

## Software Verification

# 1<sup>st</sup> Testing – System Test

Team 5

201410373 고예은

201411266 김수현

201411268 김아름

# Index

1. Specification Review .....	3
1.1. Stage 1000 Planning.....	3
1.2. Stage 2030 Analysis .....	3
1.3. Stage 2040 Design .....	4
1.4. Stage 2050/2060 Implementation & Unit Test.....	6
2. Category-partition Testing .....	8
2.1. Test Cases.....	8
2.2. Testing Result.....	10
3. Pairwise Testing .....	12
3.1. Testing Result.....	12
4. Brute Force Testing .....	13
4.1. Testing Result.....	13
4.2. Failed Case Report .....	13
5. Overall .....	14
5.1. System Test Result .....	14

# 1. Specification Review

## 1.1. Stage 1000 Planning

### 1001. Define Draft Plan

#### 4. Non Functional Requirements

- “금융 거래 프로그램이므로 보안이 잘 되어야 한다.”  
-> 이후 문서에서 보안에 대한 명세 존재하지 않음

### 1003. Define Requirements

#### 1. Functional Requirements

- “성공한 거래의 송금금액/입금계좌/이름을 보여줌”  
-> 이후 단계에서 거래 후 잔고를 보여주는 명세가 존재하지 않음  
(실제 구현된 코드에서는 자기 계좌의 남은 잔고만 보여줌)
- “ATM 잔고가 일정수준(50 만원) 이하가 되면 잔고부족 메시지를 출력”  
-> 이후 단계에서 사라짐
- 1001 단계에서는 manage\_ATM 부분이 서술되어 있지 않음  
문서 전체에서 “withdraw”에 대한 해석을 통일하고 있지 않음 (출금/인출)

## 1.2. Stage 2030 Analysis

### 2031. Define Essential Use Cases

2010 까지는 존재하는 ‘수수료’와 ‘명세서 출력’에 대한 서술이 모두 사라짐

#### 1. Withdraw

- “출금 화면을 출력한다” & “출금 금액을 입력한다” 표현 모호
- “ATM의 잔고와 출금 금액을 비교한다”는 내용 추가해야 함

#### 2. Deposit

- “계좌 잔액을 표시한다”는 내용 추가해야 함

#### 3. Remittance

- “계좌 잔액을 표시한다”는 내용 추가해야 함
- Exceptional courses of Events: “자기 자신의 계좌에는 송금할 수 없다”가 추가 되어야 함

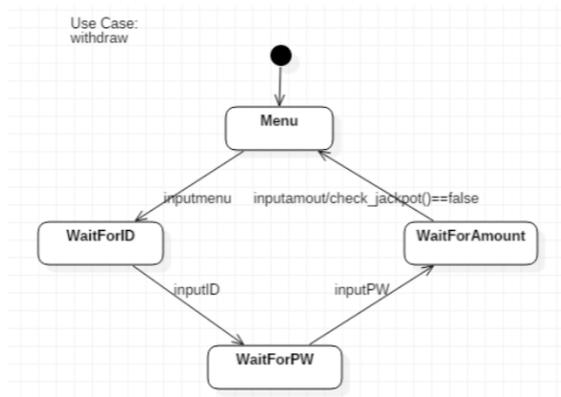
## 5. Random\_Jackpot

- “User 가 돈을 인출했을 때” 표현이 모호함
  - > “User 가 올바른 password 를 입력했을 때”
- pre-requisites: 사용자가 인출 중이어야 한다.
- “당첨되는 순서를 랜덤하게 선택한다.”
  - > Random 범위 설명 필요
- “ATM 의 잔고에서 5 만원을 감소시킨다”는 내용 추가해야 함

## 6. manage\_ATM

- “갱신 후 관리자의 입출금 선택 화면으로 돌아간다”는 내용 추가되어야 함
- Exceptional Courses of Events: ATM 잔고의 최대값에 대한 내용이 추가되어야 함

