

Object-oriented Analysis and Design  
Term Project 1B

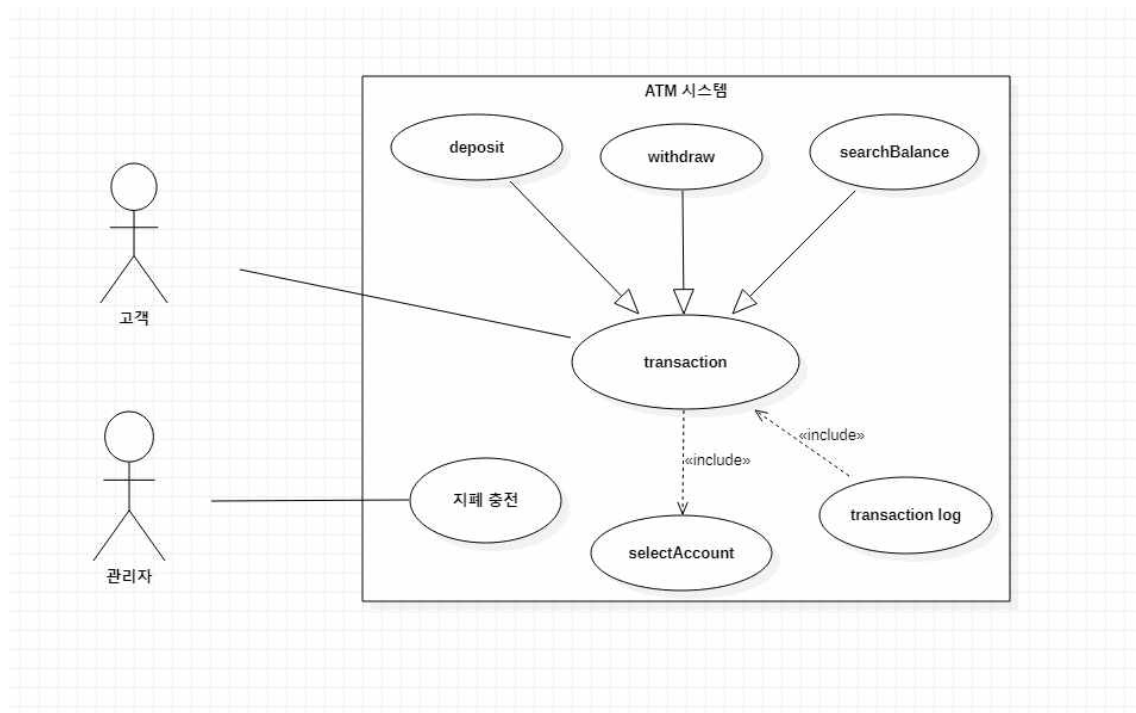
ATM 시스템 개발 보고서  
(10조)

2016125087 변동환  
2018121068 김현재  
2019125071 최지운

## < ATM 요구사항 >

- ATM 시스템에는 ATM 관리자와 고객 총 2가지 종류의 사용자가 있습니다.
- 처음 로그인 화면에서 ATM 관리자는 존재하는 PIN 번호를 이용하고 고객은 회원가입을 통해 고유 PIN 번호와 계좌를 부여받을 수 있습니다.
- 고객은 회원가입시 고유 PIN 번호, 비밀번호, 본인 계좌를 등록하게 됩니다. 계좌 등록 시에는 입출금 계좌와 정기예금 계좌 둘 중 선택하여 계좌를 개설할 수 있습니다.
- ATM 관리자는 기존의 존재하는 PIN 번호를 기반으로 관리자인증을 할 수 있습니다.
- ATM 관리자로 로그인 성공 시 ATM 업무에 필요한 지폐를 충전할 수 있는 기능을 사용할 수 있습니다.
- 사용자는 회원가입을 통해 부여받은 PIN 번호와 PASSWORD를 통하여 사용자 인증을 할 수 있습니다.
- 고객의 PIN 번호로 사용자 인증 성공 시 입금, 출금, 잔고 조회의 기능을 보여주고 고객이 필요한 업무를 선택하여 사용할 수 있도록 합니다.
- 고객이 로그인 이후 사용하는 모든 업무는 ATM 시스템에 모두 기록이 되어야 합니다.
- 고객은 하나의 기능을 수행 시 주어진 시간 동안 하나의 트랜잭션만 수행할 수 있습니다.
- 거래가 끝나면 거래의 성공/실패를 알리기 위해 적절한 메시지가 고객에게 표시되어야 합니다.
- 하나의 거래가 모두 끝나게 되면 고객은 다른 거래를 시작할 수 있습니다.
- ATM 관리자 및 고객이 ATM 업무가 종료하게 되면 자동으로 초기화면으로 변경됩니다.

