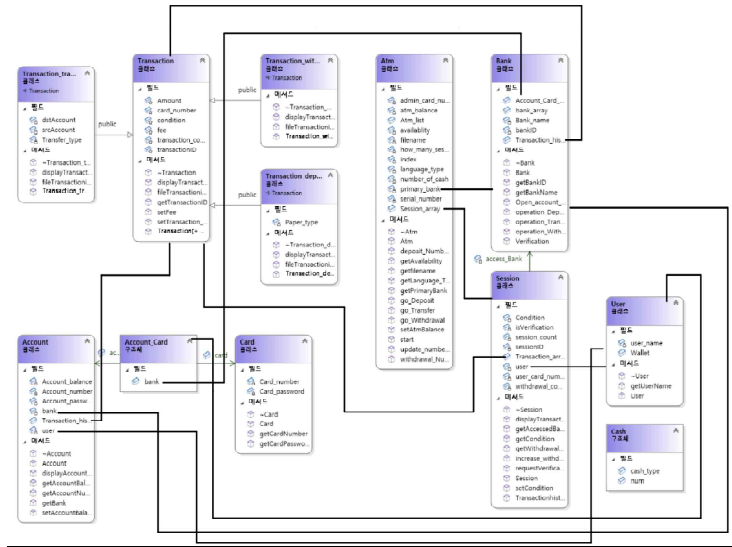


A) final class diagram



<그림 1 : Final class diagram>

위 그림은 final class diagram을 그린 것입니다. transaction class와 연결된 하얀색 세모는 상속을 나타냅니다.

B) The list of requirements that have been successfully implemented and The console screenshots to demonstrate the successfully implemented requirements.

(REQ1.1) An ATM has a 6-digit serial number that can be uniquely identified among all ATMs (e.g., 315785).

```

Atm::Atm(Bank* primary_bank, language_t language_type, available_t availability, int serial_number
: primary_bank(primary_bank), atm_balance(atm_balance), language_type(language_type), availab
serial_number(serial_number), how_many_sessions(0), index(index), admin_card_number(admin_car
Atm_list.push_back(this);
int count = (static_cast<int>(atm_balance / pow(10, 6))) == 0 ? 1 : atm_balance / pow(10,
int num[4] = {10, 8, 20, 15};
for (int i = 0; i < 4; i++)
    number_of_cash[i].num = num[i] * count;
}
    
```

<그림 2 : REQ 1-1 | atm의 serial_num 초기화 코드>

```
Atm* atm1 = new Atm(Kakao, UNI, MULTI, 123333, 12345678, 0, "atm1", 10000000);
Atm* atm2 = new Atm(Shinhan, BI, SINGLE, 123456, 23456789, 1, "atm2", 10000000);
```

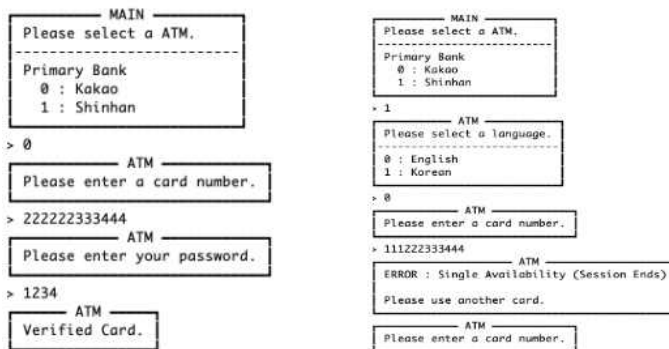
<그림 3 : REQ 1-1 | main 함수에서 atm_serial_number 지정하기>

```
class Atm {
private:
    string filename;
    int index;
    long long admin_card_number;
    Bank* primary_bank;
    language_t language_type;
    available_t availability;
    long long serial_number;
};
```

<그림 4 : REQ 1-1 | atm의 member variable에 serial_number가 있는 것을 볼 수 있다.>

Atm class를 보면 serial_number 멤버변수가 존재한다. atm생성자를 통해 이를 초기화해줄 수 있다. atm의 선언 방식은 그림-3과 같다.

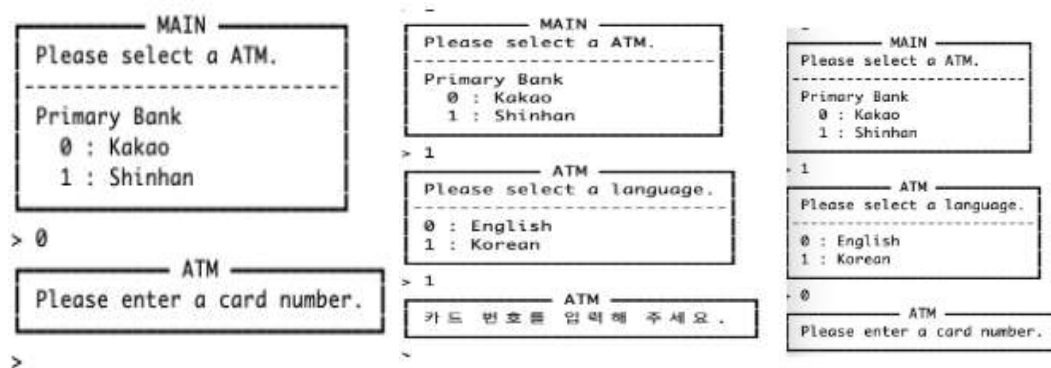
(REQ1.2) An ATM is set to one of the following types: (1) Single Bank ATM, (2) Multi-Bank ATM.



<그림 5 : REQ(1.2)의 설명 그림>

신한은행이 primary인 atm기는 single-bank-atm이고, 카카오가 primary인 atm기는 multi-bank-atm이다. 왼쪽 그림을 보면, kakao-atm에 신한은행 카드번호를 입력했음에도 불구하고, multi-bank-atm이기 때문에 문제가 없이 카드가 verification됨을 볼 수 있다. 반면, 오른쪽 그림을 보면, 신한은행 atm기에 카카오 카드번호를 입력하여 오류가 나는 것을 볼 수 있다.

(REQ1.3) An ATM may support either unilingual or bilingual languages.



<그림 6 : unilingual>

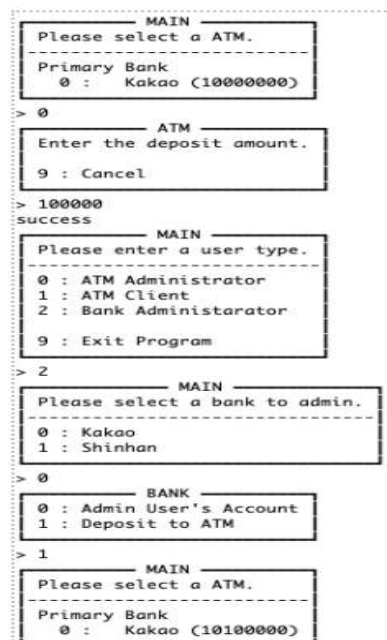
<그림 7 : bilingual-korean>

<그림 8 : bilingual-Eng>

그림6을 보면, 카카오 atm기는 unilingual atm이다. 이 atm기를 고르면 바로 영어버전 atm기가 사용되는 것을 알 수 있다.

그림7,8을 보면, 신한은행 atm기는 bilingualatm 이다. 이 atm기를 고르면 영어를 쓸 건지 한국어를 쓸 건지 물어보는 창이 나오고, 고른대로 atm기의 유형을 결정하는 것을 볼 수 있다.

(REQ1.4) A Bank deposits a certain amount of cashes to an ATM to serve users.



<그림 9 : REQ1.4>

그림 9를 보면, 은행의 관리자는 해당 은행이 관리하는 atm에 돈을 넣을 수 있음을 알 수 있다. (atm의 잔고 : 10000000 => 10100000)

(REQ1.5) A Bank can open an Account for user with necessary information to perform bank Services.

```

MAIN
Please enter a user type.
-----
0 : ATM Administrator
1 : ATM Client
2 : Bank Administrator
9 : Exit Program
> 2

MAIN
Please select a bank to admin.
-----
0 : Kakao
1 : Shinhan
> 0

BANK
Please enter a account number to admin.
> 111111111111

BANK
Bank Name : Kakao
User Name : seekmin
Account Number : 111111111111
Available Funds : 100000000
Transaction History :
  
```

<그림 10 : REQ1.5>

은행의 관리자는 account number만 알고있다면 account의 정보를 자유롭게 열람할 수 있음을 볼 수 있다.

(REQ1.6) A user may have multiple Accounts in a Bank.

```

User* user_1 = new User("seekmin");
User* user_2 = new User("seongwoo");

Bank* Kakao = new Bank("Kakao", 111);
Bank* Shinhan = new Bank("Shinhan", 222);

class Account {
    Account(Kakao, 111111111111, 100000000, 1234, user_1);
    Account* a2 = new Account(Shinhan, 222111111111, 600000, 1234, user_2);
    Account* a3 = new Account(Kakao, 111111111333, 30000, 1234, user_1);

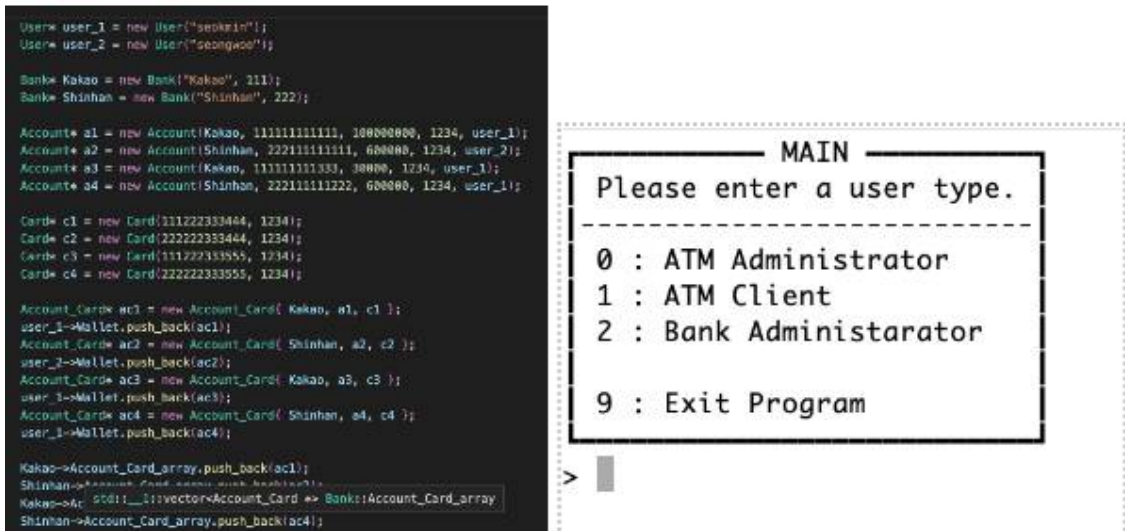
    Card* c1 = new Card(111222333444, 1234);
    Card* c2 = new Card(222222333444, 1234);
    Card* c3 = new Card(111222333555, 1234);
    Account_Card* ac1 = new Account_Card( Kakao, a1, c1 );
    user_1->Wallet.push_back(ac1);
    Account_Card* ac2 = new Account_Card( Shinhan, a2, c2 );
    user_2->Wallet.push_back(ac2);
    Account_Card* ac3 = new Account_Card( Kakao, a3, c3 );
    user_1->Wallet.push_back(ac3);
}

MAIN
Please enter a user type.
-----
0 : ATM Administrator
1 : ATM Client
2 : Bank Administrator
9 : Exit Program
>
  
```

<그림 11 : REQ1.6>

user1의 kakao account를 2개 생성했음에도, 에러가 나지 않는 것을 보아 req를 잘 충족했음을 볼 수 있다.

