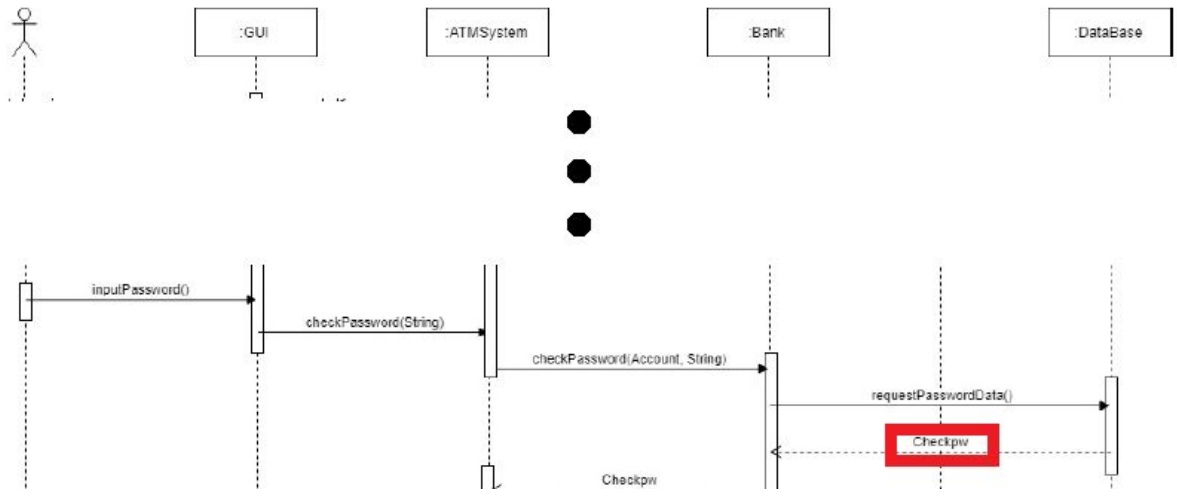
- accountState 반환에 관한 것이 명시되어있지만, 소스 코드 및 문서에 관련 메소드에 대한 정보가 나와 있지 않다.



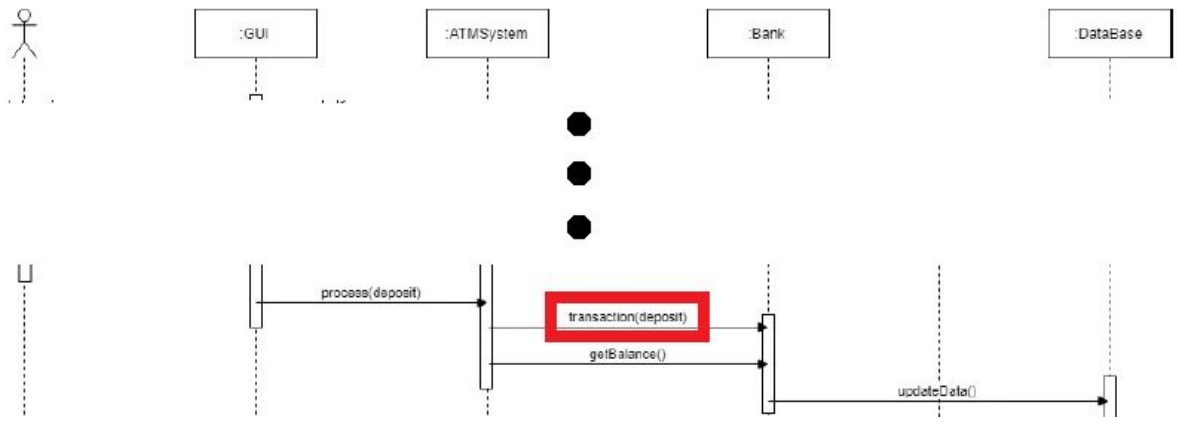
- accountState==notValid한 경우에 AtmSystem에서 error를 반환한다고 명시되어있으나, 소스 코드에서는 error가 아닌 메소드별로 false 혹은 null을 반환하도록 구현되어있다.



