

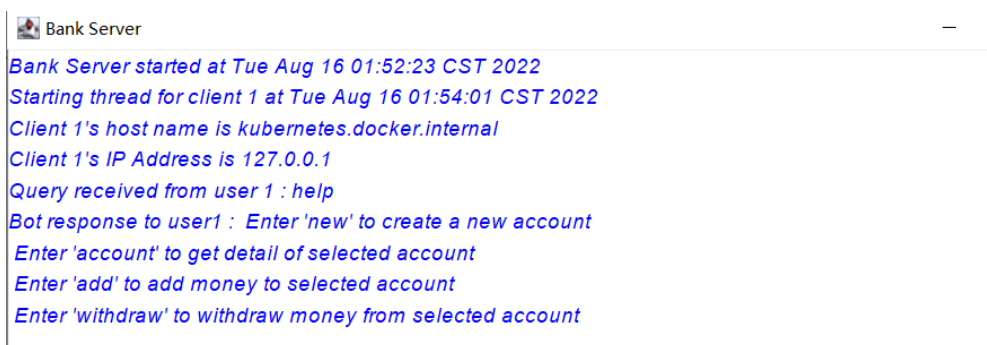
Final Project

Linghang Wang lw3150

Abstract

This project is a bank simulator where you can create your own account, add, and withdraw money from the account. The bank simulator also allows multiple users to use this program at the same time in terms of creating accounts as well as operating their accounts.

The concepts I implemented in this project are networking, multi-thread, and database. The networking is completed by building a network communication between the client (front end) and the server (back end). The server is multi-thread so that it is capable of receiving and handling multiple connections at once. A database (DB) is implemented to store the account details and store/retrieve the changes. A GUI is also implemented for users to give instructions and receive feedback.



```
Bank Server
Bank Server started at Tue Aug 16 01:52:23 CST 2022
Starting thread for client 1 at Tue Aug 16 01:54:01 CST 2022
Client 1's host name is kubernetes.docker.internal
Client 1's IP Address is 127.0.0.1
Query received from user 1 : help
Bot response to user1 : Enter 'new' to create a new account
Enter 'account' to get detail of selected account
Enter 'add' to add money to selected account
Enter 'withdraw' to withdraw money from selected account
```

Figure 1. Server interface

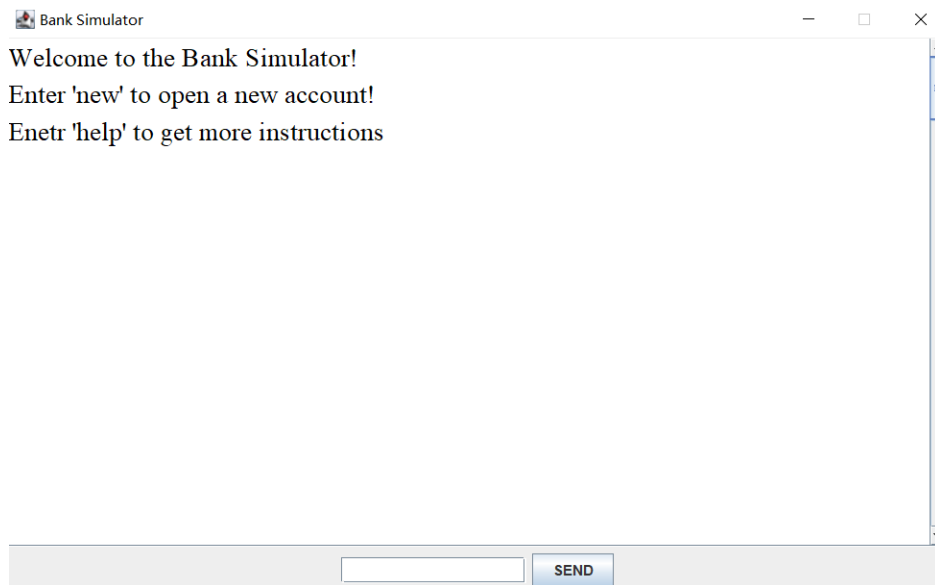


Figure 2. GUI of Bank Simulator

Table of Contents

Abstract	1
Table of Contents	3
List of Illustrations	4
Functions Description	5
Startup	5
Instructions List	6
Create Account.....	6
Add Deposit	7
Withdraw Deposit	8
Account Details.....	8
Additional Details	9
Conclusion	10

List of Illustrations

Title	Page
Figure 1. Server interface	1
Figure 2. GUI of Bank Simulator	2
Figure 3. Initial Server Interface at Started	5
Figure 4. Initial GUI of BankClient	5
Figure 5. Detailed Instructions invoked by keyword “help”	6
Figure 6. Process of Create Account	7
Figure 7. Process of Adding Money to Selected Bank Account	7
Figure 8. Withdraw Value from Selected Account	8
Figure 9. Selected Account Details	9
Figure 10. Providing Non-numeric Response when Creating Account Number	9
Figure11. Non-acceptable Instructions	10

Functions Description

Startup

1. Run BankServer.java to start the server. The server interface will show the time when the server is started in EDT time zone.



