atm-interface

September 17, 2023

[ ]:
```python
class Account:
    def __init__(self, user_id, pin):
        self.user_id = user_id
        self.pin = pin
        self.balance = 0
        self.transaction_history = []

    def deposit(self, amount):
        if amount > 0:
            self.balance += amount
            self.transaction_history.append(f"Deposited ${amount}")

    def withdraw(self, amount):
        if amount > 0 and self.balance >= amount:
            self.balance -= amount
            self.transaction_history.append(f"Withdrew ${amount}")

    def transfer(self, recipient, amount):
        if amount > 0 and self.balance >= amount:
            self.balance -= amount
            recipient.balance += amount
            self.transaction_history.append(f"Transferred ${amount} to␣
 ↪{recipient.user_id}")
            recipient.transaction_history.append(f"Received ${amount} from␣
 ↪{self.user_id}")

    def check_balance(self):
        return self.balance

    def view_transaction_history(self):
        return self.transaction_history


class ATM:
    def __init__(self):
        self.accounts = {}
```

```
    def add_account(self, user_id, pin):
        if user_id not in self.accounts:
            self.accounts[user_id] = Account(user_id, pin)

    def authenticate(self, user_id, pin):
        if user_id in self.accounts and self.accounts[user_id].pin == pin:
            return True
        return False


def main():
    atm = ATM()

    # Create user accounts
    atm.add_account("user123", "1234")
    atm.add_account("user456", "5678")

    while True:
        print("Welcome to the ATM")
        user_id = input("Enter your User ID: ")
        pin = input("Enter your PIN: ")

        if atm.authenticate(user_id, pin):
            user_account = atm.accounts[user_id]

            while True:
                print("\nATM Menu:")
                print("1. Check Balance")
                print("2. Deposit")
                print("3. Withdraw")
                print("4. Transfer")
                print("5. View Transaction History")
                print("6. Logout")

                choice = input("Enter your choice: ")

                if choice == "1":
                    print(f"Current Balance: ${user_account.check_balance()}")
                elif choice == "2":
                    amount = float(input("Enter the deposit amount: $"))
                    user_account.deposit(amount)
                elif choice == "3":
                    amount = float(input("Enter the withdrawal amount: $"))
                    user_account.withdraw(amount)
                elif choice == "4":
                    recipient_id = input("Enter recipient's User ID: ")
                    if recipient_id in atm.accounts:
```

```
                        amount = float(input("Enter the transfer amount: $"))
                        recipient_account = atm.accounts[recipient_id]
                        user_account.transfer(recipient_account, amount)
                    else:
                        print("Recipient not found.")
                elif choice == "5":
                    transactions = user_account.view_transaction_history()
                    for transaction in transactions:
                        print(transaction)
                elif choice == "6":
                    print("Logging out...")
                    break
                else:
                    print("Invalid choice. Please try again.")

        else:
            print("Authentication failed. Please try again.")


if __name__ == "__main__":
    main()
```

```
Welcome to the ATM
Enter your User ID: user123
Enter your PIN: 1234

ATM Menu:
1. Check Balance
2. Deposit
3. Withdraw
4. Transfer
5. View Transaction History
6. Logout
Enter your choice: withdraw
Invalid choice. Please try again.

ATM Menu:
1. Check Balance
2. Deposit
3. Withdraw
4. Transfer
5. View Transaction History
6. Logout
Enter your choice: Withdraw
Invalid choice. Please try again.

ATM Menu:
```

```
1. Check Balance
2. Deposit
3. Withdraw
4. Transfer
5. View Transaction History
6. Logout
Enter your choice: Check Balance
Invalid choice. Please try again.

ATM Menu:
1. Check Balance
2. Deposit
3. Withdraw
4. Transfer
5. View Transaction History
6. Logout
Enter your choice: 1
Current Balance: $0

ATM Menu:
1. Check Balance
2. Deposit
3. Withdraw
4. Transfer
5. View Transaction History
6. Logout
Enter your choice: 2
Enter the deposit amount: $10000000

ATM Menu:
1. Check Balance
2. Deposit
3. Withdraw
4. Transfer
5. View Transaction History
6. Logout
Enter your choice: 1
Current Balance: $10000000.0

ATM Menu:
1. Check Balance
2. Deposit
3. Withdraw
4. Transfer
5. View Transaction History
6. Logout
Enter your choice: Withdraw
Invalid choice. Please try again.
```

```
ATM Menu:
1. Check Balance
2. Deposit
3. Withdraw
4. Transfer
5. View Transaction History
6. Logout
Enter your choice: 3
Enter the withdrawal amount: $233023

ATM Menu:
1. Check Balance
2. Deposit
3. Withdraw
4. Transfer
5. View Transaction History
6. Logout
Enter your choice: 1
Current Balance: $9766977.0

ATM Menu:
1. Check Balance
2. Deposit
3. Withdraw
4. Transfer
5. View Transaction History
6. Logout
```

```
---------------------------------------------------------------------------
KeyboardInterrupt                         Traceback (most recent call last)
<ipython-input-12-cdc5d8822c6e> in <cell line: 102>()
    101
    102 if __name__ == "__main__":
--> 103     main()

<ipython-input-12-cdc5d8822c6e> in main()
     68                 print("6. Logout")
     69
---> 70                 choice = input("Enter your choice: ")
     71
     72                 if choice == "1":

/usr/local/lib/python3.10/dist-packages/ipykernel/kernelbase.py in␣
 ↪raw_input(self, prompt)
    849                 "raw_input was called, but this frontend does not␣
 ↪support input requests."
```

```
    850             )
--> 851         return self._input_request(str(prompt),
    852             self._parent_ident,
    853             self._parent_header,

/usr/local/lib/python3.10/dist-packages/ipykernel/kernelbase.py in␣
 ↪_input_request(self, prompt, ident, parent, password)
    893             except KeyboardInterrupt:
    894                 # re-raise KeyboardInterrupt, to truncate traceback
--> 895                 raise KeyboardInterrupt("Interrupted by user") from Non
    896             except Exception as e:
    897                 self.log.warning("Invalid Message:", exc_info=True)

KeyboardInterrupt: Interrupted by user
```

[ ]: