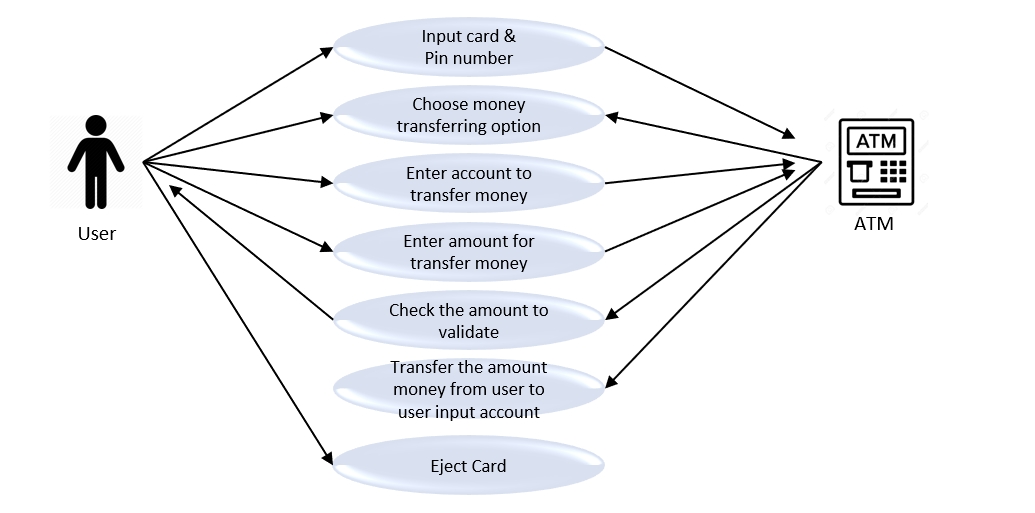
**LAB7\_assignment**

|  |  |  |  |
| --- | --- | --- | --- |
| Name: | 신민선 | Department: | 미디어학과 |
| Student ID: | 201723307 | Room Number: | 팔달관 333호 |
| Due Date: | April 20, 23 : 59 | | |

**Part 4. Exercises (15 points)**

1. **Analyze the “use case” scenario of Authenticate and Withdrawal, and then write the “use case” scenario for Transfer (5 points)**

**Attach1) “use case” diagram for Transfer**



**Attach 2) additional explanation**

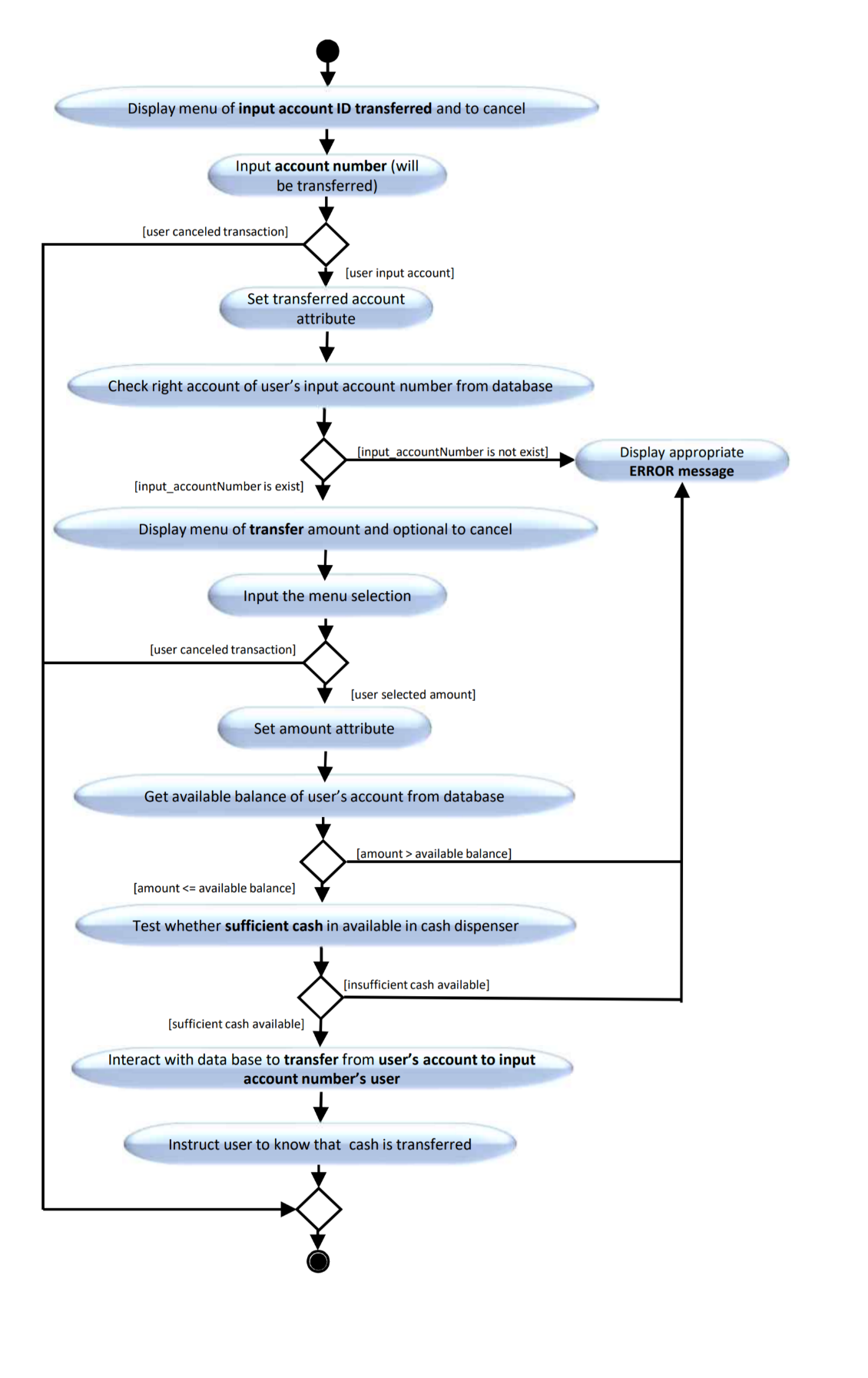
Compare of Authenticate and Withdrawal, Transfer scenario add two kinds of function.

