Bank Network

Project based on making P2P client/server banking system.

Project Documentation: Bank Network

Project Information

- **Project Name:** Bank Network
- **Author:** Jakub Hofman (C4c)
- **Contact Information:** hofmjakub@gmail.com
- **Date of Completion:** 7. 2. 2025 (Friday)
- **School:** SPŠE Ječná (Czech Republic, Prague)
- **Project type:** Academical

Description

Bank Network is application which allows users to **experience managing real bank**. All clients can connect via **LAN IP and port** number of the computer on which the app is running. Once user's connected he can **use specific commands to manage the bank's accounts** (create, remove, withdraw, deposit, and much more). Bank Network also **functions with other applications of this type**, so feel free to try it out with your friends!

User Requirements

- Computer, keyboard, mouse, monitor, cables...
- Stable **Internet Connection**

- **Java**: OpenJDK 17 or newer
- **Maven**: (source, target) 17 or newer
- **MySQL Connector**: 8.0.25
- **Operating System**: Windows 10 or newer

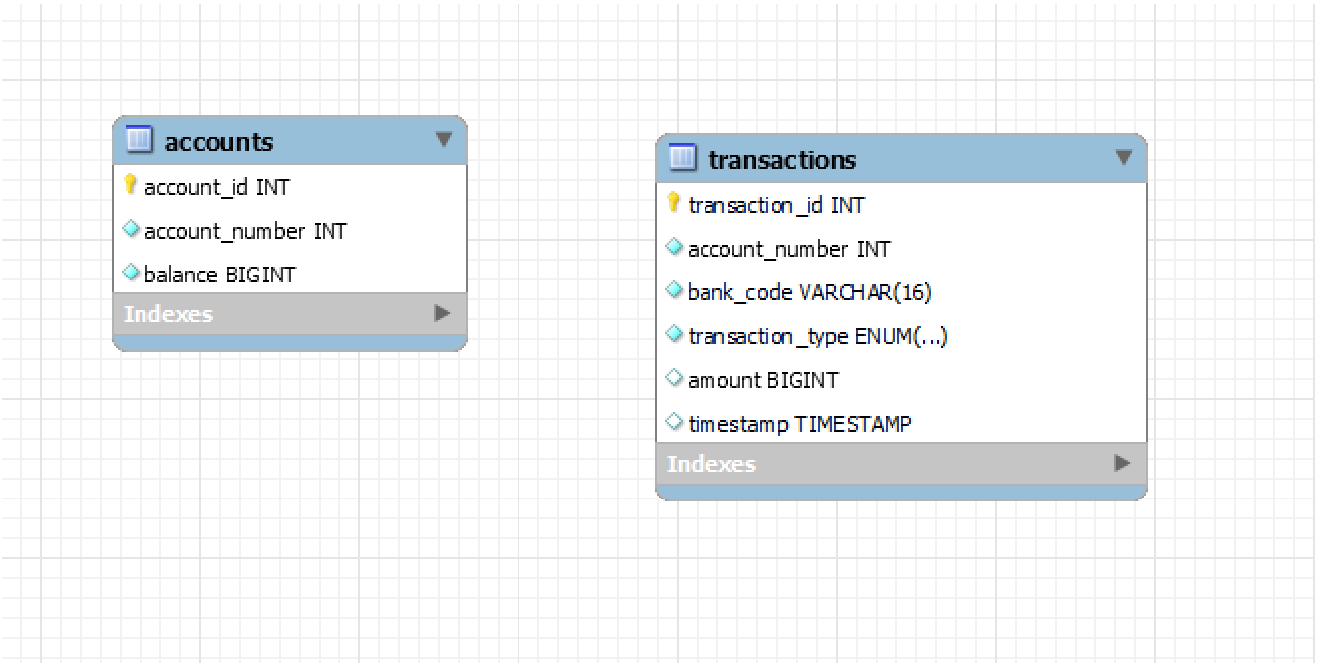
Architecture and Design

The system mainly follows an **ORM** (Object-Relation Mapping), **Singleton** (for database connection) design patterns. For all commands there is **Command design pattern** applied which perfectly fits to the architecture in BankConsole.

