

# Software Verification

## System Test Report

2st

Date

2018-06-04

---

Team 4

201411259 고수창

201411314 전소영

201412005 이세라

201511304 하지윤

<b>1. Step1. Specification Review</b>	<b>3</b>
1.2 Stage 2010 Revise Plan	4
1.3 Stage 2030 Analysis	4
1.4 Stage 2040 Design	7
1.5 Stage 2050 Implementation	9
1.6 Stage 2060 Unit test & System test	21
<b>2. Category-partition Testing Report</b>	<b>23</b>
2.1 Test Case	23
2.1.1 Testable Unit and Representative Values	23
2.1.2 Error Constraints 적용	24
2.1.3 Single Constraints 적용	24
2.1.4 Property Constraints 적용	24
<b>3. Pairwise Testing Report</b>	<b>24</b>
3.1 Testing Case	24
3.2 Testing Result	24
<b>4. Brute Force Testing Report</b>	<b>25</b>
4.1 Testing Result	25
4.2 Failed Case Report	26
<b>5. Overall</b>	<b>27</b>
5.1 System Test Result	27
5.1.1 Category-Partition-Test	27
5.1.2 Pairwise Test	27
5.1.3 Brute Force Test	27
5.1.4 Summary	27

## 1. Step1. Specification Review

### 1.1 Stage 1000 Planning

- Activity 1001. Define draft plan
  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.
  3. Operating Environments
    - Operating System- / Development Environments
      - 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 1005. Implement Prototype
  - 그림 번호 및 상황에 대한 정보가 제시되어 있지 않다.
  
- Activity 1006. Define Business Use Case
  8. Describe Use-Cases
    - 3. Deposit
      - 1003에 언급되었던 타행입금에 관한 수수료가 누락되었다.
    - 7. CheckTransactionHistory
      - CheckCriminalHistory로 수정이 필요하다.

## 1.2 Stage 2010 Revise Plan

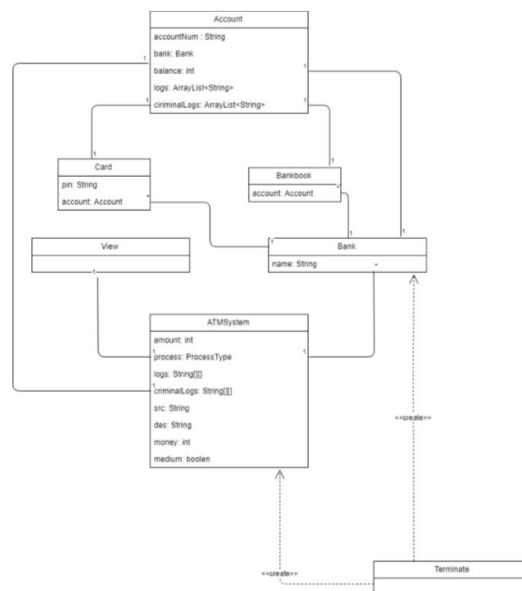
이전 문서의 버전과 달라진 것이 많으나 이를 작성하지 않았다. 요구사항의 어떤 부분이 추가, 삭제, 수정되었는지 알 수 없다. 버전별 바뀐점에 대해서 작성할 필요가 있다.

## 1.3 Stage 2030 Analysis

- 2031. Define Essential Use Case
  - 전체적인 사항
    - Typical Course of Events의 시나리오에 번호를 입력하여 순서를 명확히 할 필요가 있다.
  - 2. MediumCheck
    - Exceptional Course of Events
      - 5. “유효”의 정의가 모호하여 어떤 상황에서 유효한 것인지 알 수 없다. 카드번호와 계좌번호를 모두 입력받는데 계좌번호 오류에 대한 알림만 명시되어 있다.
  - 3. Deposit
    - Typical Courses of Events
      - 수수료를 정확히 어느정도 차감하는지 비율이 나와있지 않다.
      - 1006에는 수수료관련 언급이 없었는데 해당 단계에서 당행/타행을 비교하여 수수료를 계산하는 과정이 추가되었다.
      - 1006에는 끝내기 버튼을 눌러 메인으로 돌아간다고 명시되어 있으나 누락되었다.
      - Error에 대한 처리가 명시되지 않아 단순히 표시한다는 것인지 프로그램을 종료하는 것인지 알 수 없다.
  - 4. Withdraw
    - Typical Courses of Events
      - 1006에는 수수료관련 사항이 없었는데 해당 단계에서 당행/타행을 비교하여 수수료를 계산하는 과정이 추가되었다.
      - 1006에는 끝내기 버튼을 눌러 메인으로 돌아간다고 명시되어 있으나 누락되었다.
      - Error에 대한 처리가 명시되지 않아 단순히 표시한다는 것인지 프로그램을 종료하는 것인지 알 수 없다.
  - 5. Transfer
    - Typical Courses of Events

- 1006에는 수수료관련 사항이 없었는데 해당 단계에서 당행/타행을 비교하여 수수료를 계산하는 과정이 추가되었다.
- Error에 대한 처리가 명시되지 않아 단순히 표시한다는 것인지 프로그램을 종료하는 것인지 알 수 없다.
- 6. CheckTransactionHistory
  - 1006에는 끝내기 버튼을 눌러 메인으로 돌아간다고 명시되어 있으나 누락되었다.
- 7. CheckCriminalHistory
  - 1006에는 끝내기 버튼을 눌러 메인으로 돌아간다고 명시되어 있으나 누락되었다.

- 2033. Define Domain Model



- 1004에서 제시하였던 User 객체가 보이지 않는다.

- 2034. Refine Glossary

- Class Diagram에선 Bankbook으로 표시되어 있으나 Glossary에는 Passbook으로 표시되었다. Bankbook과 Passbook을 혼용하여 혼란을 준다.

