

Software Modeling & Analysis

Global ATM System

-Stage 2040. Design-

Project Team

1 Team

Date

2018-05-09

Team Information

201311287 엄현식

201311318 최정현

201611293 전다운

목차

1	Activity 2141 Design Real Use Cases	4
1.1	Check	4
1.2	Deposit	4
1.3	Withdraw	5
1.4	Transfer	6
1.5	Issue TrafficCard	7
1.6	Management	8
2	Activity 2142 Define Reports, UI, and Storyboards	9
2.1	. window-1	9
2.2	Window-2	10
2.3	Window-3	10
2.4	Window-4	11
2.5	Window-5	12
2.6	Window-6	12
2.7	window -7	13
2.8	Window-8	14
2.9	Window-9	14
2.10	Window-10	15
2.11	Window-11	16
3	Activity 2043 Refine System Architecture	17
4	Activity 2044 Define Interaction Diagrams	18
4.1	Check	18
4.2	Deposit	18
4.3	Withdraw.....	19

4.4	Transfer	20
4.5	Issue TrafficCard	21
4.6	Management	22
5	Activity 2045 Define Design Class Diagrams	23
6	Activity 2046 Design Traceability Analysis	24

1 Activity 2141 Design Real Use Cases

1.1 Check

Use Case	1. Check
Actors	User
Purpose	User 의 계좌 잔액이 얼마인지 확인한다.
Overview	User 가 읽힌 Item 에 맞는 계좌에 잔액이 얼마인지 은행으로부터 가져와서 User 에게 보여준다.
Type	Primary and Essential
Cross Reference	System Functions: R.2.2 Use Cases:"Status Alarm"
Pre-Requisites	User 의 사용 의사 표현, 올바른 Item 삽입
UI Widgets	Window-1, Window-2,window-3. ,window-4
Typical Courses of Events	(A) : Actor / (S) : System 1. (S) 본인인증 절차를 진행한다. 2. (S) 인증에 성공한 경우, 은행으로부터 해당 계좌의 잔액을 가져온다. 3. (S) Item 을 반환한다. 4. (S) 사용자에게 잔액을 보여준다. 5. (A) 잔액을 확인한다. 6. (S) 명세표를 출력할지 물어본다. 7. (A) 명세표를 출력할지 선택한다. 8. (S) 명세표 출력 용지가 충분한지 선택한다.
Alternative Courses of Events	E0. 한국 계좌의 경우 한국어 모드, 해외 계좌일 경우 외국어 모드 E1. 다른 거래 선택 E7. 명세표 출력시, 계좌 잔액 출력 / 미출력시 다음 단계.
Exceptional Courses of Events	E1. 인증에 실패한 경우, 종료한다 E8. 명세표 출력 용지가 부족할 경우 알림을 발생시킨다.

1.2 Deposit

Use Case	2. Deposit
Actors	User
Purpose	User 가 소지한 지폐(한화/달러)를 계좌에 입금한다.

Overview	User 가 읽힌 Item 에 맞는 계좌에 User 가 삽입한 지폐 액수 만큼의 금액을 입금시켜준다.
Type	Primary and Essential
Cross Reference	System Functions: R.2.0, R.2.2 Use Cases: "Update", "Status Alarm"
Pre-Requisites	User 의 사용 의사 표현, 올바른 Item 삽입 ATM 적정 현금 보유
UI Widgets	Window-1, Window-2, window-3. , window-5
Typical Courses of Events	(A): Actor, (S): System 1.(S) 현금 투입구를 개방한다. 2.(A) 지폐를 투입구에 넣고 투입구 폐쇄를 요청한다. 3.(S) 현금 투입구를 닫는다. 4.(S) 유효한 현금인지 검사한다. 5.(S) 계좌에 해당 금액을 입금한다. 6.(S) Item 을 반환한다. 7.(S) 거래 내역을 보여주고, 명세표를 원하는지 물어본다. 8.(A) 거래 내역을 확인하고, 명세표를 출력할지 선택한다. 9.(S) ATM 현금 보유량이 충분한지 확인한다. 10. (S) 명세표 출력 용지가 충분한지 선택한다.
Alternative Courses of Events	E0. 한국 계좌의 경우 한국어 모드, 해외 계좌일 경우 외국어 모드 E0. 다른 거래 선택 E8. 명세표 출력 시 거래 내역 출력 / 미출력시 다음 단계.
Use Case	E0. ATM 보유 현금이 초과되었을 경우 종료한다. E2. 사용자가 폐쇄요청을 하지 않을 경우, 일정 시간 후에 안내 후 자동 폐쇄한다. E4. 유효한 현금이 아닐 경우, 반환 후 종료한다. E9. 현금 보유량이 많을 경우 알림을 발생시킨다. E10. 명세표 출력 용지가 부족할 경우 알림을 발생시킨다.

1.3 Withdraw

Use Case	3. Withdraw
Actors	User
Purpose	User 가 필요한 현금을 인출한다.