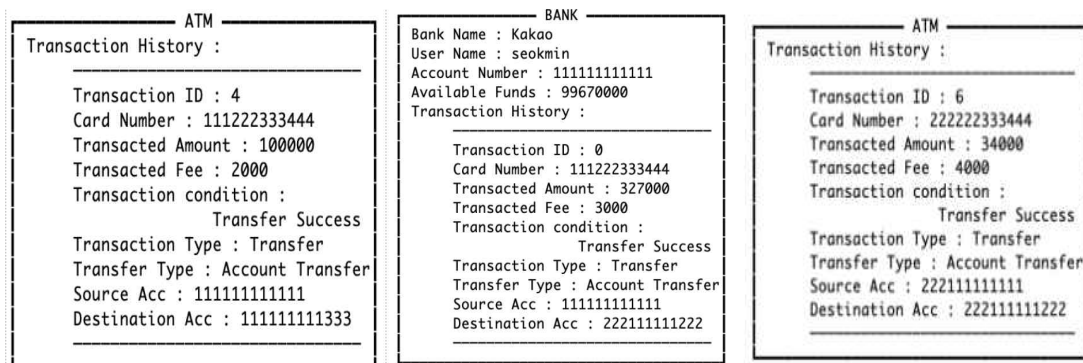
(REQ1.7) A user may have Accounts in multiple Banks.



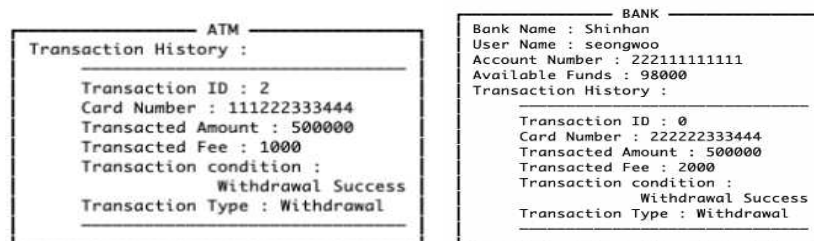
<그림 12 : REQ 1.7>

user1이 신한은행과 카카오뱅크의 계좌를 모두 가지고 있음에도 불구하고 에러가 나지 않는 것을 보아 req를 잘 충족했음을 볼 수 있다.

(REQ1.8) Each ATM have several types of transaction fees as follows



<그림 13 : transfer-2000> <그림 14 :transfer-3000> <그림 15 :transfer - 4000>



<그림 16 : withdrawal-1000> <그림 17 : withdrawal-2000>

BANK	
Bank Name :	Kakao
User Name :	seokmin
Account Number :	111111111111
Available Funds :	100110000
Transaction History :	
Transaction ID : 0	
Card Number : 111222333444	
Transacted Amount : 10000	
Transacted Fee : 0	
Transaction condition : Deposit Success	
Transaction Type : Deposit	
Deposit Paper Type : Cash	

<그림 18 : deposit - 0>

ATM	
Transaction History :	
Transaction ID : 0	
Card Number : 222222333444	
Transacted Amount : 100000	
Transacted Fee : 1000	
Transaction condition : Deposit Success	
Transaction Type : Deposit	
Deposit Paper Type : Cash	

<그림 19 : deposit - 1000>

위 그림13~19는 모든 일어날 수 있는 fee의 경우이다. transaction history를 통해 atm기가 fee를 잘 부과하고 있음을 알 수 있다. 해당 transaction들의 자세한 설명과정은 REQ4,5,6색션의 fee와 관련된 부분에 자세히 설명하였다.

(REQ1.9) An admin can access the menu of “Transaction History” via an admin card
(See REQ Display of Transaction History).

```

MAIN
Please enter a user type.
0 : ATM Administrator
1 : ATM Client
2 : Bank Administrator
9 : Exit Program
> 0

MAIN
Please select a ATM.
Primary Bank
0 : Kakao
1 : Shinhan
> 0

ATM
Please enter a card number.
> 12345678

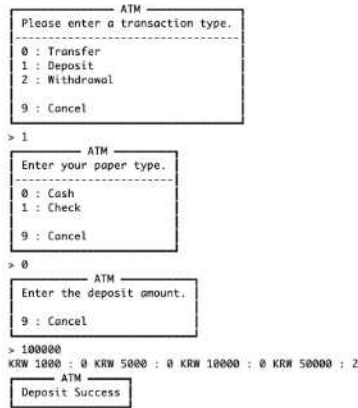
ATM
0 : Transaction History
9 : Cancel
> 0

ATM
Transaction History :
Transaction ID : 0
Card Number : 111222333444
Transacted Amount : 10000
Transacted Fee : 0
Transaction condition :
Deposit Success
Transaction Type : Deposit
Deposit Paper Type : Cash
  
```

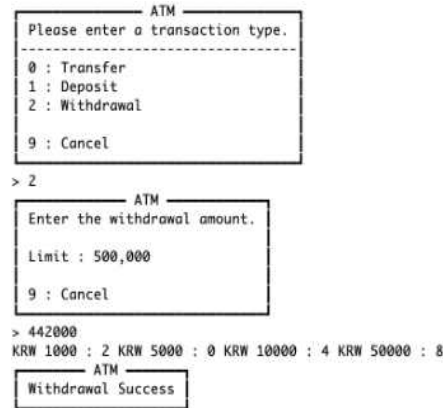
<그림 20 : REQ1.9>

admincard를 통해 atm의 transaction history를 볼 수 있음을 알 수 있다.

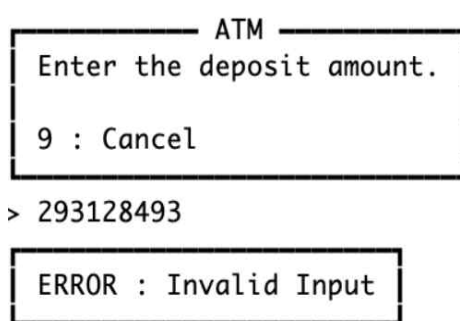
(REQ1.10) An ATM only accepts and returns the following types of cashes – KRW 1,000, KRW 5,000, KRW 10,000, KRW 50,000



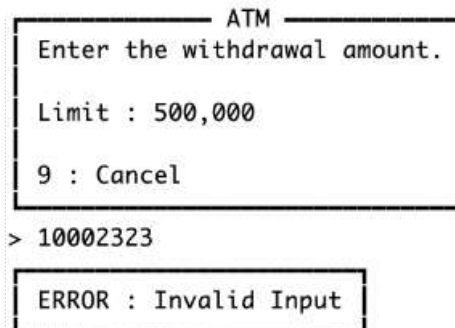
<그림 21 : deposit>



<그림 22 : withdrawal>



<그림 23 :deposit-invalid>



<그림 24 : withdrawal-invalid>

그림 21을 보면, deposit할 돈의 양을 넣어주었을 때 1000원 단위로 나누어 떨어진다면, 프로그램에서 자동으로 user가 넣어줄 1000원,5000원,10000원,50000원권의 지폐의 양을 추측하여 deposit하는 것을 볼 수 있다. 그림 22를 보면, withdrawal하고 싶은 돈의 양을 입력하였을 때 프로그램이 자동으로 1000원,5000원,10000원,50000원권의 지폐의 양을 계산하여 atm기로부터 withdrawal해주는 것을 볼 수 있다. 그림 23, 24에서는 1000원으로 나누어 떨어지지 않는 양을 입력하였을 때, deposit과 withdrwal 모두 에러가 발생하는 것을 볼 수 있다. (session 종료)

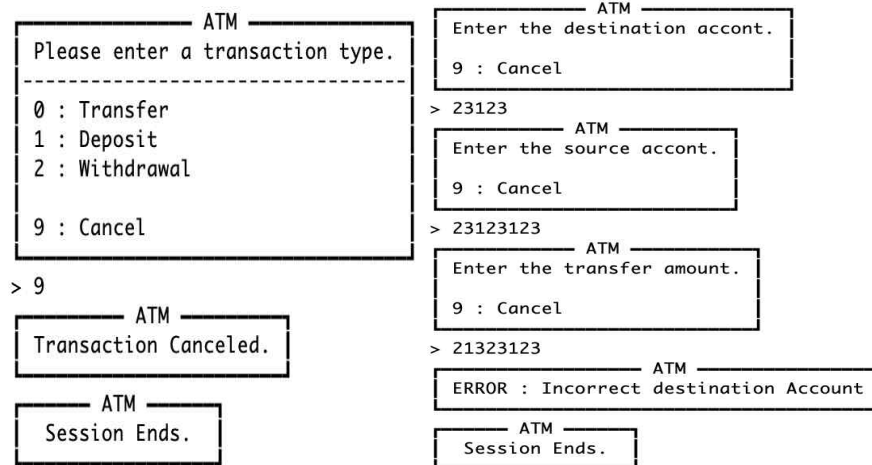
(REQ2.1) A session starts when a user inserts a card.



<그림 25 : REQ 2.1>

Card number만 입력하면(same as insert a card to atm) Session이 만들어진다.

(REQ2.2) A session ends whenever a user wishes (e.g., by choosing a cancel button) or there are some exceptional conditions detected by the ATM (e.g., no cash available).



<그림 26 : cancel>

<그림 27 : error>

그림 26,27을 보면 각각 cancel, error로 인해 session이 종료되었음을 알 수 있다.

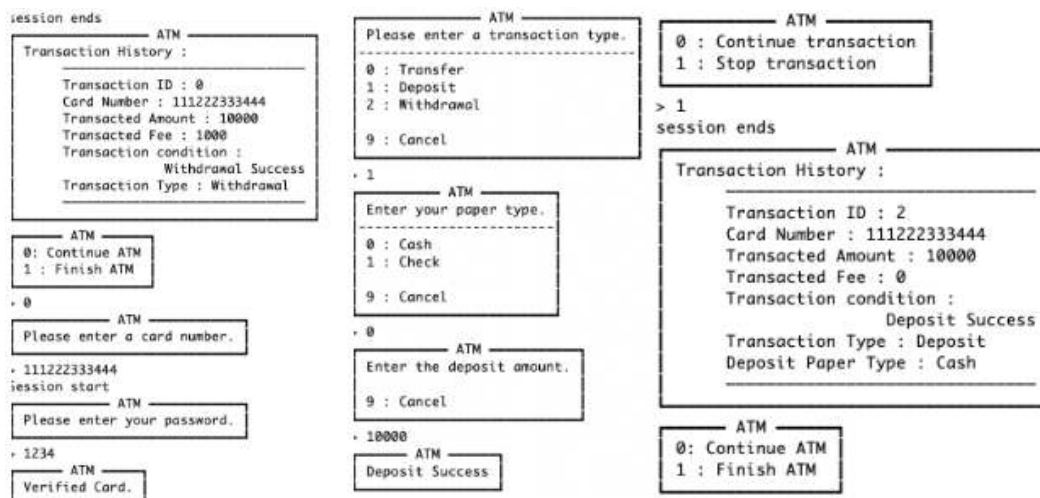
(REQ2.3) When a session ends, the summary of all transactions performed in a session must be displayed.



