# Software Requirements Specification

## 

## Introduction

### Purpose

This is being built to support a bank’s online system through the creation of a demand deposit accounting system. It will help automate the process of handling client’s accounts and ATM transactions.

### Intended Audience

The intended audience will be the IT department of the bank. They will need to be able to manage the system especially if any failures occur. The IT team will need to have enough knowledge of the system to provide maintenance.

The system will also be used by bank tellers and sales representatives. Bank tellers will need to be able to access the system to deposit and withdraw money for clients of the bank, while the sales representatives will use the system to sign new clients up with the proper banking account.

### Intended Use

The project will produce a product that covers the basic user needs when using the bank. If functional after evaluated, a final complete product will be produced.

### Scope

The product will be able to support checks, ATM transactions, electronic credits and debits, and various types of checking accounts. It will also handle overdrafts, produce statements and accrue interest. All of these functions will create a system that will make it easier to manage customer accounts. By automating the process expected actions will be able to happen faster and will make the customers experience better.

### Definitions and Acronyms

### Demand Deposit Accounting (DDA) - banking in which customers can deposit and remove money from their accounts

Success will mean, able to handle different client accounts, credits and debits, and produce statements for clients accounts.

Risk will mean the possibility of a faulty system, such as one that is easily hacked or needs a lot of maintenance to run fully functional.

## Overall Description

### User Needs

Allow users to use different types of checking accounts.

Support electronic credits, debits, checks and ATM transactions.

Produce statements for user accounts, handle statements and accrue interest.

### Assumptions and Dependencies

Knowledge of computer programming and relational databases will greatly impact this project. This knowledge needs to be complete in order to create the system. This will be dependent on factors such as support from the team and planning.

User accounts will be protected.

## 

## System Features and Requirements

### Functional Requirements

A relational database that can handle customer information, and access to the database.

A program that can handle removing money from an account.

A program that can handle adding money to an account.

A program that produces statements for an account, and also calculates interest and handles overdrafts.

### 

### Interface Requirements

This system will have an interface in the command line. This may not be user friendly but the processes being completed will not necessitate a web-based interface. The interface will also need to be able to efficiently access the database so a command line interface can help support this.

### System Features

Uses a relational database to manage customer accounts. This will allow information to be stored about the customer, and provide an easy system to handle and manage the individual funds within each customer’s account. Requires java language.

### Non-functional Requirements

The system will need to have some security requirements so that customer’s accounts are safe. This may require maintenance and updates to the system. Overall, the security system will need to ensure that outside users cannot access or make changes to the customer's accounts.

Performance will improve the customer experience. Whether they are interacting at a desk with a teller or through an online platform or ATM, the system should not have a long lag time or create a wait.