- 소스 코드와 비교했을때, insertMoney를 deposit으로 표현을 수정해야 한다.



- 소스 코드와 비교했을때, Database에서 Bank로, Bank에서 AtmSystem으로 반환되는 Checkpw는 존재하지 않는다.



- AtmSystem->Bank로 진행되면서 transaction(deposit)를 호출하게 되어있다. 하지만 실제 소스 코드에서는 AtmSystem->Account의 transaction(deposit)을 호출하며, 이후 Account에서 Bank의 transaction(deposit)을 다시 호출한다. 이와 관련된 수정이 필요하다.

- CheckTransacationHistoryView
 - Post-condition
 - 계좌가 유효한지 확인하고, 계좌 비밀번호가 유효한지 확인한다는 내용이 들어가야 한다.

- 2055. Write Unit Test Code
 - Account 클래스 내의 메소드등을 테스트하는 것이 아니라, FindAccount, Deposit처럼 특정 기능에 대한 클래스를 별도로 만들어서 테스트한다는 점이 부자연스럽다.
 - ActivateTest
 - 무엇을 테스트하는지 불분명하고 실제 프로젝트의 소스 코드를 테스트하는 것이 아니다.
 - SelectMenuTest
 - 테스트 코드에서는 각각의 transaction 작업을 switch~case의 case로 나누어 테스트를 진행하지만, 실제 프로그램 코드에서는 숫자로 나누어진 케이스에 따라서 transaction 작업을 수행하지 않는다. 즉, 테스트코드의 동작이 실제 동작과 일치하지 않는다.
 - InputMediumTest
 - InputMediumTest를 한다고 명시되어있으나, FindAccount 테스트 코드가 명시되어있다.
 - InputAccountTest
 - 테스트 코드를 살펴보면 실질적으로는 FindAccount 테스트를 수행하는 것과 같다.
 - Deposit
 - Deposit 클래스 안에 inputMedium, inputPassword, insertCash 메소드를 사용하는 것으로 나와 있으나, 실제 프로젝트의 소스 코드에서는 존재하지 않는 메소드이다. 어떤 코드를 기준으로 테스트 코드를 작성했는지 모호하다.
 - withdraw()
 - inputMedium, inputPassword, insertCash 메소드를 사용하는 것으로 나와 있으나, 실제 프로그램 코드에서는 존재하지 않는 것들이다.
 - transfer()
 - inputMedium, inputPassword, insertCash 메소드를 사용하는 것으로 나와 있으나, 실제 프로그램 코드에서는 존재하지 않는 것들이다.
 - CheckTransactionHistory, CheckCriminalHistory
 - 실질적으로 inputMedium에 대한 검사만을 수행하며, 실제 Check Transaction History 기능 및 Check Criminal History 기능의 정상 동작에 대한 검사를 수행하지 않는다.
 - _FileReader

- 어떠한 객체 생성도 없이 단순히 println만을 사용하고 있다. 무엇을 테스트한다는 것인지 명확하지 않다.
- ValidateAccount
 - validateAccount()
 - 'boolean exist'를 매개변수로 받아서 Account의 유효성을 검사하지만, 실제 프로젝트의 소스 코드에서는 'String accountNum'을 받아서 유효성을 검사한다. 즉, 테스트코드의 동작이 실제 동작과 일치하지 않는다.
- Checkpassword
 - checkPassword()
 - 매개변수로 받은 값과 비밀번호 1111을 직접 비교한다. 즉, 실제 어떤 계좌의 비밀번호를 비교하는 것이 아니므로 이에 대한 수정이 필요하다.
- GetBalance
 - getBalance()
 - 단순히 매개변수로 받은 값이 0과 같으면 1을 반환하고, 같지 않으면 2를 반환하고 있다. 무엇을 테스트한다는 것인지 명확하지 않다.
- Transaction
 - transaction()
 - 매개변수로 'int amount'를 받고있다. 하지만 실제 프로젝트의 소스 코드에서는 'int amount'와 'String msg'를 함께 받고 있다. 어떤 코드를 기준으로 테스트 코드를 작성했는지 모호하다.
- getLogs()
 - 단순히 println으로 'path'와 'fileReader'를 출력한다. 실제 프로그램에서의 getLogs 메소드는 log.txt 파일을 읽어서 반환한다. 즉, 테스트코드의 동작이 실제 동작과 일치하지 않는다.
- showLogs()
 - 단순히 println으로 'showlogs'를 출력한다. 즉, 어떤 기능을 수행한다고 보기에는 어려운 점이 있다. 또한, 실제 프로젝트의 소스 코드에서 showLogs 메소드는 존재하지 않는다.