<그림 28 : REQ2.3>

그림 28을 보면, 세션이 끝나고 transaction의 summary가 나타나는 것을 볼 수 있다.

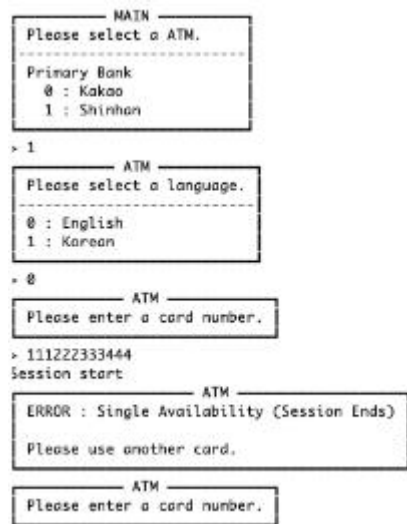
(REQ2.4) Each transaction has a unique identifier across all sessions.



<그림 29 : REQ(2.4)-1> <그림 30 :REQ(2.4)-2> <그림 31 : REQ(2.4)-3>

위 그림은 그림 29,30,31 순서로 실행된 것이다. 서로 다른 Session에서 실행된 Transaction들이 각기 다른 Transaction_ID를 가진 것을 볼 수 있다.

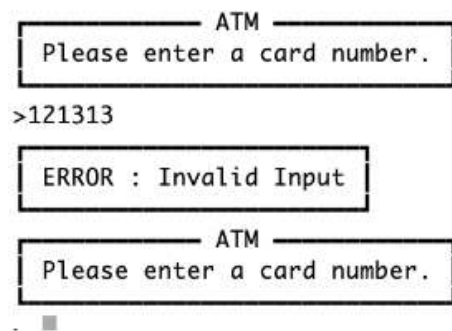
(REQ3.1) An ATM checks if the inserted card is valid for the current type of ATM.



<그림 32 : REQ3.1>

신한은행이 primary인 atm기는 single-bank-atm 이다. 위 그림을 보면, 신한은행 atm기에 카카오 카드를 넣었다. 그 결과, 세션이 종료되고 다른 카드를 넣어달라고 하는 것을 볼 수 있다.

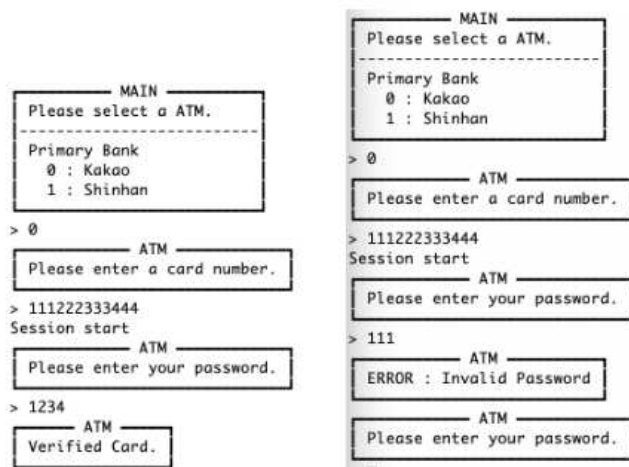
(REQ3.2) If an invalid card is inserted, the ATM shall display an appropriate error message (e.g., Invalid Card).



<그림 33 : REQ 3.2>

존재 불가능한 카드번호를 입력할 경우 적절한 에러메세지를 띄어주고, 다시 카드번호를 입력해 달라고 하는 것을 볼 수 있다.

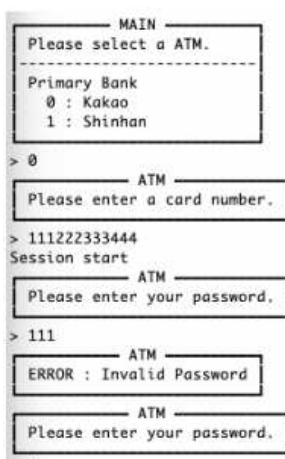
(REQ3.3) An ATM shall ask a user to enter the password (e.g., Enter Password), and verify if the password is correct



<그림 34 : REQ3.3 answer the password>

존재하는 카드번호를 입력하면 카드비밀번호를 묻는 창이 나타나는 것을 볼 수 있다. 올바른 비밀번호를 입력하면 그 다음 과정으로 넘어가고, 그렇지 않으면 에러메세지를 띄우고 다시 비밀번호를 묻는 것을 볼 수 있다.

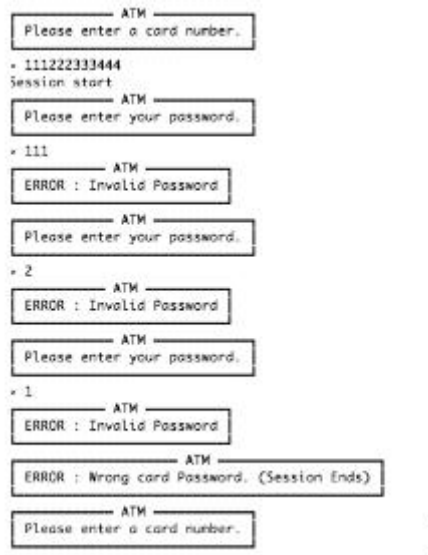
(REQ3.4) If the entered password is incorrect, the ATM shall display an appropriate error message (e.g., Wrong Password).



<그림 35 : REQ3.4-wrong password. >

존재하는 카드번호를 입력하면 카드비밀번호를 묻는 창이 나타나는 것을 볼 수 있다. 올바른 password를 입력하지 않으면 에러메세지를 띄우고 다시 비밀번호를 묻는 것을 볼 수 있다.

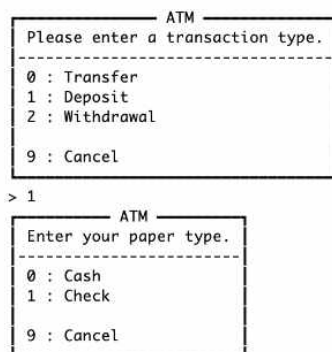
(REQ3.5) If a user enters wrong passwords 3 times in a row, a session is aborted, and return the card to the user.



<그림 36 : wrong password - 3>

3번 연속으로 올바른 password를 입력하지 않으면 세션이 종료되면서 새로운 카드번호를 묻는 것(카드를 돌려준 후, 다시 카드를 넣으라는 것과 같다.)이 실행되는 것을 볼 수 있다.

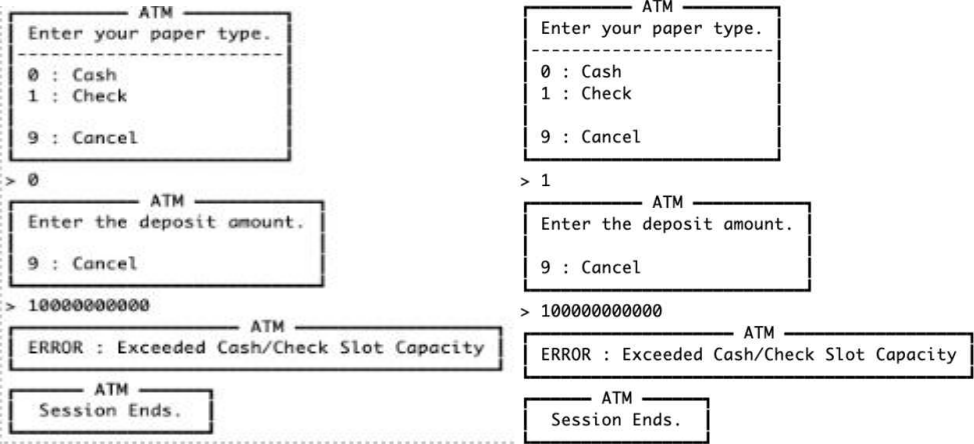
(REQ4.1) An ATM shall take either cash or check from a user.



<그림 37 : REQ4.1>

그림 37을 보면, deposit을 선택했을 때, 현금과 수표 중 선택할 수 있는 것을 볼 수 있다.

(REQ4.2) An ATM shall display an appropriate error message if the number of the inserted cash or checks exceed the limit allowed by the ATM.

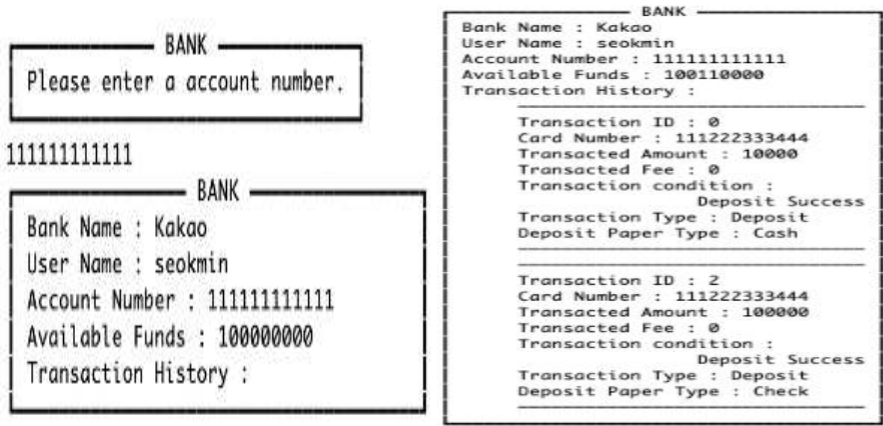


<그림 38 : cash>

<그림 39 : check>

그림 38, 39를 보면 현금과 수표 모두 atm기의 slot보다 많은 장수를 넣으면 에러를 출력하는 것을 볼 수 있다. (현금과 수표 모두 금액을 입력하면 프로그램 내에서 입력 장수를 실제상황과 가장 비슷하게 예상하여 진행한다.)

(REQ4.3) Once cash or checks are accepted by ATM, the transaction must be reflected to the bank account as well (i.e., the same amount of fund must be added to the corresponding bank account).



<그림 40 : 초기 잔고>

<그림 41 : 추후 잔고>

account의 잔고가 2번의 deposit transaction을 거친 후에 잘 증가되었음을 볼 수 있다.

(REQ4.4) Some deposit fee may be charged (See REQ in System Setup)

Transaction History :

Transaction ID : 0
Card Number : 22222333444
Transacted Amount : 100000
Transacted Fee : 1000
Transaction condition :
Deposit Success
Transaction Type : Deposit
Deposit Paper Type : Cash

<그림 42 : deposit fee>

카카오 ATM에서 신한 카드를 사용하면 1000원의 수수료가 발생하는 것을 볼 수 있다.

(REQ4.5) The deposited cash increase available cash in ATM that can be used by other users.

MAIN

Please select a ATM.

Primary Bank
0 : Kakao (10000000)

<그림 43 : atm초기 잔고>

Transaction History :

Transaction ID : 0
Card Number : 111222333444
Transacted Amount : 123000
Transacted Fee : 0
Transaction condition :
Deposit Success
Transaction Type : Deposit
Deposit Paper Type : Cash

<그림 44 : deposit transaction>

MAIN

Please select a ATM.