Overview	User 의 본인 계좌임을 확인 후, 원하는 금액만큼 현금(한화/달러)을 인출한다.
Type	Primary and Essential
Cross Reference	System Functions: R.2.0, R.2.1, R.2.2 Use Cases: "Update", "Verify Sufficient Fund", "Status Alarm"
Pre-Requisites	User 의 사용 의사 표현, 올바른 Item 삽입, ATM 적정 현금 보유
UI Widgets	Window-1, Window-2, window-3. ,window-4, window-6 , window-8
Typical Courses of Events	(A): Actor, (S): System 1.(S) 본인확인을 한다. 2.(S) 출금 하고자 하는 지폐의 종류 선택을 요청한다. 3.(A) 한화/달러 중 1 가지 종류의 지폐를 선택한다. 4.(S) 원하는 액수 입력을 요청한다. 5.(A) 원하는 출금 금액을 입력한다. 6.(S) 올바른 금액이 입력되었는지 확인한다. 7.(S) 현금을 꺼낸 뒤, Item 을 반환한다. 8.(A) Item 을 회수한다. 9.(S) 현금 투입구를 개방한다. 10.(A) 현금을 인출한다. 11.(S) 인출 후 투입구를 폐쇄한다. 12.(S) 거래 내역을 보여주고, 명세표를 원하는지 물어본다. 13.(A) 거래 내역을 확인하고, 명세표를 출력할지 선택한다. 14.(S) ATM 현금 보유량이 충분한지 확인한다. 15.(S) 명세표 출력 용지가 충분한지 선택한다.
Use Case	E0. 한국 계좌의 경우 한국어 모드, 해외 계좌일 경우 외국어 모드 E0. 다른 거래 선택 E11. 명세표 출력 시, 거래 내역 출력 / 미출력시 다음 단계.
Actors	E1. 본인확인 실패 시, 종료한다. E5. 계좌 잔액 부족 시, 종료한다. E11. 현금 인출이 일정 시간동안 이루어지지 않는 경우 투입구를 폐쇄하고 다시 입금한다. E14. 현금 보유량이 부족할 경우 알림을 발생시킨다. E15. 명세표 출력 용지가 부족할 경우 알림을 발생시킨다.

1.4 Transfer

Use Case	4. Transfer
Actors	User

Purpose	User 가 자신의 계좌로부터 타계좌로 송금한다.
Overview	User 본인확인 후, 수신 계좌 정보를 입력하고, 원하는 금액을 송금한다.
Type	Primary and Essential
Cross Reference	System Functions: R.2.0, R.2.1 Use Cases: "Update", "Verify Sufficient Fund"
Pre-Requisites	User 의 사용 의사 표현, 올바른 Item 삽입
UI Widgets	Window-1, Window-2,window-3. ,window-4. ,window-7. ,window-8
Typical Courses of Events	(A): Actor, (S): System 1.(S) 본인확인을 진행한다. 2.(S) 수신 계좌 정보 입력을 요청한다. 3.(A) 수신 계좌 정보를 입력한다. 4.(S) 수신 계좌를 확인한다. 5.(A) 수신 계좌를 확인한다. 6.(S) 송금액 입력을 요청한다. 7.(A) 원하는 송금 금액을 입력한다. 8.(S) 올바른 금액이 입력되었는지 확인한다. 9.(S) 송금한다. 10.(S) Item 을 반환한다. 11.(S) 거래 내역을 보여주고, 명세표를 원하는지 물어본다. 12.(A) 거래 내역을 확인하고, 명세표를 출력할지 선택한다. 13.(S) 명세표 출력 용지가 충분한지 선택한다.
Alternative Courses of Events	E0. 한국 계좌의 경우 한국어 모드, 해외 계좌일 경우 외국어 모드 E0. 다른 거래 선택 E12. 명세표 출력 시, 거래 내역 출력 / 미출력시 다음 단계.
Exceptional Courses of Events	E1. 본인확인 실패 시, 종료한다. E4. 올바르지 않은 계좌정보일 경우 종료한다. E8. 계좌 잔액 부족 시, 종료한다. E13. 명세표 출력 용지가 부족할 경우 알림을 발생시킨다.

1.5 Issue TrafficCard

Use Case	5. Issue TrafficCard
Actors	User

Purpose	User 의 계좌와 연동된 선불식 교통 카드를 발급한다.
Overview	User 본인 확인 후, 해당 계좌와 연동된 선불식 교통 카드를 발급한다.
Type	Primary and Essential
Cross Reference	System Functions: R.2.0, R.2.1, R.2.2 Use Cases: "Update", "Verify Sufficient Fund", "Status Alarm"
Pre-Requisites	User 의 사용 의사 표현, 올바른 Item 삽입, ATM 적정 교통카드 보유
UI Widgets	Window-1, Window-2, window-3. ,window-4 ,window-9
Typical Courses of Events	(A): Actor, (S): System 1.(S) 본인확인을 진행한다. 2.(S) 교통 카드 사용 기간 입력을 요청한다. 3.(A) 교통 카드 사용 기간을 입력한다. 4.(S) 교통 카드 발급 비용 및 관련 안내 사항 속지를 확인한다. 5.(A) 교통 카드 발급 안내 사항을 속지하고 발급을 요청한다. 6.(S) 계좌에서 발급 비용을 차감한다. 7.(S) Item 을 반환한다. 8.(S) 교통 카드를 발급한다. 9.(S) 거래 내역을 보여주고, 명세표를 원하는지 물어본다. 10.(A) 거래 내역을 확인하고, 명세표를 출력할지 선택한다. 11.(S) ATM 교통 카드 보유량이 충분한지 확인한다. 12.(S) 명세표 출력 용지가 충분한지 선택한다.
Alternative Courses of Events	E0. 한국 계좌의 경우 한국어 모드, 해외 계좌일 경우 외국어 모드 E0. 다른 거래 선택 E10. 명세표 출력 시, 거래 내역 출력 / 미출력시 다음 단계.
Exceptional Courses of Events	E1. 본인확인 실패 시 종료한다. E5. 안내사항 속지 확인을 거절할 경우 종료한다. E6. 계좌 잔액 부족 시, 종료한다 E11. 교통 카드가 부족할 경우 알림을 발생시킨다. E12. 명세표 출력 용지가 부족할 경우 알림을 발생시킨다.

