

Software Modeling

# A NEW C.S.E. ATM

1<sup>st</sup> Cycle – System Test

201311299 | 이원오  
201311301 | 이재규  
201311309 | 전홍준

A NEW C.S.E. ATM

# INDEX

- Part 1. Specification Review
- Part 2. Brute Force Test
- Part 3. Category Partition Test
- Part 4. Pairwise Combination Test
- Part 5. Conclusion

# Part 1

## Specification Review

01

02

03

04

05

Stage 1000

Date  
2017-04-17

**2.6 Other Managerial Issues**  
-2017년 6월까지 개발이 완료되어야 한다

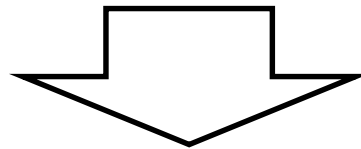
**3.2 Performance Requirements**  
-2017년 6월까지 개발이 완료되어야 한다

**5. Activity 1005. Implement Prototype**  
**6. Activity 1006. Define System Architecture**  
**7. Activity 1007. Define Business Use Case**  
**8. Activity 1008 Define Business Concept Model**  
**9. Activity 1009 Define System Test Case**  
**10. Activity 1010 Refine Plan**

**6. Activity 1006. Define Business Use Case**

**다수의 오타자**

Specification에 빈번한 오타가 발생 및 인덱스 오류가 존재한다.



Stage 1000, 2030, 2040, 2050/2060 보  
고서에 있는 모든 오타 및 인덱스 오류  
수정

01

02

03

04

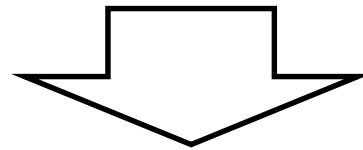
05

Stage 2000-2010

- Activity 1003의 Functional Requirements
- version1에서 잘못 표시된 use case들을 지우고 Actor의 기준에서 생각할 수 있는 case 추가

문서의 일관성 문제

Stage 1000에 반영되지 않은 내용이 2030에 수정사항으로 등장한다.



Stage 1000, 2030, 2040, 2050/2060 보고서에서 일관성이 떨어지는 부분들을 모두 수정.

01

02

03

04

05

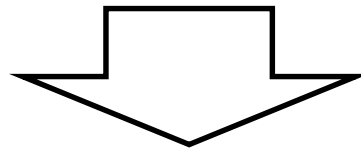
Stage 2030-2031

Use Case	1. Deposit
Actor	User
Purpose	사용자가 원하는 계좌에 입금한다.
Overview	User가 입금처리를 누른다.
Type	Primary and Essential
Cross Reference	Functional Requirements : R 1,10,11,12
Pre-Conditions	N/A
Typical courses of events	(A) Actor (U) System 1. (A) User가 프로그램(UI)에서 Deposit버튼을 누른다. 2. (S) User에게 Deposit UI를 띄운다. 3. (A) 입금할 계좌(or 카드번호)와 비밀번호를 프로그램에 입력한다. 4. (S) 시스템에서 해당 계좌의 비밀번호를 확인한다. 5. (S) 금액 입력으로 UI 변경. 6. (A) User가 입금할 금액을 입력한다. 7. (S) User가 입력한 금액을 해당 계좌나 카드에 추가한다. 8. (S) 변경 내용을 저장한다. 9. (S) 변경 내용을 User에게 보여준다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	1. (S) 잘못된 계좌정보를 입력하거나 입금금액이 일정 금액을 초과하면 입력을 취소한다.

Use Case	2. Withdraw
Actor	User
Purpose	사용자가 원하는 계좌에서 출금을 한다.
Overview	User가 출금처리를 누른다.
Type	Primary and Essential
Cross Reference	Functional Requirements : R 2,10,11,12
Pre-Conditions	N/A
Typical courses of events	(A) Actor (U) System 1. (A) User가 프로그램(UI)에서 Withdraw버튼을 누른다. 2. (S) User에게 Withdraw UI를 띄운다. 3. (A) 출금할 계좌(or 카드번호)와 비밀번호를 프로그램에 입력한다. 4. (S) 시스템에서 해당 계좌의 비밀번호를 확인한다. 5. (S) 금액 입력으로 UI 변경. 6. (A) User가 출금할 금액을 입력한다. 7. (S) User가 입력한 금액을 해당 계좌나 카드에서 빼준다. 8. (S) 변경 내용을 저장한다. 9. (S) 변경 내용을 User에게 보여준다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	1. (S) 잘못된 계좌정보 또는 비밀번호를 입력하거나 출금한 금액이 한계액이나 계좌 잔액을 초과하면 입력을 취소한다.