Primary Bank
0 : Kakao (10123000)

<그림 45 : 이후 atm잔고>

위 그림을 보면 그림 43에서 atm의 초기 잔고를 확인할 수 있다. 그림 44의 deposit transaction이 수행된 후에 atm기의 잔고가 증가한 것을 확인할 수 있다.

(REQ4.6) The deposited check does not increase available cash in ATM that can be used by other users.

```

MAIN
-----
Please select a ATM.
-----
Primary Bank
0 : Kakao (10123000)

```

<그림 46 : atm 초기잔고>

```

ATM
-----
Transaction History :
-----
Transaction ID : 2
Card Number : 111222333444
Transacted Amount : 200000
Transacted Fee : 0
Transaction condition :
                        Deposit Success
Transaction Type : Deposit
Deposit Paper Type : Check

```

<그림 47 : deposit - check>

```

MAIN
-----
Please select a ATM.
-----
Primary Bank
0 : Kakao (10123000)

```

<그림 48 : atm 추후 잔고>

위 그림들을 보면, check deposit은 atm의 사용가능 잔고를 바꾸지 않는 것을 볼 수 있다.

(REQ5.1) An ATM shall ask a user to enter the amount of fund to withdraw.

```

ATM
-----
Please enter a transaction type.
-----
0 : Transfer
1 : Deposit
2 : Withdrawal
9 : Cancel

```

> 2

```

ATM
-----
Enter the withdrawal amount.
-----
Limit : 500,000
9 : Cancel

```

<그림 49 : REQ5.1>

그림 49를 보면, withdrawal 진행중에 withdrawal할 금액을 물어보는 것을 볼 수 있다.

(REQ5.2) An ATM shall display an appropriate error message if there is insufficient fund in the account or insufficient cash in the ATM.

```

      BANK
-----
Bank Name : Kakao
User Name : seokmin
Account Number : 11111111333
Available Funds : 30000
Transaction History :
  
```

그림 50 : 초기 계좌 잔고

```

      ATM
-----
Enter the withdrawal amount.
Limit : 500,000
9 : Cancel
  
```

> 100000

```

      ATM
-----
ERROR : Not Enough Account Balance
  
```

```

      ATM
-----
Session Ends.
  
```

<그림 51 : error>

```

      MAIN
-----
Please select a ATM.
-----
Primary Bank
1 : Shinhan (10000) |
  
```

<그림 52 : 초기 atm 잔고>

```

      ATM
-----
Enter the withdrawal amount.
Limit : 500,000
9 : Cancel
  
```

> 500000

```

      ATM
-----
ERROR : Not Enough ATM Balance
  
```

```

      ATM
-----
Session Ends.
  
```

<그림 53 : error>

그림 50은 계좌의 잔고를 나타낸다. 그림 51에서는 해당 계좌에서 계좌에 있는 돈 보다 더 많은 돈을 인출하려다가 에러가 난 상황이다.

그림 52는 atm기의 잔고를 나타낸다. 그림 53에서는 해당 atm기에서 atm에 있는 돈 보다 더 많은 돈을 인출하려다가 에러가 난 상황이다.

(REQ5.3) Once the withdrawal is successful, the transaction must be reflected to the bank account as well (i.e., the same amount of fund must be deducted from the corresponding bank account).

```

      BANK
-----
Bank Name : Shinhan
User Name : seongwoo
Account Number : 22211111111
Available Funds : 600000
Transaction History :
  
```

<그림 54 : 초기 계좌 잔고>

```

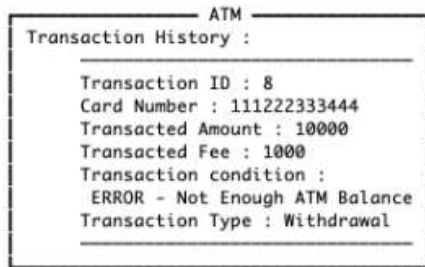
      BANK
-----
Bank Name : Shinhan
User Name : seongwoo
Account Number : 22211111111
Available Funds : 98000
Transaction History :

Transaction ID : 0
Card Number : 22222333444
Transacted Amount : 500000
Transacted Fee : 2000
Transaction condition :
Withdrawal Success
Transaction Type : Withdrawal
  
```

<그림 55 : 출금후의 계좌 잔고>

그림 54는 계좌의 초기잔고이다. 그림55의 withdrawal을 수행하고 나서 amount + fee만큼의 양이 계좌의 잔고에서 없어진 것을 볼 수 있다.

(REQ5.4) Some withdrawal fee may be charged (See REQ in System Setup).



ATM

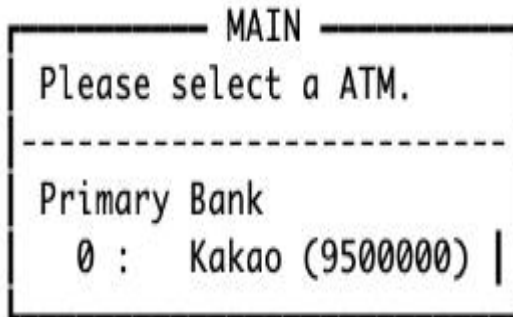
Transaction History :

Transaction ID : 8
Card Number : 111222333444
Transacted Amount : 10000
Transacted Fee : 1000
Transaction condition :
ERROR - Not Enough ATM Balance
Transaction Type : Withdrawal

<그림 56: withdrawal fee>

그림 56을 보면, withdrawal fee가 charge됨을 알 수 있다.

(REQ5.5) The cash withdrawal lower available cash in the ATM that can be used by other users.

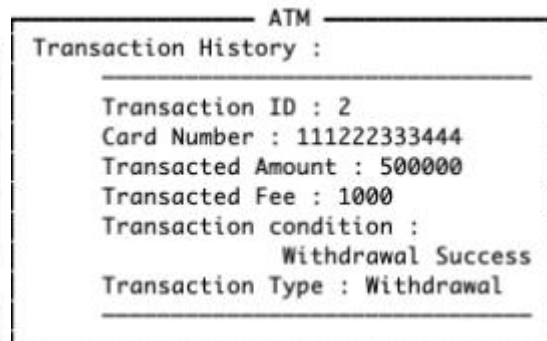


MAIN

Please select a ATM.

Primary Bank
0 : Kakao (9500000) |

<그림 57 : 초기 atm잔고>

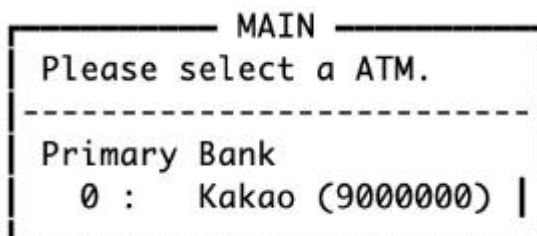


ATM

Transaction History :

Transaction ID : 2
Card Number : 111222333444
Transacted Amount : 500000
Transacted Fee : 1000
Transaction condition :
Withdrawal Success
Transaction Type : Withdrawal

<그림 58 : withdrawal transaction>



MAIN

Please select a ATM.

Primary Bank
0 : Kakao (9000000) |

<그림 59 : 추후 atm잔고>

초기 atm 잔고에서 withdrawal 한 amount 만큼 잔고가 줄어든 것을 볼 수 있다.

(REQ6.1) An ATM shall ask a user to choose the types of transfer either cash transfer or account

ATM

Please enter a transaction type.

0 : Transfer
1 : Deposit
2 : Withdrawal
9 : Cancel

> 0

ATM

Please enter a transfer type.

0 : Cash
1 : Account
9 : Cancel

<그림 60 : REQ6.1>

그림 60을 보면, transfer type중 cash transfer와 account transfer중 하나를 선택할 수 있음을 볼 수 있다.

(REQ6.2) For both cash and account transfers, an ATM shall ask the destination account number where the fund is to be transferred.

ATM

Please enter a transaction type.

0 : Transfer
1 : Deposit
2 : Withdrawal
9 : Cancel

. 0

ATM

Please enter a transfer type.

0 : Cash
1 : Account
9 : Cancel

. 0

ATM

Enter the destination account.

9 : Cancel

> 0

ATM

Please enter a transfer type.

0 : Cash
1 : Account
9 : Cancel

> 1

ATM

Enter the destination account.

9 : Cancel

<그림 61 : ask destination account>

그림 61을 보면, cash transfer와 account transfer모두 destination account를 물어봄을 알 수 있다.

(REQ6.3) For cash transfer, an ATM shall ask user to insert the cash including the transaction fees, and verify if the amount of the inserted cash is correct. All inserted cash excluding the transaction fee shall be transferred.

ATM	BANK
Please enter a transfer type. ----- 0 : Cash 1 : Account 9 : Cancel	Bank Name : Shinhan User Name : seongwoo Account Number : 22211111111 Available Funds : 600000 Transaction History :
> 0 ----- ATM Enter the destination acct. 9 : Cancel	Bank Name : Shinhan User Name : seongwoo Account Number : 22211111111 Available Funds : 650000 Transaction History : ----- Transaction ID : 0 Card Number : 111222333444 Transacted Amount : 50000 Transacted Fee : 3000 Transaction condition : Transfer Success Transaction Type : Transfer Transfer Type : Cash Transfer Source Acc : None Destination Acc : 22211111111
> 22211111111 ----- ATM Enter the transfer amount. 9 : Cancel	
> 100000 ----- ATM Please insert cashes 103000 9 : Cancel	
> 13000 ----- ATM ERROR : Invalid Amount	

<그림 62 : REQ 6.3> <그림 63 : REQ 6.3>

그림 62를 보면, cash transfer의 경우에 fee를 포함한 금액을 넣어달라고 말해주고 있다. 그 금액과 다른 금액을 넣게되면 위와같이 error를 출력해준다. 같은 금액을 넣을 경우에 올바르게 trasfer가 일어난다. 그림 63을 보면, 위의 박스는 계좌의 초기잔고를 볼 수 있다. 밑의 박스를 보면, 해당 계좌에 한번 송금이 일어났었음을 알 수 있고, 송금된 금액만큼 잔고가 늘어났음을 알 수 있다.

(REQ6.4) For account transfer, an ATM shall ask the source account number, and the amount of fund to be transferred.

The image shows a sequence of five ATM screens for an account transfer process. Each screen is titled 'ATM' and has a dashed border. The first screen prompts 'Please enter a transaction type.' with options: 0 : Transfer, 1 : Deposit, 2 : Withdrawal, and 9 : Cancel. The second screen, after input '0', prompts 'Please enter a transfer type.' with options: 0 : Cash, 1 : Account, and 9 : Cancel. The third screen, after input '1', prompts 'Enter the destination account.' with option 9 : Cancel. The fourth screen, after input '22211111111', prompts 'Enter the source account.' with option 9 : Cancel. The fifth screen, after input '11111111111', prompts 'Enter the transfer amount.' with option 9 : Cancel.

<그림 64 : REQ 6.4>

그림 64를 보면, account transfer의 경우에 source account number와 amount를 묻고 있는 것을 확인할 수 있다.

(REQ6.5) Some transfer fee may be charged (See REQ in System Setup).