- 2036. Define Operation Contracts

- inputMedium()
  - Exception
    - 유효하지 않을때의 Exception이 없음.
  - Post Condition

- “입력된 계좌번호가 DB에 존재하는지 확인한다”라고 했지만 확인하는 단계에 해당하는 System Operation에 해당되는 사항이 없다.
  - inputMoney()
    - Exceptions
      - 2031의 Essential Use Cases에 음수를 입력할 시 에러 처리를 한다고 명시되어 있으나 해당 단계에선 제시되지 않고있다.
  - inputPassword()
    - Exception
      - 유효하지 않을때의 Exception이 없음.
    - Post Condition
      - “비밀번호가 계좌에 해당하는 비밀번호인지 확인하다”라고 명시되어 있으나 비밀번호를 확인하는 System Operation이 없다.
  - inputAmount()
    - Exceptions
      - 2031의 Essential Use Cases에 음수를 입력할 시 에러 처리를 한다고 명시되어 있으나 해당 단계에선 제시되지 않고있다.
  - inputAccount()
    - Post Condition
      - “계좌번호가 유효한지 확인한다.”라고 명시되어 있으나 계좌가 유효한지 확인하는 System Operation이 없다.
      - 계좌의 유효성에 대한 기준이 없어 어떤 계좌가 유효한 것인지 알 수 없다.
- 2037. Define State Diagrams
    - 이전 검증에서 수정 권고를 하였으나 문서에서 누락되어 있다.
  - 2038. Refine System Test Case
    - ATM\_STC\_003/004/005
      - \_001: “잘”이라는 기준이 모호하다. 2031에 음수를 입력 받는 경우 Error처리한다고 명시되어 있으나 이를 확인 하는 것이 누락되어 있다.

## 1.4 Stage 2040 Design

- 2041. Design Real Use Cases
  - 2031과 동일한 이슈
- 2042. Define Reports, UI and Storyboards
  - 이미지 제목 및 설명 누락
- 2043. Refine System Architecture
  - 실제 소스상에 없는 User, Passbook, ATM 존재
  - 소스상에는 Bankbook, Terminate, View, AtmSystem 존재
- 2044. Define Interaction Diagrams
  - Alert : 실제 소스상에서, ATMSystem 안에서 alert를 호출하지 않는다.
  - 2. MediumCheck
    - 3. findCard: 실제 소스상에서 atm.setSrc를 이용해 AtmSystem 내부에서 findCard를 자체적으로 호출한다.
    - 4. validateCard : return이 명시되어있지 않다.
    - 5. getPin : 실제 소스상에서 getPin은 card를 return 하지 않고 validateCard가 card를 return한다.
    - 7. findBankbook : 실제 소스상에서 atm.setSrc를 이용해 AtmSystem 내부에서 findBankbook을 자체적으로 호출한다.
    - 8. validateBankbook : return이 명시되어있지 않다.
  - 3. Deposit
    - 12. transaction : process -> transfer로 가는 부분이 명시되어 있지 않다.
    - 14. getBalance : 호출하고 return되는 과정이 생략되어 있다.
    - 15. fileReader : 실제 소스에서 FileWriter를 호출한다.
  - 4. Withdraw
    - 12. transaction : process -> transfer로 가는 부분이 명시되어 있지 않다.
    - 14. getBalance : 호출하고 return되는 과정이 생략되어 있다.
    - 15. fileReader : 실제 소스에서 FileWriter를 호출한다.
  - 5. Transfer
    - 5. findCard : 실제 소스상에서 atm.setSrc를 이용해 AtmSystem 내부에서 findCard를 자체적으로 호출한다.
    - 6. validateCard : return이 명시되어있지 않다.

- 7. getPin : 실제 소스상에서 getPin은 card를 return 하지 않고 validateCard가 card를 return한다.
  - 9. findBankbook : 실제 소스상에서 atm.setSrc를 이용해 AtmSystem 내부에서 findBankbook을 자체적으로 호출한다.
  - 10. validateBankbook : return이 명시되어있지 않다.
  - 23. transaction : process -> transfer로 가는 부분이 명시되어 있지 않다.
  - 25. getBalance : 호출하고 return되는 과정이 생략되어 있다.
  - 26. fileReader : 실제 소스에서 FileWriter를 호출한다.
- 2045. Define Design Class Diagrams
    - \*Bold로 처리한 항목은 소스와 문서가 일치하는 항목이다.*
    - 실제 Class와 일치하지 않다.
      - Account
        - 소스 : **accountNum, bank**, password, **Account**, getBank, getAccountNum, getBalance, **checkPassword, transaction**, getLogs, getCriminalLogs
        - 문서 : **accountNum, bank**, balance, logs, criminalLogs, **Account, checkPassword, transaction**
      - Card
        - 소스 : **pin, account**, password, **Card**, getAccount, getPin, **checkPassword**
        - 문서 : **pin, account, Card, checkPassword**
      - Bankbook
        - 소스 : **account, Bankbook**, getAccount
        - 문서 : **account, Bankbook**
      - Bank
        - 소스 : **name, Bank, loadDataBase, fileReader, validateBankbook, validateCard, getBalance, transaction, getLogs, getCriminalLogs**
        - 문서 : **name**, path, cards, bankbooks, **Bank, loadDataBase, fileReader, validateBankbook, validateCard, getBalance, transaction, getLogs, getCriminalLogs, getName**



- View
  - GUI에 대한 의존성이 있으므로 생략한다.
  