추상적 표현

Exceptional Courses of Events에서 한계 금액 -> 구체적 표현의 사용 필요



보고서에 있는 추상적인 표현들  
 (ex. 한계 금액)을 구체적으로 표현.  
 (ex. 한계금액 -> 100만원이내)

01

02

03

04

05

Stage 2030-2031

R.1	Deposit	Evident
R.2	Withdraw	Evident
R.3	Check Balance	Evident
R.4	Transfer	Evident
R.5	Language	Evident
R.6	Robbery Report	Evident
R.7	Size Up	Evident
R.8	Add Cash	Evident
R.9	Check Cash	Evident
R.10	Load	Hidden
R.11	Store	Hidden
R.12	Check	Hidden

Use Case	3 Transfer
Actor	User
Purpose	사용자가 원하는 계좌로 자신의 계좌에서 돈을 이체한다.
Overview	User가 계좌이체 버튼을 누른다.
Type	Primary and Essential
Cross Reference	Functional Requirements : R 3.10.11.12

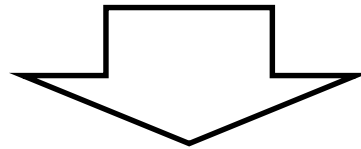
Use Case	4 Check Balance
Actor	User
Purpose	사용자가 원하는 계좌의 잔액을 보여준다.
Overview	User가 잔액확인 버튼을 누른다.
Type	Primary and Essential
Cross Reference	Functional Requirements : R 5.10.11.12

Use Case	5 Robbery Report
Actor	User
Purpose	사용자가 도난 신고를 한다.
Overview	User가 도난 신고 버튼을 누른다.
Type	Primary and Essential
Cross Reference	Functional Requirements : R 7.10.11.12

Use Case	6 Add Cash
Actor	Admin
Purpose	관리자가 ATM에 현금을 추가한다.
Overview	관리자가 비밀번호를 입력해서 현금을 추가한다.
Type	Primary and Essential
Cross Reference	Functional Requirements : R 8.10.11.12

요구사항의 불일치

Cross Reference-Stage1000 항목과 2031의 내용이 일치하지 않음  
2036에서도 같은 사항 발생



Stage 1000에서 요구사항과 2030보고서에서의 내용(ex. Language, Size Up)을 삭제해서 일치시킴.

01

02

03

04

05

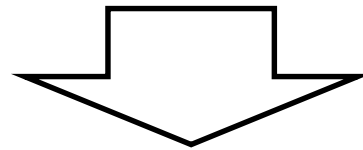
General Problems

GUI 미구현

GUI가 구현되지 않고 CUI로 구현되어 있다.

문서 불완전

2043에서 Refine System Architecture,  
2046, 2067에서 traceability 부분이 작성되지 않았다.



GUI 미구현 -> 설계 및 구현 단계  
문서 불완전 -> 보고서에서 빠져있거나  
적절치 않은 부분들을 전체적으로 추가  
및 수정 완료.



# Part 2

## Brute Force Test

01

02

03

04

05

06

07

08

09

10

Fail Cases	Modified
1. 메뉴 선택 화면에서 잘못된 입력(number / String)	1. 메뉴 선택 시 잘못된 입력에 대한 예외처리 추가.
2. Database의 계좌와 비밀번호의 형식(숫자 6자리/4자리)를 명시하지 않아서 예외의 경우에 대한 처리 불분명	2. 계좌번호와 비밀번호를 받는 메소드인 getPw()와 getAccount() 메소드에 계좌번호와 비밀번호에 대한 정보 등록 및 예외처리 완료.
3. 자기자신에게 이체 하는 경우	3. Transfer 메소드에서 출금 계좌와 이체 계좌가 동일한 경우에 대한 예외처리 완료.
4. 프로그램 종료, 재시작 후 Database 보존 여부	4. Text 파일 입출력을 통해서 변경된 정보들을 저장 및 관리.