The image shows three ATM screens displaying transaction history. Each screen is titled 'ATM' and has a dashed border. The first screen shows Transaction ID : 2, Card Number : 111222333444, Transacted Amount : 23000, Transacted Fee : 3000, and Transaction condition : Transfer Success. The second screen shows Transaction ID : 4, Card Number : 111222333444, Transacted Amount : 100000, Transacted Fee : 2000, and Transaction condition : Transfer Success. The third screen shows Transaction ID : 6, Card Number : 222222333444, Transacted Amount : 34000, Transacted Fee : 4000, and Transaction condition : Transfer Success. All three screens also show Transaction Type : Transfer and Source Acc : 11111111111.

<그림 65 : transfer fees>

그림 65를 보면 상황에 따라 여러 종류의 transfer fee가 발생함을 알 수 있다.

(REQ6.6) The inserted cash for transfer increase available cash in ATM that can be used by other Users.

```

MAIN
Please select a ATM.
-----
Primary Bank
0 : Kakao (10000000)

```

```

ATM
Transaction History :
-----
Transaction ID : 0
Card Number : 111222333444
Transacted Amount : 234000
Transacted Fee : 3000
Transaction condition :
Transfer Success
Transaction Type : Transfer
Transfer Type : Cash Transfer
Source Acc : None
Destination Acc : 222111111111

```

```

MAIN
Please select a ATM.
-----
Primary Bank
0 : Kakao (10237000)

```

<그림 66 : REQ6.6>

그림 66을 보면, cash transfer의 경우에 atm의 잔고가 증가했음을 볼 수 있다.

(REQ6.7) Once the transfer is successful, the transaction must be reflected to the bank account as well (i.e., the same amount of fund must be deducted from the source bank account, and then added to the destination bank account)

<pre> BANK Bank Name : Kakao User Name : seokmin Account Number : 111111111111 Available Funds : 100000000 Transaction History : ----- Transaction ID : 0 Card Number : 111222333444 Transacted Amount : 327000 Transacted Fee : 3000 Transaction condition : Transfer Success Transaction Type : Transfer Transfer Type : Account Transfer Source Acc : 111111111111 Destination Acc : 222111111222 </pre>	<pre> BANK Bank Name : Shinhan User Name : seokmin Account Number : 222111111222 Available Funds : 600000 Transaction History : ----- Transaction ID : 0 Card Number : 111222333444 Transacted Amount : 327000 Transacted Fee : 3000 Transaction condition : Transfer Success Transaction Type : Transfer Transfer Type : Account Transfer Source Acc : 111111111111 Destination Acc : 222111111222 </pre>
---	--

<그림 67 : transfer할 때 계좌 잔고의 변화>

그림 67을 보면 한번의 transfer로 source 계좌와 destination계좌의 잔고가 모두 변하는 것을 알 수 있다. source 계좌는 amount + fee만큼 잔고가 줄고, desitination 계좌는 amount만큼 잔고가 증가하는 것을 확인할 수 있다.

(REQ7.1) When a session is started by an admin by inserting an admin card (See REQ in System Setup), an ATM displays a menu of “Transaction History” only.

```

      MAIN
-----
Please enter a user type.
-----
0 : ATM Administrator
1 : ATM Client
2 : Bank Administrator
9 : Exit Program
-----
> 0
      MAIN
-----
Please select a ATM.
-----
Primary Bank
0 : Kakao
1 : Shinhan
-----
> 0
      ATM
-----
Please enter a card number.
-----
> 12345678
      ATM
-----
0 : Transaction History
9 : Cancel
-----

```

<그림 68 : REQ 7.1>

올바른 admincard를 입력하면 atm의 transaction history만을 볼 수 있음을 알 수 있다.

(REQ7.2) When the “Transaction History” menu is selected, an ATM display the information of all transactions from all users from the beginning of the system start.

```

      ATM
-----
0 : Transaction History
9 : Cancel
-----
> 0
      ATM
-----
Transaction History :
-----
Transaction ID : 0
Card Number : 111222333444
Transacted Amount : 100000
Transacted Fee : 0
Transaction condition : Deposit Success
Transaction Type : Deposit
Deposit Paper Type : Cash
-----
      ATM
-----
Transaction History : There is No History
-----
      ATM
-----
Transaction History :
-----
Transaction ID : 2
Card Number : 111222333444
Transacted Amount : 500000
Transacted Fee : 1000
Transaction condition : Withdrawal Success
Transaction Type : Withdrawal
-----
      ATM
-----
Transaction History :
-----
Transaction ID : 4
Card Number : 111222333444
Transacted Amount : 200000
Transacted Fee : 0
Transaction condition : Deposit Success
Transaction Type : Deposit
Deposit Paper Type : Cash
-----

```

<그림 69 : REQ 7.2>

transaction history menu가 골라지면, 해당 atm의 모든 transaction history가 표시됨을 알 수 있다.

(REQ7.3) The “Transaction History” information shall be outputted to the external file (e.g., txt file).

```

M_history > E atm.txt
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1
```

<그림 70 : REQ7.3>

위 그림을 보면, transaction history가 atm1.txt에 잘 저장되는 것을 볼 수 있다.

(REQ8.1) An ATM that is configured with the bilingual support shall provide an option for a user to choose the preferred language either English or Korean.

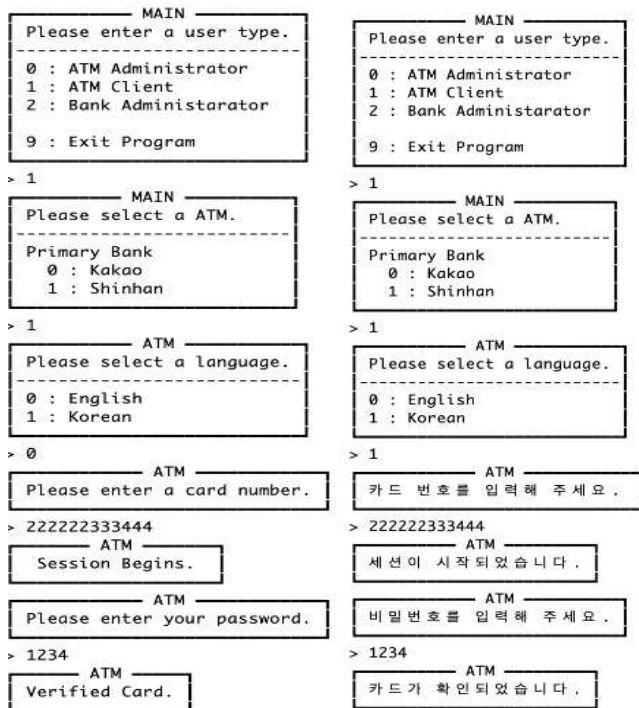
The figure consists of four sequential screenshots of an ATM interface, labeled 1 through 4. Each screenshot shows a terminal window with a title bar and a dashed line separating the title from the content.

- Screenshot 1:** The title bar says "MAIN". The content says "Please select a ATM." followed by a dashed line and two options: "0 : Kakao" and "1 : Shinhan".
- Screenshot 2:** The title bar says "ATM". The content says "Please select a language." followed by a dashed line and two options: "0 : English" and "1 : Korean".
- Screenshot 3:** The title bar says "ATM". The content says "카드 번호를 입력해 주세요." (Please enter the card number).
- Screenshot 4:** The title bar says "ATM". The content says "Please enter a card number." followed by a dashed line and two options: "0 : English" and "1 : Korean".

<그림 71 : bilingual support ATM>

신한은행 atm기는 bilingual ATM이다. 이 ATM을 선택하면, 영어와 한국어중 하나를 선택할 수 있음을 알 수 있다.

(REQ8.2) Once a certain language is chosen, all menus must be displayed using the chosen language.

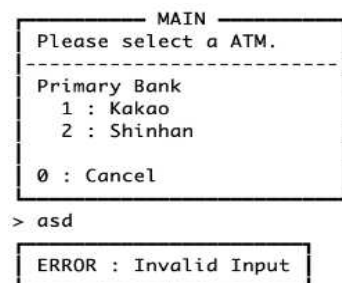


<그림 72 : REQ 8.2> <그림 73 : REQ 8.2>

한번 언어를 고르고 나면, 나머지 메뉴들이 각 언어에 맞게 표시되는 것을 볼 수 있다.

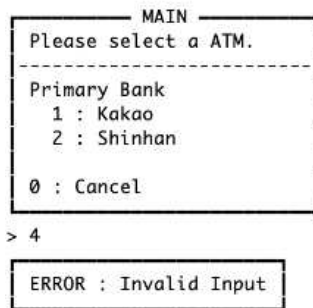
(REQ9.1) An ATM shall display an appropriate message for each exception scenario (both explicitly stated in this document and implicitly assumed ones), and take an appropriate action (e.g., end a session).

모든 case의 exception은 100가지 경우의 수가 넘어서 모든 콘솔창에서 발생할 수 있는 exception을 적지는 못했습니다.(구현은 전부 완료하였습니다.) 대표적인 exception들을 골라 20가지 정도 보고서에 넣었습니다.



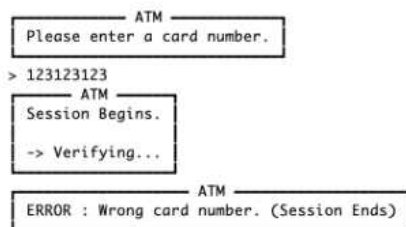
<그림 74 : exception_1>

숫자를 입력해야 하는데, 문자가 들어온 경우 invalid input이라는 error message를 출력하고, 다시 입력하라는 문구가 출력됩니다.



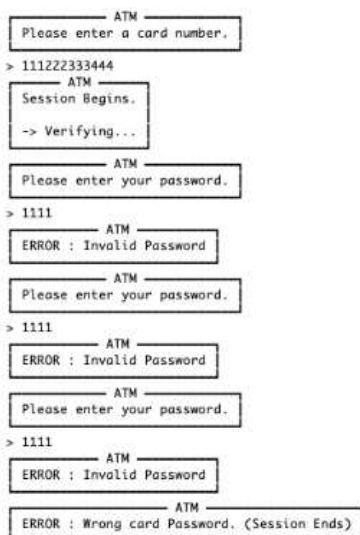
<그림 75 : exception_2>

숫자를 입력 했지만, 메뉴에 존재하지 않는 숫자를 입력한 경우 invalid input이라는 error message를 출력하고, 다시 입력하라는 문구가 출력됩니다.



<그림 76 : exception_3>

카드번호를 잘못 입력할 경우 세션이 종료되며 다시 카드번호를 입력받습니다.



<그림 77 : exception_4>

카드번호를 잘 입력하였지만 비밀번호를 3번 연속 틀리게 된 경우 세션이 종료되며 다시 카드번호를 입력받습니다.

```

      ATM
-----
Please enter a transfer type.
-----
1 : Cash
2 : Account
0 : Cancel

> 1

      ATM
-----
Enter the destination acct.
-----
0 : Cancel

> 123123123123

      ATM
-----
Enter the transfer amount.
-----
0 : Cancel

> 100000

      ATM
-----
Please insert cashes 103000_
-----
0 : Cancel

> 103000

      ATM
-----
ERROR : Incorrect Destination Account

      ATM
-----
Session Ends.