1.6 Management

Use Case	9. Management
Actor	Admin
Purpose	Admin 이 ATM 의 현금/교통카드/명세표출력용지를 관리한다.
Overview	관리자임을 확인한 후, ATM 의 현금/교통카드/명세표출력용지를 관리할 수 있도록 한다.
Type	Optional
Cross Reference	
Pre-Requisites	
UI Widgets	
Typical Courses of (A): Actor (S):System Events	<p>1. (A) 관리자임을 인증한다.</p> <p>2. (S) ATM 의 현금/교통카드/명세표출력용지 저장소를 개방한다.</p>
Alternative Courses of Events	
Exceptional Courses of Events	E1. 잘못된 접근의 경우, 담당 은행지점 및 관리자에게 알림을 발생시킨다.

2 Activity 2142 Define Reports, UI, and Storyboards

2.1 . window-1

카드/통장 삽입(초기화면)

카드나 통장을 넣어주세요
Please insert Card or Bankbook



GLOBAL ATM

2.2 Window-2
로딩화면

작업중입니다, 잠시만 기다려주세요
Working, Please wait



GLOBAL ATM

2.3 Window-3
서비스 선택

서비스를 선택하세요

- 1 조회
- 2 입금
- 3 출금
- 4 송금
- 5 교통카드 발급



6 취소

GLOBAL ATM

2.4 Window-4

비밀번호 입력

비밀번호를 입력하세요

비밀번호:

1

1	2	3
4	5	6
7	8	9
←	0	

2 취소

GLOBAL ATM

2.5 Window-5

현금 입금



2.6 Window-6

출금 지폐 선택

어떤 출금을 할 것인지 선택하세요

1

원

2

달러

3

취소

GLOBAL ATM

2.7 송금 대상 정보 입력

송금 대상 정보를 입력하세요

1

국민

신한

기업

농협

2

계좌번호:

1

2

3

4

5

6

7

8

9

←

0

3

취소

GLOBAL ATM

2.8 Window-8

금액 입력

금액을 입력하세요

2 **확인**

금액:

1

1	2	3
4	5	6
7	8	9
←	0	만원

3 **취소**

GLOBAL ATM

2.9 Window-9

교통카드 사용기간 입력

언제까지 사용할 것인지 선택하세요

1

05 | May 2018

일 SUN	월 MON	화 TUE	수 WED	목 THU	금 FRI	토 SAT
		1 근로자의날	2	3	4	5 어린이날 / 입하
6	7 대체휴일	8 어린이날	9	10	11	12
13	14	15 수능의날	16	17	18 5.18민주화운동기념일	19
20	21 휴일	22 석가탄신일	23	24	25	26
27	28	29	30	31		

상디닷컴 sang-d.blog.me

2

취소

GLOBAL ATM

2.10 Window-10

교통카드 결제 및 계좌 연동 동의

1

확인

현재 계좌와 교통카드 연동에 동의하고
교통 카드비용 3000원 결제하시겠습니까?

2

취소

GLOBAL ATM

2.11 Window-11

명세표 출력 여부 선택

명세표를 받으시겠습니까?

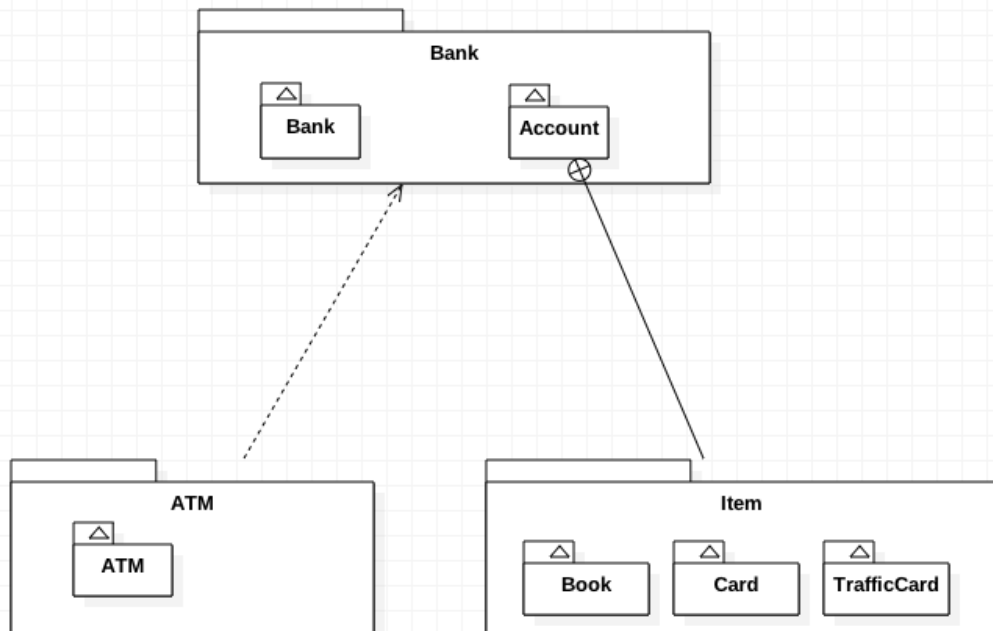
1 명세표 출력

2 명세표 생략

거래 금액:
수수료:
거래후잔액:

