Ryan Pepe

35/35

Professor Arias

Software Development I

26 April 2019

Project Milestone

Abstract

Currently, I have implemented each of the classes that will be required for a successful banking system. These classes include: “AccountHolder,” “Account,” “CheckingAccount,” “SavingsAccount,” and “Transaction.” There is a hierarchy between the account holder and the abstract accounts. In addition to implementing the classes, I have also downloaded the necessary libraries to begin making queries through Hibernate. I have no previous experience with neither databases, nor Hibernate, so I watched an entire Youtube tutorial for getting started with Hibernate. Lastly, I created the SQL database using MySql Workbench, and added a new user called “hbuser” who is able to access everything through the database.

Introduction

The reason that I chose to create a digital banking application is because I knew that it would challenge my abilities as a novice programmer. I did not have any experience in databasing, so I knew that I would have to teach myself a tremendous amount of information in order to implement the features that I wished to be implemented. In short, I wanted to create an application that would allow a user to create as many checking/savings accounts as they wish and to make various transactions with each account — each aspect of the process being

translated directly into a database. Lastly, I would like the user to be able to retrieve transaction information from the account.

Detailed System Description

As mentioned previously, the digital banking system will allow a user to perform the actions that anyone could perform by going to a bank — create an account, withdraw funds, deposit funds, transfer funds, and cancel an account. The Account class is an abstract class because I do not want people to be able to create generic bank accounts. One must choose to create either a checking account or savings account — both of which are subclasses of the Account superclass. In addition, the Account class extends the AccountHolder class because I wanted each account to contain the fields of its account holder. Include the UML diagrams

---

Requirements

This project is addressing the problem of creating bank accounts digitally and making transactions through a desktop application. In addition, it is addressing the problem of keeping track of an influx of information that is being thrown through the system. Each piece of data will be stored into a database in order for the bank administrators to monitor account activity and see user information.

Literature Survey

Chase bank has created applications to transfer funds between accounts and view account activity; however, they do not allow one to create accounts and delete accounts. I seek to allow

the user to create accounts remotely and generate a unique account number that will be stored in the database. Many modern banks are trying to implement similar features as the world begins to become more digitized.

User Manual

The bank application will have a visually appealing GUI which will make the process of performing bank actions fairly easy. There will be a tab on the desktop application for each of the functions of the program: Add User, Create Account, Remove Account, Make a Transaction, View Records.

Conclusion

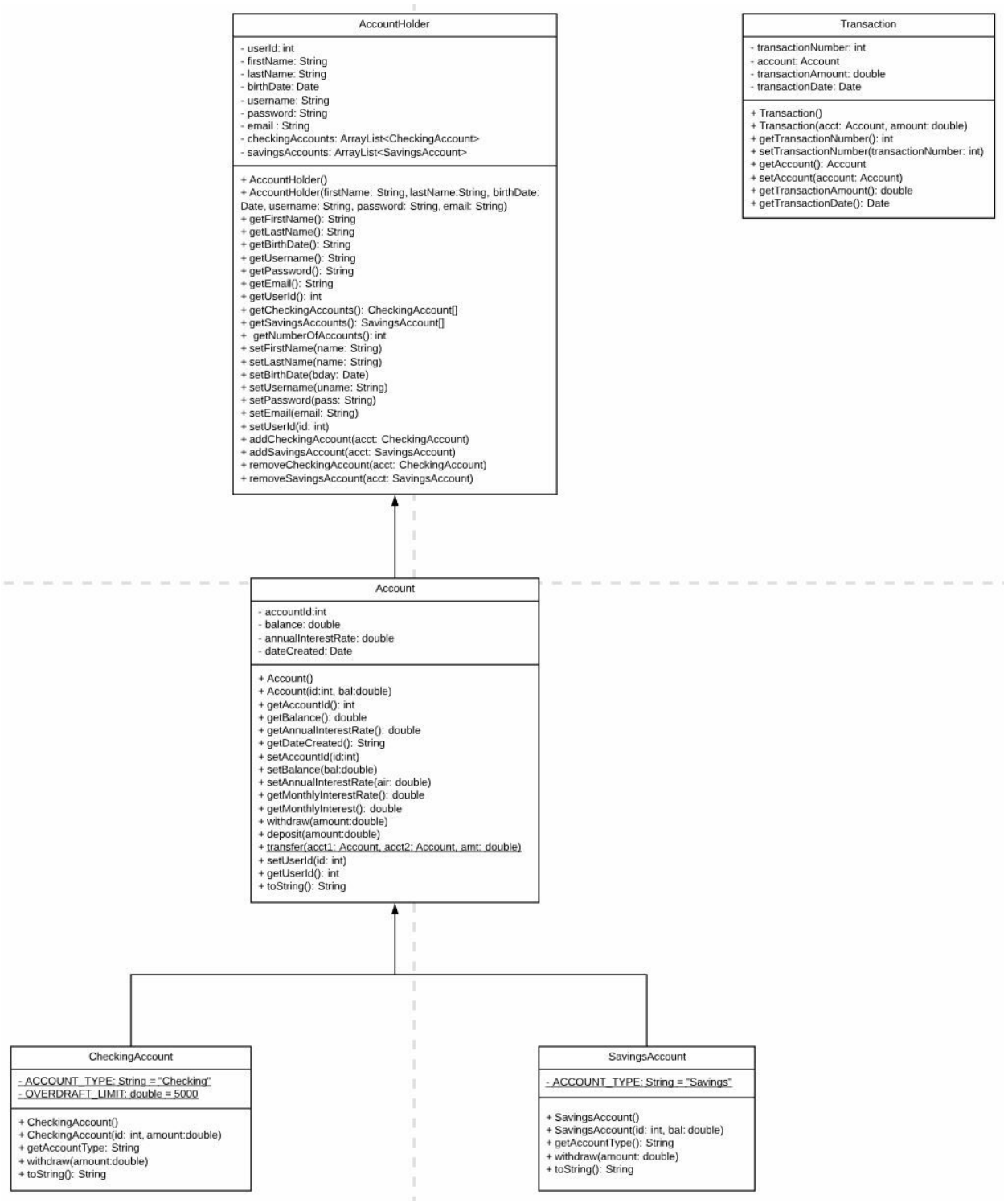
The system simply seeks to make an easier banking experience for the user. Many banks require one to physically go to the bank to do many bank procedures; however, this application will make life much easier for them. If I am successful with creating this program, it will be a testament to the amount of work that I put in to learning the necessary skills to complete the project. I sought to go above and beyond my current technical abilities in this program, so hopefully my hard work will pay off.

References/Bibliography

1. **Luv2Code Hibernate Tutorials**
    a. **https://www.youtube.com/playlist?list=PLEAQNNR8IlB7fNkRsUgzrR346i-UqE5CG**

2. **MySQL Workbench Manual**
    a. **https://dev.mysql.com/doc/workbench/en/**
3. **Hibernate ORM Final User Guide**
    a. **https://docs.jboss.org/hibernate/orm/5.2/userguide/html_single/Hibernate_User_Guide.html**

UML CLASS DIAGRAM



UML ER DIAGRAM