

Arizona State University
CSE 434 SP Group 89

Design Document

Sean Berrios
Spring 2023

Table of Contents

1 Description of Message Format	2
1.1 Required Commands	3
1.1.1 Open Command	3
1.1.2 New-Cohort Command	3
1.1.3 Delete-Cohort Command	3
1.1.4 Exit command	4
1.2 Commands added for Testing/Integration	4
1.2.1 Close-Bank Command	4
1.2.2 Listen-for-cohort	4
1.2.3 Listen	4
1.2.4 Print-cohort	4
1.2.5 Test	5
2 Time-Space Diagram	6
3 Data Structures and algorithms/ Design Implementations	6
3.1 Data Structures Used For Bank	6
3.1.1 Customer Data Structure	7
3.1.2 Cohort Data Structure	7
3.2 Data Structure Used For Customer	7
3.2.1 Cohorts Data Structure	7
3.3 Helper Methods	7
4 Screenshot of Commits	8
4.1 Migration From Local to general.asu.edu	8
5 Link to Video Demo	8
5.1 Video Timestamps	8

1 Description of Message Format

The way I set up my Customer Servers was to create a user application that could be started after the initial commands from the command line were set to set up the Banks Information. In order to run the program you will need to use the following format when running either the customer or bank applications.

```
java 'Customer/Bank' 'Bank IP' 'Bank Port'
```

Once this command is run you can input the following commands using stdin.

1.1 Required Commands

1.1.1 Open Command

The open command is as required it takes in a command formatted as follows

open 'customer' 'balance' 'customer ip' 'bank port' 'customer port'

The parameters are then parsed as a string and sent to the bank as a message. The bank will take this message and parse it and add a customer via a HashMap which is the Data Structure I have chosen to store all Customer Info. If the Customer info is stored correctly then the Bank sends a 'SUCCESS' message back to the customer and the customer will go back to waiting for a command.

1.1.2 New-Cohort Command

The new-cohort command is implemented as required and takes in a command that is formatted as follows:

new-cohort 'customer name' 'size of cohort'

The new cohort command will parse the message and issue a new cohort message to the bank. When the bank receives the command it will create a new cohort with the issuing customer and a random number of customer for the remaining number of customers available up to the specified size.

Once the cohort is created by the Bank the bank will send the Cohort data Structure as an arraylist. The arraylist is generated from the cohorts data structure inside the bank which is a hashmap that has an arraylist of a cohorts as the value.

Once the customer receives this list it will parse the list and create a new temporary list which does not contain itself and will send the array list to the other customers that are included in the cohort.

1.1.3 Delete-Cohort Command

The delete cohort command is implemented as required and takes in a command formatted as follows:

delete-cohort 'customer name'

The delete-cohort command parses the message and issues the delete-cohort command to the Bank. When the bank receives this command It will first issue a notification for each customer within the cohort so that they can each delete their cohort database. Once every customer has issued a 'SUCCESS' notification to the Bank then the bank will delete the cohort from its database.

1.1.4 Exit command

The exit command is implemented as required and takes a command formatted as follows:

exit 'Customer name'

The exit command is parsed by the customer and sent to the bank. When the Bank receives the command it will delete the customer from its database and return a 'SUCCESS' message back to the customer. When the customer receives this response it will then close the application.

1.2 Commands added for Testing/Integration

1.2.1 Close-Bank Command

This command is a non require command and is used for testing or for closing the bank from a customer process. The command is formatted as follows:

close-bank

This command must be issued from a customer, once issued it will close the Bank server.

1.2.2 Listen-for-cohort

This command is added so that cohorts who are not issuing a new-cohort command can receive messages from their peers for their cohort information. The command is formatted as follows:

listen-for-cohort

The command is issued by all non new-cohort issuing customers. Since the cohort selection is random all customers will need to use this command to receive their cohort information.