1.5 Stage 2060 Unit test & System Test

- 2063. System Testing
 - System Function 목록에 대해 참조할 수 있는 자료가 없다.
 - 1_6
 - '오류처리'가 정확히 어떤 오류를 말하는 것인지 명확하지 않다.
 - 2_1
 - '매체'라는 표현이 명확하지 않다.
 - 실제 프로젝트의 소스 코드에서는 추상적으로 '매체'가 아닌 단순히 계좌번호만을 입력하게 되어있다.
 - 3_2
 - '비밀번호를 입력하는 화면이 나타나고, 알맞은 비밀번호 입력이 제대로 진행되는지 확인한다.'로 표현을 바꾸는 것이 적절하다.
 - 3_3
 - '비밀번호를 입력하는 화면이 나타나고, 잘못된 비밀번호를 입력했을 때 오류처리가 되었는지 확인한다.'로 표현을 바꾸는 것이 적절하다.
 - 4_1
 - '돈을 입력하고 그만큼 제대로 계산되는지 확인 한다.'라고 명시되어있으나, 정확히 어떤 계산을 제대로 한다는 것인지 불분명하다.
 - 4_2
 - '거래하거나 출금할 양을 입력하고 그만큼 정확히 계산되는지 확인한다.'라고 명시되어있으나, '거래'가 뜻하는 것이 정확히 출금인지 입금인지 송금인지 불분명하다.
 - 정확히 계산한다는 것이 무엇을 정확히 계산한다는 것인지 불분명하다.
 - 4_4_1 ~ 4_4_4
 - 당행, 타행에서 거래시 수수료가 정확히 계산된다고 명시되어있으나, 실질적으로 당행에서는 수수료가 차감되지 않으며 타행에서 거래시 수수료가 차감된다.
 - 정확히 몇 %의 수수료를 차감하는 것인지 명시되어있지 않다.
 - 5_1
 - 예금할 돈을 입력한 뒤 비밀번호를 입력받는다라는 system test case에 대한 정의가 누락되어있다.
 - 5_2

- 출금할 돈을 입력한 뒤 비밀번호를 입력받는다라는 system test case에 대한 정의가 누락되어있다.
- 5_3
 - 송금할 돈을 입력한 뒤 비밀번호를 입력받는다라는 system test case에 대한 정의가 누락되어있다.
- 5_4_2
 - ‘알맞은 매체를 입력’한다고 명시되어있으나, 실제로는 계좌번호만을 입력받으므로, ‘알맞은 계좌번호를 입력’한다고 수정해야 한다.

2. Category-partition Testing Report

2.1 Test Case

2.1.1 Testable Units and Representative Values

Group	Category	Description	Value	Number
Validate	Account	계좌의 존재 유무 확인	계좌번호가 존재할때	1000
			계좌번호가 존재하지 않을때	1001
	Password	비밀번호 일치 확인	일치하는 비밀번호	1100
			일치하지 않는 비밀번호	1101
	ATM Balance	ATM 기계 내 잔고 확인	ATM의 잔고가 출금하려는 잔고보다 많다(충분)	1200
			ATM의 잔고가 출금하려는 잔고보다 적다(불충분)	1201
	Account Balance	계좌 내 잔고 확인	계좌의 잔고가 출금하려는 잔고보다 많다(충분)	1300
			계좌의 잔고가 출금하려는 잔고보다 적다(불충분)	1301
Transaction	Deposit	입금	타행인경우 거래 금액에서 수수료를 제한 금액을 입금한다(잔액 추가)	2000
			당행인경우 거래 금액을 입금한다(잔액 추가)	2001
	Withdraw	출금	타행인경우 거래 금액에서 수수료를 더한 금액을 출금한다(잔액 감소)	2100
			당행인경우 거래 금액을 출금한다(잔액 감소)	2101