More detailed description of how the these patterns are used in (among classes) is in the code itself.

Database Structure

E-R Model:



- ***accounts*** = table for storing the bank's accounts with their **number and balance**
- ***transactions*** = table for storing all the interesting **transactions (operations) that were processed by the users** (creating/removing accounts, deposits, withdrawals)

Configuration

The following configuration options are available:

1. **Database Configurations** (editable in `config.properties` in `/resources` directory):
   - `db.url` : Database connection string.
   - `db.username` : Username for database.
   - `db.password` : Password for database.
   - `db.driver` : Specify MySQL driver (*Optional*)
2. **Application-Specific Configurations**:
   - Server settings:
       - `server.port` : Port on which the app runs
       - `server.port.min.range` : Starting port for searching the Network for available bank applications
       - `server.port.max.range` : Ending port for searching the Network for available bank applications
       - `server.thread.pool.size` : Maximum number of clients that can be connected at a time
       - `server.backlog` : Maximum amount of clients who can wait for the connection
       - `server.client.timeout` : Timeout for Client when being idle
       - `server.client.connect.timeout` : Connection timeout
   - Client settings:
       - `client.command.timeout` : Command timeout
       - `client.account.number.min.range` : Minimum account number value

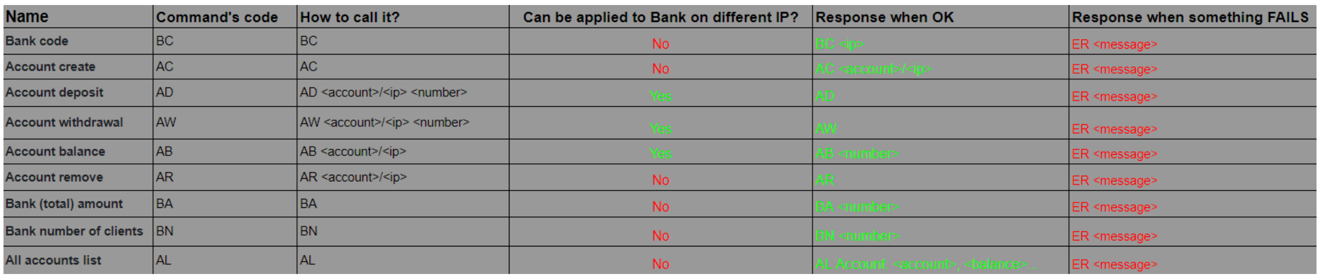
- `client.account.number.max.range` : Maximum account number value
- `client.account.balance.min` : Starting bank account amount

How it works?

1. **Launch the program on your PC** by following the README.md on my GitHub.
2. **Connect to it** via PuTTY or Telnet enter the IP and Port on which it runs.
3. **Manage the bank by using commands** and enjoy!

