# Software Requirements Specification

## Introduction

### Purpose

State why this product is being built. What need does it fulfill?

This product is used as a demand deposit accounting (DDA) system that allows bank customers to readily access their money. It allows them to deposit checks, use the ATM, and use electronic credits (both credits & debits). Additionally, the system supports multiple types of checking accounts depending on what is necessary for the customer. Apart from customer interactions, this product handles overdrafts, accrues interest, and produces statements.

### Intended Audience

State who in your organization should use this document. This typically includes software engineers, test engineers and project managers, but may also include stakeholders in other organizations, such as leadership teams, sales, marketing and data center operations.

I don't know? I suppose that if this were actually for a company, software engineers, test engineers, and project mangers would need to know about this. In addition, the people at the bank would need to know so that they can help implement the system as well as aid customers once they were able to use it.

### Intended Use

State how the audience should interpret and use what is outlined herein. For example: Build a prototype that addresses 50% of the users’ needs so it can be evaluated for suitability before building the final, fully functional deliverable.

The intended use of this product is to let banking customers access their accounts and perform any necessary operations with their money. A prototype could potentially handle one kind of account and show basic deposits and withdrawls so that the core idea of the project is there.

### Scope

Describe the software product to be built; include how it addresses expected benefits, objectives, and goals. This should relate to overall business goals, especially if teams outside of development will have access to this SRS.

This software will have customers' bank account information and will keep tracking of money moving from account to account. There will need to be a database that stores all the information as well as a separate piece of software that keeps track of things like overdrafts. There are a lot of different aspects to this product, as outlined earlier, and each piece needs to be able to work with the other.

### Definitions and Acronyms

Explain any commonly used acronyms and define any relevant terms. Define what *success* means (looks like), as well as what *risk* means.

DDA (Demand Deposit Accounting) -- accounts that are kept in commercial banks in which money can be deposited or withdrawn from when necessary

## Overall Description

### User Needs

Define the users’ expectations and needs addressed by this software product.

The user expects to be able to easily & simply access the funds they store in their checking accounts. This product allows them to add or remove funds whether that be through an ATM or credit/debit system.

### Assumptions and Dependencies

Explains factors that impact your ability to fulfill the requirements outlined herein.

Note any assumptions being made that may have a negative impact on this deliverable

Finally, you should note if your project is dependent on any external factors. This might include software components you are reusing from another project.

Dependant on the Internet.

## 

## System Features and Requirements

### Functional Requirements

Describe what is absolutely essential for the product to function.

The system relies on the internal banking system, so as long as that is functioning, this product will as well. If there are any external transactions, those systems will need to be utilized, but this product will still work even if they are down.

### Interface Requirements

Describe how the product interfaces with the user or other components and systems. For example: A web-based interface supports individual user interaction; an RPC layer supports intranet system requests; a web service XML API is used to fulfill external intersystem requests.

This product uses a web-based interface so that individual customers can readily access their own accounts with the use of a computer or mobile device.

### System Features

Describe the major features of the product (e.g. uses relational database, hive, logger, journal, distributed architecture, AI); Use as much detail as necessary to explain each feature and supplement with graphics where appropriate.

This product will use a database system to keep record of an individual customer's financial records.

### Non-functional Requirements

Describe the less tangible but still important requirements, such as performance and security.

This product needs to be incredibly secure because it is dealing with people's banking accounts and the money within them. There are not really any room for mistakes and vulnerabilities in such a product. It also needs to be able to perform quickly so customers do not get frustrated with the system, but security is a far more prevalent concern.