

Software Verification

2nd 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	3
1.4. Stage 2050/2060 Implementation & Unit Test.....	4
2. Category-partition Testing	6
2.1. Test Cases.....	6
2.2. Testing Result.....	8
3. Pairwise Testing	10
3.1. Testing Result.....	10
4. Brute Force Testing	11
4.1. Testing Result.....	11
4.2. Failed Case Report	11
5. Overall	11
5.1. System Test Result	12

1. Specification Review

1.1. Stage 1000 Planning

1001. Define Draft Plan

2. Project Objective

- Jackpot 기능의 악용가능성(ex. 한사람이 만원씩 지속적으로 출금)을 제거

-> 이후 단계에서 관련 명세가 존재하지 않음

이후 단계에서 시간 차가 있는 출금만 카운트하거나, 5 만원 이상 출금만 카운트하는 식 등의 명확한 방법으로 명세 추가 필요

1.2. Stage 2030 Analysis

2031. Define Essential Use Cases

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

1. Withdraw

- "ATM의 잔고와 출금 금액을 비교한다"는 내용 추가해야 함
- Typical Courses of Events 항목에 13번이 두 개임.
- 수수료에 대한 서술 추가 필요

5. Random_Jackpot

- "당첨되는 순서를 랜덤하게 선택한다."
- > Random 범위 설명 필요
- "ATM 의 잔고에서 5 만원을 감소시킨다"는 내용 추가해야 함

2034. Refine Glossary

- 2040단계에서 View의 부재로 인해 GUI_Interface 관련 항목 수정 필요

2038. Refine System Test Case

- 1009에서 정의한 System test case 와 항목 및 내용이 완전히 달라짐
- > 관련된 test case 끼리 묶고 넘버링 필요

1.3. Stage 2040 Design

2041. Design Real Use Cases

3. remittance

- Exceptional Courses of Events 에 2030 단계에서 존재하는
“E5: 자기 자신의 계좌에는 송금할 수 없다.” 는 내용 누락.

6. manage_ATM

- 다음 단계에서 기술되는 GUI_Interface 가 존재함

1.4. Stage 2050/2060 Implementation & Unit Test

2051. Implement Class & Method Definitions

Class Definitions – Manager

1. Manager.checkID

- Abstract Operation 오타: inputMID > inputID, ManagerIID > ManagerID

2. Manager.CheckPW

- Abstract Operation 오타 : inputMPW > inputPW

Class Definitions – Timer

1. Timer

- 메소드 start 에 대한 명세는 존재하나 소스코드 상에 존재하지 않음.

Class Definitions – MoneyCount

1. ATMBalanceUpdate

- 명세와 소스코드의 output 값의 자료형 불일치. Abstract Operation 에서는 input_count 이나 소스코드 상에서는 inputcount 임

Class Definitions – Bank

1. SearchID

- 명세와 소스코드의 output 값의 자료형 불일치

2. getAccountBalance

- 명세와 소스코드의 output 값의 자료형 불일치

Class Definitions – Account

1. Account.checkID

- Abstract Operation 에서 '찾음'이라는 표현이 모호함.

2. searchCardID

- Abstract Operation 에서 '찾음'이라는 표현이 모호함.

3. balanceAccount

- Abstract Operation 에서 '변화시킨다'는 표현이 증가인지 감소인지 불분명함.

2052. Implement Windows

- MainGUI 라는 표현이 소스코드 상에 존재하지 않음

1. View Account Detail

- Insert_ID 의 Pre-Conditions 가 적합하지 않음.

2. Category-partition Testing

2.1. Test Cases

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

- User

Environments:	
Program Mode:	
Main Menu.	[single]
Withdraw.	[property Withdraw]
Deposit.	[property Deposit]
Transfer.	[property Transfer]
View.	[property View]
Parameters:	
Account ID:	
Invalid ID.	[error]
Correct ID.	
Incorrect ID.	[error]
Receiver ID:	
Invalid ID.	[error][if Transfer]
Correct ID.	[if Transfer]
Incorrect ID.	[error][if Transfer]
Bill:	
50000.	[if Deposit Withdraw]
10000.	[if Deposit Withdraw]
5000.	[if Deposit]
1000.	[if Deposit]
Amount:	
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]
Password Input:	
Invalid Password.	[error]
Correct Password.	[if !Deposit]
Incorrect Password.	[error]
Return:	
Click Return.	[single]

