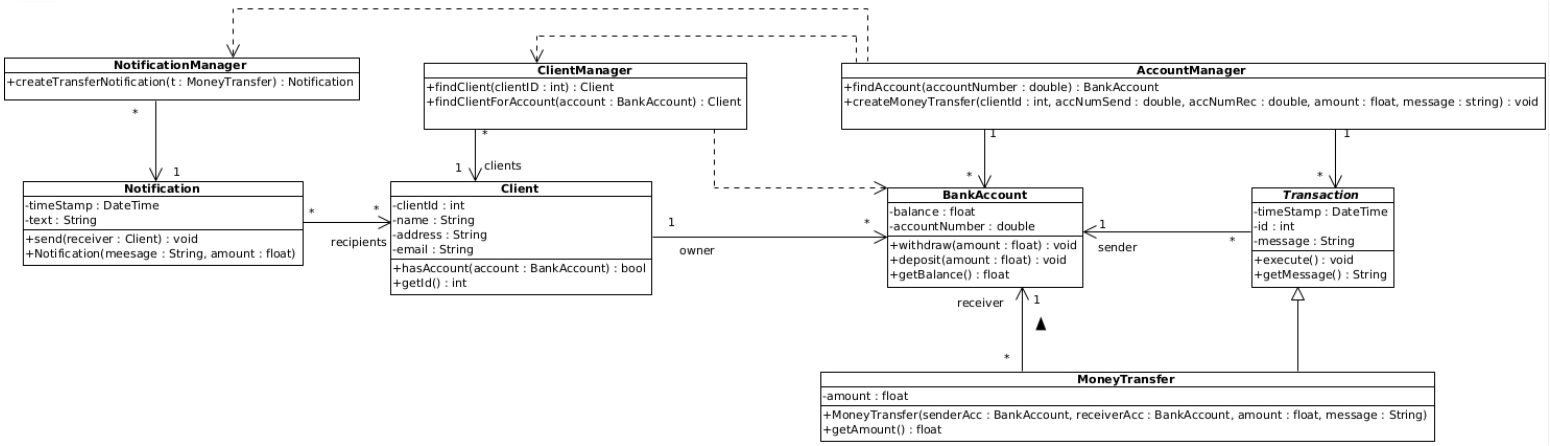
Task: Create a sequence diagram based on the provided pseudocode

Money transfer use case - main flow of events

1. Client selects option in the menu to transfer money to another account.
2. System displays form for input information about the transfer
3. Client fills the money amount, recipient account number and description.
2. System searches for client's and recipient's accounts.
3. System checks whether Client is the owner of the account
4. System checks whether client's account has sufficient balance.
5. IF the balance is sufficient
    5.1. System creates a new transaction.
    5.2. System updates the balance in both accounts by executing the transaction.
    5.3 System notifies both clients about the transaction.

Design class diagram



\

Pseudocode

Note: indentation represents the location where the code is executed, for example `findClient()` and `notif.send()` are called inside of the `createMoneyTransfer()` and `clientAcc.withdraw()` is called within the `execute()` method;

```
AccountManager.createMoneyTransfer(clientID, accountNumberSender, accountNumberReceiver, amount, msg);
                client = ClientManager.findClient(clientID);
                                        foreach (Client c in clients){
                                            cId = c.getId()
                                            if (cId == clientID)
                                                return c;
                                                break;
                                        return null;
                clientAcc = AccountManager.findAccount(accountNumberSender);
                receiverAcc = AccountManager.findAccount(accountNumberReceiver);
                if (client.hasAccount(clientAcc)){
                    balance = clientAcc.getBalance();
                    if (balance >= amount)
                        t = new Transaction (clientAcc, receiverAcc, amount, message);
                        t. execute()
                            clientAcc.withdraw(amount);
                            receiverAcc.deposit(amount);
                receiverClient = ClientManager.findClientForAccount(receiverAcc);
                notif = NotificationManager.createNotification(t);
                                                m = t.getMessage();
                                                a = t.getAmount();
                                                notif = new Notification(m, a);
                                                return notif;
                notif.send(client);
                notif.send(receiverUser);
```