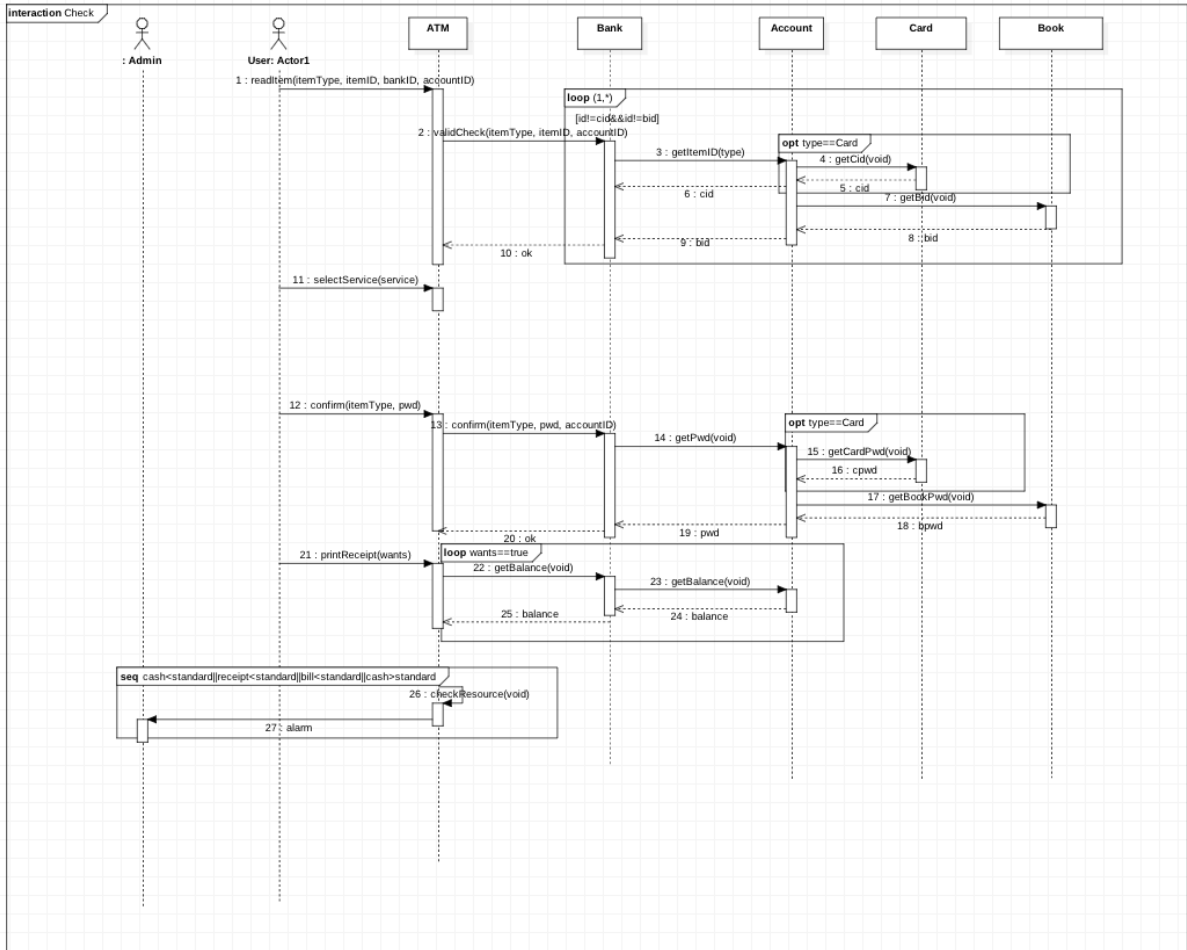
GLOBAL ATM

3 Activity 2043 Refine System Architecture

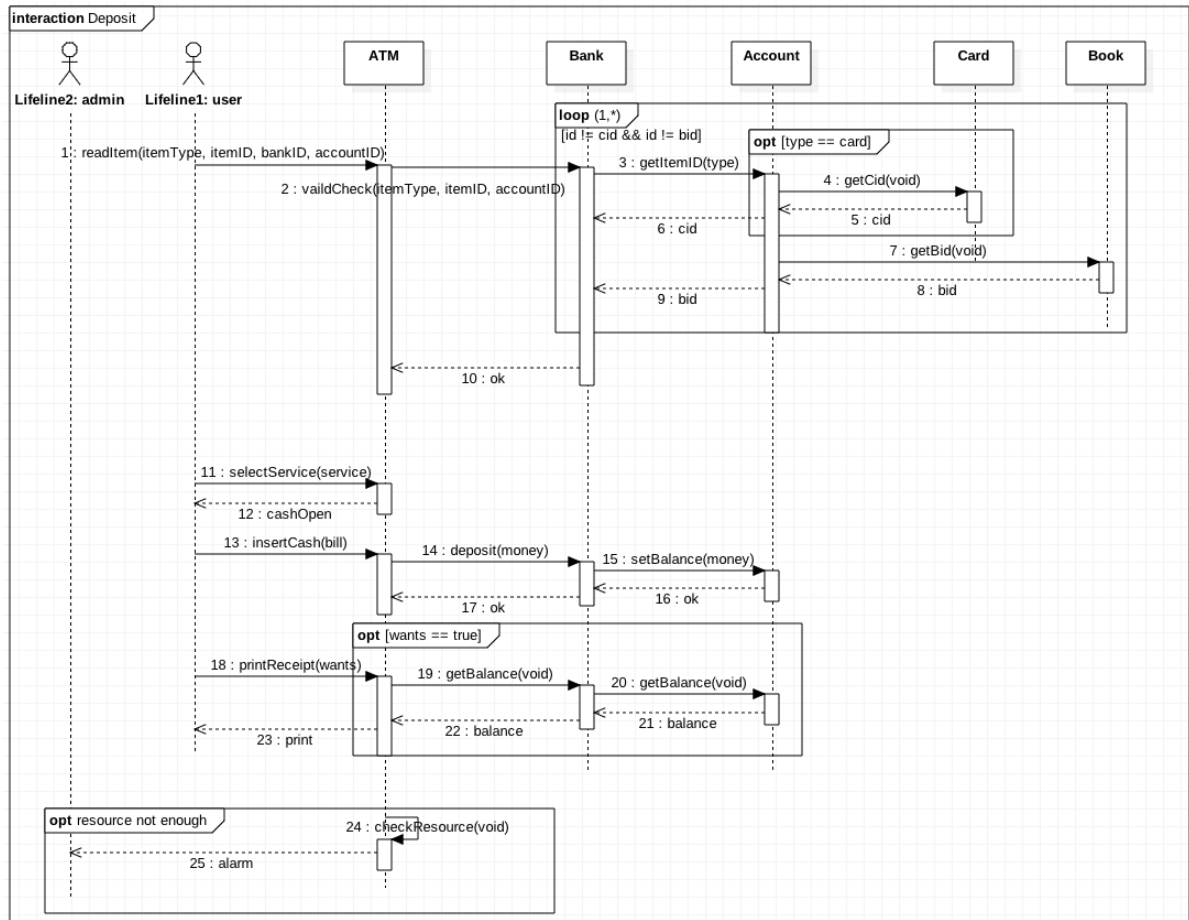


4 Activity 2044 Define Interaction Diagrams

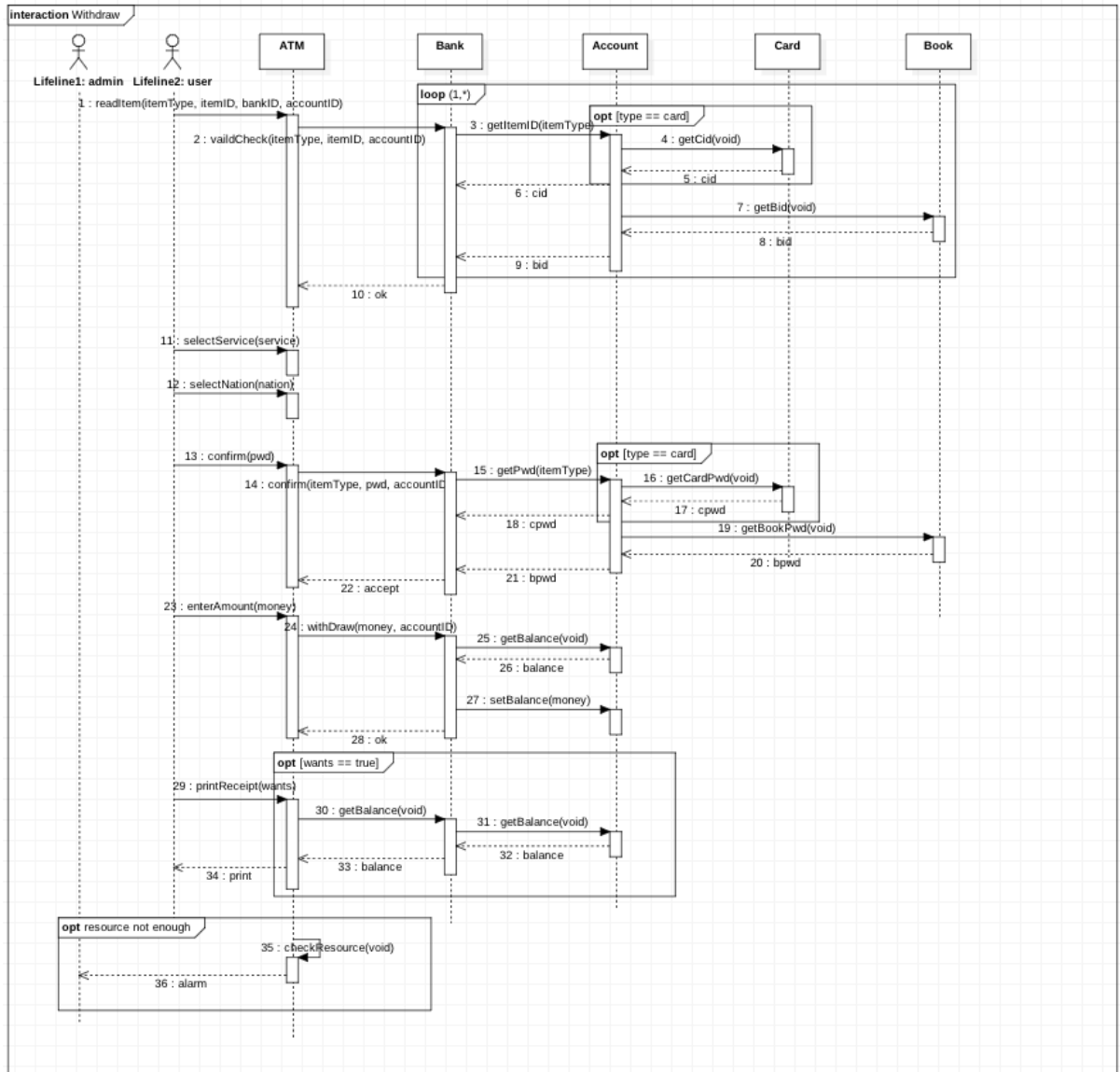
4.1 Check



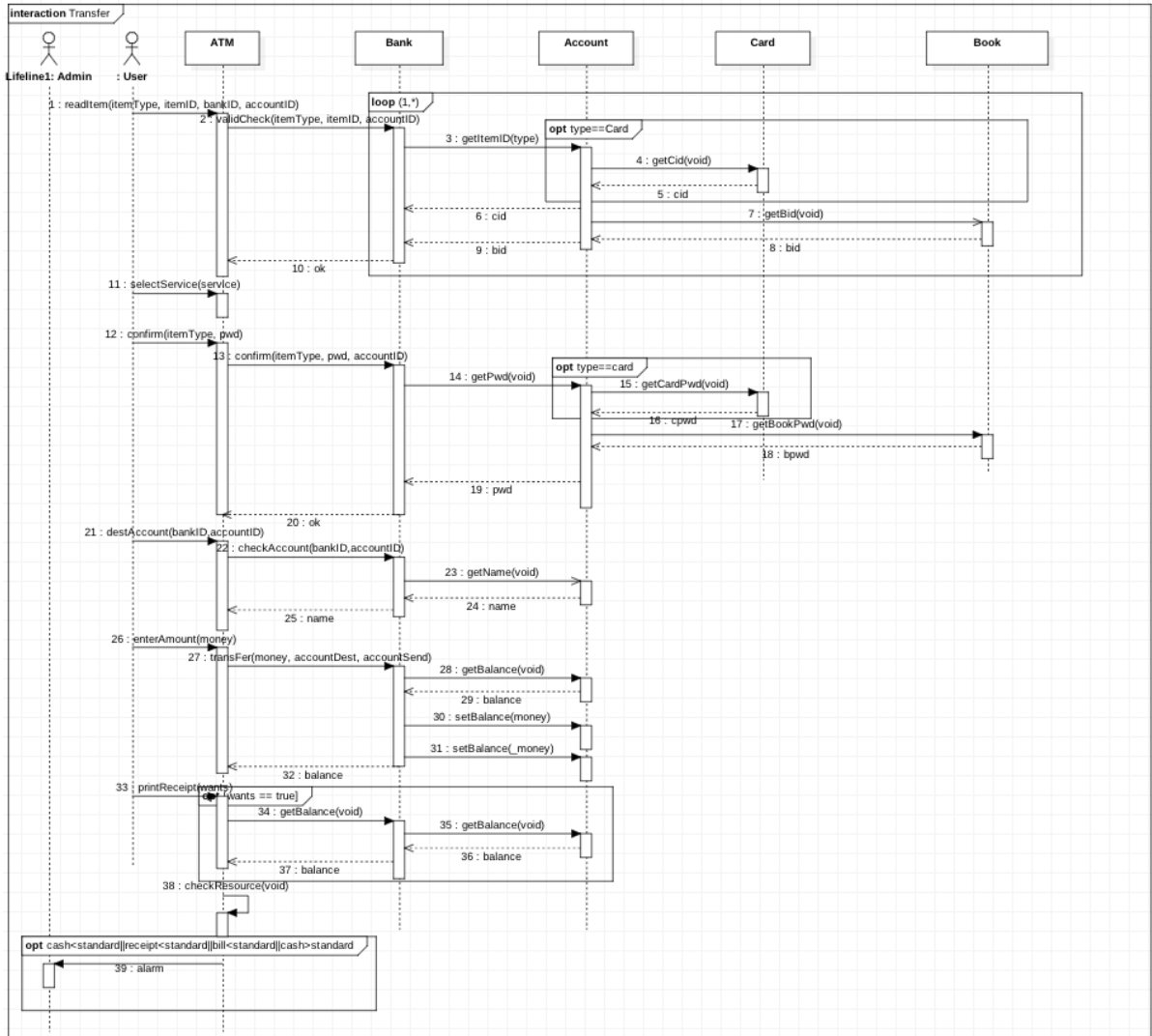
4.2 Deposit



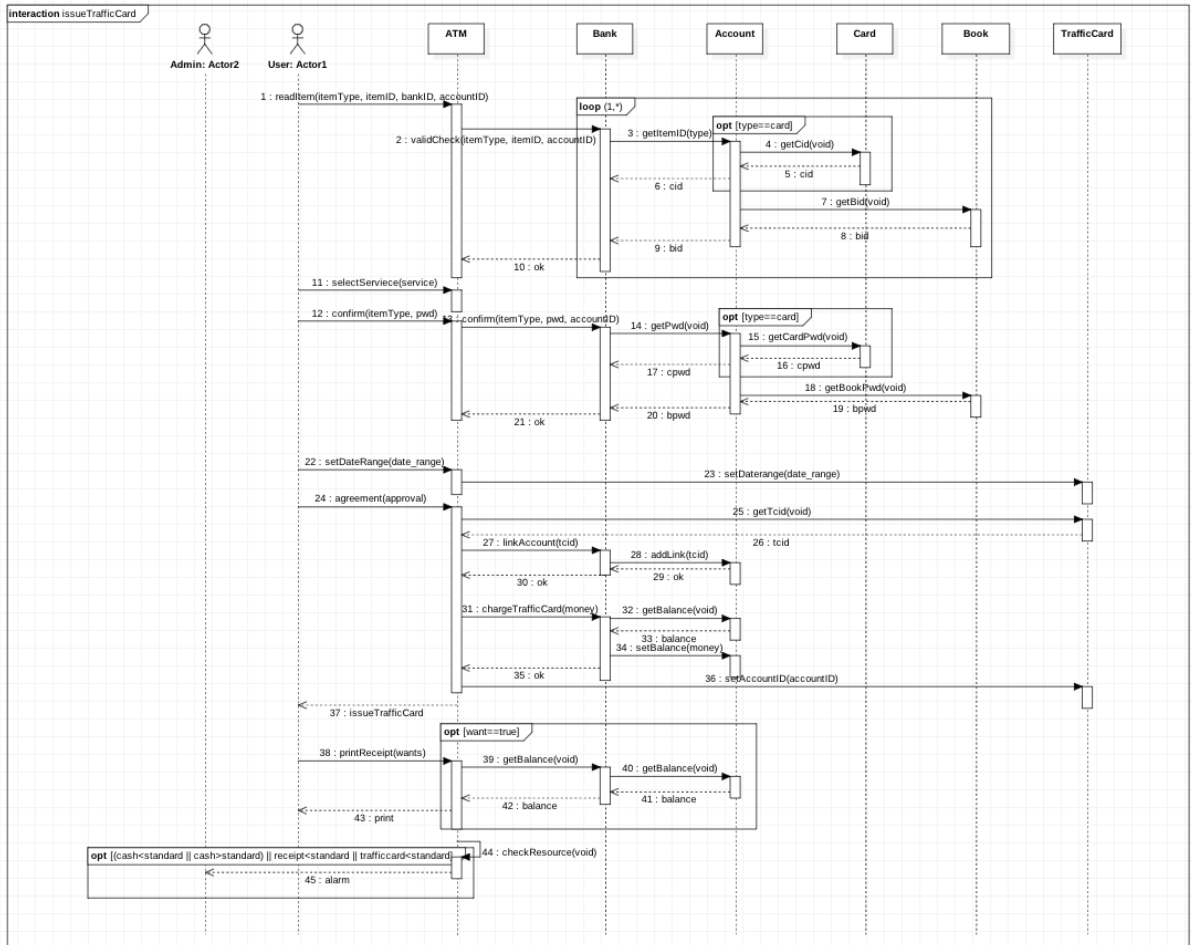
4.3 Withdraw



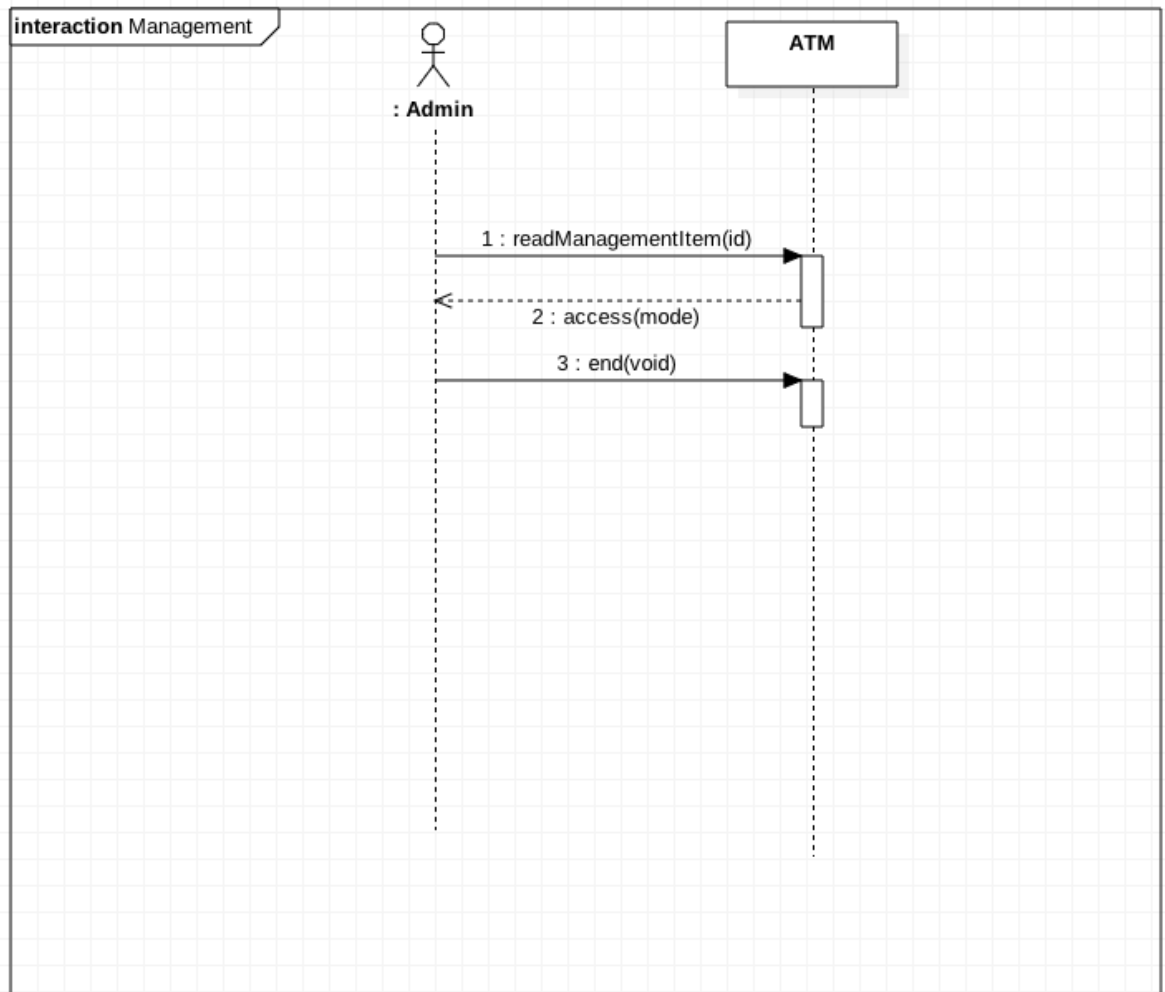
4.4 Transfer



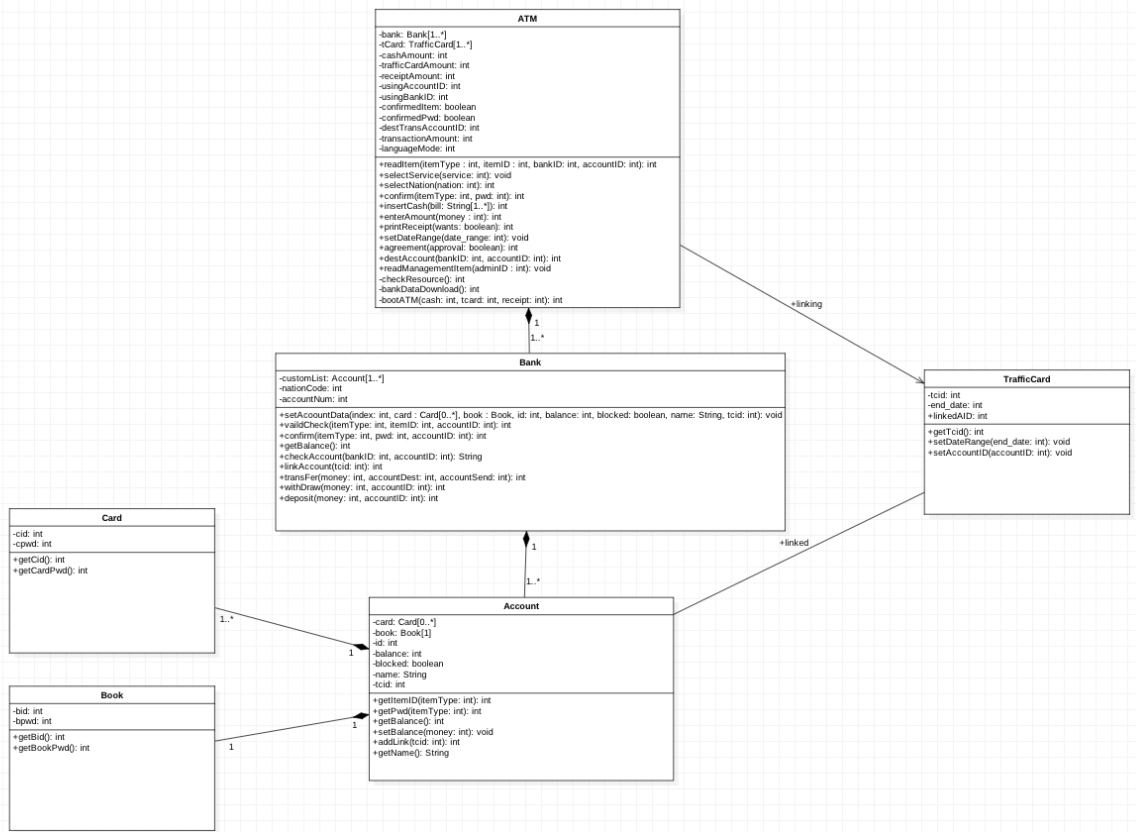
4.5 Issue TrafficCard



4.6 Management



5 Activity 2045 Define Design Class Diagrams



6 Activity 2046 Design Traceability Analysis

Operation in Sequence diagram	Operation in interaction diagram	Method	Class
