

Software Verification

<1st Testing – System Test>

T3

201414184 송지연

201611275 윤상혁

201511290 장서연

2018.05.27

List

Specification Review

- 1 Stage 1000
- 2 Stage 2030
- 3 Stage 2040
- 4 Stage 2050 & 2060
- 5 Conclusion

System Testing

- 1 Brute Force Testing
- 2 Category-Partition Testing
- 3 Pairwise Testing
- 4 Conclusion

Specification Review

1 Stage 1000

- 1.1 1페이지, 표지: Date가 2017-04-17로 기재되어 있음
- 1.2 2페이지, 목차: 6. Activity 1006. Define System Architecture 항목의 부재 및 6~10 항목과 실제 문서상 인덱스의 매칭이 안됨.
- 1.3 6페이지, 1002.6: 2017년도 6월까지 개발이 완료되어야 한다.
 - 1.3.1.1 ->2018년도 6월까지
- 1.4 7페이지, 1003.2: 2017년도 6월까지 개발이 완료되어야 한다.
 - 1.4.1.1 ->2018년도 6월까지

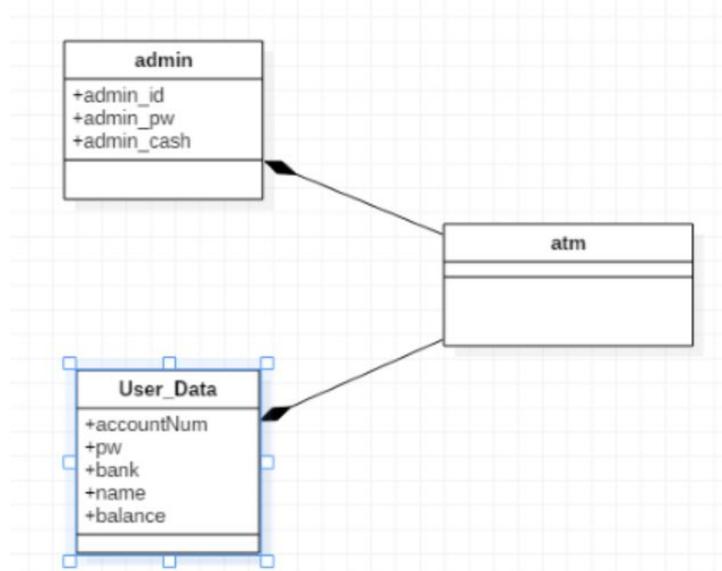
2 Stage 2030

- 2.1 Stage 1000를 수정 후 수정을 요함.
- 2.2 3페이지, 2010: Language와 Size Up의 삭제 등 변경 사항이 stage 1000 문서에 반영되어 있지 않음.
- 2.3 4페이지~7페이지, 2031: Exceptional Courses of Events 항목에서 일정 금액을 초과하면 등의 추상적 표현
-> 구체적인 범위를 적용해야 함. 실제 코드에서는 백만원 이상 출금하면 error처리
- 2.4 5페이지, 2031-2: Exceptional Courses of Events 항목에서 출금 금액이 한계 금액
-> 계좌의 한계 금액인지 ATM 내부의 한계 금액인지 정확한 표현이 필요함.
- 2.5 6페이지, 2031-3: Cross Reference 항목에서 R.3,10,11,12
->R.3,10,11,12로 변경
- 2.6 7페이지, 2031-4: Cross Reference 항목에서 R.5,10,11,12
->R.3,10,11,12로 변경
- 2.7 8페이지, 2031-5: Cross Reference 항목에서 R.7,10,11,12
->R.6,10,11,12로 변경
- 2.8 13페이지, 2033: User_Data 항목에서 stage 1000부터 언급되던 카드의