-> Test Case 개수 32 개

- Manager

Environment:	
Program Mode:	
Main Menu.	[single]
Manage ATM.	[property Manage_ATM]
Parameters:	
ID Input:	
Invalid ID.	[error]
Correct ID.	[if Manage_ATM]
Incorrect ID.	[error]
Password Input:	
Invalid Password.	[error]
Correct Password.	[if Manage_ATM]
Incorrect Password.	[error]
Select:	
Click Deposit.	[property Deposit]
Click Withdraw.	[property Withdraw]
Bill:	
50000.	
10000.	
5000.	
1000.	
Amount:	
Invalid.	[error]
Non.	[error]
One less than Integer boundary.	
More than One.	
Over balance.	[error][if Withdraw]
Over Integer boundary.	
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	Passed
User	atm-7:Test Case 3	Passed
User	atm-8:Test Case 4	Passed
User	atm-9:Test Case 5	Passed
User	atm-10:Test Case 6	Passed
User	atm-11:Test Case 7	Passed
User	atm-12:Test Case 8	Passed
User	atm-13:Test Case 9	Passed
User	atm-14:Test Case 10	Passed
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	Passed
User	atm-19:Test Case 15	Passed
User	atm-20:Test Case 16	Passed
User	atm-21:Test Case 17	Passed
User	atm-22:Test Case 18	Passed
User	atm-23:Test Case 19	Passed
User	atm-24:Test Case 20	Passed
User	atm-25:Test Case 21	Passed
User	atm-26:Test Case 22	Passed
User	atm-27:Test Case 23	Passed
User	atm-28:Test Case 24	Passed
User	atm-29:Test Case 25	Passed
User	atm-30:Test Case 26	Passed
User	atm-31:Test Case 27	Passed
User	atm-32:Test Case 28	Passed
User	atm-33:Test Case 29	Passed
User	atm-34:Test Case 30	Passed
User	atm-39:Test Case 31	Passed
User	atm-40:Test Case 32	Passed
Manager	atm-41:Test Case 1	Passed
Manager	atm-42:Test Case 2	Passed
Manager	atm-43:Test Case 3	Passed
Manager	atm-44:Test Case 4	Passed
Manager	atm-45:Test Case 5	Passed
Manager	atm-46:Test Case 6	Passed
Manager	atm-47:Test Case 7	Passed

Manager	atm-48:Test Case 8	Passed
Manager	atm-49:Test Case 9	Passed
Manager	atm-50:Test Case 10	Passed
Manager	atm-51:Test Case 11	Passed
Manager	atm-52:Test Case 12	Passed
Manager	atm-53:Test Case 13	Passed
Manager	atm-54:Test Case 14	Passed
Manager	atm-55:Test Case 15	Passed
Manager	atm-56:Test Case 16	Passed
Manager	atm-57:Test Case 17	Passed
Manager	atm-58:Test Case 19	Passed
Manager	atm-59:Test Case 20	Passed
Manager	atm-60:Test Case 21	Passed
Manager	atm-61:Test Case 22	Passed
Manager	atm-62:Test Case 23	Passed
Manager	atm-63:Test Case 24	Passed
Manager	atm-64:Test Case 25	Passed
Manager	atm-65:Test Case 26	Passed
Manager	atm-66:Test Case 27	Passed
Manager	atm-67:Test Case 28	Passed
Manager	atm-68:Test Case 29	Passed
Manager	atm-69:Test Case 30	Passed
Manager	atm-70:Test Case 31	Passed
Manager	atm-71:Test Case 32	Passed
Manager	atm-72:Test Case 33	Passed

64/64 = 100% 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	Passed
User	atm-75:[Pairwise]Test Case3	Passed
User	atm-76:[Pairwise]Test Case4	Passed
User	atm-77:[Pairwise]Test Case5	Passed
User	atm-78:[Pairwise]Test Case6	Passed
User	atm-79:[Pairwise]Test Case7	Passed
User	atm-80:[Pairwise]Test Case8	Passed
User	atm-81:[Pairwise]Test Case9	Passed
User	atm-82:[Pairwise]Test Case10	Passed
User	atm-83:[Pairwise]Test Case11	Passed
User	atm-84:[Pairwise]Test Case12	Passed
User	atm-85:[Pairwise]Test Case13	Passed
User	atm-86:[Pairwise]Test Case14	Passed
Manager	atm-87:[Pairwise]Test Case1	Passed
Manager	atm-88:[Pairwise]Test Case2	Passed
Manager	atm-89:[Pairwise]Test Case3	Passed
Manager	atm-90:[Pairwise]Test Case4	Passed
Manager	atm-91:[Pairwise]Test Case5	Passed
Manager	atm-92:[Pairwise]Test Case6	Passed
Manager	atm-93:[Pairwise]Test Case7	Passed
Manager	atm-94:[Pairwise]Test Case8	Passed
Manager	atm-95:[Pairwise]Test Case9	Passed
Manager	atm-96:[Pairwise]Test Case10	Passed
Manager	atm-97:[Pairwise]Test Case11	Passed
Manager	atm-98:[Pairwise]Test Case12	Passed

