$implicity: A Simple Financial Tracking Software

Software Requirements Specification

1.0

24 November 2014

Engineers:

Michael Esry

Bill Kolega

Nicholas Roudebush

Tyler Steiner

Prepared for

KU EECS 448—Software Engineering I

Instructor: Swapan Chakrabarti, Ph.D.

Fall 2014

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Description** | **Author** | **Comments** |
| 24 Nov 14 | Preliminary Version | Tyler Steiner | This is a working requirements to be submitted for approval. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

|  |  |  |  |
| --- | --- | --- | --- |
| **Signature** | **Printed Name** | **Title** | **Date** |
|  | Michael Esry | Engineer |  |
|  | Bill Kolega | Engineer |  |
|  | Nicholas Roudebush | Engineer |  |
|  | Tyler Steiner | Engineer |  |

**Table of Contents**

Revision History iii

Document Approval iii

1. Introduction 1

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience 1

1.4 Additional Information 1

1.4.1 Definitions, Acronyms, and Abbreviations 1

1.5 Contact Information/SRS Team Members 1

1.6 References 2

2. Overall Description 2

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 2

2.5 User Environment 2

2.6 Design/Implementation Constraints 2

2.7 Assumptions and Dependencies 2

3. External Interface Requirements 3

3.1 User Interfaces 3

3.2 Hardware Interfaces 3

3.3 Software Interfaces 3

4. System Features 3

4.1 Description and Priority 3

4.2 Action/Result 3

4.3 Functional Requirements 3

5. Other Nonfunctional Requirements 4

5.1 Performance Requirements 4

5.2 Software Quality Attributes 4

5.3 Project Documentation 4

5.4 User Documentation 4

# 1. Introduction

## 1.1 Purpose

This Software Requirements Specification (SRS) specifies the requirements of the “$implicity: A simple financial tracking software” software. This software will provide for tracking financial transactions through a web portal. It will allow the user to save their accounting information from session to session. This document will be useful for the clients in order to ensure that all specifications and requirements are completed as mentioned. This document will be useful for the engineers in designing the system.

## 1.2 Document Conventions

*Main Section Titles*

* Font: Times
* Face: Bold
* Size: 16

*Sub Section Titles*

* Font: Times
* Face: Bold
* Size: 14

*Other Text Explanations*

* Font: Times
* Face: Normal
* Size: 12

## 1.3 Intended Audience

**Clients:** The users of the system will get a clear idea of the software and hardware requirements of the system and how the software will provide the end product of a financial tracking solution.

**Developers:** The developers have an advantage of quickly understanding the methodology enabled and any personalization of the product.

## 1.4 Additional Information

### 1.4.1 Definitions, Acronyms, and Abbreviations

* SRS: Software Requirement Specification

## 1.5 Contact Information/SRS Team Members

Michael Esry – [mdesry@ku.edu](mailto:mdesry@ku.edu)

Bill Kolega – [bkolega@ku.edu](mailto:bkolega@ku.edu)

Nicholas Roudebush – [nroudebush@ku.edu](mailto:nroudebush@ku.edu)

Tyler Steiner – [tsteiner@ku.edu](mailto:tsteiner@ku.edu)

## 1.6 References

Websites:

* <http://www.w3schools.com/>
* <http://www.codecademy.com/>
* <http://github.com/>

# 2. Overall Description

## 2.1 Product Perspective

The Client will supply an account number, password, and the transactions they would like to keep track of in $implicity.

## 2.2 Product Functions

The software will first be given an account number/password combination in order to open prior work the user may have done. The user may also create a new account number/password combination to keep track of a new account. The user may then view and/or enter transactions within the opened account.

## 2.3 User Classes and Characteristics

General Users: Every user will either be able to open a pre-existing account that they started or create a new account.

## 2.4 Operating Environment

The designed system will consist of a HTML/CSS/Javascript combination. It will reside on a Windows Server platform. Any modern web browser should be able to view the software/service and interact with the software.

## 2.5 User Environment

This software will be used in an Academic University Environment. It is not designed for commercial use other than that of the initial client.

## 2.6 Design/Implementation Constraints

A continuously operating server platform is required to host the software. A modern web browser is required to interact with the software.

## 2.7 Assumptions and Dependencies

* Web Browser is installed
* The user has internet access
* User has a financial record they would like to track

# 3. External Interface Requirements

## 3.1 User Interfaces

A webpage operating with a combination of HTML/CSS/Javascript. The user should see a simple, intuitively designed GUI where all information transfer between user and software is done.

## 3.2 Hardware Interfaces

--No required Hardware Interfaces--

## 3.3 Software Interfaces

The software will be hosted on a web server and will provide secure transmission. Therefore, a HTTPS ability and internet connection will be required.

# 4. System Features

## 4.1 Description and Priority

The system will consist of 3 main web pages that the user will interact with. The first page will be the welcome/login page. Here, the user can enter in an account number/password if they have one already and the software will load their previously entered data. If the user does not have an account already, there will be an option to take them to a page to create one.

The second page the user will likely interact with is the New Account page. Here, the user will enter a password they would like attached to the account as well as the starting balance of the account. The software will tell them their account number (unique to this software).

The third page the user will interact with is the Account View page. On this page, the user will be able to view past logged transactions and the current account balance, as well as having the ability to log new transactions to the ledger.

## 4.2 Action/Result

The result will be a ledger/view of the transactions that the user has logged for the account as well as the current account balance.

## 4.3 Functional Requirements

**Input:** Financial transaction information

**Output:** Organized ledger showing the transaction history of an account

# 5. Other Nonfunctional Requirements

## 5.1 Performance Requirements

This software is a very low requirement software. It will not take much computing power. The average computer a user may have should have zero difficulty running this software.

## 5.2 Software Quality Attributes

Since the user interface will be housed in a web-based GUI, the interaction should be quite smooth and the average user should not have any difficulty with it.

## 5.3 Project Documentation

Documentation will be done within the code itself as well as external documentation that will be stored on github.com.

## 5.4 User Documentation

End User documentation will be provided explaining the final product.