존재가 사라짐

2.9 13페이지, 2034: User_Data 항목에서 정보들에 카드의 존재가 사라짐

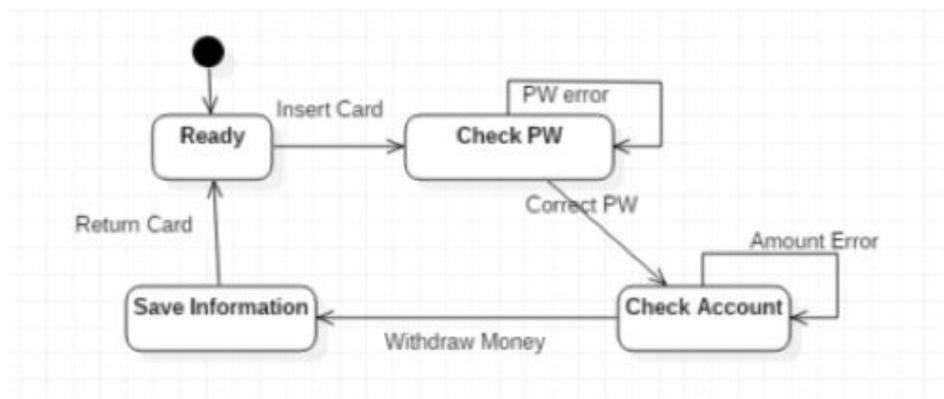
2.10 13페이지, 2034: 다이어그램이 미완성되어 있음



2.11 15페이지, 2036: 본 문서 2.5,6,7과 같은 사항 수정 요함.

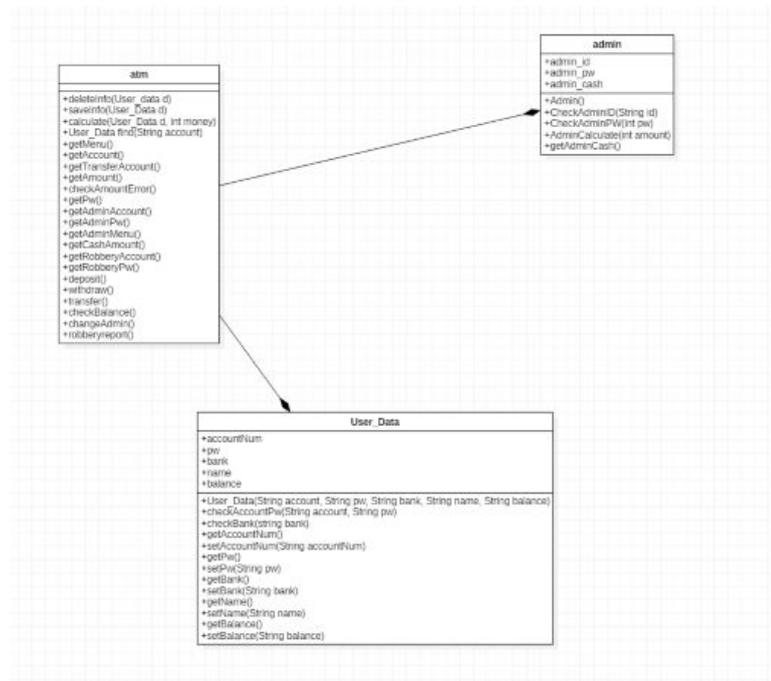
2.12 17페이지, 2036: Withdraw() 테이블에서 Exception에 잔액보다
-> 계좌의 잔액인지 ATM의 잔액인지 명세 필요

2.13 20페이지, 2037: withdraw항목 밖에 작성되지 않았다. 또한 실제 구현에서는 PW Error시 다시 Check PW로 돌아가지 않아 해당 화살표가 Ready를 향해야 한다. Amount Error 또한 같다. Card의 존재가 없으므로 Card 대신 Account라고 수정하여야 한다.



3 Stage 2040

- 3.1 Stage 1000, 2030 수정 후 전체적 수정이 필요
- 3.2 3~9페이지, 2041-1~7: Type 항목이 2030과 달리 Primary and Real로 변경됨
- 3.3 6~7페이지, 2041-4~5: 본 문서 2.6,7과 같은 수정 요함.
- 3.4 12페이지,2043: Refine System Architecture 작성되지 않음
- 3.5 20페이지, 2045: UML에 많은 부분이 추가되고 문서에서 계속 언급되던 카드의 존재 부재



- 3.6 21페이지, 2046: traceability 작성되지 않음

4 Stage 2050 & 2060

- 4.1 3페이지, 2051: Admin class에서 stage 1000를 비롯한 이전 문서에서는 ATM 금액 관리가 DB를 기반한다고 작성 됨
-> DB사용 대신 클래스 변수로 구현 됨
- 4.2 5~8페이지, 2052: GUI가 실제 코드에서 구현되어 있지 않음

-> CUI로 대체 됨

4.3 8페이지, 2067: traceability 작성되지 않음

4.4 이전 문서에서 언급되었던 카드가 구현되지 않았다.