Figure 3. Initial Server Interface at Started

2. Run BankClient.java to start the client GUI where most interacts are executed. The detailed description of “new” and “help” will be discussed below at [Create Account](#) and [Instructions List](#).

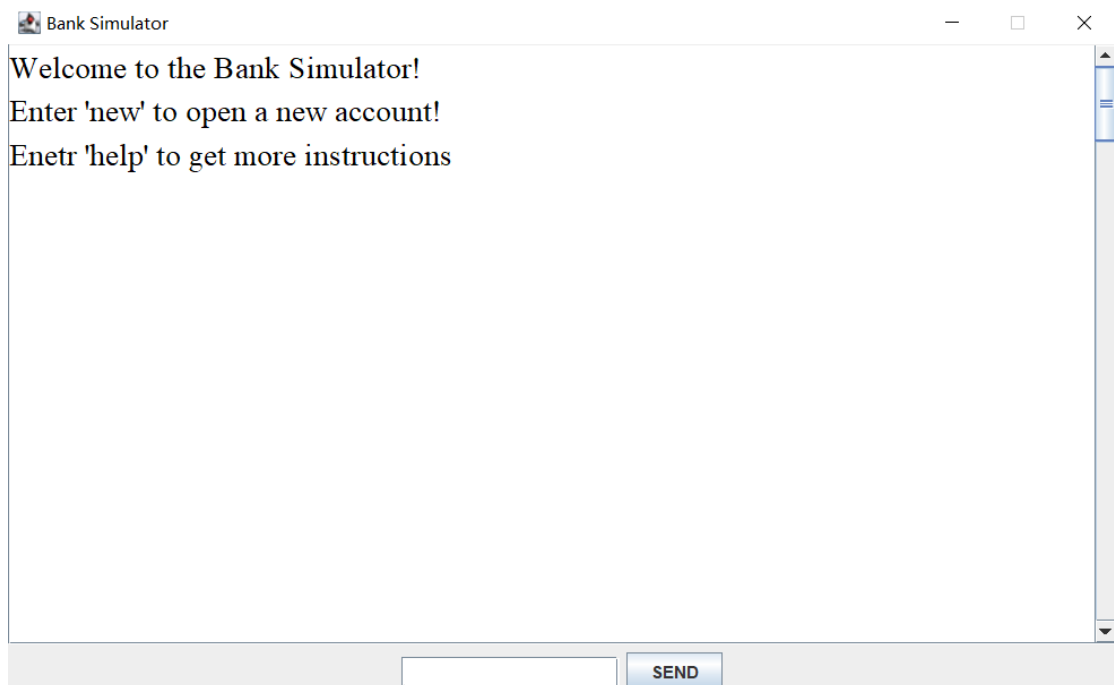



Figure 4. Initial GUI of BankClient

Instructions List

When typing in “help” to the server, detailed instructions will be provided to users for reference. Instructions that are not listed here are not acceptable and will not realize any functions.

