Oasis Infobyte

Task1:

```
        import java.time.LocalDateTime;
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

public class Main{
    private static String userId = "user123";
    private static String userPin = "1234";
    private static double balance = 100000.0;
    private static List<Transaction> transactionHistory = new
ArrayList<>();

    private static class Transaction {
        private String type;
        private double amount;
        private LocalDateTime timestamp;

        public Transaction(String type, double amount) {
            this.type = type;
            this.amount = amount;
            this.timestamp = LocalDateTime.now();
```

```
    }

    public String getType() {
        return type;
    }

    public double getAmount() {
        return amount;
    }

    public LocalDateTime getTimestamp() {
        return timestamp;
    }
}

private static void showOptions() {
    System.out.println("Available options:");
    System.out.println("1. Transaction History");
    System.out.println("2. Withdraw");
    System.out.println("3. Deposit");
    System.out.println("4. Transfer");
    System.out.println("5. Quit");
}
```

```
    private static void showTransactionHistory() {
        System.out.println("Transaction History:");
        for (Transaction transaction : transactionHistory) {
            System.out.println(
                    transaction.getType() + " - " + transaction.getAmount() + "
- " + transaction.getTimestamp()
            );
        }
    }

    private static void withdraw(double amount) {
        if (amount > 0 && amount <= balance) {
            balance -= amount;
            Transaction withdrawalTransaction = new
Transaction("Withdrawal", amount);
            transactionHistory.add(withdrawalTransaction);
            System.out.println("Withdrawal successful. New balance: " +
balance);
        } else {
            System.out.println("Invalid withdrawal amount or insufficient
balance.");
        }
    }

    private static void deposit(double amount) {
```

```
        if (amount > 0) {
            balance += amount;
            Transaction depositTransaction = new Transaction("Deposit",
amount);
            transactionHistory.add(depositTransaction);
            System.out.println("Deposit successful. New balance: " +
balance);
        } else {
            System.out.println("Invalid deposit amount.");
        }
    }

    private static void transfer(String recipientUserId, double amount) {
        // Assuming a simple transfer where the recipient is another
user with a known user ID
        if (amount > 0 && amount <= balance) {
            balance -= amount;
            Transaction transferTransaction = new Transaction("Transfer to
" + recipientUserId, amount);
            transactionHistory.add(transferTransaction);
            System.out.println("Transfer successful. New balance: " +
balance);
        } else {
            System.out.println("Invalid transfer amount or insufficient
balance.");
```

```
    }
  }

  public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);

    System.out.println("Welcome to the ATM!");
    System.out.print("Enter your user ID: ");
    String inputUserId = scanner.nextLine();
    System.out.print("Enter your user PIN: ");
    String inputUserPin = scanner.nextLine();

    if (inputUserId.equals(userId) && inputUserPin.equals(userPin))
{
      System.out.println("Login successful! Welcome, " +
inputUserId + ".");
      showOptions();
      boolean quit = false;

      while (!quit) {
        System.out.print("Enter your choice: ");
        int choice = scanner.nextInt();
        scanner.nextLine(); // Consume the newline character after
reading the integer
```

```
            switch (choice) {
                case 1:
                    showTransactionHistory();
                    break;
                case 2:
                    System.out.print("Enter the amount to withdraw: ");
                    double withdrawAmount = scanner.nextDouble();
                    withdraw(withdrawAmount);
                    break;
                case 3:
                    System.out.print("Enter the amount to deposit: ");
                    double depositAmount = scanner.nextDouble();
                    deposit(depositAmount);
                    break;
                case 4:
                    System.out.print("Enter the user ID to transfer: ");
                    String transferUserId = scanner.next();
                    System.out.print("Enter the amount to transfer: ");
                    double transferAmount = scanner.nextDouble();
                    transfer(transferUserId, transferAmount);
                    break;
                case 5:
                    System.out.println("Thank you for using the ATM.
Goodbye!");
```

```
                    quit = true;
                    break;
                default:
                    System.out.println("Invalid choice. Please try again.");
            }
        }
    } else {
        System.out.println("Login failed. Please check your user ID and PIN.");
    }

    scanner.close();
  }
}
```

Output:

```
Welcome to the ATM!
Enter your user ID: user123
Enter your user PIN: 1234
Login successful! Welcome, user123.
Available options:
1. Transaction History
2. Withdraw
3. Deposit
4. Transfer
5. Quit
Enter your choice: 2
Enter the amount to withdraw: 8764
Withdrawal successful. New balance: 91236.0
Enter your choice: 1
Transaction History:
Withdrawal - 8764.0 - 2023-08-06T17:54:40.109128
Enter your choice: 3
Enter the amount to deposit: 2532
Deposit successful. New balance: 93768.0
Enter your choice: 4
Enter the user ID to transfer: user123
Enter the amount to transfer: 78366
Transfer successful. New balance: 15402.0
Enter your choice: 5
Thank you for using the ATM. Goodbye!
```