- ATMSystem
  - 소스 : own, view, **src, des, amount, money, fee, medium, banks, process, srcCard, AtmSystem, loadBanks, process, checkBalance, checkPassword, deposit, withdraw, transfer, getLogs, getCriminalLogs, setProcess, setMedium, setSrc, validateSrc, setDes, validateDes, setAmount, setMoney, findCard, findBankbook**
  - 문서 : **src, money, medium, des, process, criminalLogs, logs, amount, AtmSystem, loadBanks, process, checkBalance, checkPassword, deposit, withdraw, transfer, validateSrc, validateDes, findCard, findBankbook**

### 1.5 Stage 2050 Implementation

- 2051. Implement Class & Methods Definition
  - class Account
    - Account()
      - purpose
        - "Account 의 객체 생성한다."를 수정해야 한다.  
Account는 생성자로서 단순히 attribute를 초기화해주는 역할만 수행하며, 별도의 객체를 생성하지 않기 때문이다.
        - 이 부분은 이전 보고서에서 찾아낸 부분이지만, 수정이 안된 것으로 보인다.
    - getBalance()
      - abstract operation
        - 'bank DB 에서 balance 를 불러와 반환한다.'를 수정해야한다. bankDB라는 표현이 정의되어있지 않아 정확히 무엇을 의미하는지 모호하다.
        - 이 부분은 이전 보고서에서 찾아낸 부분이지만, 수정이 안된 것으로 보인다.
    - checkPassword()
      - abstract operation

- 'bankDB 에서 비밀번호 일치 여부를 반환한다.'를 수정해야 한다. bankDB라는 표현이 정의되어있지 않아 정확히 무엇을 의미하는지 알 수 없다.
- 이 부분은 이전 보고서에서 찾아낸 부분이지만, 수정이 안된 것으로 보인다.
- getCriminalLogs()
  - abstract operation
    - 계좌의 범죄 이력과 횡수를 반환한다고 했으나, 실제 프로그램을 실행시켜 보면 횡수가 아닌 범죄 내용 및 범죄 시간에 대해서만 반환한다.
- class AtmSystem
  - Overview
    - attribute
      - View view가 누락되어있다.
  - AtmSystem()
    - purpose
      - 'AtmSystem 의 객체 생성한다.'를 수정해야 한다. AtmSystem은 생성자로서 단순히 attribute를 초기화해주는 역할만 수행하며, class View의 객체를 생성하기 때문이다.
      - 이 부분은 이전 보고서에서 찾아낸 부분이지만, 수정이 안된 것으로 보인다.

```

37     private void loadBanks() {
38         File file = new File(System.getProperty("user.dir") + "/res");
39         File[] banks = file.listFiles();
40         if (banks != null) {
41             for (File bank : banks) {
42                 this.banks.add(new Bank(bank.getName()));
43             }
44         }
45     }

```

- 문서에는 Bank own을 input으로 받도록 나와있으나, 실제 코드에서는 어떠한 파라미터도 받지 않는다.
- checkBalance()
  - Output

```

67     public boolean checkBalance() {
68         return (amount + fee < src.getBalance()) && (money - fee + src.getBalance() > 0);
69     }

```

- boolean (amount + fee) < src.getBalance()로 명시되어있으나, 이는 코드와 다르다.
- Abstract operation
  - 코드와 실제 문서에 나온 내용이 다르므로 이에 대한 통일이 필요하다.
- deposit()

```

79     public void deposit() {
80         src.transaction(money - fee, "입금");
81     }

```

- '계좌의 balance를 amount-fee 만큼 증가시킨다'는 표현이 나와있으나, 실제 코드에서는 money라는 이름의 변수로 표현되어있다. 이에 대한 통일이 필요하다.
- setDes()

```

134     public void setDes(String accountNum) {
135         if (medium == MediumType.CARD) {
136             src = findCard(accountNum);
137         } else {
138             src = findBankbook(accountNum);
139         }
140     }

```

- '매체에 따라 findCard 또는 findBankBook의 결과를 des에 저장한다'고 명시되어있으나, 실제 코드에서는 src를 설정한다.
- setAmount()
  - Abstract operation
    - amount 값을 설정할 때, 타행 계좌인 경우 fee(수수료)를 설정한다는 설명이 누락되어있다.
- setMoney()
  - Abstract operation
    - money 값을 설정할 때, 타행 계좌인 경우 fee(수수료)를 설정한다는 설명이 누락되어있다.

- findCard()
  - 문서 전체적으로 'bank DB'라는 표현이 반복되어 나온다. 실제 DB를 사용하지도 않고, 텍스트 파일 형태의 단순 파일 시스템을 사용하기 때문에 표현을 바꾸거나, 정의를 새로 해줘야 할 필요가 있다.
- class Bank
  - Bank()
    - purpose
      - 'Bank 의 객체를 생성한다.'를 수정해야 한다. Bank는 생성자로서 단순히 attribute를 초기화해주는 역할만 수행하며, 별도의 객체를 생성하지 않기 때문이다.
      - 이 부분은 이전 보고서에서 찾아낸 부분이지만, 수정이 안된 것으로 보인다.
  - loadDataBase()
    - 문서 전체적으로 DB라는 표현이 반복되어 나온다. 실제 DB를 사용하지도 않고, 텍스트 파일 형태의 단순 파일 시스템을 사용하기 때문에 표현을 바꾸거나, 정의를 새로 해줘야 할 필요가 있다.
  - validateBankbook()
    - bank DB라는 표현이 등장하는데, 실제 DB를 사용하지도 않고, 이에 대한 정의가 모호하여 특정 변수를 말하는 것인지 분명한 표현으로 바꿀 필요가 있다.
  - validateCard()
    - bank DB라는 표현이 등장하는데, 실제 DB를 사용하지도 않고, 이에 대한 정의가 모호하여 특정 변수를 말하는 것인지 분명한 표현으로 바꿀 필요가 있다.

```

82     public Card validateCard(String pin) {
83         for (Card card : this.cards) {
84             if (card.getPin().equals(pin)) {
85                 return card;
86             }
87         }
88         return null;
89     }