	Transfer	송금	타행인경우 거래 금액에서 수수료를 더한 금액을 출금하고(잔액 감소), 송금 대상 계좌의 잔액을 거래 금액만큼 증가시킨다	2200
			당행인경우 거래 금액을 출금하고(잔액 감소), 송금 대상 계좌의 잔액을 거래 금액만큼 증가시킨다	2201
Print	Transaction Result	입,출, 송금 결과 출력	입금, 출금, 송금 후 계좌에 남은 실제 잔액을 화면에 표시한다	3000
			입금, 출금, 송금 후 계좌에 남은 실제 잔액을 화면에 표시하지 않는다	3001
	Criminal History	범죄기록 출력	범죄기록이 있는 경우	3100
			범죄기록이 없는 경우	3101
	Transaction History	거래내역 출력	거래내역이 있는 경우	3200
			거래내역이 없는 경우	3201
Input	Account Number	계좌번호 입력	숫자로만 입력한 경우	4000
			기타 문자와 숫자를 섞어 쓴 경우(숫자 이외의 문자가 들어간 경우)	4001
			입력받지 않은 경우	4002
	Password Number	계좌 비밀번호 입력	계좌 비밀번호를 입력받는다	4100
			입력받지 않은 경우	4101
	Transaction Amount	거래할 금액 입력	숫자로만 입력한 경우	4200
			기타 문자와 숫자를 섞어 쓴 경우(숫자 이외의 문자가 들어간 경우)	4201
			입력받지 않은 경우	4202
	Access	Transaction Log	txt 파일에 접근	거래기록 파일이 있는 경우
거래기록 파일이 없는 경우				5001
Criminal Log		txt 파일에 접근	범죄기록 파일이 있는 경우	5100
			범죄기록 파일이 없는 경우	5101
Total Test Case			22184	

2.1.2 Error Constraints 적용

- Total Test Case: 22184 -> 777 (96.50% 감소)

2.1.3 Single Constraints 적용

- Total Test Case: 777 -> 111 (85.71% 감소)

2.1.4 Property Constraints 적용

- Total Test Case: 111 -> 51 (53.15% 감소)

2.2 Testing Result

Test Case No.	Test Case	Result
1	1001	F
2	1101	F
3	1201	F
4	1301	F
5	4001	F
6	4002	P
7	4101	P
8	4201	F
9	4202	F
10	1000.1100.1200.2000.3000.3100.3200.4000.4100.4200.5000.5100.	F
11	1000.1100.1200.2100.3000.3100.3200.4000.4100.4200.5000.5100.	F
12	1000.1100.1200.2200.3000.3100.3200.4000.4100.4200.5000.5100.	F
13	1000.1100.1200.2001.3000.3100.3200.4000.4100.4200.5000.5100.	F
14	1000.1100.1200.2101.3000.3100.3200.4000.4100.4200.5000.5100.	F
15	1000.1100.1200.2201.3000.3100.3200.4000.4100.4200.5000.5100.	F
16	1000.1100.1200.2000.3000.3101.3200.4000.4100.4200.5000.5100.	P
17	1000.1100.1200.2100.3000.3101.3200.4000.4100.4200.5000.5100.	P
18	1000.1100.1200.2200.3000.3101.3200.4000.4100.4200.5000.5100.	P
19	1000.1100.1200.2001.3000.3101.3200.4000.4100.4200.5000.5100.	P
20	1000.1100.1200.2101.3000.3101.3200.4000.4100.4200.5000.5100.	P
21	1000.1100.1200.2201.3000.3101.3200.4000.4100.4200.5000.5100.	P
22	1000.1100.1200.2000.3000.3100.3201.4000.4100.4200.5000.5100.	F
23	1000.1100.1200.2100.3000.3100.3201.4000.4100.4200.5000.5100.	F
24	1000.1100.1200.2200.3000.3100.3201.4000.4100.4200.5000.5100.	F