First, add account number whose person will be transferred.

Second, add transfer method that withdraw in user’s account and then deposit in i nput\_accountNumber. So I additionally draw two diagram in withdrawal.

1. **Analyze the “activity diagram” of Withdrawal, and write the “activity diagram” for Transfer. (5 points)**

**Attach1) “activity diagram” for Transfer**



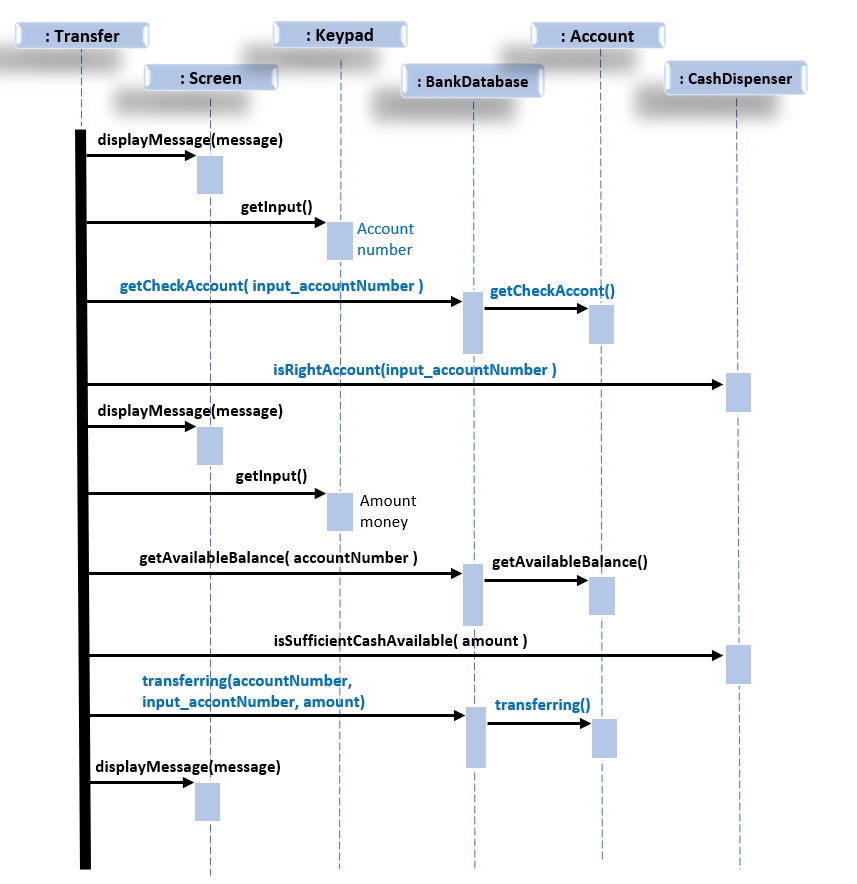
**Attach 2) additional explanation**

Compare to withdraw diagram, display transfer amount menus and cancel menu. After then, get input account data and check that account exist. If it doesn’t exist, display error message.

If It exist, then put the amount menu and get amount data. As same as accountNumber, check if user has enough money to transfer in own account. And then on database, they transfer money from user to user input account. And then exit.

1. **Write the “sequence diagram” for Transfer. (5 points)**

**Attach1) “sequence diagram” for Transfer**



**Attach 2) additional explanation**

To run Transfer(), I add some kinds of functions like getChenckAccount(), isRightAccount() and so on.

getCheckAccount() method is checking about input\_accountNumber really exist or not.

IsRightAccount() method is also checking about input\_accountNumber really exist in BankData.

Transferring() method is transferring money from user’s account to user input account

So, they can run transferring in ATM.

So eventually Transfer.java will **extends** Transaction like **class** Withdrawal