### 2037. Define State Diagram



- WaitForAmount 와 Menu 사이에 잔고를 보여주는 상태가 존재

### 2038. Refine System Test Case

- 1009 에서 정의한 System test case 와 항목 및 내용이 완전히 달라짐

## 1.3. Stage 2040 Design

### 2041. Design Real Use Cases

2031 바탕으로 전체적인 수정 필요

#### 1. Withdraw

- TotalMoney 뿐만 아니라 지폐권에 따른 각 잔고도 감소시킨다는 내용이 추가되어야 함

## 2. Deposit

- TotalMoney 뿐만 아니라 지폐권에 따른 각 잔고도 감소시킨다는 내용이 추가되어야 함

## 6. manage\_ATM

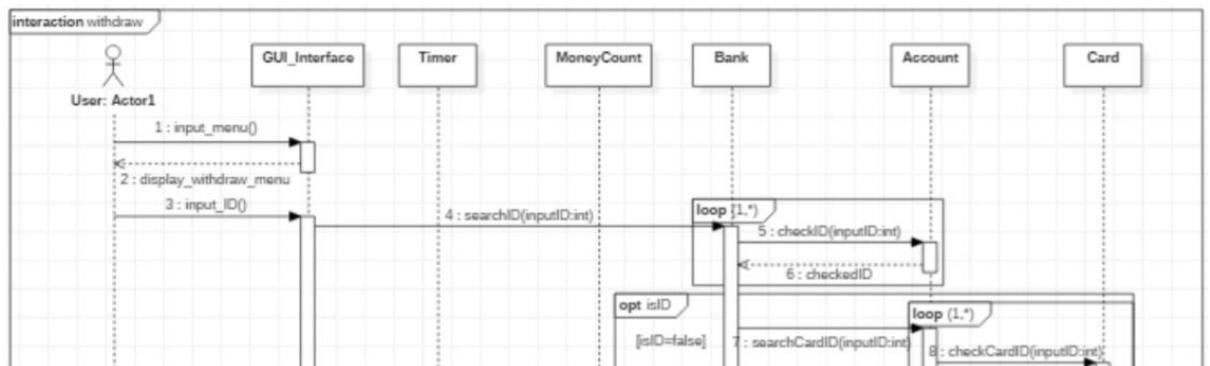
- TotalMoney 뿐만 아니라 지폐권에 따른 각 잔고도 증가/감소시킨다는 내용이 추가되어야 함

## 2042. Define Reports UI, and Storyboards



- '돌아가기' 버튼에 대한 명세가 전혀 없었으나 UI 단계에서 등장
- 금액을 입력할 때 지폐권을 나누어서 입력한다는 명세가 없었음

## 2044. Define Interaction Diagrams



- 실제로 GUI\_Interface 라는 클래스는 존재하지 않음
  - > 대신 System\_Control & MainFrame 존재
- Timer 클래스 사용하지 않는 diagrams에서는 생략 필요

## 1.4. Stage 2050/2060 Implementation & Unit Test

### 2051. Implement Class & Method Definitions

#### Class Definitions – System\_Control

- 명세에 정의되지 않은 메소드가 소스코드 상에 존재  
(getATMcount, get\_inputMoney, inputRID, get\_isjackpot, set\_usercount)

#### 1. System\_Control

- 생성자 메소드에 대한 명세 누락

#### 2. input\_menu

- 명세와 소스코드의 input 값 자료형 불일치
- Abstract Operation: “사용자가 입력한 기능”이라는 표현 모호  
분명한 attribute 로 수정 필요

#### 3. input\_ID

- 명세와 소스코드의 input, output 값 자료형 불일치

#### 4. input\_amount

- 명세와 소스코드의 input, output 값 자료형 불일치
- Purpose, Abstract Operation 에서는 inputAmount attribute 가 존재하나  
소스코드 상에서는 존재하지 않음.