```

- 'accountNum'에 해당하는 카드를 확인한다고 하였으나, 실제 코드에서는 pin을 확인한다.
- class Bankbook

- Bankbook()
  - Purpose, Abstract operation
    - ‘Bankbook의 객체를 생성한다’는 표현이 명시되어 있으나, 이는 생성자로서 단순히 attribute를 초기화해주는 역할만을 수행한다. 또한, 별도의 객체를 생성하지 않기 때문에 수정해야한다.
- class Card
  - Card()
    - Purpose
      - ‘Card의 객체를 생성한다’고 명시되어 있으나, 이는 생성자로서 단순히 attribute를 초기화해주는 역할만을 수행한다. 또한, 별도의 객체를 생성하지 않기 때문에 수정해야한다.
    - Abstract operation

```

6     public Card(Account account, String pin, String password) {
7         this.pin = pin;
8         this.password = password;
9         this.account = account;
10    }

```

- ‘Bankbook의 객체를 생성한다’고 명시되었으나, 실제 코드에서는 객체를 새로이 생성하지도 않고, 인자값으로 받아서 초기화 해주지도 않는다.

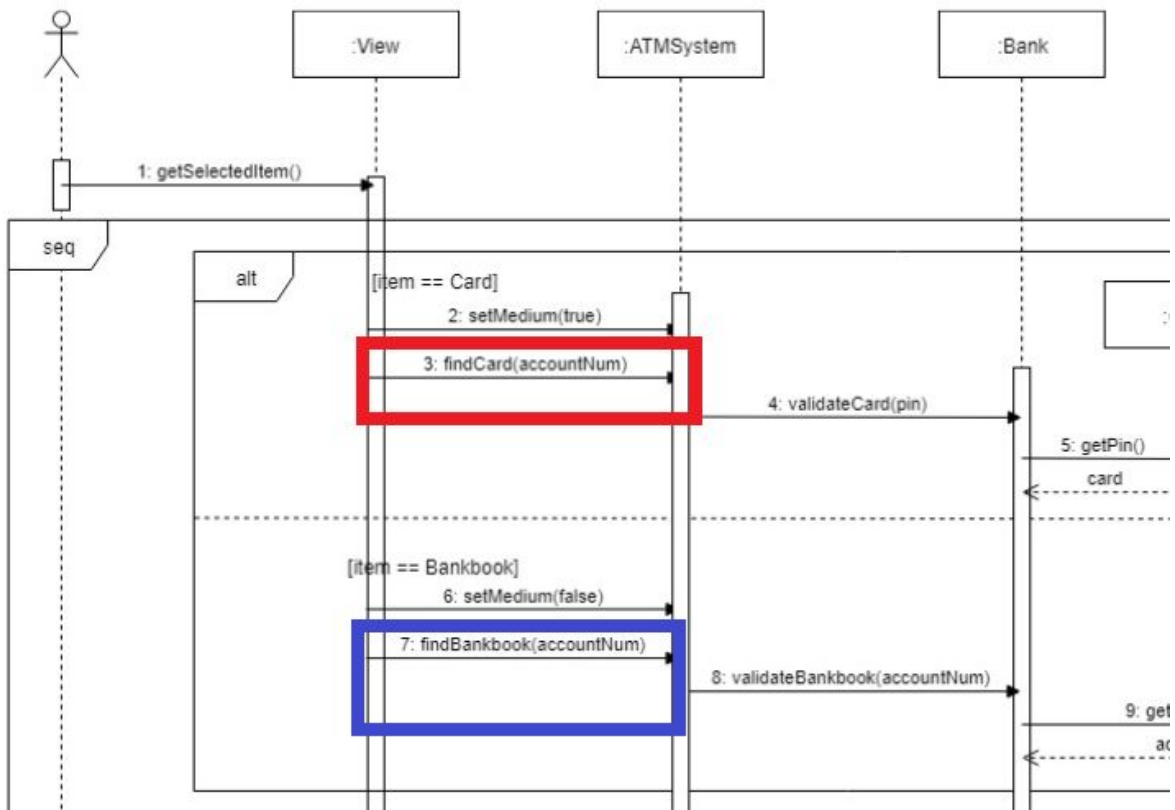
- 2052. Implements Windows

- SelectMenuView
  - 거래 내역 조회를 단순히 ‘조회’라고 명시하였으나, 이는 추상적인 측면이 있기 때문에 ‘거래 내역 조회’로 변경하는 것이 바람직하다.
  - Post-Condition
    - 계좌를 입력할 수 있다고는 하나, 신용카드등 매체를 선택하는 과정도 포함되어있기에, ‘선택한 버튼의 기능으로 진행한다’로 수정해야 한다.
- InputSrcView
  - Diagram

```

460         Object item = selectMedium.getSelectedItem();
461         if (item == null) {
462             item = "Card";
463         }
464         if (item.equals("Card")) {
465             atm.setMedium(true);
466         } else {
467             atm.setMedium(false);
468         }

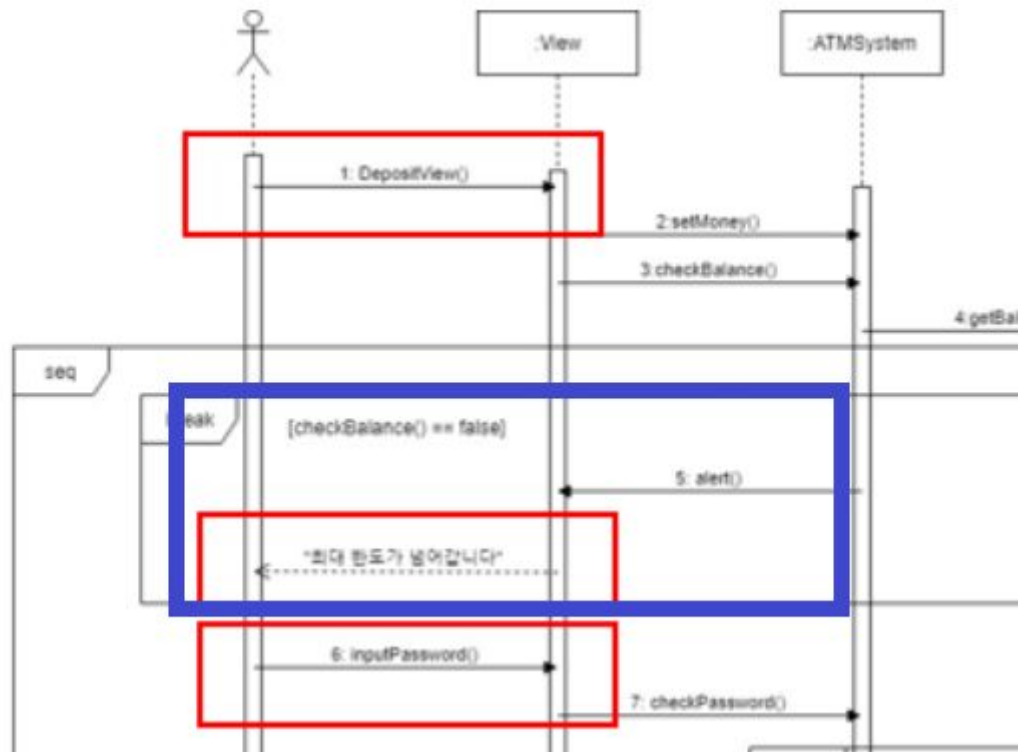
```



- 코드와 다이어그램 내용의 차이가 존재한다.
- Post-Condition
  - '선택한 버튼의 기능으로 계속 진행한다'라고 명시되어 있으나, 정확히 어떤 기능으로 진행한다는지 모호하므로 '이전 단계에서 선택한 기능의 다음 단계로 계속 진행한다'로 수정해야 한다.
- DepositView()
  - Pre-Condition
    - '계좌 번호가 유효한 상태여야 한다'를 '해당 번호가 유효한 상태여야 한다'로 바꿔야 한다.

- alert “최대한도가 넘어갑니다”
  - Diagram

```
191     private class DepositView extends JPanel {
192         private DepositView() {
193             setLayout(null);
194             setBackground(bgColor);
195             setForeground(fontColor);
196
197             InputLabel amountLabel = new InputLabel("금액",80);
198             add(amountLabel);
199
200             NumberInput amountInput = new NumberInput(80);
201             add(amountInput);
202
203             NextBtn nextBtn = new NextBtn();
204             nextBtn.addActionListener((ActionEvent e) -> {
