March 18, 2016

Achal Parikh, Stweart Gibson, Andrew Lau

Team 4

Phase 4: Back-End Design Document

SoftwareQuality

# Introduction

This document is design to be a reference for any individual who wishes to understand or implement the structure of the back-end of the bank transaction system created by team 4. This document describes the overall structure of the system as well as provides a structural class diagram to show the relationships between each class. Additionally, this document contains a description of each method and class and the specific purpose they serve within the system. The document also includes a sample of input and output files of one system iteration.

# Overall Structure

## UML class diagram:

## Description:

The back-end of the system consists of three classes:

The first class is called the backend.java, this class takes in three different files as argument and outputs the new master bank account file as well as the new current bank account file.

The second class is called backFiles.java, this class is used by the backend.java to read the files and load the accounts and the transaction into a vector and a queue respectively.

The third class is called backData.java, /\*………\*/

# Class method and intentions:

|  |  |  |
| --- | --- | --- |
| Type | Name | Description |
| **Class** | backEnd | The class is holds the main method for the back-end of the banking system and is responsible for output of new master bank account file and new current accounts file` |
| Method | Main | The main method is responsible for creating the objects for backFiles and backData which will carry out the computation of new files as well as checking that all files have valid |

|  |  |  |
| --- | --- | --- |
| Type | Name | Description |
| **Class** | backFiles | The class is responsible for taking in the input and generating the output for the system |
| Constructor | backFiles | The constructor of this class contains three strings for transaction file, master accounts file, and current accounts file |
| Method | loadAccounts | This method takes in the master accounts files and loads the accounts into a Vector |
| Method | loadTransactions | This method takes in the transaction file and loads them into a queue |

|  |  |  |
| --- | --- | --- |
| Type | Name | Description |
| **Class** | backData |  |

# Input and output Files

## Current bank account file:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Account number | Name | Active/disabled | Balance | Plan |
| 00001 | Achal Parikh | A | 01001.01 | Student |
| 00002 | Andrew Law | A | 20000.00 | Student |
| 00003 | Jack | A | 30000.00 | Non-student |
| 00004 | Bruce Banner | A | 40000.00 | Non-student |
| 00005 | Hawk eye | A | 50000.00 | Non-student |
| 00006 | Black Widow | A | 00001.00 | Student |
| 00007 | Steve Rogers | D | 05000.00 | Non-student |
| 00008 | Tony Stark | A | 99999.99 | Non-student |
| 00009 | Thor Odinson | A | 00500.00 | Non-student |
| 00010 | Student | A | 00500.00 | Student |
| 00011 | John Doe | A | 00500.05 | Non-student |
| 00012 | John Doe | D | 00500.05 | Non-student |
| 00000 | End of file | A | 00000.00 | Non-student |

## Transactions Files:

Each transaction is made up of three rows where:

The first row indicates a successful login.

The second row consists of data regarding the transaction.

The third row indicates a successful logout.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 10 | | | | |
| 01 | BLACK WIDOW | 00006 | 00025.00 | S |
| 00 | | | | |
| 10 | | | | |
| 03 | ACHAL PARIKH | 00001 | 020000.00 | A |
| 00 | | | | |
| 10 | | | | |
| 02 | STUDENT | 000010 | 00500.00 | A |
| 00 | | | | |

## Sample master bank accounts file:

|  |
| --- |
| 00001 ACHAL PARIKH 01001.00 0001 |
| 00006 BLACK WIDOW 00001.00 0001 |
| 00010 STUDENT 00500.00 0001 |