26/26 = 100% 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.	Passed
Manager	관리자 프로세스에서 10 번 반복하여 입/출금 할 수 있다.	Passed
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.	Passed
User	1 일 1 명만 Jackpot 이 터진다.	Passed
User	출금 시에만 Jackpot 이 터진다.	Passed
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.	Passed
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.	Passed
User	세 번 이상 연속적으로 송금할 수 있다.	Passed
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.	Passed
User	입금 시 금액 입력 값에 minus 정수를 입력하면 다음 단계로 넘어가지 않거나, 오류 메시지가 출력된다.	Passed
User	송금 시 금액 입력 값에 minus 정수를 입력하면 다음 단계로 넘어가지 않거나, 오류 메시지가 출력된다.	Passed
Manager	ATM 잔고가 일정수준(50 만원) 이하가 되면 잔고부족 메시지를 출력한다.	Passed
User	출금 시 수수료 1200 원을 더하여 출금한다.	Passed
User	송금 시 수수료 1200 원을 더하여 출금한다.	Passed
User	입금 시 ATM 잔고 지폐 별 500 개를 초과하면 오류 메시지를 출력한다.	Passed
User	입금 시 계좌 잔고 10 억원을 초과하면 오류 메시지를 출력한다.	Passed
User	송금 시 상대방 계좌 잔고 10 억원을 초과하면 오류 메시지를 출력한다.	Failed
User	Jackpot 이 터질 때 ATM 잔고가 부족하면 사용자의 계좌로 5 만원을 입금하고 알림 메시지를 출력한다.	Passed
User	복수의 화폐에서 최대치(500 매)를 넘어선 매수를 입금하려고 하면, 초과된 모든 지폐에 대해 오류 메시지를 출력한다.	Failed
Manager	복수의 화폐에서 최대치(500 매)를 넘어선 매수를 입금하려고 하면, 초과된 모든 지폐에 대해 오류 메시지를 출력한다.	Failed

23/27 = 85% Pass

문서 변경으로 인하여 이전 Case에서 8개 추가됨

4.2. Failed Case Report

Test Suite	Test Case
User	여러 화폐의 최대 수량(500)을 동시에 넘었을 때, 복수의 오류 메시지를 출력하지 않는다.
User	같은 계좌로 출금을 여러 번 반복했을 때, 다시 Jackpot 이 터진다.
User	송금 시 상대방 계좌 잔고 10 억원을 초과하면 잘못된 오류 메시지가 출력된다.

5. Overall

5.1. System Test Result

- Category Partition Test

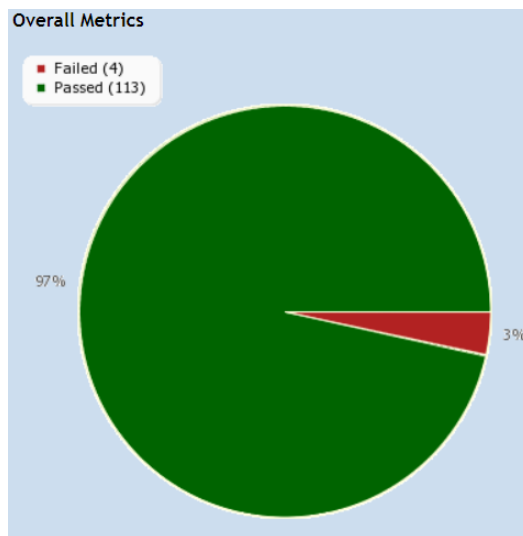
64/64 = 100% Pass

- Pairwise Test

26/26 = 100% Pass

- Brute Force Test

23/27 = 85% Pass



5.2. Summary

- 문서들의 단계마다 일치하지 않는 부분들이 있다.
- 실제 구현된 소스코드와 문서 간 차이점이 존재한다.
- 첫번째 System Testing에 비해 예외처리들이 잘 구현되어 pass의 비율이 높아졌다.
- 아직 Jackpot 중복 당첨 오류가 존재한다.
- 전체적인 문서의 내용이 구체적이지 않다. 89