25	1000.1100.1200.2001.3000.3100.3201.4000.4100.4200.5000.5100.	F
26	1000.1100.1200.2101.3000.3100.3201.4000.4100.4200.5000.5100.	P
27	1000.1100.1200.2201.3000.3100.3201.4000.4100.4200.5000.5100.	P
28	1000.1100.1200.2000.3000.3101.3201.4000.4100.4200.5000.5100.	P
29	1000.1100.1200.2100.3000.3101.3201.4000.4100.4200.5000.5100.	P
30	1000.1100.1200.2200.3000.3101.3201.4000.4100.4200.5000.5100.	P
31	1000.1100.1200.2001.3000.3100.3200.4000.4100.4200.5000.5100.	P
32	1000.1100.1200.2101.3000.3100.3200.4000.4100.4200.5000.5100.	P
33	1000.1100.1200.2201.3000.3100.3200.4000.4100.4200.5000.5100.	P
34	1000.1100.1200.2000.3000.3101.3200.4000.4100.4200.5000.5101.	P
35	1000.1100.1200.2100.3000.3101.3200.4000.4100.4200.5000.5101.	P
36	1000.1100.1200.2200.3000.3101.3200.4000.4100.4200.5000.5101.	P
37	1000.1100.1200.2001.3000.3101.3200.4000.4100.4200.5000.5101.	P
38	1000.1100.1200.2101.3000.3101.3200.4000.4100.4200.5000.5101.	P
39	1000.1100.1200.2201.3000.3101.3200.4000.4100.4200.5000.5101.	P
40	1000.1100.1200.2000.3000.3101.3201.4000.4100.4200.5000.5101.	P
41	1000.1100.1200.2100.3000.3101.3201.4000.4100.4200.5000.5101.	P
42	1000.1100.1200.2200.3000.3101.3201.4000.4100.4200.5000.5101.	F
43	1000.1100.1200.2001.3000.3101.3201.4000.4100.4200.5000.5101.	P
44	1000.1100.1200.2101.3000.3101.3201.4000.4100.4200.5000.5101.	F
45	1000.1100.1200.2201.3000.3101.3201.4000.4100.4200.5000.5101.	F
46	1000.1100.1200.2000.3001.3100.3201.4000.4100.4200.5001.5100.	P
47	1000.1100.1200.2100.3001.3100.3201.4000.4100.4200.5001.5100.	P
48	1000.1100.1200.2200.3001.3100.3201.4000.4100.4200.5001.5100.	P
49	1000.1100.1200.2001.3001.3100.3201.4000.4100.4200.5001.5100.	P
50	1000.1100.1200.2101.3001.3100.3201.4000.4100.4200.5001.5100.	P
51	1000.1100.1200.2201.3001.3100.3201.4000.4100.4200.5001.5100.	P

3. Pairwise Testing Report

3.1 Testing Case

Bank	Action	Log File	PW	Account
Same	Deposit	Exist both	Vaild	Vaild
Different	Withdraw	Exist Criminal only	Not Vaild	Not Vaild

	Transfer	Exist History only		
	History	Non		
	Criminal History			

3.2 Testing Result

Num.	Bank	Action	Log File	PW	Account	Result
1	Same	Criminal	Non	-	Not Vaild	F
2	Different	Criminal	Exist both	-	Vaild	F
3	Same	History	Exist History Only	Vaild	Vaild	F
4	Different	Deposit	Exist History Only	Not Vaild	Not Vaild	F
5	Same	Withdraw	Exist both	Vaild	Not Vaild	F
6	Different	Criminal	Exist History Only	-	Not Vaild	F
7	Same	Criminal	Exist Criminal Only	-	Vaild	F
8	Different	Withdraw	Exist Criminal Only	Not Vaild	Vaild	F
9	Different	History	Non	Not Vaild	Vaild	F
10	Same	Transfer	Exist Criminal Only	Vaild	Not Vaild	F
11	Different	Withdraw	Non	Not Vaild	Vaild	F
12	Same	History	Exist both	Vaild	Not Vaild	F
13	Same	History	Exist Criminal Only	Not Vaild	Not Vaild	F
14	Same	Deposit	Non	Vaild	Vaild	F
15	Different	Transfer	Exist both	Not Vaild	Vaild	F
16	Same	Withdraw	Exist History Only	Vaild	Vaild	F
17	Same	Deposit	Exist both	Vaild	Vaild	F
18	Same	Deposit	Exist Criminal Only	Vaild	Vaild	F
19	Same	Transfer	Exist History Only	Vaild	Not Vaild	F
20	Different	Transfer	Non	Not Vaild	Not Vaild	F