## < 유스케이스 다이어그램 >



< 유스케이스 시나리오 >

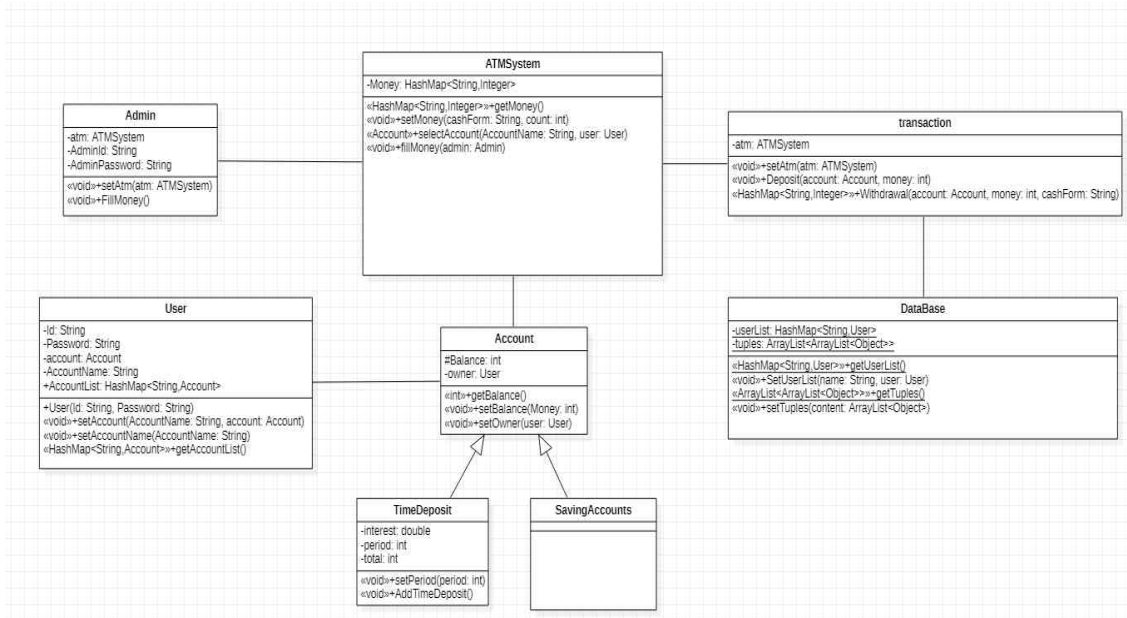
명칭		◇ 입금
(액터) 고객	정의	사용자가 입력한 금액이 계좌에 입금된다.
	기본 흐름	<ol style="list-style-type: none"> <li>1. 사용자는 ATM의 로그인 화면에서 고유 PIN 번호와 PASSWORD를 이용하여 사용자 인증을 한다.</li> <li>2. 사용자 인증 성공 시 ATM 업무를 선택하는 화면을 제공한다.</li> <li>3. 사용자가 금액을 입력하고 입금 버튼을 클릭 시 입금이 성공되었습니다. 메시지와 함께 입금한 금액만큼 증가한 계좌 잔액이 화면에 표시된다.</li> <li>4. 입금 업무가 성공적으로 되었으므로 사용자는 추가업무가 없을 시 뒤로 가기 버튼을 통하여 메인화면으로 돌아간다.</li> </ol>
	예외 흐름	<p>사용자가 입력한 고유 PIN 번호와 PASSWORD가 틀릴 경우</p> <ul style="list-style-type: none"> <li>- “PASSWORD가 틀렸습니다” 메시지를 출력한 후 다시 입력을 받는다.</li> </ul> <p>사용자가 입금화면에서 “뒤로 가기”를 버튼은 클릭하는 경우</p> <ul style="list-style-type: none"> <li>- “뒤로 가기”를 클릭하면 메인화면으로 돌아간다.</li> </ul> <p>사용자가 입금할 금액을 입력하지 않고 입금 버튼을 클릭 시</p> <ul style="list-style-type: none"> <li>- “금액을 입력해주세요”라는 경고 메시지를 뜬다.</li> </ul>