```

<그림 78 : exception_5>

destination account number가 잘못 입력되었을 경우에 세션이 종료되고 다시 카드번호를 입력받는 창(카드를 atm기에 넣는 창)으로 돌아갑니다.

```

      ATM
-----
Please enter a transfer type.
-----
1 : Cash
2 : Account
0 : Cancel

> 1

      ATM
-----
Enter the destination acct.
-----
0 : Cancel

> 222111111111

      ATM
-----
Enter the transfer amount.
-----
0 : Cancel

> 130

      ATM
-----
ERROR : Invalid Input

```

<그림 79 : exception_6>

현금 이체 시에 이체 하는 돈이 100원이나 10원 단위가 있는 경우 (1원도 포함) 에러를 출력하고 다시 입력을 요청합니다.

```

      ATM
-----
Please enter a transfer type.
-----
1 : Cash
2 : Account
0 : Cancel

> 1
      ATM
-----
Enter the destination account.
-----
0 : Cancel

> 222111111111
      ATM
-----
Enter the transfer amount.
-----
0 : Cancel

> 100000
      ATM
-----
Please insert cashes 103000__
-----
0 : Cancel

> 200000
      ATM
-----
ERROR : Invalid Amount

```

<그림 80 : exception_7>

현금 이체 시에 수수료 포함해서 알맞은 금액을 입력하지 않은 경우 다시 입력을 요청합니다.

]

```

      ATM
-----
Please enter a transfer type.
-----
1 : Cash
2 : Account
0 : Cancel

> 2
      ATM
-----
Enter the destination account.
-----
0 : Cancel

> 222111111111
      ATM
-----
Enter the source account.
-----
0 : Cancel

> 123123123123
      ATM
-----
Enter the transfer amount.
-----
0 : Cancel

> 10000
      ATM
-----
ERROR : Incorrect Source Account

      ATM
-----
Session Ends.

```

<그림 81 : exception_8>

계좌 이체 시, source account를 틀린 경우 세션이 종료되고 다시 카드번호를 입력받는 창(카드를 atm기에 넣는 창)으로 돌아갑니다.

```

      ATM
-----
Please enter a transfer type.
-----
1 : Cash
2 : Account
0 : Cancel

> 2
      ATM
-----
Enter the destination account.
-----
0 : Cancel

> 222111111111
      ATM
-----
Enter the source account.
-----
0 : Cancel

> 11111111333
      ATM
-----
Enter the transfer amount.
-----
0 : Cancel

> 1000000
      ATM
-----
ERROR : Not Enough Account Balance

      ATM
-----
Session Ends.

```

<그림 82 : exception_9>

계좌 이체 시, 계좌의 잔고보다 더 많은 돈을 입력한 경우 세션을 종료시키고 카드를 삽입하는 곳으로 돌아갑니다. (카드번호 입력 창)

```

      ATM
-----
Please enter a transaction type.
-----
1 : Transfer
2 : Deposit
3 : Withdrawal
0 : Cancel

> 2
      ATM
-----
Enter your paper type.
-----
1 : Cash
2 : Check
0 : Cancel

> 2
      ATM
-----
Enter the deposit amount.
-----
0 : Cancel

> 130000
      ATM
-----
ERROR : Invalid Deposit Type(Not Check)

      ATM
-----
Session Ends.

```

<그림 83 : exception_10>

수표 입금 시, 수표 단위와 맞지 않는 경우 (수표 단위는 100,000원, 500,000원, 1000,000원) 세션이 종료되고, 카드를 삽입하는 곳으로 돌아갑니다.(카드번호 입력 창)

```

      ATM
Enter your paper type.
-----
1 : Cash
2 : Check
0 : Cancel
> 1
      ATM
Enter the deposit amount.
0 : Cancel
> 2478000
      ATM
ERROR : Exceeded Cash/Check Slot Capacity
      ATM
Session Ends.

```

<그림 84 : exception_11>

현금 입금 시, slot의 limit인 50장을 초과한 경우 ($2478000 = 3(1000) + 1(5000) + 2(10000) + 49(500000) = 55$ 장) 세션을 종료시키고 카드번호를 입력하는 창으로 돌아갑니다.

```

      ATM
Please enter a transaction type.
-----
1 : Transfer
2 : Deposit
3 : Withdrawal
0 : Cancel
> 3
      ATM
Enter the withdrawal amount.
Limit : 500,000
0 : Cancel
> 1000000
      ATM
ERROR : Exceed Limit
      ATM
Session Ends.

```

<그림 85 : exception_12>

출금 시, 한도 500,000원을 초과해서 출금하려고 하는 경우 세션을 종료시키고 카드번호를 입력하는 창으로 돌아갑니다.

```

      ATM
Enter the withdrawal amount.
Limit : 500,000
0 : Cancel
> 500000
      ATM
ERROR : Not Enough ATM Balance
      ATM
Session Ends.
      ATM
Transaction History :
Transaction ID : 0
Card Number : 22222333444
Transacted Amount : 500000
Transacted Fee : 1000
Transaction condition :
      ERROR - Not Enough ATM Balance
Transaction Type : Withdrawal

```

<그림 86 : exception_13>

출금 시, ATM에 돈이 없는 경우 세션을 종료시키고 카드번호를 입력하는 창으로 돌아갑니다.

```

      ATM
Enter the withdrawal amount.
Limit : 500,000
0 : Cancel
> 500000
      ATM
ERROR : Not Enough Account Balance
      ATM
Session Ends.

```

<그림 87 : exception_14>

출금 시, 계좌에 돈이 부족한 경우 세션을 종료시키고 카드번호를 입력하는 창으로 돌아갑니다.

```

      ATM
Enter the withdrawal amount. (1/3)
Limit : 500,000
0 : Cancel
> 500000
      ATM
Withdrawal Success
KRW1000 : 0
KRW5000 : 0
KRW10000 : 0
KRW50000 : 10

      ATM
Please enter a transaction type.
1 : Transfer
2 : Deposit
3 : Withdrawal
0 : Cancel
> 3
      ATM
Withdraw only 3 times per Session.
--> Session Successfully Ends.

```

<그림 88 : exception_15>

한 세션에 출금 3번 초과시 강제로 세션을 종료시키고 카드번호를 입력하는 창으로 돌아갑니다.

ATM

Enter the withdrawal amount. (3/3)

Limit : 500,000

0 : Cancel

> 500000

ATM

ERROR : Exceeded Cash/Check Slot Capacity

ATM

Session Ends.

ATM

Transaction History :

Transaction ID : 6

Card Number : 111222333444

Transacted Amount : 500000

Transacted Fee : 1000

Transaction condition :
Withdrawal Success

Transaction Type : Withdrawal

Transaction ID : 8

Card Number : 111222333444

Transacted Amount : 500000

Transacted Fee : 1000

Transaction condition :
Withdrawal Success

Transaction Type : Withdrawal

Transaction ID : 10

Card Number : 111222333444

Transacted Amount : 500000

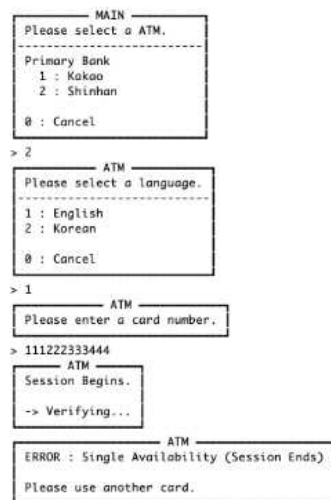
Transacted Fee : 1000

Transaction condition :
ERROR - Exceeded Cash/Check Slot Capacity

Transaction Type : Withdrawal

<그림 89 : exception_16>

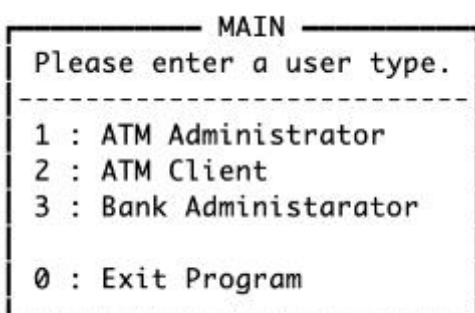
출금 되는 현금 50장 넘어가는 경우 세션을 종료시키고 카드번호를 입력하는 창으로 돌아갑니다.



<그림 90 : exception_17>

신한 ATM은 single인데 카카오 은행 카드를 입력한 경우 세션을 종료시키고 카드번호를 입력하는 창으로 돌아갑니다.

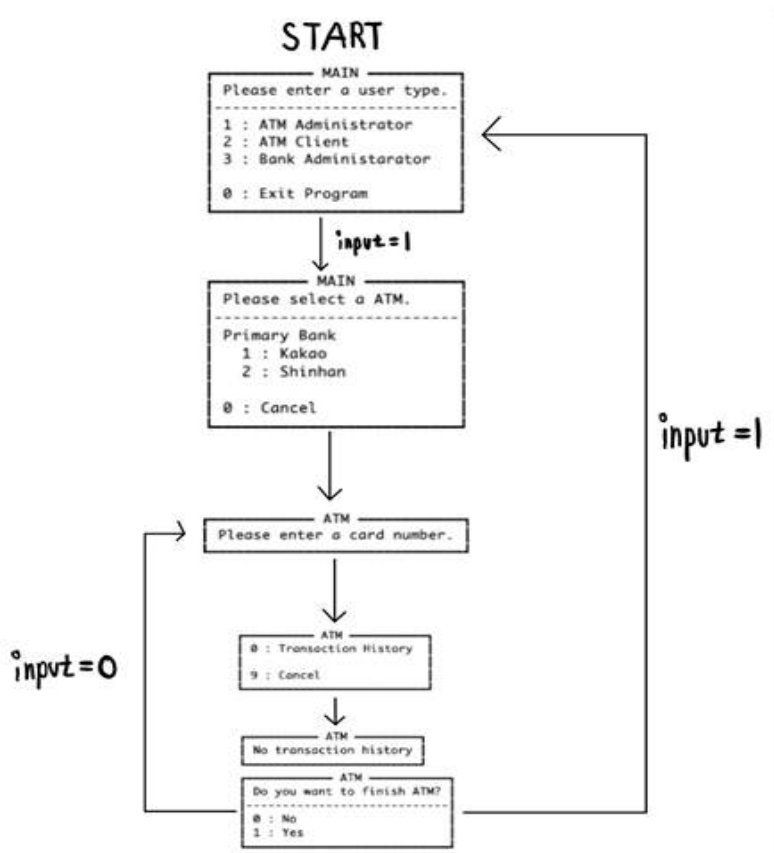
C) Instruction to run the source code



<그림 91 : entire_start>

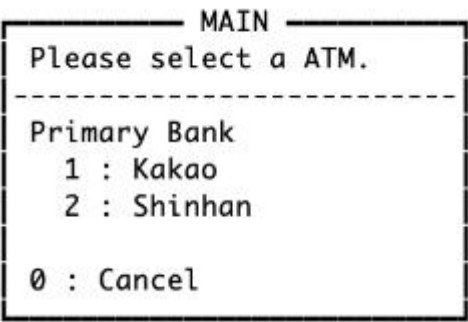
위 그림은 시작 페이지를 나타낸다. 1번은 ATM관리자 창, 2번은 ATM 고객 창, 3번은 BANK 관리자 창을 나타낸다.