#### 5. input\_PW

- 명세와 소스코드의 input, output 값 자료형 불일치
- Purpose, Abstract Operation 에서는 inputPW attribute 가 존재하나  
소스코드 상에서는 존재하지 않음.

#### 6. updateBalance

- 명세와 소스코드의 output 값 자료형 불일치
- Abstract Operation 에서는 Account.Balance 가 존재하나 소스코드 상에서는  
존재하지 않음.

#### 7. input\_MID

- 명세와 소스코드의 input, output 값 자료형 불일치
- Abstract Operation 에서는 inputID 가 존재하나 소스코드 상에서는 존재하지  
않음.

## 8. input\_POM

- 명세와 소스코드의 input 값 자료형 불일치
- Abstract Operation 에는 inpuPW 가 존재하나 소스코드 상에는 존재하지 않음.
- 오타 : 입력력 > 입력, inpuPW > inputPW

## 9. input\_MPW

- 명세와 소스코드의 input, output 값 자료형 불일치

## Class Definitions - Manager

### 1. Manager

- 생성자 메소드에 대한 명세 누락

### 2. Manager.checkID

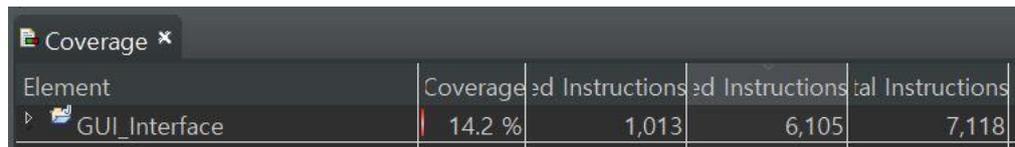
- 소스코드 상에서는 checkID
- Abstract Operation 오타: inputMID > inputID, ManagerIID > ManagerID

### 3. Manager.CheckPW

- 소스코드 상에서는 checkPW
- Abstract Operation 오타 : inputMPW > inputPW

## 2055. Write Unit Test Code

- Unit Test 를 거치지 않은 함수들 존재



Element	Coverage	Covered Instructions	Uncovered Instructions	Total Instructions
GUI_Interface	14.2 %	1,013	6,105	7,118

## 2063. System Testing

- System Testing 결과 차이 존재 (2-1, 3-1, 5-1, 6-1, 7-1: Pass -> Fail)
  - > 숫자가 아닌 다른 character 를 입력하거나 Integer 범위 밖에 있을 때는 오류 메시지를 출력하지 않는다.
- 8-1 “plus 를 입력하였을 때” 표현 매우 모호
- 8-2 “minus 를 입력하였을 때” 표현 매우 모호
- test case 의 내용 자체가 굉장히 광범위하게 작성됨

## 2. Category-partition Testing

### 2.1. Test Cases

- TSL 작성 (Error Constraints, Single Constraints, Property Constraints 적용)

- User

Environments:	
<b>Program Mode:</b>	
Main Menu.	[single]
Withdraw.	[property Withdraw]
Deposit.	[property Deposit]
Transfer.	[property Transfer]
View.	[property View]
Parameters:	
<b>Account ID:</b>	
Invalid ID.	[error]
Correct ID.	
Incorrect ID.	[error]
<b>Receiver ID:</b>	
Invalid ID.	[error][if Transfer]
Correct ID.	[if Transfer]
Incorrect ID.	[error][if Transfer]
<b>Bill:</b>	
50000.	[if Deposit    Withdraw]
10000.	[if Deposit    Withdraw]
5000.	[if Deposit]
1000.	[if Deposit]
<b>Amount:</b>	
Invalid.	[error]
Non.	[error]
One less than Integer boundary.	[if !View]
More than One.	[if Withdraw    Deposit]
Over balance.	[error][if Withdraw    Transfer]
Over Integer boundary.	[if !View]
<b>Password Input:</b>	
Invalid Password.	[error]
Correct Password.	[if !Deposit]
Incorrect Password.	[error]
<b>Return:</b>	
Click Return.	[single]

-> Test Case 개수 32 개

- Manager

**Environment:**

**Program Mode:**

Main Menu.  
Manage ATM.

