Lab W05 – Online Bank Continued

Overview

This project continues to develop an online banking system using a multi-server, multi-client, socket-based architecture.

- The project follows an iterative development model
- You will make design decisions that may change over time
- Requirements may be refined to better support later phases
- Learning and implementation happen at the same time

I am coding this assignment alongside you. I will try to minimize major changes, but some may be unavoidable.

Expectations & Grading

- Each assignment includes core requirements (sometimes general, sometimes specific)
- You have some flexibility in design decisions, but that does not mean anything goes
- Discovering and fixing earlier decisions is part of the learning process
- A significant portion of your grade is based on:
    - Applying concepts from lectures and the textbook
    - Applying them correctly and appropriately
    - Minor decisions that need revision later will unlikely be penalized
- Code that merely "works" is not enough, and it should reflect the concepts we discuss.
- Example: If a value should be an Integer object, and you use a primitive int:
    - It is unlikely that points will be deducted
    - If it causes issues later, you will have to fix it in a future iteration anyway
    - This is how you learn the semantics, subtleties, and nuances of the language.

Code Visibility

After grading, your code will be visible to the combined course.

- Students may learn from each other's ideas and design approaches
- Code may not be copied
- Submit your best work both to learn from others and to avoid being an example of what not to do

Questions & Communication

Have questions? Please ask.

- Email: short, quick questions only, and combine multiple small questions into one message
- Longer or complex questions: office hours or a scheduled meeting
- I am unlikely to respond in the evenings or weekends

Managing your time and planning around limited access to your "boss" is part of the experience.

Generative AI Policy

Do NOT use generative AI for these labs.

- Eclipse IDE-generated boilerplate is acceptable
- Generative AI is not acceptable

My experience with AI-generated solutions for this project has been consistently bad:

- It makes incorrect assumptions
- Uses concepts and features we have not covered
- Increases complexity and application size with buried layers of Java library objects
- Feels like the AI programmers are trying to show off complex code
- Reduces performance

Do not be a sloperator. You are fully capable of producing cleaner, more efficient, and more thoughtful code than AI. That is a goal of this assignment and future-proofing your job.

Instructions

This assignment will focus on two important aspects: collections and data files. First, create a copy of your Lab W04 project and name this project lastnameLabW05. Note that you may need to continue working on this lab series without feedback for Lab W04.

Part 1

You will decide which Java Collections Framework implementations are appropriate for storing bank data. To do this, read the chapter on collections and select the collection(s) that best fit the purpose and context of our labs.

Think carefully about performance and scalability. Assume you are designing software for a multi-server, multi-client online banking system that may process millions of transactions per week, especially during peak usage. Not all collections are equal in this context, and some choices are correct, while others are incorrect. Hint: ArrayList is not the best option here.

For each concrete collection discussed in the chapter, write a brief justification explaining:

- A short overview of how the collection works
- Why it should or should not be used to store our bank data
- Any relevant considerations

Then rank the collections from best to worst for our banking system.

I am not expecting pages of documentation. However, your work should demonstrate clear reasoning and be something you could confidently explain during a technical meeting (or perhaps a technical job interview?).

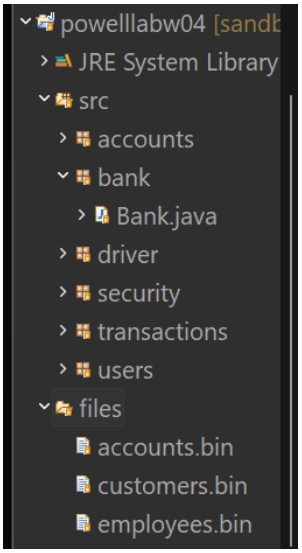
Requirements:

- Your name at the top of the document
- Writing that is clear and organized
- You do not need to write formal paragraphs. You may use tables, bullet lists, short sections, diagrams, or any format that helps clearly explain your reasoning.

After the due date, all submissions will be shared with the class so we can all learn different perspectives on it.

Once you select the best collection, implement it into the bank to hold the data.

Part 2



We recently fired our data files expert because he was earning a six-figure salary, and generative AI can obviously convert any text files we have into the binary files we need for our bank. We had it create the binary files you will find in the repository for this assignment. The following describes the original CSV files (note that this may be different from what you saw in the previous lab instructions). Place the files in a subfolder of the project.

File: employees.csv

Header row:
`id,first,last,email,username,password,pin,title,type,vault`

Examples:

`1264838,Calum,Van Laar,VanLaar@CISBank.edu,vanlaarca,29M<F8S6hzJ]~7xM,3029,Administrative Assistant,Associate,`

`1934208,Inigo,Benton,Benton@CISBank.edu,bentonin,dQJHn$8D0YcEu&TO,1246,Loan Manager,Manager,1489626`

Note: Only managers have vault pin numbers.

File: customers.csv

Header row: `id,firstname,lastname,email,username,password,pin`

Examples:

`2846341,Alina,Santiago,Santiago@cis3970.edu,santiagoal,x7zMHkCOG09fX%sR,3311`

`2203845,Amanda,Emerson,Emerson@cis3970.edu,emersonam,C&$L2td@fiS39!gG,2257`

File: accounts.csv

Header row: `number,balance,APY,authorized,authorized1,authorized2,authorized3,authorized4`

`7222671,81000.97,4.15,2,2300011,2394174,,`

`9660676,85028.25,4.25,3,2203845,2683025,2262403,`

Note: “authorized” is the number of authorized customers of the account (the original file had a max of 4). The first one is always the account owner; the others can also use the account. Every account has an owner, but not all accounts will have other authorized owners.

Your bank class will need methods (I would expect one for each file) that read the binary files and store the data in the accounts and users collections.

Submission

Submit your project to your cis3970work repository. As before, you will have a lastnameLabW05 folder in the repository. Also, place your collections document from part 1 in that folder.