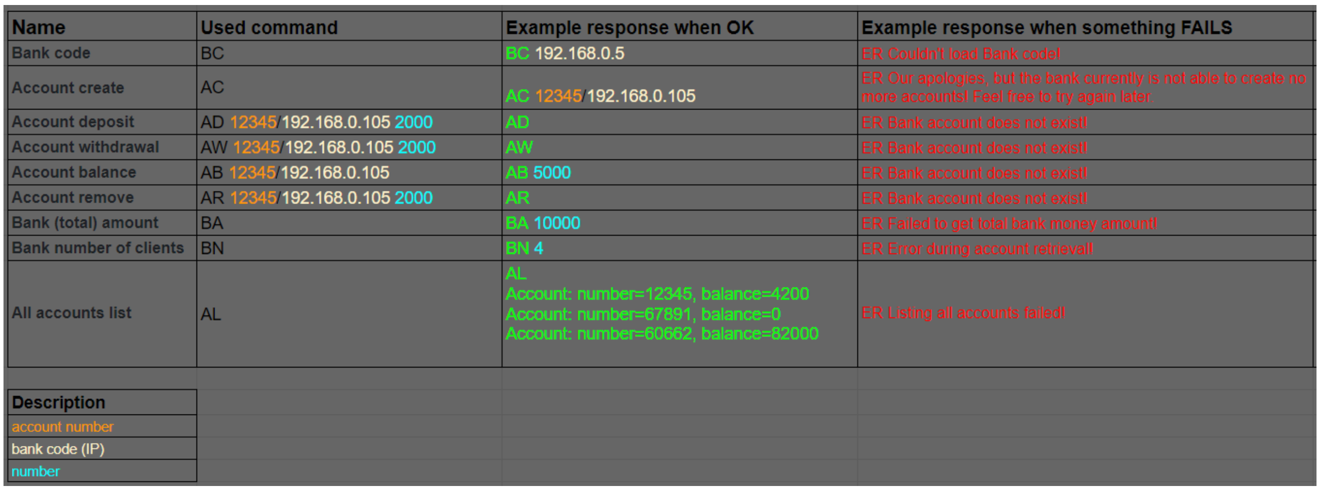
Commands

- Once you have the PuTTY (or other client) opened just **write the command's code** with attributes (if required) and press **"Enter"** key
- Then you get the result



Name	Command's code	How to call it?	Can be applied to Bank on different IP?	Response when OK	Response when something FAILS
Bank code	BC	BC	No	BC <ip>	ER <message>
Account create	AC	AC	No	AC <account>/<ip>	ER <message>
Account deposit	AD	AD <account>/<ip> <number>	Yes	AD	ER <message>
Account withdrawal	AW	AW <account>/<ip> <number>	Yes	AW	ER <message>
Account balance	AB	AB <account>/<ip>	Yes	AB <number>	ER <message>
Account remove	AR	AR <account>/<ip>	No	AR	ER <message>
Bank (total) amount	BA	BA	No	BA <number>	ER <message>
Bank number of clients	BN	BN	No	BN <number>	ER <message>
All accounts list	AL	AL	No	AL Account: <account>, <balance>...	ER <message>

Example usage:



Name	Used command	Example response when OK	Example response when something FAILS
Bank code	BC	BC 192.168.0.5	ER Couldn't load Bank code!
Account create	AC	AC 12345/192.168.0.105	ER Our apologies, but the bank currently is not able to create no more accounts! Feel free to try again later.
Account deposit	AD 12345/192.168.0.105 2000	AD	ER Bank account does not exist!
Account withdrawal	AW 12345/192.168.0.105 2000	AW	ER Bank account does not exist!
Account balance	AB 12345/192.168.0.105	AB 5000	ER Bank account does not exist!
Account remove	AR 12345/192.168.0.105 2000	AR	ER Bank account does not exist!
Bank (total) amount	BA	BA 10000	ER Failed to get total bank money amount!
Bank number of clients	BN	BN 4	ER Error during account retrieval!
All accounts list	AL	AL Account: number=12345, balance=4200 Account: number=67891, balance=0 Account: number=60662, balance=82000	ER Listing all accounts failed!

Description
account number
bank code (IP)
number

How to install?

The GitHub repository contains a detailed `README.txt` file with further instructions.

- GitHub: https://github.com/Mithynite/BankNetwork

Third-Party Libraries

- **MySQL Connector/J** (a JDBC driver for MySQL databases)

Conclusion

This application provides an option to experience banking system via your own LAN network. You can use series of commands to manage to bank. Also the app can be applied to multiple local networks. Also many parts of the code are reusable in my future projects.

Documentation prepared by Jakub Hofman on 7. 2. 2025

Sources

- https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/net/Socket.html
- https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/util/concurrent/ExecutorService.html
- https://www.baeldung.com/java-executor-service-tutorial
- https://youtu.be/Nb85yJ1fPXM?feature=shared
- https://medium.com/@vikas.taank_40391/understanding-java-future-and-completable-future-eeef49fb430f

- https://www.geeksforgeeks.org/java-net-inetaddress-class-in-java/
- ChatGPT chat: https://chatgpt.com/share/67a66bf3-4204-800f-ad44-50aa4347d24f