[single]  
[property Manage\_ATM]

**Parameters:**

**ID Input:**

Invalid ID.  
Correct ID.  
Incorrect ID.

[error]  
[if Manage\_ATM]  
[error]

**Password Input:**

Invalid Password.  
Correct Password.  
Incorrect Password.

[error]  
[if Manage\_ATM]  
[error]

**Select:**

Click Deposit.  
Click Withdraw.

[property Deposit]  
[property Withdraw]

**Bill:**

50000.  
10000.  
5000.  
1000.

**Amount:**

Invalid.  
Non.  
One less than Integer boundary.  
More than One.  
Over balance.  
Over Integer boundary.

[error]  
[error]  
[error][if Withdraw]

**Return:**

Click Return.

[single]

-> Test Case 개수 32 개

## 2.2. Testing Result

Test Suite	Test Case	Latest Exec Result
User	atm-5:Test Case 1	Passed
User	atm-6:Test Case 2	Failed
User	atm-7:Test Case 3	Failed
User	atm-8:Test Case 4	Failed
User	atm-9:Test Case 5	Failed
User	atm-10:Test Case 6	Failed
User	atm-11:Test Case 7	Passed
User	atm-12:Test Case 8	Failed
User	atm-13:Test Case 9	Failed
User	atm-14:Test Case 10	Failed
User	atm-15:Test Case 11	Passed
User	atm-16:Test Case 12	Passed
User	atm-17:Test Case 13	Passed
User	atm-18:Test Case 14	Failed
User	atm-19:Test Case 15	Passed
User	atm-20:Test Case 16	Passed
User	atm-21:Test Case 17	Failed
User	atm-22:Test Case 18	Failed
User	atm-23:Test Case 19	Failed
User	atm-24:Test Case 20	Failed
User	atm-25:Test Case 21	Failed
User	atm-26:Test Case 22	Failed
User	atm-27:Test Case 23	Failed
User	atm-28:Test Case 24	Failed
User	atm-29:Test Case 25	Failed
User	atm-30:Test Case 26	Failed
User	atm-31:Test Case 27	Failed
User	atm-32:Test Case 28	Failed
User	atm-33:Test Case 29	Failed
User	atm-34:Test Case 30	Failed
User	atm-39:Test Case 31	Failed
User	atm-40:Test Case 32	Passed
Manager	atm-41:Test Case 1	Passed
Manager	atm-42:Test Case 2	Failed
Manager	atm-43:Test Case 3	Failed
Manager	atm-44:Test Case 4	Failed
Manager	atm-45:Test Case 5	Failed
Manager	atm-46:Test Case 6	Failed
Manager	atm-47:Test Case 7	Passed
Manager	atm-48:Test Case 8	Failed
Manager	atm-49:Test Case 9	Passed
Manager	atm-50:Test Case 10	Failed

Manager	atm-51:Test Case 11	Failed
Manager	atm-52:Test Case 12	Failed
Manager	atm-53:Test Case 13	Failed
Manager	atm-54:Test Case 14	Failed
Manager	atm-55:Test Case 15	Failed
Manager	atm-56:Test Case 16	Passed
Manager	atm-57:Test Case 17	Failed
Manager	atm-58:Test Case 19	Passed
Manager	atm-59:Test Case 20	Failed
Manager	atm-60:Test Case 21	Failed
Manager	atm-61:Test Case 22	Passed
Manager	atm-62:Test Case 23	Passed
Manager	atm-63:Test Case 24	Failed
Manager	atm-64:Test Case 25	Passed
Manager	atm-65:Test Case 26	Passed
Manager	atm-66:Test Case 27	Failed
Manager	atm-67:Test Case 28	Passed
Manager	atm-68:Test Case 29	Passed
Manager	atm-69:Test Case 30	Failed
Manager	atm-70:Test Case 31	Passed
Manager	atm-71:Test Case 32	Passed
Manager	atm-72:Test Case 33	Failed