1.2.3 Listen

The listen command is used for customers who are awaiting information from the bank for a delete-cohort message. This command is formatted as follows:

listen

This command will listen for the bank to issue a delete-cohort command which the customer will then process and give the bank the result of the operation.

1.2.4 Print-cohort

This command is used for testing and is formatted as follows:

print-cohort

This command can be issued by a customer to view the contents of its current cohort information. It is used for testing to verify whether a cohort was actually deleted or not.

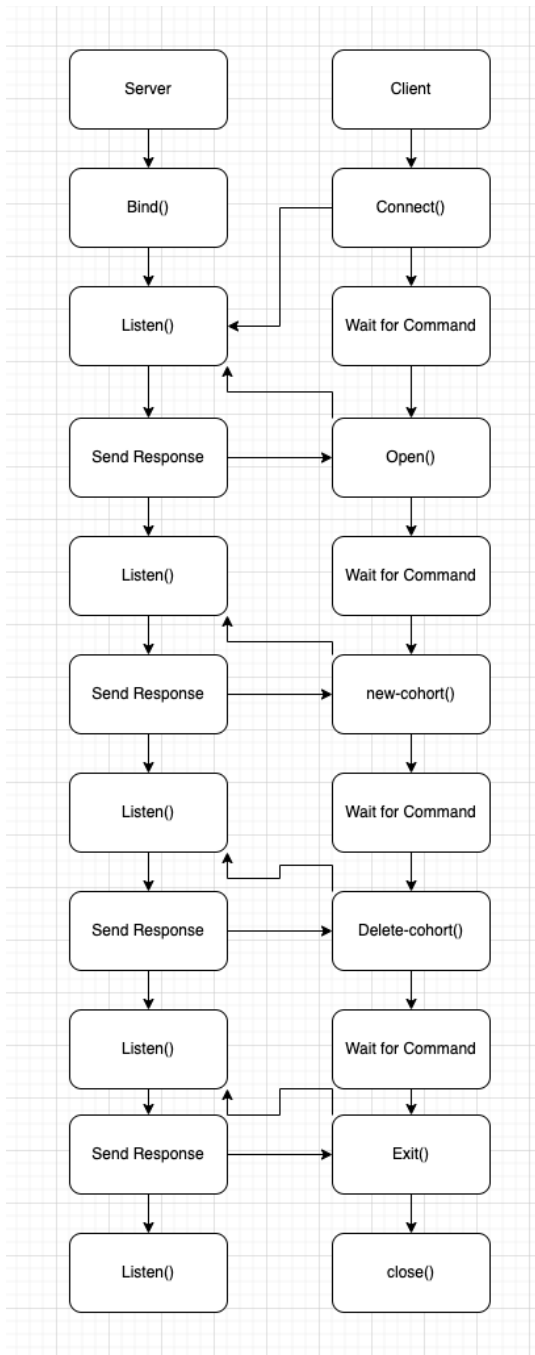
1.2.5 Test

This command is used for testing and is formatted as follows:

Test

This command is issued by a customer and is used for the purpose of printing out all of the data structure in the bank to verify whether a customer/cohort was added/deleted.

2 Time-Space Diagram



3 Data Structures and algorithms/ Design Implementations

3.1 Data Structures Used For Bank

3.1.1 Customer Data Structure

The customer Data Structure I used for the Bank is a HashMap Data Structure. The Hashmap contains a key as the name of the customer and value as another Hashmap that contains all of the customers information.

I chose this format because it makes it easier to be able to move data around and format data as I need for all the different functionality needed for the assignment. Having a hashmap with a hashmap was helpful because I could access all of the data I needed with only the customers name.

3.1.2 Cohort Data Structure

The Cohort Data Structure I used was also a hashmap but for this map I used a different implementation than the customers data structure. For this structure I still used the customers name, but this time it is only the issuing customer that is used as the key of the cohort structure. And then for the value I have an ArrayList the contains a hashmap of all of the list members information.