5 문서 분석 결과

- 최근 문서(2050-2060)의 변경사항이 이전 문서(1000, 2030, 2040)에 반영되지 않아 내용의 일관성이 지켜지지 않았다. 변경사항에 대해 모든 문서의 수정이 요구된다.
- Specification의 모든 입력에 대해 기준이 모호하다. 예를 들어, 비밀번호의 Type과 자릿수가 생략되어있다. 이 부분에서 구체적인 기준의 작성이 요구된다.
- 기능 표현에서도 구체적인 작성이 필요하다. 예를 들어, 도난 신고(Robbery Report)의 경우, 도난 신고 이후에 도난 신고 해제의 기능이 없고 신고 후 일어나는 일들에 대해 내용이 생략되어 어떻게 동작되는지 Specification만으로 알 수 없다.

System Testing

- 1 Brute Force Testing
 - Testing Result

#	Testcases	Result
1	메뉴 선택 화면에서 잘못된 입력(number)	Failed
2	메뉴 선택 화면에서 잘못된 입력(malformed)	Failed
3	Database의 계좌를 12자리로 변경 후, 올바른 계좌 번호 입력	Failed
4	Database의 계좌를 malformed(6자리)로 변경 후, 올바른 계좌 번호 입력	Failed
5	Database의 계좌 보유액을 조정(1,000,000,000,000,000) 후, 900,000,000,000,000 출금	Failed
6	Database의 계좌 보유액을 조정(1,000,000,000,000,000) 후, 900,000,000,000,000 송금	Failed
7	Database의 비밀번호를 8자리로 수정 후, 올바른 비밀번호 입력	Failed
8	Database의 비밀번호를 malformed(4자리)로 수정 후, 올바른 비밀번호 입력	Failed
9	Database 고객 이름 변경(4자리 이상)	Failed
10	자기 자신에게 올바른 금액 이체	Failed
11	반복 입금으로 ATM 보유액이 long int 범위 이상	Failed
12	프로그램 종료, 재시작 후 Database 보존 여부 확인	Failed

결과: 0/12 Success 12/12 Failed

2 Category-Partition Testing

○ Testing Result

Category	#	Testcases	Result
Deposit	101	잘못된 계좌번호(number) 입력	Success
	102	잘못된 계좌번호(malformed) 입력	Success
	103	올바른 계좌번호 입력했으나, 잘못된 금액(< 0) 입력	Failed
	104	올바른 계좌번호 입력했으나, 잘못된 금액(malformed) 입력	Failed
	105	올바른 계좌번호 입력 후, 올바른 금액 입력	Success
Withdraw	201	잘못된 계좌번호(number) 입력	Success
	202	잘못된 계좌번호(malformed) 입력	Success
	203	올바른 계좌번호 입력 후, 잘못된 비밀번호(number) 입력	Success
	204	올바른 계좌번호 입력 후, 잘못된 비밀번호(malformed) 입력	Success
	205	올바른 계좌번호와 비밀번호 입력 후, 잘못된 금액(< 0) 입력	Failed
	206	올바른 계좌번호와 비밀번호 입력 후, 잘못된 금액(> AccountAmount) 입력	Success
	207	올바른 계좌번호와 비밀번호 입력 후, 잘못된 금액(> ATMamount) 입력	Failed
	208	올바른 계좌번호와 비밀번호 입력 후, 올바른 금액 입력	Success
Transfer	301	잘못된 출금 계좌번호(number) 입력	Success
	302	잘못된 출금 계좌번호(malformed) 입력	Success
	303	올바른 출금 계좌 입력 후, 잘못된 비밀번호(number) 입력	Success
	304	올바른 출금 계좌 입력 후, 잘못된 비밀번호(malformed) 입력	Success
	305	올바른 출금계좌 와 비밀번호 입력 후, 잘못된 입금 계좌(number) 입력	Success
	306	올바른 출금 계좌와 비밀번호 입력 후, 잘못된 입금 계좌(malformed) 입력	Success
	307	올바른 출금 계좌, 비밀번호, 입금 계좌 입력 후, 잘못된 금액(< 0) 입력	Failed
	308	올바른 출금 계좌, 비밀번호, 입금 계좌 입력 후, 잘못된 금액(< 출금 계좌 보유액) 입력	Success
	309	올바른 출금 계좌, 비밀번호, 입금 계좌 입력 후, 잘못된 금액(malformed) 입력	Success
	310	올바른 출금 계좌, 비밀번호, 입금 계좌 입력 후, 올바른 금액 입력	Success

