Team #6: Saul Gonzalez, Gian Hernandez, Bruno Valencia
December 1, 2024
CS 3331 – Advanced Object-Oriented Programming – Fall 2024
Instructor: Dr. Bhanukiran Gurijala
Project Part 2 Lab Report

This work was done individually/as a team and completely on my/our own. I/we did not share, reproduce, or alter any part of this assignment for any purpose. I/we did not share code, upload this assignment online in any form, or view/received/modified code written from anyone else. All deliverables were produced entirely on my/our own. This assignment is part of an academic course at The University of Texas at El Paso and a grade will be assigned for the work I/we produced.

1. Program Explanation

For part 3 of the assignment, we were tasked with revising/refactoring our code and making sure everything from the previous parts were complete. We started with finishing a couple of functions that were missing from part 2, then we moved on to implementing the one function included in part 3, which was a basic login prompt for the user. From here we focused on refactoring, which involved making sure the code was written in a more efficient way, making sure it followed the design principles, and doing various tests for the new functionality.

2. What did I learn?

In part 3 we gained a deeper understanding of how refactoring and design patterns work. We also gained experience making a state diagram for an actual program. One problem we kept running into with our program involved the Accounts csv file, in which the first row of data was getting erased randomly, for no apparent reason. This was one way our solution could have been improved, but unfortunately, we weren't able to solve the problem and we ran out of time. This final part of the assignment was the easiest so far, and we were able to put our ideas together in about a week.

3. Solution Design

To be more specific, in this part of the program we implemented the login functionality for the user, where the menu asks the user for a password to log in to their account. We implemented a second design pattern (Singleton design pattern) for our logging function. We added a few more jUnit tests for further testing of the code, and we revised our program to minimize any violations of the design principles. Finally, we added any documentation that was missing for the code.

4. Testing

This time around we used a combination of manual and automatic jUnit testing. Some functionality was tested more than others. One that was tested extensively was the User Transactions file reader functionality, as it had a lot of moving parts. This did involve breaking

the program, which helped to get it working smoothly for the most part. We used jUnit testing for the smaller things. This time we added tests for our Customer class, which has various getter and setter-like functions that we wanted to ensure worked properly, as it is one of the crucial parts of our program.

5. Test results

Testing the various functions of the BankManager class including: Reading the transactions file, creating a new user, and generating a bank statement for a specific user.

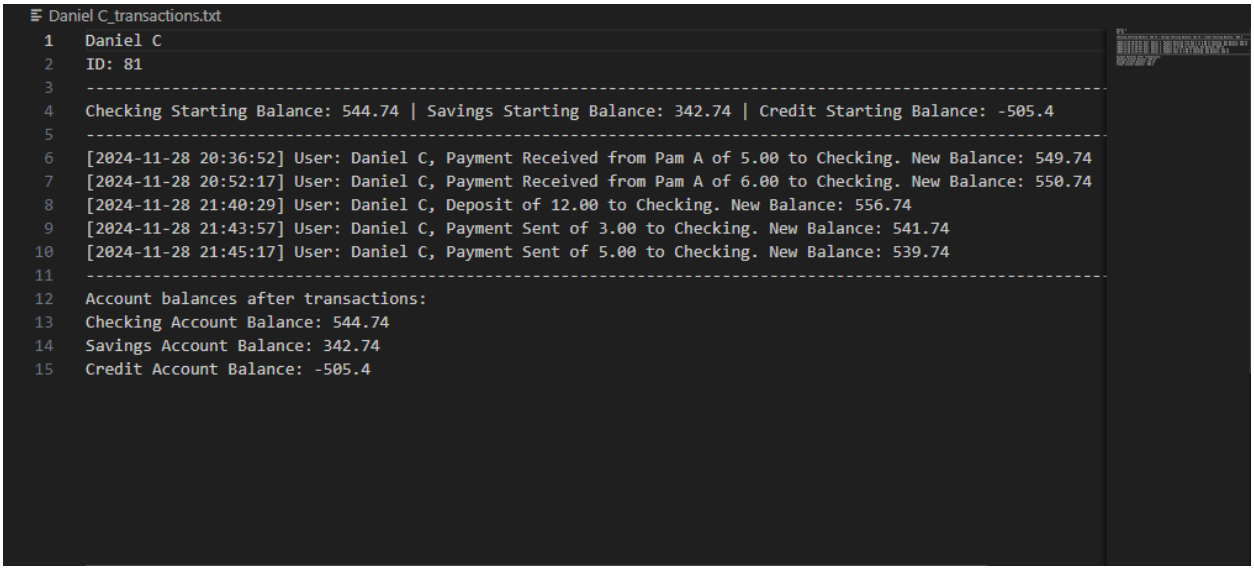
```
Welcome, Bank Manager. What would you like to do?
1. View All Transactions
2. Process Transactions File
3. Create a New User
4. Generate a Bank Statement
5. Exit
Choose an option: 2
Transaction #3 failed: Invalid action
Transaction #4 failed: Invalid action
Transaction #7 failed: Invalid action
Transaction #86 failed: Invalid action
Transactions complete.

Welcome, Bank Manager. What would you like to do?
1. View All Transactions
2. Process Transactions File
3. Create a New User
4. Generate a Bank Statement
5. Exit
Choose an option: 3
Enter First Name: Gian
Enter Last Name: Hernandez
Enter Date of Birth (DD-MM-YYYY): 29-11-2002
Enter Street Address: 1213 Utep Ave
Enter City: El Paso
Enter State: TX
Enter Zip Code: 79901
Enter Phone Number: 9152586351
Enter Credit Score: 700
User created successfully with User ID: 1008

Welcome, Bank Manager. What would you like to do?
1. View All Transactions
2. Process Transactions File
3. Create a New User
4. Generate a Bank Statement
5. Exit
```