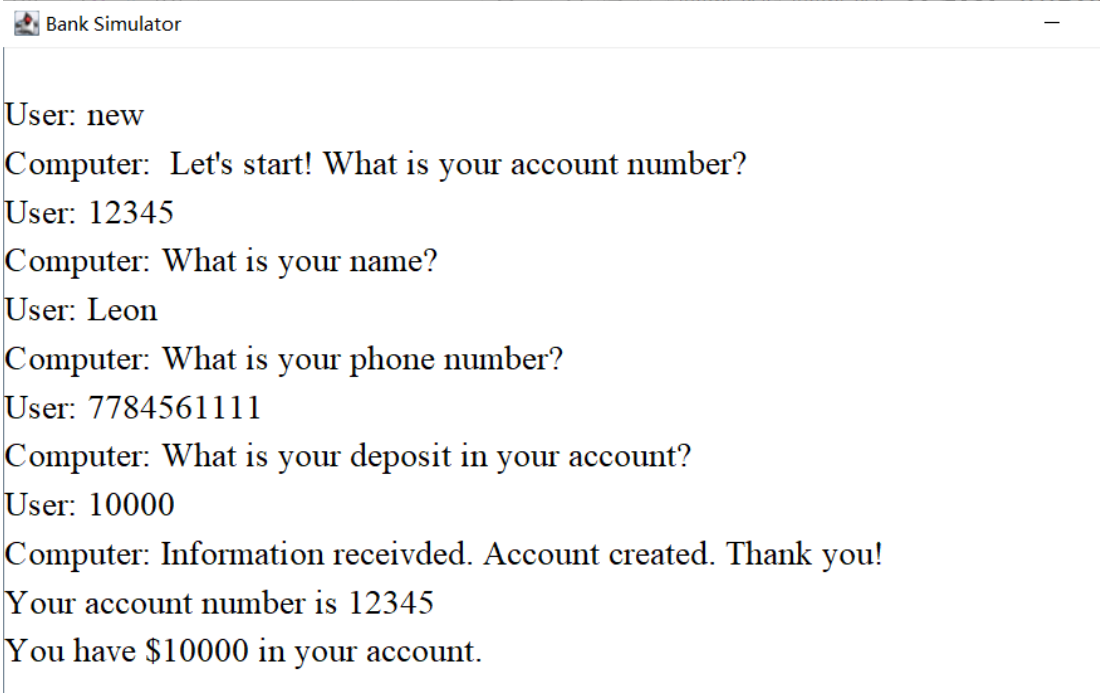
 Bank Simulator

Welcome to the Bank Simulator!
Enter 'new' to open a new account!
Enter 'help' to get more instructions
User: help
Computer: Enter 'new' to create a new account
Enter 'account' to get detail of selected account
Enter 'add' to add money to selected account
Enter 'withdraw' to withdraw money from selected account

Figure 5. Detailed Instructions invoked by keyword “help”

Create Account

When user types in “new” in the text field and send it to the server, the server invokes `createAccount()` method to guide the user to provide information step by step. After finishing gathering information, the server will insert user’s information to Registration table in the database.



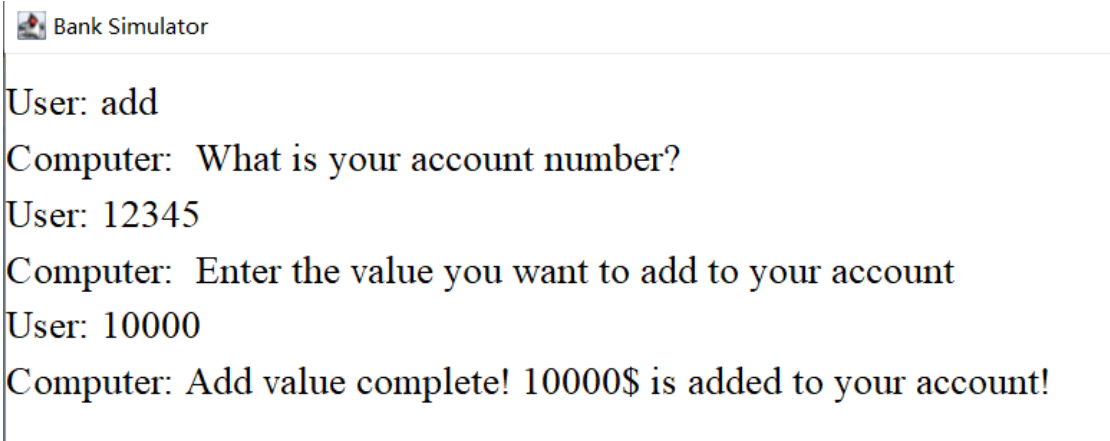
Bank Simulator

User: new
Computer: Let's start! What is your account number?
User: 12345
Computer: What is your name?
User: Leon
Computer: What is your phone number?
User: 7784561111
Computer: What is your deposit in your account?
User: 10000
Computer: Information received. Account created. Thank you!
Your account number is 12345
You have \$10000 in your account.

Figure 6. Process of Create Account

Add Deposit

As listed in the instructions, when user send “add” to the server, the server invokes addMoney() method to allow user add desired value to selected bank account and change the value in “money” field in the database.



Bank Simulator

User: add
Computer: What is your account number?
User: 12345
Computer: Enter the value you want to add to your account
User: 10000
Computer: Add value complete! 10000\$ is added to your account!