명칭		◇ 출금
(액터) 고객	정의	사용자가 입력한 금액만큼 계좌에서 출금한다.
	기본 흐름	<ol style="list-style-type: none"> <li>1. 사용자는 ATM의 로그인 화면에서 고유 PIN번호와 PASSWORD를 이용하여 사용자 인증을 한다.</li> <li>2. 사용자 인증 성공 시 ATM 업무를 선택하는 화면을 제공한다.</li> <li>3. 출금할 금액을 입력하는 필드를 통해서 사용자는 출금금액을 입력하고 실제 출금은 만원 또는 5만원권 지폐 단위로 선택할 수 있어야 합니다.</li> <li>4. 사용자가 출금 버튼을 클릭하면 입력 필드의 금액에 맞는 지폐 단위로 변환됩니다.</li> <li>6. 정상적으로 출금이 완료되었으면 사용자가 출금하려는 금액에 따라 변환된 “지폐 단위와 출금이 완료되었습니다.” 메시지와 함께 출금한 금액만큼 감소한 계좌 잔액이 화면에 표시된다.</li> <li>7. 출금 업무가 성공적으로 되었으므로 추가적인 업무가 없으면 사용자는 “뒤로 가기” 버튼을 통하여 메인화면으로 돌아간다.</li> </ol>
	예외 흐름	<p>사용자가 출금하려는 금액이 계좌잔고보다 큰 경우</p> <ul style="list-style-type: none"> <li>- “잔액이 부족합니다” 메시지와 함께 현재 계좌 잔고를 보여준다.</li> </ul> <p>사용자 고유 PIN 번호와 PASSWORD가 틀릴 경우</p> <ul style="list-style-type: none"> <li>- “PASSWORD가 틀렸습니다” 메시지를 출력한 후 다시 입력을 받는다.</li> </ul> <p>사용자가 입금화면에서 “뒤로가기”를 요청하는 경우</p> <ul style="list-style-type: none"> <li>- “뒤로가기”를 요청하면 메인화면으로 돌아간다.</li> </ul>