01

02

### Fail Case 1

03

04

05

06

07

08

09

10

```
Main (2) [Java Application] C:\Program Files\Java\jdk-9.0.1\bin\javaw.exe (2018. 5. 29. 오후 6:26:52)
메뉴를 입력하십시오 : (0:종료/1:입금 /2:출금 /3:조회/4:이체/5:관리자/6:도난신고)1234897
Wrong Approach!
메뉴를 입력하십시오 : (0:종료/1:입금 /2:출금 /3:조회/4:이체/5:관리자/6:도난신고)dskjhgdjfjhge
Wrong Approach!
메뉴를 입력하십시오 : (0:종료/1:입금 /2:출금 /3:조회/4:이체/5:관리자/6:도난신고)
```

### Fail Case 2

```
메뉴를 입력하십시오 : (0:종료/1:입금 /2:출금 /3:조회/4:이체/5:관리자/6:도난신고)2
계좌번호를 입력하십시오 : (6자리)940621
Password : (4자리)4839
금액을 입력하십시오 : 20000
잔액은 599339
성공!
메뉴를 입력하십시오 : (0:종료/1:입금 /2:출금 /3:조회/4:이체/5:관리자/6:도난신고)
```

01

02

03

04

05

06

07

08

09

10

### Fail Case 3

```

Main (2) [Java Application] C:\Program Files\Java\jdk-9.0.1\bin\javaw.exe (2018. 5. 29. 오후 6:34:03)
메뉴를 입력하십시오 : (0:종료/1:입금 /2:출금 /3:조회/4:이체/5:관리자/6:도난신고)4
계좌번호를 입력하십시오 : (6자리)940621
Password : (4자리)4839
이체할 계좌번호 입력하십시오 : (6자리)940621
자기 자신의 계좌로는 이체할 수 없습니다.
메뉴를 입력하십시오 : (0:종료/1:입금 /2:출금 /3:조회/4:이체/5:관리자/6:도난신고)

```

### Fail Case 4

```

database.txt - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
940621 4839 국민 이재규 619339
930512 2478 신한 한태원 1000000000

```

```

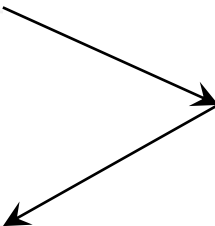
database.txt - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
940621 4839 국민 이재규 649339
930512 2478 신한 한태원 1000000000

```

```

<terminated> Main (2) [Java Application] C:\Program Files\Java\jdk-9.0.1\bin\javaw.exe (2018. 5. 29. 오후 6:37:10)
메뉴를 입력하십시오 : (0:종료/1:입금 /2:출금 /3:조회/4:이체/5:관리자/6:도난신고)1
계좌번호를 입력하십시오 : (6자리)940621
금액을 입력하십시오 : 30000
잔액은 649339
성공!
메뉴를 입력하십시오 : (0:종료/1:입금 /2:출금 /3:조회/4:이체/5:관리자/6:도난신고)0
Exit the ATM...
Finished!

```



# Part 3

## Category Partition Test

# Category Partition Test

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 출금 계좌와 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

01

02

03

04

05

06

07

08

09

10

Fail Cases	Modified
103. 올바른 계좌번호 입력했으나, 잘못된 금액(<0) 입력	103. 금액을 확인하는 메소드인 checkAmountError(int amount)를 수정 및 deposit() 메소드에서 금액 입력 부분 수정.
104. 올바른 계좌번호 입력했으나, 잘못된 금액 (malformed) 입력	104. 금액정보를 받는 메소드인 getAmount()메소드에 금액 정보에 대한 예외처리 추가.
205. 올바른 계좌번호와 비밀번호 입력 후, 잘못된 금액(<0) 입력	205. 금액에 대해서 확인하는 메소드인 checkAmountError() 메소드에 음수에 대한 예외처리 추가.
207. 올바른 계좌번호와 비밀번호 입력 후, 잘못된 금액(ATM amount) 입력	207. 금액에 대해서 확인하는 메소드인 checkAmountError() 메소드에 한계금액에 대한 예외처리 추가.
307. 올바른 출금 계좌, 비밀번호, 입금 계좌 입력 후, 잘못된 금액(<0) 입력	307. 금액에 대해서 확인하는 메소드인 checkAmountError() 메소드에 음수에 대한 예외처리 추가.
603. 올바른 관리자 아이디를 입력 했으나, 잘못된 비밀번호(malformed) 입력	603. 관리자 비밀번호를 받는 메소드인 getAdminPw() 메소드에 잘못된 형식의 비밀번호에 대한 예외처리 추가.
604. 올바른 관리자 아이디 비밀번호를 입력 했으나, 잘못된 금액(<0) 입력	604. 관리자 모드를 관리하는 메소드인 changeAdmin() 메소드에 음수에 대한 예외처리 추가.
703. 올바른 관리자 아이디를 입력 했으나, 잘못된 관리자 비밀번호(malformed) 입력	703. 관리자 비밀번호를 받는 메소드인 checkAdminPw() 메소드에서 pw에 대한 형식을 int에서 String으로 변경.