4. Brute Force Testing Report

4.1 Testing Result

Num.	Test Case	Result
1	입금 시 입금 금액에 음수를 넣고 확인버튼을 누른다.	F
2	출금 시 출금 금액에 음수를 넣고 확인버튼을 누른다.	F
3	송금 시 송금 금액에 음수를 넣고 확인버튼을 누른다.	F
4	입금 시 입금 금액에 숫자 이외의 문자를 넣고 확인버튼을 누른다.	F
5	출금 시 출금 금액에 숫자 이외의 문자를 넣고 확인버튼을 누른다.	F
6	송금 시 송금 금액에 숫자 이외의 문자를 넣고 확인버튼을 누른다.	F
7	입금 시 입금 금액에 int 범위 이외의 숫자를 넣고 확인버튼을 누른다.	F
8	출금 시 출금 금액에 int 범위 이외의 숫자를 넣고 확인버튼을 누른다.	F
9	송금 시 송금 금액에 int 범위 이외의 숫자를 넣고 확인버튼을 누른다.	F
10	입금 시 입금 금액에 아무것도 넣지 않고 확인버튼을 누른다.	F
11	출금 시 출금 금액에 아무것도 넣지 않고 확인버튼을 누른다.	F
12	송금 시 송금 금액에 아무것도 넣지 않고 확인버튼을 누른다.	F
13	입금 시 비밀번호를 3회 이상 틀렸을 때 예외처리를 확인 한다.	F
14	출금 시 비밀번호를 3회 이상 틀렸을 때 예외처리를 확인 한다.	F
15	송금 시 비밀번호를 3회 이상 틀렸을 때 예외처리를 확인 한다.	F
16	입금 내역이 많을 경우 1초 이내에 처리 되는지 확인 한다.	F
17	출금 내역이 많을 경우 1초 이내에 처리 되는지 확인 한다.	F
18	송금 내역이 많을 경우 1초 이내에 처리 되는지 확인 한다.	F
19	거래 내역이 많을 경우 1초 이내에 처리 되는지 확인 한다.	F
20	범죄 이력이 많을 경우 1초 이내에 처리 되는지 확인 한다.	F
21	거래 내역이 많을 경우, 거래 내역 조회 기능에서 Simple한 UI/UX를 가지는지 확인한다.	F
22	범죄 내역이 많을 경우, 범죄 이력 조회 기능에서 Simple한 UI/UX를 가지는지 확인한다.	F