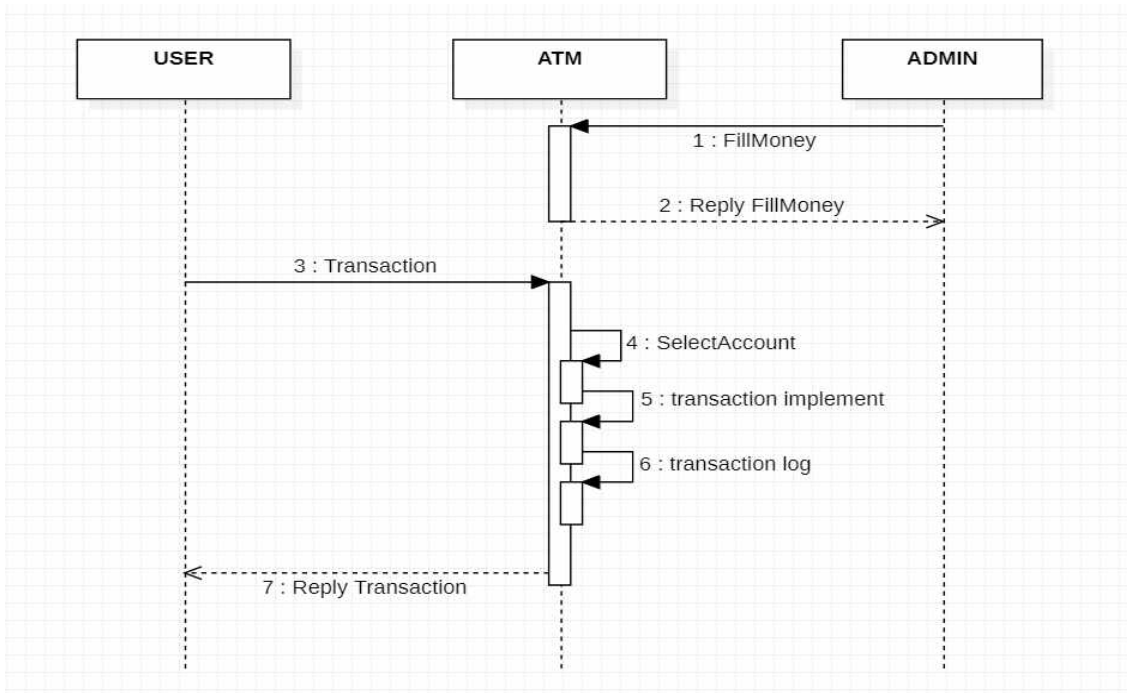
명칭		◇ 잔고 조회
(액터) 고객	정의	사용자의 현재 계좌잔고를 확인할 수 있다.
	기본 흐름	<ol style="list-style-type: none"> <li>1. 사용자는 ATM의 로그인 화면에서 고유 PIN 번호와 PASSWORD를 이용하여 사용자 인증을 한다.</li> <li>2. 사용자 인증 성공 시 ATM 업무를 선택하는 화면을 제공한다.</li> <li>3. 잔고 조회를 요청하면 잔고 조회화면이 나타나고 현재 계좌의 잔액이 표시된 메시지를 보여준다.</li> <li>4. 잔고 금액의 확인이 끝나고 추가적인 업무가 없으면 사용자는 “뒤로 가기” 버튼을 통해서 메인화면으로 돌아간다.</li> </ol>

명칭		◇ ATM 지폐 충전
(액터) 관리자	정의	ATM 관리자는 ATM의 출금을 위한 금액을 넣어놓는다.
	기본 흐름	<ol style="list-style-type: none"> <li>1. 관리자는 ATM의 로그인 화면에서 고유 PIN 번호와 PASSWORD를 이용하여 관리자인증을 한다.</li> <li>2. 관리자인증 성공 시 ATM 업무를 선택하는 화면을 제공한다.</li> <li>3. 관리자가 ATM 지폐 충전 요청 시 ATM에서 1만원권 1000장, 5만원권 200장을 채워 넣는다.</li> <li>4. 업무가 완료되면 관리자는 “뒤로 가기” 버튼을 통해서 메인화면으로 돌아간다.</li> </ol>
	예외 흐름	<p>관리자 고유 PIN 번호와 PASSWORD가 틀릴 경우</p> <p>- “ PASSWORD가 틀렸습니다” 메시지를 출력한 후 다시 입력을 받는다.</p>

## < 클래스 다이어그램 >



## < 시퀀스 다이어그램 >



클래스 명칭	◇ ATM System	
정의	ATM의 전반적인 업무를 수행하는 클래스이다.	
필드 (field)	Money: HashMap< String, int >	클래스의 필드를 HashMap으로 정의하여 key값으로 1만원, 5만원 지폐를 구분하여 value값을 저장한다.
메소드 (method)	SelectAccount (user:User): void	입출금, 정기예금 계좌 둘중 하나를 선택할 수 있는 메소드이다.
	FillMoney (admin:Admin): void	ATM관리자가 매일 아침 업무 시작을 위해 ATM에 1만원권, 5만원권을 충전해주는 메소드이다.
	TransactionMethod (trans:Transaction): void	계좌의 트랜잭션(입금, 출금, 잔고조회)를 할 수 있는 메소드이다.

