**Software Requirements Specification for**

**Brian’s Big Bank**

**Version 1.0**

Prepared by Brian Howard

DePauw University

**1 Introduction**

**1.1 Purpose**

The Brian’s Big Bank (BBB) system will allow the simulation of a bank with a variety of kinds of accounts and ways of moving money around. The user will interact with BBB in the role of one or more of the bank’s customers.

**1.2 Intended Audience and Reading Suggestions**

This document was developed both as a guideline for student developers, and also as a reference for potential simulated bank customers.

**1.3 Project Scope**

The goal of BBB is to support the simulation of a traditional bank with checking and savings accounts. Interaction will be through a series of menus displayed on the console. The implementation should be flexible enough that new kinds of accounts and transactions could be added with relatively little programming effort.

**2 Overall Description**

**2.1 Product Perspective**

This software product is being developed by students of the Object-Oriented Software Development course at DePauw University and is intended for use by anyone wanting to simulate a bank. The goal of this project is to develop a feature-rich application which will serve as a functioning prototype for a more comprehensive application which could be developed by extending the codebase.

**2.2 Product Features**

The main features of this product are:

• managing the menu-oriented interface to the simulation

• maintaining the state of a number of customer accounts of varying types as transactions are performed

• providing a flexible design to support expansion or revision of the simulation

**2.3 User Classes and Characteristics**

The users will include those who simply wish to use the simulation, as well as those (the developers) who are designing and testing the simulation or extensions to the simulation.

**2.4 Operating Environment**

This application is designed to work with a Java Virtual Machine in a desktop environment. Users of this application are expected to be running either a Windows, MacOS, or Linux desktop operating system.

**2.5 Design and Implementation Constraints**

This application may use the filesystem as a means of saving bank state and simulation history.

**2.6 User Documentation**

A programmer’s guide to working with the software components developed as part of this application will be provided for those who wish to continue development on this application. Additionally, a user’s guide will be provided for those who wish to use this application to simulate bank transactions. Within the simulation, a limited amount of help information will also be available.

**3 System Features**

The following features, with their associated requirements, will be implemented in the final revision of this software system:

**3.1 Menu Interface**

**3.1.1 Description and Priority**

A user can interact with the simulation through a series of menus.

**3.1.2 Functional Requirements**

**REQ-1:** There will be a main menu from which the user can select common operations or navigate to more specialized menus.

**REQ-2**: The user will interact with the menu by typing numbers into the console.

**REQ-3:** Any operations that need additional information will also communicate with the user through the console.

**REQ-4:** It should be easy to navigate through the menu system, and to get help from the system in doing so.

**3.2 Account Types**

**3.2.1 Description and Priority**

The system provides several types of accounts, including checking and savings. It should be easy to add additional account types with different characteristics.

**3.2.2 Functional Requirements**

**REQ-1:** TODO

**3.3 Simulation State**

**3.3.1 Description and Priority**

The simulation keeps track of all of the (simulated) money in the accounts in such a way that no money is lost or created.

**3.3.2 Functional Requirements**

**REQ-1:** TODO

**3.4 Transaction History**

**3.4.1 Description and Priority**

The system keeps track of the history of transactions, so that reports could be generated to simulate various accounting requirements.

**3.4.2 Functional Requirements**

**REQ-1:** TODO

**4 External Interface Requirements**

**4.1 Hardware Interfaces**

The software will run on a desktop or laptop and no additional hardware is needed.

**4.2 Software Interfaces**

The simulation does not need to interface with any software other than the Java platform.

**5 Other Nonfunctional Requirements**

**5.1 Performance Requirements**

Management of the menu interface and simulation state must consume minimal system resources so as to be accessible in real-time by users of the system. This application is intended to be used interactively, so users should not be expected to wait for the completion of any of the operations provided by the application.

**5.2 Security Requirements**

The application will not require any sensitive information from the user. It will rely on existing user-based security on the host operating system to keep saved simulation states private.

**5.3 Software Quality Attributes**

This application will ship with a suite of tests which insure its proper function, even if third-party updates to the source-code are integrated. Additionally, at run time, this application will verify the correctness of any data files it uses, or of any input provided by the user, and issue appropriate error messages in the cases of unexpected or erroneous input.