1. readItem(account)	+readItem(itemType, itemID, accountID)	+readItem(itemType : int , itemID : int , bankID : int , accountID : int): int	ATM
2. selectService(service)	+validCheck(itemType,itemID,accountID)	+selectService(service : int): void	ATM
3. confirm(password)	+getItem(type)	+selectNation(nation : int):int	ATM
4. insertCash(type,amount)	+getCid(void)	+confirm(itemType : int, pwd : int):int	ATM
5. selectMoneyType(type)	+getBid(void)	+insertCash(bill : String[1..*]):int	ATM
6. enterAmounts(amounts)	+seleceService(service)	+enterAmount(money :int):int	ATM
7. destAccount(bank,account)	+confirm(itemType,pwd)	+printReceipt(wants : boolean) : int	ATM
8. setDataRange(data_range)	confirm(itemType,pwd ,accountID)	+setDateRanger(data_range : int):void	ATM
9. agreement(approval)	+getPwd(void)	+agreement(approal : boolean) : int	ATM
10. printReceipt(print)	+getCardPwd(void)	+destAccount(bankID : int, accountID : int) : int	ATM
11. readManagementItem(id)	+getBookPwd(void)	+readManagementItem(adminID : int) : void	ATM
12. end()	+insertCash(bill)	+checkResource() : int	ATM
	+deposit(money)	+bankDataDownload() : int	ATM
	+setBalance(money)	+bootATM(): int	ATM
	+selectNation(nation)		Bank