클래스 명칭	◇ Transaction ( ATM System의 내부 클래스 )	
정의	ATM의 입금, 출금, 잔고 조회를 수행하는 클래스이다.	
필드 (field)	atm: ATMSystem	고객이 출금 시 ATM에서 저장된 지폐를 사용하기 위한 ATM객체이다.
메소드 (method)	setAtm ( atm:ATMSystem ): void	현재 ATM정보를 필드에 설정한다.
	Deposit ( account:Account): void	선택된 고객의 계좌에 사용자가 입력한 금액을 입금하는 메소드이다.
	Withdrawal ( account:Account): void	선택된 고객의 계좌에 사용자가 입력한 금액을 출금하는 메소드이다.
	CheckBalance ( account:Account): void	고객의 계좌 잔고를 출력해주는 메소드이다.

클래스 명칭	◇ Account	
정의	고객의 계좌를 정의하는 클래스이다.	
필드 (field)	Balance : int	계좌의 잔고를 정의하는 클래스 필드이다.
	owner : User	클래스 자료형으로 User class의 객체를 필드에 가져온다
메소드 (method)	getBalance() : int	고객의 계좌의 잔고를 반환하는 메소드이다.
	setBalance(int Money) : void	입금후에 계좌 잔고에 돈을 추가하는 메소드이다.
	setOwner(User user) : void	이 계좌의 고객을 설정하는 메소드이다.

클래스 명칭	◇ User	
정의	고객을 정의하는 클래스이다.	
필드 (field)	Id: String	고객의 아이디를 나타내는 필드이다.
	passWord: String	고객의 비밀번호를 나타내는 필드이다.
	account Account	클래스 자료형으로 고객의 계좌정보를 나타내는 필드이다.
	AccountName String	계좌종류의 이름을 나타내는 필드이다.
	AccountList HashMap<String,Account>	고객의 계좌정보를 저장하는 필드
메소드 (method)	setAccount (String AccountName, Account account): void	사용자의 계좌를 등록하는 메소드이다
	setAccountName (String AccountName):void	계좌의 종류를 설정하는 메소드이다.
	getAccountList(): HashMap<String,Account>	고객의 계좌정보를 반환하는 메소드이다.

클래스 명칭	◇ Admin	
정의	ATM 관리자를 정의하는 클래스이다.	
필드 (field)	atm: ATMSysstem	ATMSysstem class의 객체를 필드에 저장한다.
	AdminId: String	관리자의 Id를 나타내는 필드이다.
	AdminPassword: String	관리자의 password를 나타내는 필드이다.
메소드 (method)	setAtm (ATMSysstematm):void	현재의 ATM정보를 필드에 있는 객체에 저장해주는 메소드이다.
	FillMoney() :void	ATM에 돈을 채워넣는 메소드이다.

클래스 명칭	◇ DataBase	
정의	ATM안에서 일어나는 업무(transaction)을 저장하는 클래스이다.	
필드 (field)	<u>userList</u> : HashMap<String,User>	고객들을 저장하는 필드이다.
	<u>tuples</u> : ArrayList<ArrayList<Object>>	table을 이용하여 ATM의 모든 트랜잭션을 기록하는 필드이다.
메소드 (method)	<u>getUserList()</u> : HashMap<String,User>	고객정보들을 반환해주는 메소드이다.
	<u>setUserList</u> (String name, User user): void	고객정보를 저장하는 메소드이다
	<u>setTuples</u> (ArrayList <Object> content): void	행 단위로 테이블에 트랜잭션을 저장하는 메소드이다.
	<u>getTuples()</u> : ArrayList<ArrayList<object>>	테이블을 반환하는 메소드이다.

클래스 명칭	◇ TimeDeposit	
정의	정기예금계좌를 나타내는 클래스이다.	
필드 (field)	interest: double	월별 이자율을 나타내는 필드이다.
	period: int	사용자가 선택한 기간을 나타내는 필드이다.
	total: int	전체 금액을 저장하는 필드이다.
메소드 (method)	<u>setPeriod</u> (int period):void	기간을 설정하는 메소드이다.
	<u>AddTimeDeposit</u> (int money, int period) : int	현재 몇 개월째 예금하고 있는지를 반환하는 메소드이다.