21/64 = 32% Pass

### 3. Pairwise Testing

#### 3.1. Testing Result

Test Suite	Test Case	Latest Exec Result
User	atm-73:[Pairwise]Test Case1	Passed
User	atm-74:[Pairwise]Test Case2	Failed
User	atm-75:[Pairwise]Test Case3	Failed
User	atm-76:[Pairwise]Test Case4	Passed
User	atm-77:[Pairwise]Test Case5	Failed
User	atm-78:[Pairwise]Test Case6	Failed
User	atm-79:[Pairwise]Test Case7	Failed
User	atm-80:[Pairwise]Test Case8	Failed
User	atm-81:[Pairwise]Test Case9	Failed
User	atm-82:[Pairwise]Test Case10	Failed
User	atm-83:[Pairwise]Test Case11	Failed
User	atm-84:[Pairwise]Test Case12	Failed
User	atm-85:[Pairwise]Test Case13	Failed
User	atm-86:[Pairwise]Test Case14	Failed
Manager	atm-87:[Pairwise]Test Case1	Failed
Manager	atm-88:[Pairwise]Test Case2	Passed
Manager	atm-89:[Pairwise]Test Case3	Failed
Manager	atm-90:[Pairwise]Test Case4	Passed
Manager	atm-91:[Pairwise]Test Case5	Failed
Manager	atm-92:[Pairwise]Test Case6	Failed
Manager	atm-93:[Pairwise]Test Case7	Failed
Manager	atm-94:[Pairwise]Test Case8	Failed
Manager	atm-95:[Pairwise]Test Case9	Failed
Manager	atm-96:[Pairwise]Test Case10	Failed
Manager	atm-97:[Pairwise]Test Case11	Failed
Manager	atm-98:[Pairwise]Test Case12	Failed

4/26 = 15% Pass

## 4. Brute Force Testing

### 4.1. Testing Result

Test Suite	Test Case	Result
Manager	관리자 프로세스에서 5 번 반복하여 입/출금 할 수 있다.	Failed
Manager	관리자 프로세스에서 10 번 반복하여 입/출금 할 수 있다.	Failed
User	프로그램 초기 실행 시 난수가 제대로 생성된다.	Passed
User	출금 10 번 이내에 Jackpot 이 터지는 지 확인한다.	Passed
User	Jackpot 이 터지면 해당 User 에게 5 만원을 추가 인출하며 ATM 의 잔고에서 5 만원이 빠져나간다.	Failed
User	1 일 1 명만 Jackpot 이 터진다.	Failed
User	출금 시에만 Jackpot 이 터진다.	Failed
User	하나의 통장으로 여러 번 출금할 시 Jackpot 당첨이 여러 번 되지 않게 한다.	Failed
User	모든 입력에 대한 기기의 반응은 1 초 이내로 이루어진다.	Failed
User	거래는 1 분 이내에 이루어져야 한다.	Passed
User	사용자가 보기 편한 화면을 제공해야 한다.	Passed
User	세 번 이상 연속적으로 출금할 수 있다.	Passed
User	세 번 이상 연속적으로 입금할 수 있다.	Failed
User	세 번 이상 연속적으로 송금할 수 있다.	Failed
User	세 번 이상 연속적으로 조회할 수 있다.	Passed
User	출금 시 금액 입력 값에 minus 정수를 입력하면 다음 단계로 넘어가지 않거나, 오류 메시지가 출력된다.	Failed
User	입금 시 금액 입력 값에 minus 정수를 입력하면 다음 단계로 넘어가지 않거나, 오류 메시지가 출력된다.	Failed
User	송금 시 금액 입력 값에 minus 정수를 입력하면 다음 단계로 넘어가지 않거나, 오류 메시지가 출력된다.	Failed
Manager	ATM 잔고가 일정수준(50 만원) 이하가 되면 잔고부족 메시지를 출력한다.	Failed