C-1) ATM 관리자 창



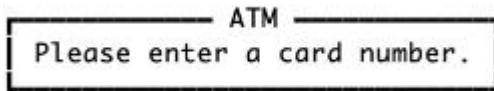
<그림 92 : ATM administrator 창 STATE-DIAGRAM>

위 그림은 ATM 관리자 PAGE의 STATE DIAGRAM을 나타낸 것이다. EXCEPTION CONDITION은 너무 많아서 따로 표시하지는 않았다. 밑에 STATE DIAGRAM을 바탕으로 자세하게 설명하였다.



<그림 93 : ATM 관리자_1>

ATM 관리자 시작 페이지에는 어떤 ATM을 관리하는지 먼저 물어본다.

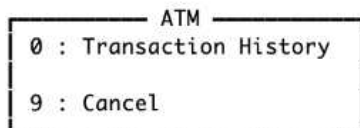


ATM

Please enter a card number.

<그림 94 : atm 관리자_2>

관리하는 ATM을 선택하고 나면, admin card number를 물어본다.



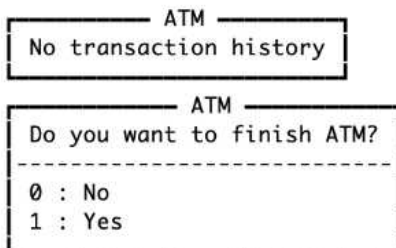
ATM

0 : Transaction History

9 : Cancel

<그림 95 : ATM 관리자_3>

올바른 admin card number를 입력하면, transaction history를 볼것인지 묻는 창이 나타난다.



ATM

No transaction history

ATM

Do you want to finish ATM?

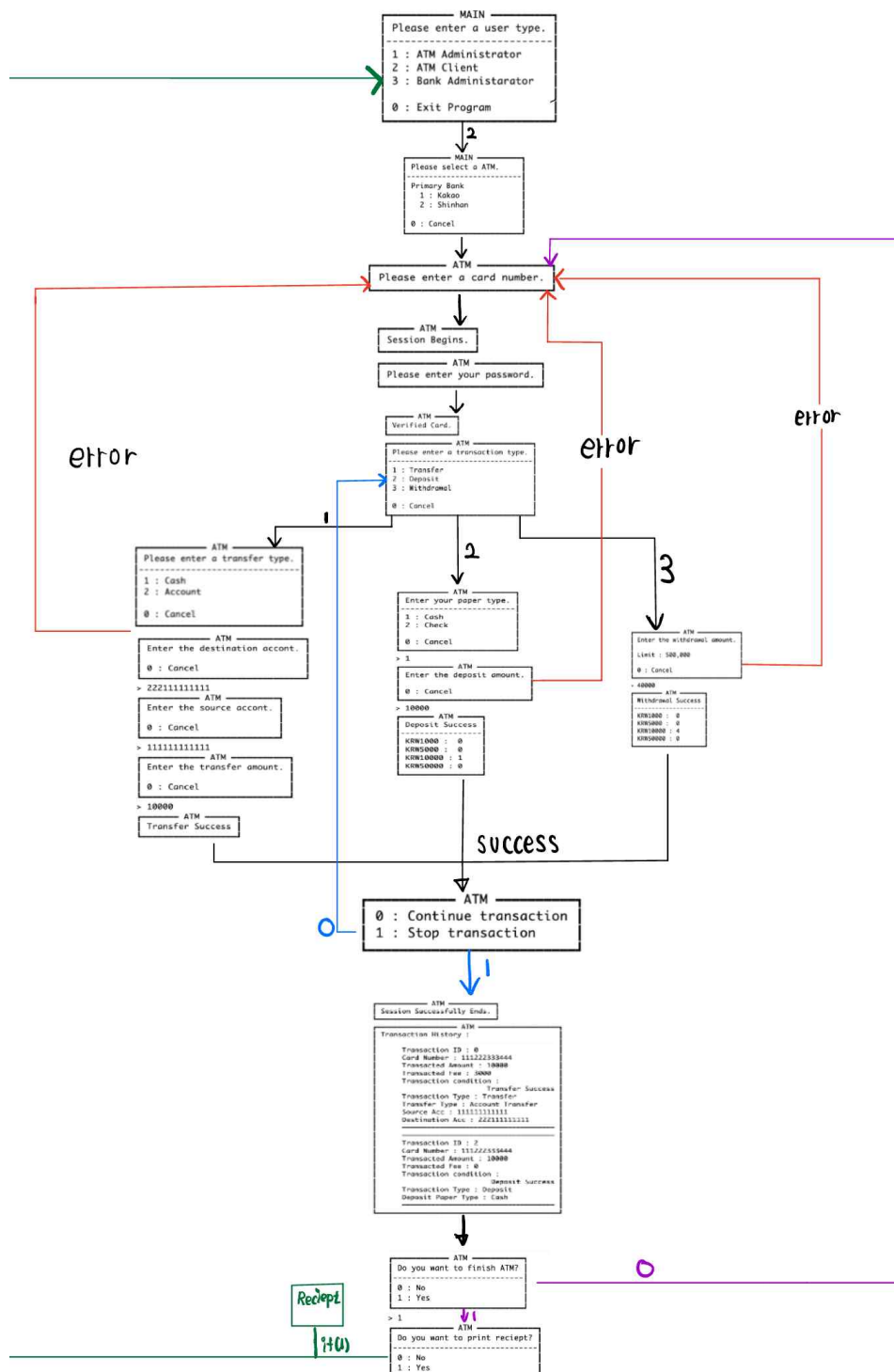
0 : No

1 : Yes

<그림 96 : ATM 관리자_4>

transaction history를 보겠다고 하면, transaction history를 보여주고, ATM을 그만 쓸 것인지 물어본다. 만약 그만 사용하겠다고 하면, 프로그램의 맨 처음 시작 페이지로 돌아가고, 계속 사용하겠다고 하면, admin card number를 묻는 창으로 돌아간다.

C-2) ATM client 창



<그림 97 : ATM_Client_STATE_DIAGRAM>

위 그림은 atm_client창의 state diagram이다. 모든 exception을 표시하지는 않았다. 자세한 설명은 밑에 있다.

```

      MAIN
    Please select a ATM.
-----
Primary Bank
  1 : Kakao
  2 : Shinhan
  0 : Cancel
  
```

<그림 98 : client_1>

ATM client 창으로 들어가게 되면, 어떠한 ATM기를 쓸 것인지 물어보게 된다.

```

      ATM
    Please enter a card number.
  
```

<그림 99 : client_2 >

ATM기를 선택하고 나면, 카드를 넣어달라고 부탁한다.(카드번호를 입력하라고 한다.)

```

      ATM
    Session Begins.
-----
      ATM
    Please enter your password.
  
```

<그림 100 : client_3>

올바른 card number를 입력하면, 세션이 시작되고 카드 비밀번호를 물어보는 것을 볼 수 있다.

```

      ATM
    Verified Card.
-----
      ATM
    Please enter a transaction type.
-----
  1 : Transfer
  2 : Deposit
  3 : Withdrawal
  0 : Cancel
  
```

<그림 101 : client_4>

올바른 카드 비밀번호를 입력하고 나면, card 가 verification 됐다는 message를 출력하고, 어떠한 종류의 거래를 할 것인지 물어본다.

Client_5-1) ATM client-Transfer 창

```

      ATM
-----
Please enter a transaction type.
-----
1 : Transfer
2 : Deposit
3 : Withdrawal
0 : Cancel
  
```

> 1

```

      ATM
-----
Please enter a transfer type.
-----
1 : Cash
2 : Account
0 : Cancel
  
```

```

      ATM
-----
Enter the destination acct.
0 : Cancel
  
```

> 222111111111

```

      ATM
-----
Enter the source acct.
0 : Cancel
  
```

> 111111111111

```

      ATM
-----
Enter the transfer amount.
0 : Cancel
  
```

> 10000

```

      ATM
-----
Transfer Success
  
```

<그림 102 : transfer_1>

<그림 103 : transfer_2>

transfer를 고르게 되면, transfer type이 무엇인지 물어보고, destination account, source account, transfer amount를 물어본다. 만약 cash transfer라면, amount + fee 만큼의 돈을 넣어달라는 메시지도 출력된다. transfer가 성공하면 transfer success라는 메시지가 출력된다. 실패하면 세션이 끝났다는 메시지가 출력되면서 다시 카드를 insert하는 창으로 돌아간다.

Client_5-2) ATM client-Deposit 창

```

      ATM
-----
Enter your paper type.
-----
1 : Cash
2 : Check
0 : Cancel
  
```

> 1

```

      ATM
-----
Enter the deposit amount.
0 : Cancel
  
```

> 10000

```

      ATM
-----
Deposit Success
-----
KRW1000 : 0
KRW5000 : 0
KRW10000 : 1
KRW50000 : 0
  
```

<그림 104 : deposit>

deposit을 고르게 되면, paper type과 amount를 물어보고, 성공했다면 deposit success라는 문구와 함께 deposit한 (천원,오천원,만원,오만원권)의 개수를 각각 출력해준다. 만약 deposit이 실패한다면, 세션이 종료되었다는 문구와 함께 카드를 insert하는 창으로 돌아간다.

Client_5-3) ATM client-Withdrawal 창

ATM

Please enter a transaction type.

1 : Transfer

2 : Deposit

3 : Withdrawal

0 : Cancel

3

ATM

Enter the withdrawal amount.

Limit : 500,000

0 : Cancel

40000

ATM

Withdrawal Success

KRW1000 : 0

KRW5000 : 0

KRW10000 : 4

KRW50000 : 0

<그림 105 : Withdrawal>

withdrawal을 고르게 되면, amount를 물어보고, 성공하면 withdrawal success라는 문구를 출력해주는 것을 볼 수 있다. 실패하면 세션을 종료시키고 card를 insert 하는 clinet_2 과정으로 돌아간다.

ATM

0 : Continue transaction

1 : Stop transaction

<그림 106 : clinet_6>

transaction이 성공적으로 끝나면, transaction을 계속 할 것인지 묻는 창이 나온다. 계속하겠다고 하면, transaction type을 고르는 client_4 과정으로 돌아간다. 만약 transaction stop을 원하는 경우는 밑에 설명해두었다.

```

      ATM
    0 : Continue transaction
    1 : Stop transaction
  > 1
      ATM
    Session Successfully Ends.

      ATM
    Transaction History :
    Transaction ID : 0
    Card Number : 111222333444
    Transacted Amount : 10000
    Transacted Fee : 3000
    Transaction condition :
      Transfer Success
    Transaction Type : Transfer
    Transfer Type : Account Transfer
    Source Acc : 111111111111
    Destination Acc : 222111111111

    Transaction ID : 2
    Card Number : 111222333444
    Transacted Amount : 10000
    Transacted Fee : 0
    Transaction condition :
      Deposit Success
    Transaction Type : Deposit
    Deposit Paper Type : Cash

      ATM
    Do you want to finish ATM?
    0 : No
    1 : Yes
  
```

<그림 107 : client_7>

만약, transaction을 그만하기를 원하면, session을 종료시키고, 해당 session에서 수행한 transaction history를 출력해주는 것을 볼 수 있다. 그리고 atm을 그만 쓸 것인지를 물어본다. 만약 atm을 그만쓰지 않는다고 하면, 카드를 insert하는 client_2부분으로 돌아간다. ATM을 종료하기를 원할 경우는 밑에 설명해 두었다.

```

      ATM
    Do you want to finish ATM?
    0 : No
    1 : Yes
  > 1
      ATM
    Do you want to print reciept?
    0 : No
    1 : Yes

      ATM
    Transaction History :
    Transaction ID : 0
    Card Number : 111222333444
    Transacted Amount : 10000
    Transacted Fee : 3000
    Transaction condition :
      Transfer Success
    Transaction Type : Transfer
    Transfer Type : Account Transfer
    Source Acc : 111111111111
    Destination Acc : 222111111111

    Transaction ID : 2
    Card Number : 111222333444
    Transacted Amount : 10000
    Transacted Fee : 0
    Transaction condition :
      Deposit Success
    Transaction Type : Deposit
    Deposit Paper Type : Cash

      ATM
    Transaction History :
    Transaction ID : 4
    Card Number : 111222333444
    Transacted Amount : 10000
    Transacted Fee : 0
    Transaction condition :
      Deposit Success
    Transaction Type : Deposit
    Deposit Paper Type : Cash

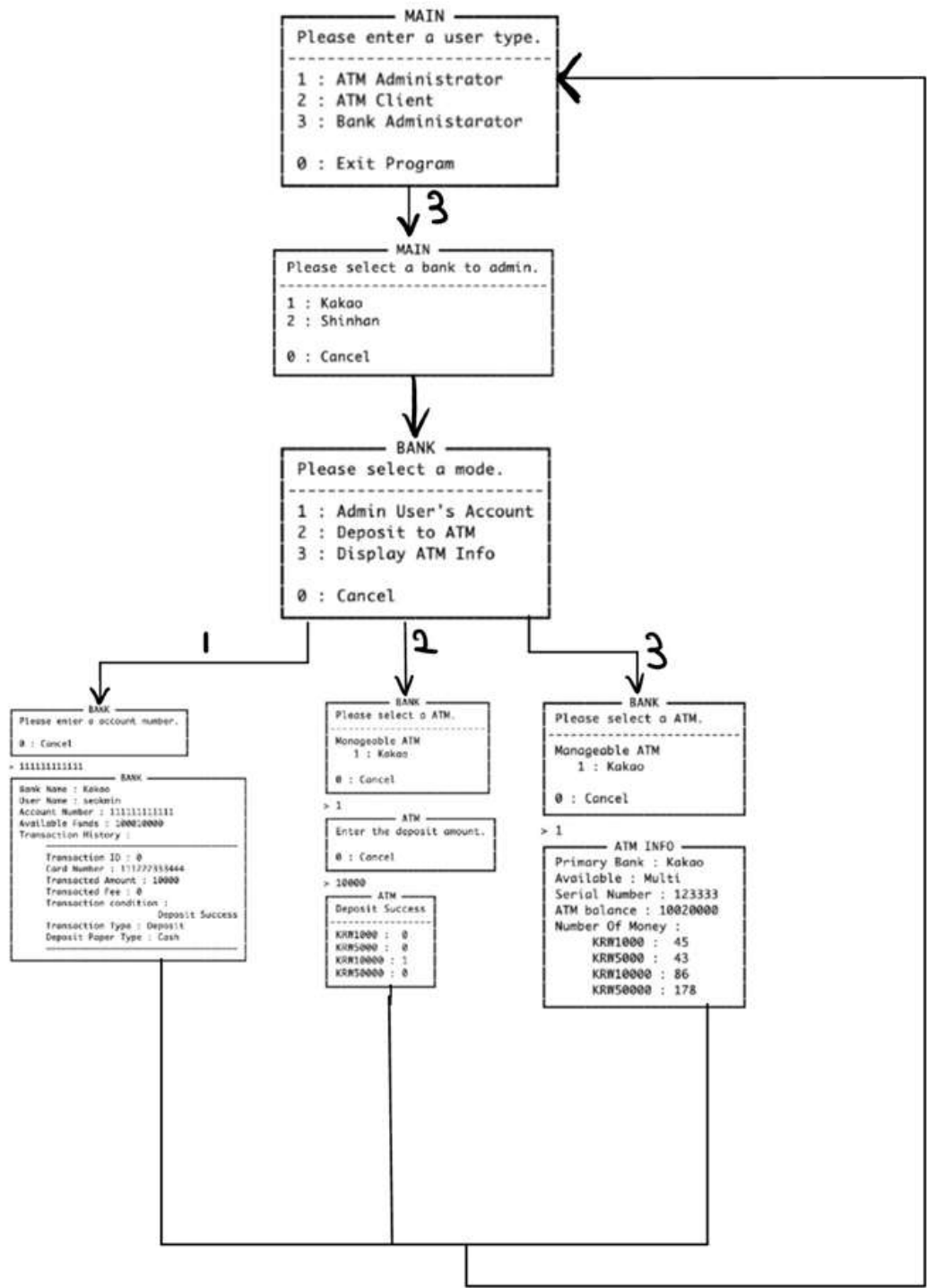
    Transaction ID : 6
    Card Number : 111222333444
    Transacted Amount : 40000
    Transacted Fee : 1000
    Transaction condition :
      Withdrawal Success
    Transaction Type : Withdrawal
  
```

<그림 108 : client_8>

<그림 109 : client_9_receipt>

만약, reciept를 출력하기를 원한다면, 한 사람이 atm을 시작하고, 끝낼 때 까지의 모든 transaction history를 출력해주고(원하지 않으면 출력하지 않음) 프로그램의 초기화면(entire_start)으로 돌아간다.

C-3) Bank administrator 창



<그림 110 : bank administrator state diagram>

위 그림은 bank administrator의 state diagram이다. exception은 표시하지 않았다.

자세한 설명은 밑에 적었다.

```

      MAIN
-----
Please select a bank to admin.
-----
1 : Kakao
2 : Shinhan
0 : Cancel

```

<그림 111: bank_1>

맨 처음으로는 어떤 bank를 admin 하고 있는지 물어본다.

```

      BANK
-----
Please select a mode.
-----
1 : Admin User's Account
2 : Deposit to ATM
3 : Display ATM Info
0 : Cancel

```

<그림 112 : bank_2>

admin하는 bank를 클릭하고 나면, 3가지 작업을 할 수 있다. 해당 bank의 user의 account에 접근할 수 있고, 관리하는 atm의 잔고를 늘려줄 수 있으며, 관리하는 atm의 information을 출력할 수 있다.

bank_3-1) Admin User's Account 창

```

      BANK
-----
Please enter a account number.
-----
0 : Cancel

```

> 111111111111

```

      BANK
-----
Bank Name : Kakao
User Name : seokmin
Account Number : 111111111111
Available Funds : 100010000
Transaction History :
-----
Transaction ID : 0
Card Number : 111222333444
Transacted Amount : 10000
Transacted Fee : 0
Transaction condition :
                        Deposit Success
Transaction Type : Deposit
Deposit Paper Type : Cash
-----

```

<그림 113 : bank_3_1>

관리하는 account의 account number만 알고있다면 해당 account의 정보를 열 수 있다.

bank_3-2) Deposit to ATM 창

```

      BANK
    -----
    Please select a ATM.
    -----
    Manageable ATM
      1 : Kakao
    0 : Cancel
  
```

> 1

```

      ATM
    -----
    Enter the deposit amount.
    0 : Cancel
  
```

> 10000

```

      ATM
    -----
    Deposit Success
    -----
    KRW1000 : 0
    KRW5000 : 0
    KRW10000 : 1
    KRW50000 : 0
  
```

<그림 114 : bank_3-2>

Deposit to atm을 클릭하면, 해당 bank가 관리하는 atm을 띄어주고, 어떤 atm에 돈을 넣을 것인지 묻는다. atm을 선택하고 나면, 입력한 amount만큼 해당 atm의 잔고를 늘려준다.

bank_3-3) Display ATM Info 창

```

      BANK
    -----
    Please select a ATM.
    -----
    Manageable ATM
      1 : Kakao
    0 : Cancel
  
```

> 1

```

      ATM INFO
    -----
    Primary Bank : Kakao
    Available : Multi
    Serial Number : 123333
    ATM balance : 10020000
    Number Of Money :
      KRW1000 : 45
      KRW5000 : 43
      KRW10000 : 86
      KRW50000 : 178
  
```

Display ATM Info를 클릭하면, 관리하는 atm의 목록을 출력한 후, 해당 atm의 정보를 출력해준다.

bank_3의 작업이 끝나면, 프로그램의 초기화면(entire_start)으로 돌아간다.

D) The list of concepts of object-oriented programming (e.g., inheritance, polymorphism), and explanation how they are used.

D-1) Abstraction

1. 카드를 넣는 행위를 카드번호를 입력하는 행위로 추상화 하였다.
2. 사용자가 atm기를 찾아가는 행위를 존재하는 atm중에서 atm을 고르는 것으로 추상화 하였다.
3. 사용자가 atm기기에 1000원권, 5000원권, 10000원권, 50000원권을 넣는 행위를 amount를 입력하는 것으로 추상화 하였다. (amount를 입력 받아 프로그램 내에서 실제라면 사용자가 각 지폐를 몇장 들고왔을지 예측하여 deposit 하였다.)
4. 사용자가 atm기기로부터 지폐를 받는 행위를 withdrawal이 끝난후, atm기기에서 1000원권, 5000원권, 10000원권, 50000원권이 빠져나간 장수를 보여주는 것으로 추상화 하였다.

D-2) Encapsulation

가능한 한에서는 class의 member variable을 private로 설정하여 class 외부에서 값에 무단으로 접근하여 member variable의 값이 부적절하게 손상되는 것을 차단하였다.

D-3) Inheritance

transaction의 종류는 deposit, withdrawal, transfer가 있는데, 이 3 종류의 transaction을 class로 구현할 때, transaction이라는 base class로부터 상속받아, 각각 transaction_deposit, transaction_withdrawal, transaction_transfer 라는 child class로 구현하였다.

공통된 member variable은 base class에서 다루고, 서로 다른 variable(transfer의 경우에 source account)만 child class 의 member variable로 설정하였다.

display_transaction 함수의 경우에는 공통된 코드는 base class의 함수에 넣어놓고, 서로다른 코드만 child class에 넣어서 후, virtual keyword로 인해 child class의 함수가 불리게 되면, child class의 함수에서 다시 base class의 함수를 부르게 하여, 각 transaction마다 적절한 정보가 출력되도록 디자인하였다.

Session에 transaction을 배열 형태로 저장할때는 서로다른 3개의 class를 upcasting을 이용하여 한 배열 안에 저장할 수 있었다.

D-4) Polymorphism

위에 Inheritance에서 설명했듯이, transaction_deposit, transaction_withdrawal, transaction_transfer class의 display_transaction 함수를 이용할 때 function overriding을 활용하여 같은 interface로 여러 기능을 사용할 수 있었다.

또 여러 class들의 생성자에서 function overloading을 사용하여 같은 interface로 상황에 따라 적절하게 instance를 만들 수 있었다.

E) The final version of source code (copy and paste from cpp file)

모든 코드를 캡처하여 이 섹션에 넣으면 너무 보고서가 난잡해 질 것 같아서 코드 캡처본을 따로 전체 zip파일에 저장해놓았습니다!

F) Member student contribution table and note (e.g., Student A: 50%, Student B: 20% or all students equally contributed)

student seongwoo kim : 50%, student seokmin lee : 50%