205                 int amount = amountInput.getAmount();
206                 if (amount != 0) {
207                     atm.setMoney(amount);
208                     if (atm.checkBalance()) {
209                         nextPanel(process.next());
210                     } else {
211                         alert("최대한도가 넘어갑니다.");
212                     }
213                 }
214             });
215         }
216     }
217 }
```



(파란색으로 표시한 부분)

- 코드와 다이어그램 내용의 차이가 존재한다.
- inputPassword
  - Responsibilities
    - '계좌의 비밀번호'를 '이전 단계에서 선택한 매체(통장 혹은 카드)에 대한 비밀번호'로 수정해야 한다.
- alert "wrong password"
  - Diagram

```

503         nextBtn.addActionListener((ActionEvent e) -> {
504             String pw = String.valueOf(pwInput.getPassword());
505             if (atm.checkPassword(pw)) {
506                 atm.process();
507                 nextPanel(process.next());
508                 pwInput.setText(null);
509             } else {
510                 alert("wrong password");
511             }
512         });
  
```

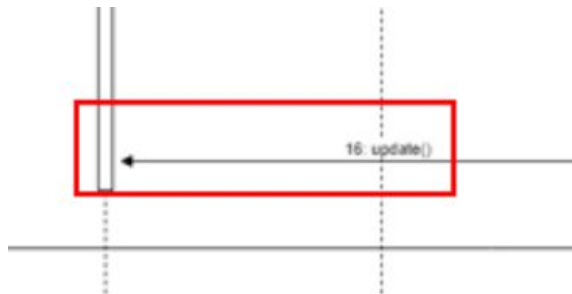




(파란색으로 표시한 부분)

- 코드와 다이어그램 내용의 차이가 존재한다.
- update

Name	Update
Type	GUI
Responsibilities	입금이 완료된 후 결과



- 'update'가 맞는 표현이나, 문서에는 'Update'로 표기되어 있다.
- WithdrawView()
  - Pre-Condition
    - '계좌 번호가 유효한 상태여야 한다'를 '해당 번호가 유효한 상태여야 한다'로 바뀌어야 한다.
- alert "잔액이 부족합니다"
  - Diagram

```

234         nextBtn.addActionListener((ActionEvent e) -> {
235             int amount = amountInput.getAmount();
236             if (amount != 0) {
237                 atm.setAmount(amount);
238                 if (atm.checkBalance()) {
239                     nextPanel(process.next());
240                 } else {
241                     alert("잔액이 부족합니다.");
242                     process.init();
243                     nextPanel(selectMenuView);
244                 }
245             }
246         });

```



(파란색으로 표시한 부분)

- 코드와 다이어그램 내용의 차이가 존재한다.
- Note

Name	alert "잔액이 부족합니다"
Type	GUI
Responsibilities	계좌에서 출금할 돈이 잔고보다 더 클 때 알린다.
Cross Reference	Functional Requirement: R 4
Notes	계좌에서 출금할 돈이 잔고보다 더 클 때 알린다.