6/19 = 31% Pass

### 4.2. Failed Case Report

Test Suite	Test Case
Manager	잔고가 어느 일정 범위 이상 커지면 "잔고 확인하세요"라는 오류 메시지가 출력된다. (잔고 최대값 명세에 존재하지 않음)
User	Jackpot 이 터져도 ATM 의 잔고에는 변화가 생기지 않는다.
User	초기 실행 후 Jackpot 이 한 번 터진 후에도 다시 터진다.
User	송금 시 사용자의 비밀번호를 입력하고 나서도 Jackpot 이 터지는 경우 존재
User	integer 이내의 정수값이 아니라면, 1 초 이내로 반응이 없다(반응이 아예 없다)
User	같은 계좌로 출금을 5 회 이상 반복 했을 때, 연속으로 2 번 잭팟이 터지는 경우가 있다
User	잔고가 어느 일정 범위 이상 커지면 "비밀번호 확인하세요"라는 오류 메시지가 출력된다.
User	3 번 이상 송금 하려고 하면 알맞은 비밀번호여도 오류 메시지가 나온다
User	출금 시 minus 정수 값을 금액으로 입력하면 해당 정수의 절대값만큼 계좌의 잔액이 증가한다. 예외처리 필요
User	입금 시 minus 정수 값을 금액으로 입력하면 해당 정수의 절대값만큼 계좌의 잔액이 감소한다. 예외처리 필요
User	송금 시 minus 정수 값을 금액으로 입력하면 해당 정수의 절대값만큼 계좌의 잔액이 증가한다. 예외처리 필요
Manager	ATM 잔고가 50 만원 이하가 되어도 아무 메시지도 출력되지 않는다.

## 5. Overall

### 5.1. System Test Result

- Category Partition Test

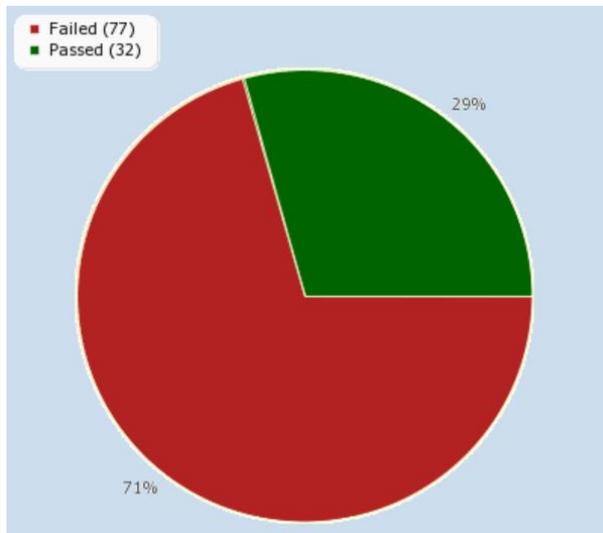
21/64 = 32% Pass

- Pairwise Test

4/26 = 15% Pass

- Brute Force Test

6/19 = 31% Pass



### 5.2. Summary

- 문서들의 단계마다 일치하지 않는 부분들이 있다.
- 실제 구현된 소스코드와 문서 간 차이점이 존재한다.
- 초반 설계한 Functional Requirements 중 구현하지 않은 부분이 있다.
- 보안 등의 Non Functional Requirements에 대한 명세가 Stage 1000에서만 존재, 구현상으로도 minus 입력이 가능한 점 등의 소홀함이 보인다.
- Keyboard 입력에 대한 예외처리가 아예 되어있지 않다.
- Specification이 대체로 명확하지 않고 모호함. 추가로 작성해야 할 것들이 많다.
- 실생활에 활용되는 ATM이라는 점을 고려했을 때 기능이 너무 부실하다.