UML Sequence Diagram for ATM Using System

User → **ATM**: Insert Card
ATM → **Bank Server**: Verify Card

alt [if card is valid]
- **Bank Server** → **ATM**: Card OK
- **ATM** → **User**: Request PIN

[else]
- **Bank Server** → **ATM**: Card Invalid
- **ATM** → **User**: Eject Card

User → **ATM**: PIN Entered
ATM → **Bank Server**: Verify PIN

alt [if PIN is valid]
- **Bank Server** → **ATM**: PIN OK
- **ATM** → **User**: Request Amount

[else]
- **Bank Server** → **ATM**: PIN Invalid
- **ATM** → **User**: Eject Card

User → **ATM**: Amount Entered
ATM → **Bank Server**: Start Transaction
Bank Server → **Bank Account**: Sufficient Fund?

alt [if Fund OK]
- **Bank Account** → **Bank Server**: Fund OK
- **Bank Server** → **Bank Account**: Withdraw Amount
- **Bank Account** → **Bank Server**: Withdraw Successful
- **Bank Server** → **ATM**: Transaction Successful
- **ATM** → **User**: Recieve Cash

[else]
- **Bank Account** → **Bank Server**: Insufficient Fund
- **Bank Server** → **ATM**: Transaction Unsuccessful

ATM → **User**: Eject Card
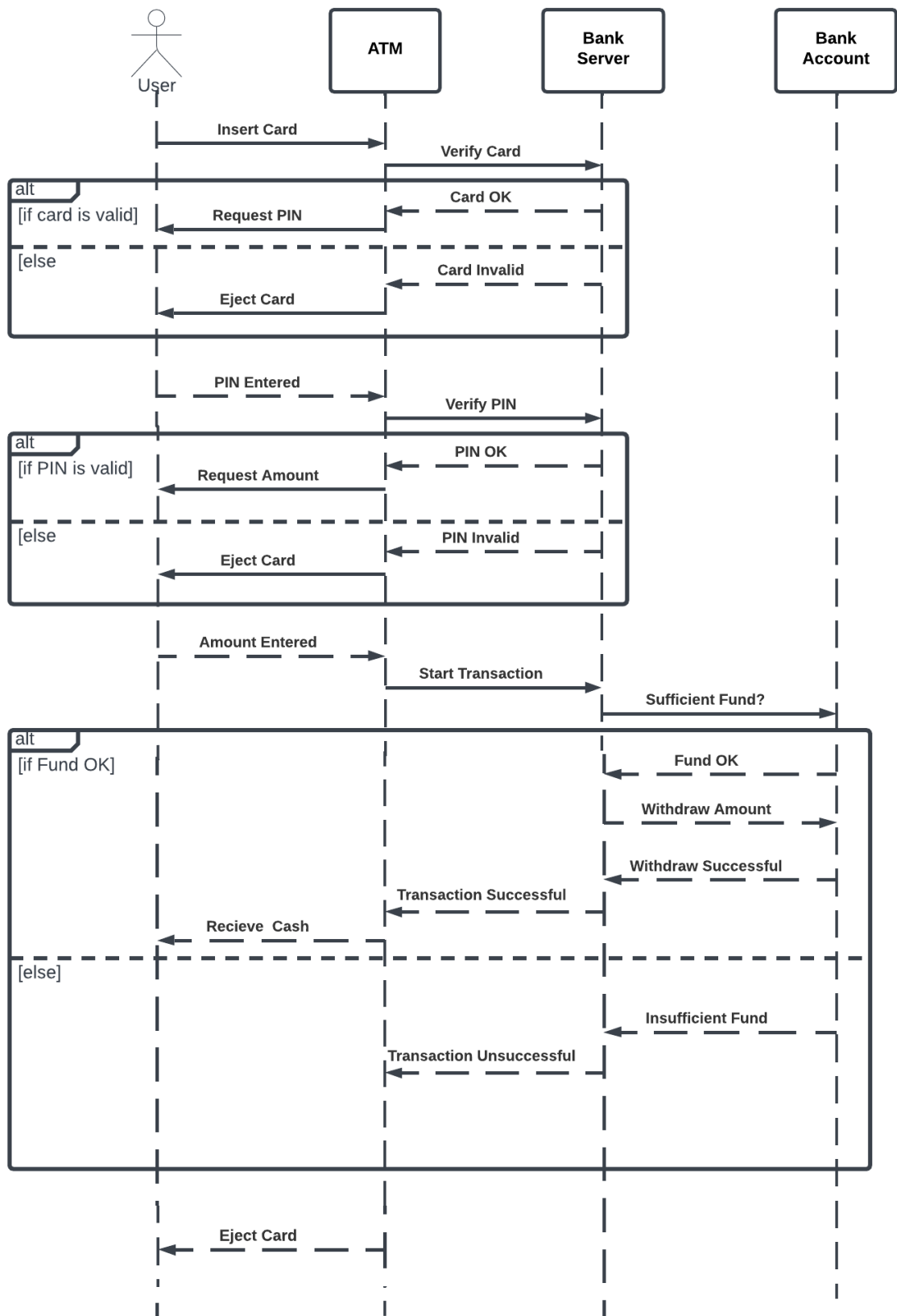
Lab #4 Assignment:

Create a Python program that simulates a basic ATM system which has been described on the UML sequence diagram above. The program should have the following classes:

Customer: represents a bank customer, with attributes name, PIN, and account_number.

BankAccount: represents a customer's bank account, with attributes account_number and balance, and methods deposit(), withdraw(), and get_balance().

BankServer: represents a bank server, with a dictionary of accounts and methods add_account() and get_account().

ATM: represents an ATM machine, with a bank server instance and methods authenticate(), display_menu(), withdraw(), deposit(), and main().

In the main program, create a BankServer instance and add a BankAccount object to it with the account_number "1234", a balance of 10000, and an account number of "12345678". Create an ATM instance with the BankServer instance, and call its main method to start the ATM system.

In the main method of the ATM class, prompt the user to enter their name, PIN, and account number. Create a Customer instance with the user input and call the authenticate method with the customer instance to obtain the BankAccount associated with that customer.

Then, display the main menu and prompt the user to enter a choice. If the user selects 1, display the account balance. If the user selects 2, prompt the user to enter an amount to withdraw and call the withdraw method with the BankAccount instance and the amount. If the user selects 3, prompt the user to enter an amount to deposit and call the deposit method with the BankAccount instance and the amount. If the user selects 4, exit the program.

Sample Output:

```
Enter your name: Jane
Enter your PIN: 8877
Enter your account number: 123
Welcome to ATM!
Click 1: Check Balance
Click 2: Withdraw
Click 3: Deposit
Click 4: Exit
Enter your choice: 1
You balance:  500

Welcome to ATM!
Click 1: Check Balance
Click 2: Withdraw
Click 3: Deposit
Click 4: Exit
Enter your choice: 3
Enter amount to deposit: 1000
Deposit successful. Current Balance: 1500
```