

Project 0 : Banking App (Pt 1)

Due: Nov 9th 10am

Sandra O

sandrao@yonderwire.com

Abstract—This Bank app is a console-based application that simulates banking operations.

TABLE OF CONTENTS

1. [Purpose](#)
2. [Description](#)
3. [Requirements](#)

1 PURPOSE

We want to see that you can meet deadlines and that you can code. You are expected to complete the following requirements and give a *5-minute presentation or demo of your project*.

2 DESCRIPTION

1. **A CUSTOMER can:**
 - (a) Apply for an account.
 - (b) View their balance.
 - (c) Make Withdrawals.
 - (d) Make Deposits.
2. **An EMPLOYEE can:**
 - (a) Approve accounts
 - (b) Deny accounts
 - (c) View account balances for their customers.

3 REQUIREMENTS

3.1 PART I: Customers

1. Build the application as a [Maven Project](#) using [Java 8](#).
2. All interaction with the user should be done through the console using the `Scanner` class.

3. Customers of the bank should be able to Register with a username and password.
4. Customers should be able to Apply:
 - (a) For joint accounts.
 - (b) To open an account.
5. Once the account is open, Customers should be able to:
 - (a) Withdraw
 - (b) Deposit
 - (c) Transfer funds between accounts
6. All basic validation should be done, such as:
 - (a) Trying to input negative amounts
 - (b) Overdrawing from accounts
 - (c) etc.

3.2 PART II: Employees

1. Employees of the bank should be able to view all of their customers information, this includes Account:
 - (a) Information
 - (b) Balances
 - (c) Personal information
2. Employees should be able to approve or deny open applications for accounts.
3. Bank admins should be able to view and edit all accounts, this includes:
 - (a) Approving
 - (b) Denying
 - (c) Withdrawing
 - (d) Depositing
 - (e) Transferring from all accounts
 - (f) Canceling accounts
4. All information should be *persisted* using text files, and *serialization* via Object Input/Output Stream.
5. 100% test coverage is expected using J-Unit.
6. You should be using [Test Driven Development\(TDD\)](#): Writing test first, then writing code to pass that test.
7. Logging should be accomplished using [Log4J](#).
8. All transactions should be logged.

4 REFERENCES

(Oracle, 2019, Project, 2002–2020, Team, 2020, Wikipedia, 2005-2020, Wikipedia, 2005-2020)

- [1] Oracle (2019). “Java SE Development Kit 8 Downloads”. In: Oracle. URL: <https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html>.
- [2] Project, Maven Apache (2002–2020). “Introduction to the Standard Directory Layout”. In: The Apache Software Foundation. URL: <http://maven.apache.org/guides/introduction/introduction-to-the-standard-directory-layout.html>.
- [3] Team, JUnit (2020). “https://junit.org/junit5/”. In: The JUnit Team. URL: <https://junit.org/junit5/>.
- [4] Wikipedia (2005-2020a). “Log4j”. In: en.wikipedia.org. URL: <https://en.wikipedia.org/wiki/Log4j>.
- [5] Wikipedia (2005-2020b). “Test-driven development”. In: en.wikipedia.org. URL: https://en.wikipedia.org/wiki/Test-driven_development.