- Responsibilities 내용과 중복되므로, 지워야 한다.
- alert "wrong password"
  - Diagram

- 코드와 다이어그램 내용의 차이가 존재한다. 이는 위의 InputPassword 클래스에서 발생했던 것과 같다.

- update

Name	Update
Type	GUI
Responsibilities	출금이 완료된 후 결고
Cross Reference	R 4



- 'update'가 맞는 표현이나, 문서에는 'Update'로 표기되어 있다.

- TransferView()

- Pre-Condition

- '계좌 번호가 유효한 상태여야 한다'를 '해당 번호가 유효한 상태여야 한다'로 바뀌어야 한다. 카드 번호를 입력하는 경우가 있기 때문.

- alert "계좌가 존재하지 않습니다"

- Diagram

```

270         NextBtn nextBtn = new NextBtn();
271         nextBtn.addActionListener((ActionEvent e) -> {
272             int amount = amountInput.getAmount();
273             if (amount != 0) {
274                 atm.setAmount(amount);
275                 atm.setDes(desInput.getText());
276                 desInput.setText("");
277                 if (!atm.validateDes()) {
278                     alert("계좌가 존재하지 않습니다.");

```



- 코드와 다이어그램 내용의 차이가 존재한다.

- alert "잔고가 부족합니다"

- Diagram

```

277         if (!atm.validateDes()) {
278             alert("계좌가 존재하지 않습니다.");
279         } else if (!atm.checkBalance()) {
280             alert("잔고가 부족합니다.");
281             process.init();
282             nextPanel(selectMenuView);
283         } else {
284             nextPanel(process.next());
285         }

```

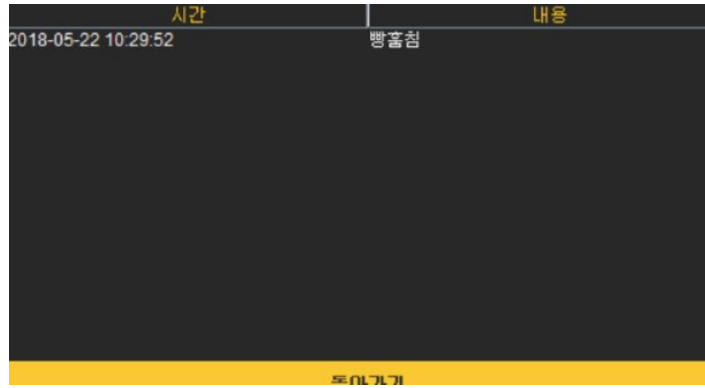


- 코드와 다이어그램 내용의 차이가 존재한다.
- alert "wrong password"
  - Diagram
    - 코드와 다이어그램 내용의 차이가 존재한다. 이는 위의 InputPassword 클래스에서 발생했던 것과 같다.
- update

Name	Update
Type	GUI
Responsibilities	송금이 완료된
Cross Reference	R 5



- 'update'가 맞는 표현이나, 문서에는 'Update'로 표기되어 있다.
- update
  - Post-Condition



- '돌아가기 버튼 또는 끝내기 버튼을 누를 수 있다'라고 명시되어 있으나, 실제 프로그램 동작에서는 돌아가기 버튼만 구현되어있다.

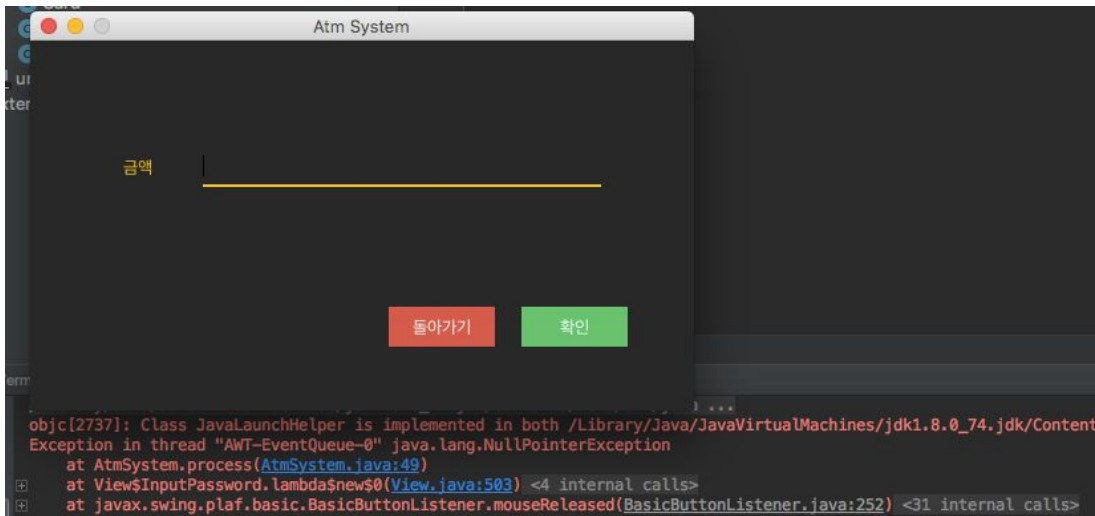
### 1.6 Stage 2060 Unit test & System test

- 2063. System Testing

- ATM\_STC\_003~005
  - '계좌번호에 해당되는 비밀번호를 입력되었을 때 수수료를 제외하고 출금되는지 확인 한다'를 '계좌번호에 해당되는 비밀번호를 입력되었을 때 수수료를 더하고 출금되는지 확인 한다'로 고쳐야 한다.
- ATM\_STC\_005