	+enterAmount(money)	+validCheck(itemType : int,itemType : itemID : int) : int	Bank
	+withDraw(money,accountID)	+confirm(itemType : int, pwd: int , accountID : int) : int	Bank
	+getBalance(void)	+getBalance(tcid :int) : int	Bank
	+getBalance(money)	+checkAccount(bankID : int , accountID : int) : String	Bank
	+destAccount(bankID,accountID)	+linkAccount(tcid :int):String	Bank
	+checkAccount(bankID,accountID)	+transfer(money :int , accountDest : int ,accountSend : int) : int	Bank
	+getName(void)	+withDraw(money : int . accountID : int) : int	Bank
	+setDateRange(date_range)	+deposit (money : int , accountID : int) : int	Bank
	+setDateRange(date_range)	+getTcid() : int	TrafficCard
	+getTcid(void)	+setDateRange(end_date : int) : void	TrafficCard
	+linkAccount(tcid)	+setAccountID(accountID : int) :void	TrafficCard
	+addLink(tcid)	+getTcid() :int	Card
	+chargeTrafficCard(money)	+getCardPwd() :int	Card
	+getBalance(void)	+getBid () : int	Book
	+setBalance(money)	+getBookPwd() : int	Book
	+setAccountID(accountID)	+getItemID(itemType : int) : int	Account
	+printReceipt(wants)	+getPwd(itemType : int) :int	Account
	+getBalance(void)	+getBalance() : int	Account
	+readManagementItem(id)	+setBalance(money : int) : void	Account
	+end(void)	+addLink(tcid :int): int	Account
		+getName() : String	Account