01

02

03

04

05

06

07

08

09

10

01

02

03

04

05

06

07

08

09

10

## Fail Case 103, 104

```

Main (2) [Java Application] C:\Program Files\Java\jdk-9.0.1\bin\javaw.exe (2018. 5. 29. 오후 7:37:02)
메뉴를 입력하십시오 : (0:종료/1:입금 /2:출금 /3:조회/4:이체/5:관리자/6:도난신고)1
계좌번호를 입력하십시오 : (6자리)940621
금액을 입력하십시오 : -20000
잔액을 입력하십시오!
메뉴를 입력하십시오 : (0:종료/1:입금 /2:출금 /3:조회/4:이체/5:관리자/6:도난신고)1
계좌번호를 입력하십시오 : (6자리)940621
금액을 입력하십시오 : dsfgdgdg
잔액을 입력하십시오!
  
```

## Fail Case 205,207

```

Main (2) [Java Application] C:\Program Files\Java\jdk-9.0.1\bin\javaw.exe (2018. 5. 29. 오후 7:40:23)
메뉴를 입력하십시오 : (0:종료/1:입금 /2:출금 /3:조회/4:이체/5:관리자/6:도난신고)2
계좌번호를 입력하십시오 : (6자리)940621
Password : (4자리)4839
금액을 입력하십시오 : -20000
잔액을 입력하십시오!
메뉴를 입력하십시오 : (0:종료/1:입금 /2:출금 /3:조회/4:이체/5:관리자/6:도난신고)2
계좌번호를 입력하십시오 : (6자리)940621
Password : (4자리)4839
금액을 입력하십시오 : 50000000
잔액을 입력하십시오!
메뉴를 입력하십시오 : (0:종료/1:입금 /2:출금 /3:조회/4:이체/5:관리자/6:도난신고)
  
```



01

02

## Fail Case 307

03

```
Main (2) [Java Application] C:\Program Files\Java\jdk-9.0.1\bin\javaw.exe (2018. 5. 29. 오후 7:44:50)
메뉴를 입력하십시오 : (0:종료/1:입금 /2:출금 /3:조회/4:이체/5:관리자/6:도난신고)4
계좌번호를 입력하십시오 : (6자리)940621
Password : (4자리)4839
이체할 계좌번호 입력하십시오 : (6자리)971111
금액을 입력하십시오 : -20000
잘못된 금액 입력
메뉴를 입력하십시오 : (0:종료/1:입금 /2:출금 /3:조회/4:이체/5:관리자/6:도난신고)|
```

04

05

06

## Fail Case 604, 605, 703

07

08

09

10

```
Main (2) [Java Application] C:\Program Files\Java\jdk-9.0.1\bin\javaw.exe (2018. 5. 29. 오후 7:45:32)
메뉴를 입력하십시오 : (0:종료/1:입금 /2:출금 /3:조회/4:이체/5:관리자/6:도난신고)5
관리자 아이디를 입력하십시오: sklee08
관리자 비밀번호 : (4자리)askldjalsd
Wrong Pw!
메뉴를 입력하십시오 : (0:종료/1:입금 /2:출금 /3:조회/4:이체/5:관리자/6:도난신고)5
관리자 아이디를 입력하십시오: sklee08
관리자 비밀번호 : (4자리)1234
which function would you like to choose? (1:check cash / 2:add cash)2
추가 금액을 입력하십시오 : -20000
cannot enter minue value!
메뉴를 입력하십시오 : (0:종료/1:입금 /2:출금 /3:조회/4:이체/5:관리자/6:도난신고)
```

# Part 4

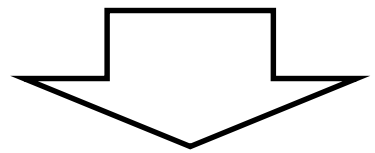
## Pairwise Combination Test

# Pairwise Combination Test

01  
02  
03  
04  
05  
06  
07  
08  
09  
10

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

• 도난신고 후 계좌를 모두 지워 버림 으로서 명세에 적혀있는 기능들이 모두 구현이 안되어 있다.



보고서에서 도난신고에 대한 명세가 적혀있는 부분을 수정 완료. (도난신고가 들어온 계좌는 Database에서 삭제.)

# Part 5

## Conclusion

01

02

03

04

05

06

07

08

09

10

예외 처리

구현 완성

Specification 재작성

Software Modeling

A NEW C.S.E. ATM

Thank you

201311299 | 이원오  
201311301 | 이재규  
201311309 | 전홍준