- 송금 대상 계좌를 입력하면, 정상적인 계좌임에도 불구하고 계좌가 존재하지 않는다는 경고가 발생한다. 결과적으로 송금 기능이 동작하지 않는다. (송금 대상 계좌를 입력할 때, 입력창에 콤마(,)가 찍혀서 출력되는데, 이것이 문제의 원인인 것 같다)
- ATM\_STC\_005\_002~005까지 모두 Pass에서 Fail로 표기해야한다.
- ATM\_STC\_006



- 초기 프로그램 실행 이후 거래기록 조회를 실행해보면, AtmSystem.process에서 에러가 발생한다. 따라서 거래기록을 조회할 수 없다.
- 입금 이후 거래기록을 조회하면 동작하는 것을 확인할 수 있지만, 직전의 입금 기록에 대해서는 조회할 수 없다. 프로그램을 종료하고 새로운 거래를 실행하면, 프로그램 종료 이전의 거래 기록을 확인할 수 있다. 즉, 최신 거래 기록을 바로 확인할 수 없다.
- 프로그램을 실행하는 동안에 발생한 거래 기록에 대해서는 확인할 수 없고, 프로그램 재실행 이후에 확인이 가능하다.
- 여러 오류가 발생하고 있기 때문에 Pass가 아닌 Fail로 표기해야 한다.
- CheckTransactionHistory, CheckCriminalHistory
  - 이력을 조회하고 싶은 계좌를 입력하라고 명시되어있으나, 실제 카드 pin번호를 통한 이력 조회도 가능하기 때문에 System test case의 추가가 필요하다.
- CheckCriminalHistory와 Safe Transaction의 정확한 차이가 무엇인지 모호하다. Safe Transaction 항목은 제거해야할 것으로 보인다.
- OS-Independent
  - 1003에서 Windows 7, Windows 10, Mac OS가 구동 가능한 OS로 명시되어있으나, 2063에서는 'OS에 무관하게'라고 표현되어 있으므로 이에 대한 수정이 필요하다. (실제 시스템케이스를 통과하였으나, 정확히 어떤 OS에 대해서 테스트를 진행했는지 명시해야 한다)
- Simple한 UI/UX에 대한 System test case가 누락되어있다.

## 2. Category-partition Testing Report

### 2.1 Test Case

#### 2.1.1 Testable Unit and Representative Values

Group	Category	Description	Value	Number
Validate	Account	계좌의 존재 유무 확인	계좌번호가 존재할때 / 계좌번호가 존재하지 않을때	1000 / 1001
	Card	카드의 존재 유무 확인	카드가 존재할때 / 카드가 존재하지 않을때	1100 / 1101
	Password	비밀번호 일치 확인	일치하는 비밀번호/일치하지 않는 비밀번호	1200 / 1201
	Account Balance	계좌 내 잔고 확인	계좌의 잔고가 출금하려는 잔고보다 많다(충분) / 계좌의 잔고가 출금하려는 잔고보다 적다(불충분)	1300 / 1301
Transaction	Transaction	입금	타행인경우 거래 금액에서 수수료를 제한 금액을 입금한다(잔액 추가) / 당행인경우 거래 금액을 입금한다(잔액 추가)	2000 / 2001
		출금	타행인경우 거래 금액에서 수수료를 더한 금액을 출금한다(잔액 감소) / 당행인경우 거래 금액을 출금한다(잔액 감소)	2100 / 2101
		송금	타행인경우 거래 금액에서 수수료를 더한 금액을 출금하고(잔액 감소), 송금 대상 계좌의 잔액을 거래 금액만큼 증가시킨다 / 당행인경우 거래 금액을 출금하고(잔액 감소), 송금 대상 계좌의 잔액을 거래 금액만큼 증가시킨다	2200 / 2201
Print	Transaction Result	입,출, 송금 결과 출력	입금, 출금, 송금 후 계좌에 남은 실제 잔액을 화면에 표시한다	3000
	Criminal History	범죄기록 출력	범죄기록이 있는 경우 / 범죄기록이 없는 경우	3100 / 3101
	Transaction History	거래내역 출력	거래내역이 있는 경우 / 거래내역이 없는 경우	3200 / 3201
Select	Medium	매체 선택 (카드/통장)	카드를 선택한 경우 / 통장을 선택한 경우	4000 / 4001
Input	Account Number	계좌번호 입력	숫자로만 입력한 경우 / 기타 문자와 숫자를 섞어 쓴 경우(숫자 이외의 문자가 들어간 경우) / 미입력	5000 / 5001 / 5002
	Card number	카드번호 입력	숫자로만 입력한 경우 / 기타 문자와 숫자를 섞어 쓴 경우(숫자 이외의 문자가 들어간 경우) / 미입력	5100 / 5101 / 5102
	Password Number	비밀번호 입력	비밀번호를 입력받는다 / 입력받지 않은 경우 / 미입력	5200 / 5201 / 5202

	Transaction Amount	거래할 금액 입력	숫자로만 입력한 경우 / 기타 문자와 숫자를 섞어 쓴 경우(숫자 이외의 문자가 들어간 경우) / 미입력/ 음수	5300 / 5301 / 5302 / 5303
Access	Transaction Log	txt 파일에 접근	거래기록 파일이 있는 경우 / 거래기록 파일이 없는 경우	6000 / 6001
	Criminal Log	txt 파일에 접근	범죄기록 파일이 있는 경우 / 범죄기록 파일이 없는 경우	6100 / 6101

### 2.1.2 Error Constraints 적용

- Total Test Case : 442368 -> 143(99.97% 감소)

### 2.1.3 Single Constraints 적용

- Total Test Case : 143 -> 80(44.06% 감소)