Figure 7. Process of Adding Money to Selected Bank Account

Withdraw Deposit

By typing and sending keyword “withdraw” to the server, withdrawMoney() method will be invoked. Same as adding money to the account, withdraw deposit has very similar process and outcome. But it is worth noting that the value withdrawn cannot be greater than the value in the selected account. In other words, deposit in a bank account cannot be negative to guarantee that it is conformity with the reality.

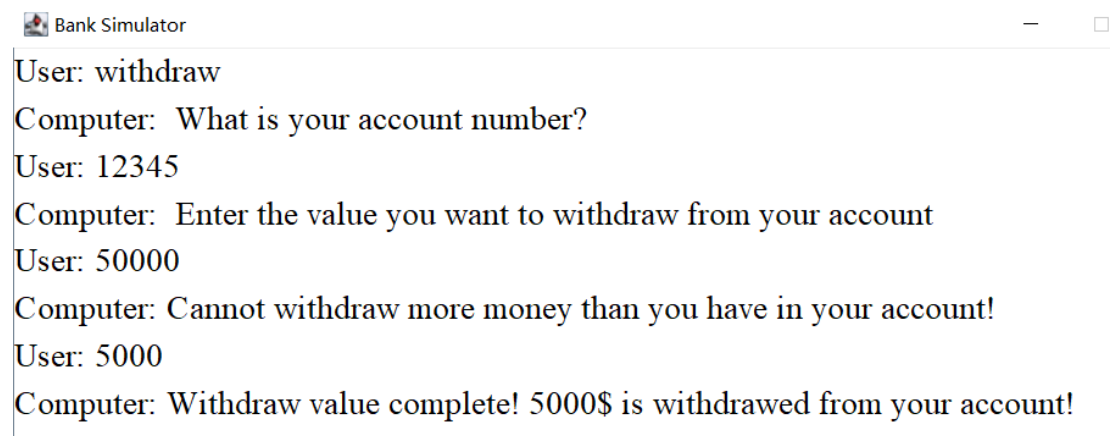



Figure 8. Withdraw Value from Selected Account

Account Details

Whenever users want to know the details of a certain bank account, they can type in “account” to invoke the checkAccount() method that returns the detailed information of selected account, such as account number, user’s Name, phone number and deposit value.

 Bank Simulator


User: account
Computer: What is your account number?
User: 12345
Computer:
Account number = 12345
User name = leon
Phone number = 7784561111
Value = 15000

Figure 9. Selected Account Details

Additional Details

When communicating with the server, there are some restrictions to make sure that the program runs without errors.

Firstly, whenever server requires a numeric response (e.g. phone number, deposit etc.), if users provide response that is not numeric, the program will notify the users that the response is not acceptable.

 Bank Simulator

User: new
Computer: Let's start! What is your account number?
User: lo
Computer: It is not number, please re-enter your account number.
User: number
Computer: It is not number, please re-enter your account number.

Figure 10. Providing Non-numeric Response when Creating Account Number

Secondly, when users send unknown instructions, the server will advise

that the instruction is not connected to any functions in this program and will require users to re-enter instructions.

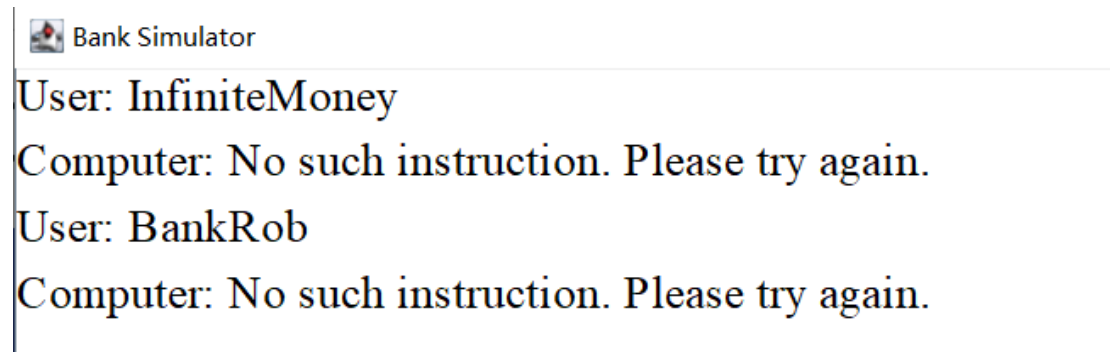


Figure 11. Non-acceptable Instructions

Conclusion

This project provides me a chance to study and implement the advanced Java concepts by myself. It is not only challenging when dealing with the difficulties during coding and debugging, but also fascinating because I was completely immersed myself in the journey of exploring and advancing.