4.2 Failed Case Report

Number	Test Case	Result
1	입력 금액에 음수를 넣고 입금을 진행할 경우 잔고가 감소한다.	F
2	출금 금액에 음수를 넣고 출금을 진행할 경우 잔고가 증가한다.	F
3	송금 금액에 음수를 넣고 송금을 진행할 경우 돈을 받는 사람의 잔고는 감소하고, 돈을 보내는 사람의 잔고는 증가한다.	F
4	입금 시 입금 금액에 숫자 이외의 문자를 넣고, 확인 버튼을 눌렀을 때 아무 반응이 없다.	F
5	출금 시 출금 금액에 숫자 이외의 문자를 넣고, 확인 버튼을 눌렀을 때 아무 반응이 없다.	F
6	송금 시 송금 금액에 숫자 이외의 문자를 넣고, 확인 버튼을 눌렀을 때 아무 반응이 없다.	F
7	입금 시 입금 금액에 int 범위 이외의 숫자를 넣고 확인 버튼을 눌렀을 때 아무 반응이 없다.	F
8	출금 시 출금 금액에 int 범위 이외의 숫자를 넣고 확인 버튼을 눌렀을 때 아무 반응이 없다.	F
9	송금 시 송금 금액에 int 범위 이외의 숫자를 넣고 확인 버튼을 눌렀을 때 아무 반응이 없다.	F
10	입금 시 입금 금액에 아무것도 넣지 않고 확인 버튼을 눌렀을 때 아무 반응이 없다.	F
11	출금 시 출금 금액에 아무것도 넣지 않고 확인 버튼을 눌렀을 때 아무 반응이 없다.	F
12	송금 시 송금 금액에 아무것도 넣지 않고 확인 버튼을 눌렀을 때 아무 반응이 없다.	F
13	입금 시 비밀번호를 3회 이상 틀렸을 때 에러를 발생시키지 않고, 비밀번호가 틀렸을 때의 에러와 동일한 에러가 발생한다.	F
14	출금 시 비밀번호를 3회 이상 틀렸을 때 에러를 발생시키지 않고, 비밀번호가 틀렸을 때의 에러와 동일한 에러가 발생한다.	F
15	송금 시 비밀번호를 3회 이상 틀렸을 때 에러를 발생시키지 않고, 비밀번호가 틀렸을 때의 에러와 동일한 에러가 발생한다.	F
16	입금메뉴 선택 후, log.txt 파일이 50MB가 넘는 계좌번호를 입력하고 확인 버튼을 누르면 약 5초 후 메뉴가 표시된다.	F
17	출금메뉴 선택 후, log.txt 파일이 50MB가 넘는 계좌번호를 입력하고 확인 버튼을 누르면 약 5초 후 메뉴가 표시된다.	F
18	송금메뉴 선택 후, log.txt 파일이 50MB가 넘는 계좌번호를 입력하고 확인 버튼을 누르면 약 5초 후 메뉴가 표시된다.	F
19	조회메뉴 선택 후, log.txt 파일이 50MB가 넘는 계좌번호를 입력하고 확인 버튼을 누르면 약 5초 후 메뉴가 표시된다.	F

20	범죄이력 조회 메뉴 선택 후, criminalLog.txt 파일이 50MB가 넘는 계좌번호를 입력하고 확인 버튼을 누르면 약 5초 후 메뉴가 표시된다.	F
21	조회 버튼을 눌러 1,622,925개 내역이 있는 거래 계좌를 입력하고 확인 버튼을 눌렀을 경우 페이징 처리가 되어 있지 않고 단순 스크롤로 약 160만여개의 거래 내역을 보여주기 때문에 Simple한 UI/UX를 가진다고 보기 어렵다.	F
22	범죄 이력 조회 버튼을 눌러 1,563,312개 내역이 있는 거래 계좌를 입력하고 확인 버튼을 눌렀을 경우 페이징 처리가 되어 있지 않고 단순 스크롤로 약 150만여개의 거래 내역을 보여주기 때문에 Simple한 UI/UX를 가진다고 보기 어렵다.	F

5. Overall

5.1 System Test Result

5.1.1 Category-Partition-Test

31/51 = 60.78% Pass

5.1.2 Pairwise Test

0/20 = 0 % Pass

5.1.3 Brute Force Test

0/22 = 0% Pass

5.1.4 Summary

- 입금액을 입력할 때 음수 값을 입력을 해주면 계좌 잔액이 차감되는등의 현상이 발생하므로 SUPER SAFE ATM이라는 소프트웨어에 적합하지 않다.
- Specification의 구체적인 기준이 제시되어 있지 않고 모호한 것이 많다.
- 문서가 일관성 없이 작성되어 각 단계마다 글을 작성하는 방식, 내용 등 달라지는 것이 많이 많다.
- Refine 부분을 명확히 할 필요가 있으며 이에 대한 문서 작성자와 개발자의 명확한 인식이 필요해 보인다.
- 성능과 사용자 인터페이스와 같은 non functional requirement를 만족시키지 못하였다.
- 문서와 실제 작성된 코드간에 일치하지 않는 부분이 다수 존재한다.