- Generated Bank Statement txt file:

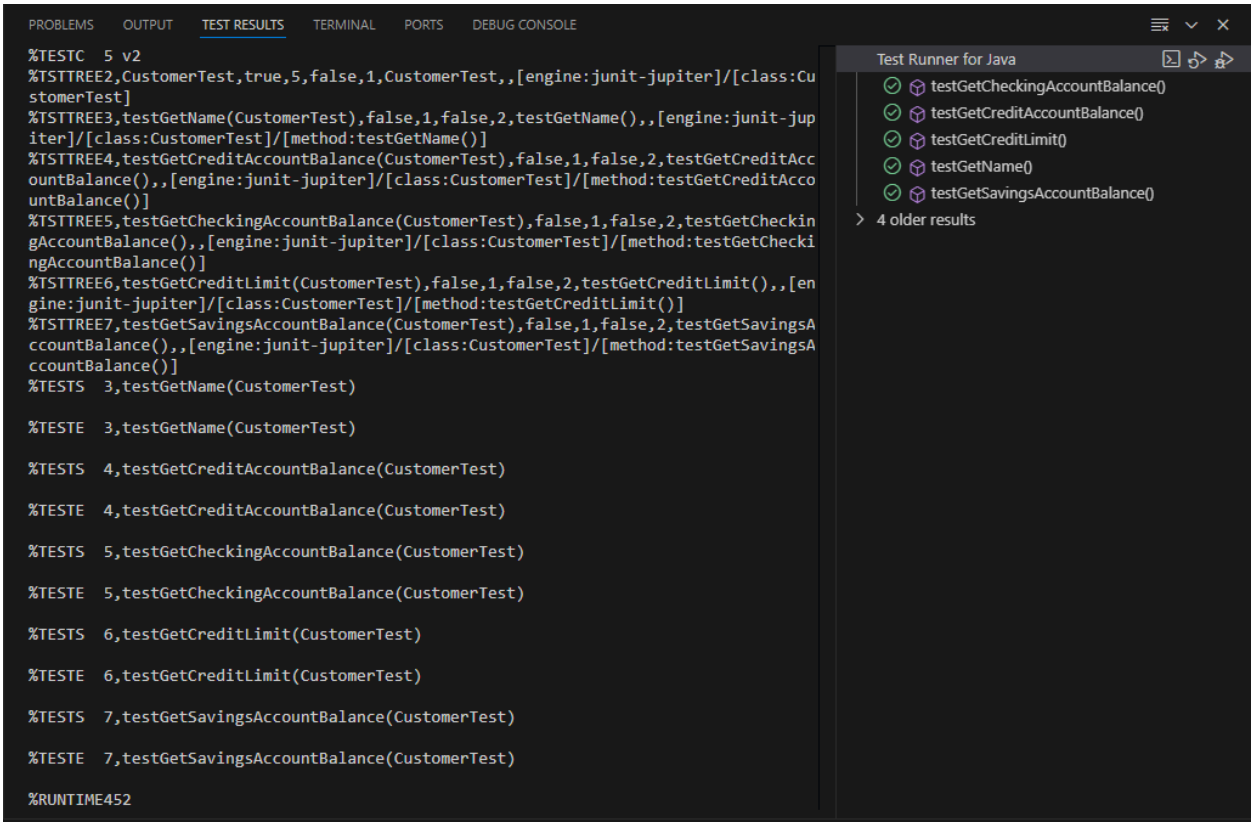
Daniel_C_BankStatement.txt

```
Bank Statement
======================
Name: Daniel C
Address: 500 W. University Ave, El Paso, TX 79968
Phone Number: (915) 747-5044

--- Account Details ---
Checking Balance: 544.74
Savings Balance: 342.74
Credit Balance: -505.4
Credit Limit: 9400

--- Transactions ---
No transactions available.
```

- Generated User Transactions file:



JUnit test results for the CustomerTest class:



6. Code Review

As mentioned before, refactoring was a big part of this assignment, so we put a lot of time into it. Before this part of the program, our code was very lengthy and was violating some of the basic design principles. After gaining a deeper understanding of these principles (and taking some good advice from our TA) we had a lot of revising to do. Now our code is much more concise and efficient and is not violating any of the design principles (to our knowledge). Our program is not perfect, but we are confident that it is the best version so far, and we are proud of what we were able to put together.