### 2.1.4 Property Constraints 적용

- Total Test Case : 80 -> 32(32.00% 감소)

## 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

Bank	Action	Log File	PW	Account	Result
Same	Criminal	Non	-	Not Vaild	P
Different	Criminal	Exist both	-	Vaild	P
Same	History	Exist History Only	Vaild	Vaild	F
Different	Deposit	Exist History Only	Not Vaild	Not Vaild	P
Same	Withdraw	Exist both	Vaild	Not Vaild	P
Different	Criminal	Exist History Only	-	Not Vaild	P
Same	Criminal	Exist Criminal Only	-	Vaild	P



Different	Withdraw	Exist Criminal Only	Not Vaild	Vaild	P
Different	History	Non	Not Vaild	Vaild	F
Same	Transfer	Exist Criminal Only	Vaild	Not Vaild	F
Different	Withdraw	Non	Not Vaild	Vaild	P
Same	History	Exist both	Vaild	Not Vaild	F
Same	History	Exist Criminal Only	Not Vaild	Not Vaild	F
Same	Deposit	Non	Vaild	Vaild	F
Different	Transfer	Exist both	Not Vaild	Vaild	P
Same	Withdraw	Exist History Only	Vaild	Vaild	P
Same	Deposit	Exist both	Vaild	Vaild	F
Same	Deposit	Exist Criminal Only	Vaild	Vaild	P
Same	Transfer	Exist History Only	Vaild	Not Vaild	F
Different	Transfer	Non	Not Vaild	Not Vaild	F

#### 4. Brute Force Testing Report

##### 4.1 Testing Result

Number	Test Case	Result
1	입금 시 입금 금액에 음수를 넣고 확인버튼을 누른다.	P
2	출금 시 출금 금액에 음수를 넣고 확인버튼을 누른다.	P
3	송금 시 송금 금액에 음수를 넣고 확인버튼을 누른다.	P
4	입금 시 입금 금액에 숫자 이외의 문자를 넣고 확인버튼을 누른다.	P
5	출금 시 출금 금액에 숫자 이외의 문자를 넣고 확인버튼을 누른다.	P
6	송금 시 송금 금액에 숫자 이외의 문자를 넣고 확인버튼을 누른다.	P
7	입금 시 입금 금액에 int 범위 이외의 숫자를 넣고 확인버튼을 누른다.	P
8	출금 시 출금 금액에 int 범위 이외의 숫자를 넣고 확인버튼을 누른다.	P
9	송금 시 송금 금액에 int 범위 이외의 숫자를 넣고 확인버튼을 누른다.	P
10	입금 시 입금 금액에 아무것도 넣지 않고 확인버튼을 누른다.	P
11	출금 시 출금 금액에 아무것도 넣지 않고 확인버튼을 누른다.	P
12	송금 시 송금 금액에 아무것도 넣지 않고 확인버튼을 누른다.	P
13	입금 내역이 많을 경우 1초 이내에 처리 되는지 확인 한다.	F
14	출금 내역이 많을 경우 1초 이내에 처리 되는지 확인 한다.	F
15	송금 내역이 많을 경우 1초 이내에 처리 되는지 확인 한다.	F

16	거래 내역이 많을 경우 1초 이내에 처리 되는지 확인 한다.	F
17	범죄 이력이 많을 경우 1초 이내에 처리 되는지 확인 한다.	F
18	거래 내역이 많을 경우, 거래 내역 조회 기능에서 Simple한 UI/UX를 가지는지 확인한다.	F
19	범죄 내역이 많을 경우, 범죄 이력 조회 기능에서 Simple한 UI/UX를 가지는지 확인한다.	F
20	반복적인 거래를 하는 경우, Simple한 UI/UX를 가지는지 확인한다.	F

#### 4.2 Failed Case Report

Number	Test Case	Result
13	입금메뉴 선택 후, log.txt파일이 50MB가 넘은 계좌번호를 입력하고 확인 버튼을 누르면 약 5초 후 메뉴가 표시된다	F
14	출금메뉴 선택 후, log.txt 파일이 50MB가 넘은 계좌번호를 입력하고 확인 버튼을 누르면 약 5초 후 메뉴가 표시된다.	F
15	송금메뉴 선택 후, log.txt 파일이 50MB가 넘은 계좌번호를 입력하고 확인 버튼을 누르면 약 5초 후 메뉴가 표시된다.	F
16	조회메뉴 선택 후, log.txt 파일이 50MB가 넘은 계좌번호를 입력하고 확인 버튼을 누르면 약 5초 후 메뉴가 표시된다.	F
17	범죄이력 조회 메뉴 선택 후, criminalLog.txt 파일이 50MB가 넘은 계좌번호를 입력하고 확인 버튼을 누르면 약 5초 후 메뉴가 표시된다.	F
18	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
19	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
20	여러번 기능을 반복 할 경우 JPanel의 UI Component가 Clear 되지 않고 화면에 겹쳐서 보이는 현상이 나타난다.	F

## 5. Overall

### 5.1 System Test Result

#### 5.1.1 Category-Partition-Test

- 21/32 = 67.74% Pass

#### 5.1.2 Pairwise Test

- 11/20 = 55% Pass

#### 5.1.3 Brute Force Test

- 8/20 = 40% Pass

#### 5.1.4 Summary

- Specification의 구체적인 기준이 제시되어 있지 않고 모호한 것이 많다.
- 문서에서 용어의 통일이 되지 않은 경우가 종종 존재한다.
- 이전 단계에서 수정 사항을 권고하였으나, 수정되지 않은 부분이 다수 존재한다.
- 성능과 사용자 인터페이스와 같은 non functional requirement를 만족시키지 못하였다.
- 문서와 실제 작성된 코드간에 일치하지 않는 부분이 다수 존재한다.