Check Balance	401	잘못된 계좌번호 입력	Success
	403	올바른 계좌번호 입력 후, 잘못된 비밀번호 입력	Success
	405	올바른 계좌번호 입력 후, 올바른 비밀번호 입력	Success
Robbery Report	501	잘못된 계좌번호 입력	Success
	503	올바른 계좌번호 입력 후, 잘못된 비밀번호 입력	Success
	505	올바른 계좌번호 입력 후, 올바른 비밀번호 입력	Success
AddCash	601	잘못된 관리자 아이디 입력	Success
	602	올바른 관리자 아이디를 입력 했으나, 잘못된 관리자 비밀번호(number) 입력	Success
	603	올바른 관리자 아이디를 입력 했으나, 잘못된 관리자 비밀번호(malformed) 입력	Failed
	604	올바른 관리자 아이디 비밀번호를 입력 했으나, 잘못된 금액(< 0) 입력	Failed
	605	올바른 관리자 아이디 비밀번호를 입력 했으나, 잘못된 금액(malformed) 입력	Success
	606	올바른 관리자 아이디, 비밀번호, 금액 입력	Success
Check Cash	701	잘못된 관리자 아이디 입력	Success
	702	올바른 관리자 아이디를 입력 했으나, 잘못된 관리자 비밀번호(number) 입력	Success
	703	올바른 관리자 아이디를 입력 했으나, 잘못된 관리자 비밀번호(malformed) 입력	Failed
	704	올바른 관리자 아이디, 비밀번호 입력	Success

결과: 31/39 Success 8/39 Failed

3 Pairwise testing
 ○ Testing Result

#	Testcases	Result
105-704	입금 후 ATM 잔액 확인	Success
105-405	입금 후 계좌 잔액 확인	Success
208-704	출금 후 ATM 잔액 확인	Success
208-405	출금 후 계좌 잔액 확인	Success
310-405	송금 후 출금 계좌 잔액 확인	Success
310-405	송금 후 입금 계좌 잔액 확인	Success
505-505	도난 신고 후 도난신고	Failed
505-208	도난 신고 후 출금	Failed
505-105	도난 신고 후 입금	Failed
505-310	도난 신고 후 송금	Failed
505-405	도난 신고 후 계좌 잔액 확인	Failed

결과: 11/11 Succcss 0/11 Failed

4 Conclusion

- GUI 외의 부분만 완성되어, 직접 build하는 방식을 통해 Test가 진행 되어야 했다.
- Category-Partition Testing과 Pairwise Testing에서 진행한 일반적인 기능 부분에서는 어느 정도 작동하나, Malformed나 Invalid 값이 의도적으로 입력되는 상황에서는 예외 처리가 되어있지 않다. 그 예로, 출금 상황에서 금액을 음수로 입력하면 보유 계좌의 잔액이 증가하는 결과를 보인다. Failed한 Test Cases를 기반으로 예외 상황을 방지하는 구현이 요구된다.
- 또한, 대부분의 Failed Test Cases가 Specification의 미비에서 발생한다. 예를 들어, 비밀번호의 경우 실제로는 4자리만 가능하고 숫자, 문자, 특수문자가 관계가 없이 작동하지만 Specification에서는 비밀번호의 형식에 대해 언급되지 않아 Test에 이 부분을 반영할 수 없어 이와 같은 경우에 대해 모두 Failed로 작성하였다.

- 결론적으로 ① 예외 처리 ② Specification 재작성 이 필요하다.