I chose this format because it still allows me to use hashmap functionality for the bank. Then when I need to access a specific cohort I could look through all of the customers listed and use another method to initiate an ArrayList that I can send as a packet to a customer.

3.2 Data Structure Used For Customer

3.2.1 Cohorts Data Structure

For the customer data structures I only used an ArrayList of hashmaps to store the customers cohort information. I chose this implementation because it is how I designed the cohorts in the bank. It also allows for a more compact list of customer information that I could loop through and be able to break off sections when needed to remove a customer so that the customer can send messages to the rest of the cohort without needing to include itself.

3.3 Helper Methods

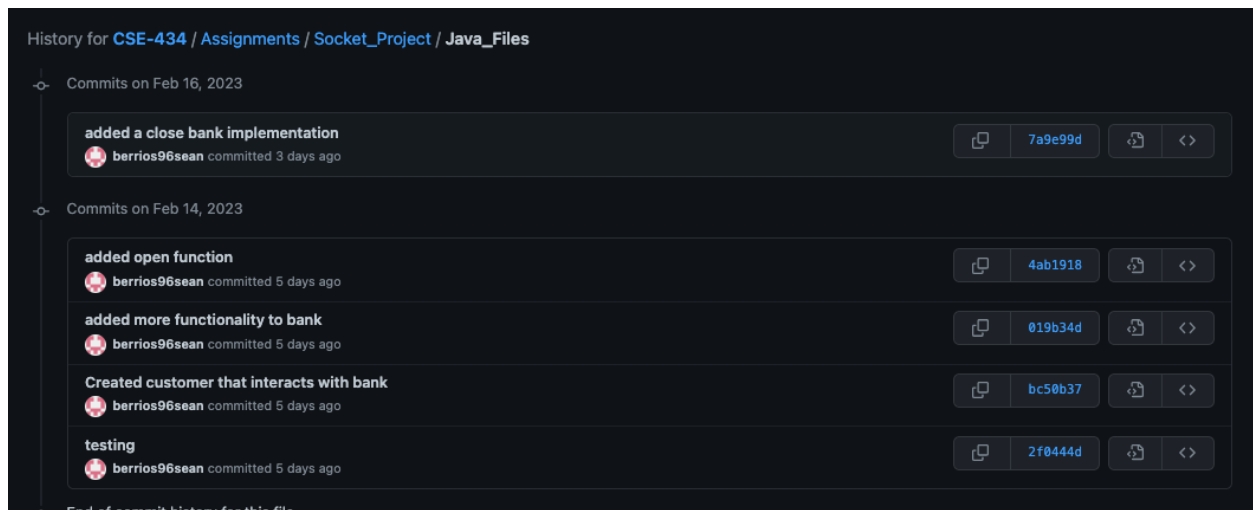
I used a large variety of helper methods which are documented within my code and would be too large to include here.

4 Screenshot of Commits

4.1 Migration From Local to general.asu.edu

All commits in screenshot are prior to my migration from local to the general.asu.edu server. As a result only the initial connections I was able to make on my local server are included in the commits.

Due to having to need a token to clone my repository on the general.asu.edu server I chose to not use my github repository for the implementation on the general server.



5 Link to Video Demo

The link to the youtube video was still processing at time of submission. The shareable link that youtube provided is what is included. I listed this video as public on my personal channel which I will also provide a link for in the event that the shareable link is not accessible.

Personal Channel Link: <https://www.youtube.com/@seanberrios7827/featured>

Demonstration Video Link: <https://youtu.be/ixCxLqumfCM>

5.1 Video Timestamps

00:00 - 00:50 Explanation of General Server Set Up
00:50 - 01:12 Compile Bank/Customer applications
01:12 - 01:30 Start Bank Server
01:30 - 03:45 Open a customer and delete with exit method
03:45 - 06:24 Open a customer on Three distinct end Hosts
06:24 - 08:30 New-cohort Demonstation

08:30 - 10:05 Delete-cohort Demonstation

10:05 - 11:12 Extra Commands/ Exit commands