

# **Software Requirements Specification**

Team 6

February 15, 2017

Version 1

Prepared by:  
Jeremiah Trahern, Jonathan Whipple, Derek Wilson

## Contents

<b>1</b>	<b>Introduction</b>	<b>5</b>
1.1	Overview	5
1.2	Goals and Objectives	5
1.3	Scope	5
1.6	Assumptions	6
<b>2</b>	<b>General Design Constraints</b>	<b>7</b>
2.1	Product Environment	7
2.2	User Characteristics	7
2.3	Mandated Constraints	7
2.4	Potential System Evolution	7
<b>3</b>	<b>Nonfunctional Requirements</b>	<b>8</b>
3.1	Usability Requirements	8
3.2	Operational Requirements	8
3.3	Performance Requirements	8
3.4	Security Requirements	8
3.5	Safety Requirements	9
3.6	Legal Requirements	9
3.7	Other Quality Attributes	9
3.8	Documentation and Training	9
3.9	External Interface	9
3.9.2	Software Interface	10
<b>4</b>	<b>System Features</b>	<b>11</b>
4.1	Feature: Budgeting Overview	11
4.1.1	Description and Priority	11
4.1.2	Use Case: See Account Information	11
4.1.3	Use Case: Manipulate Data View	11
4.1.4	Use Case: Transaction Categorization	11
4.1.5	Use Case: Set a Budget	12
4.2	Feature: Goals	12
4.2.1	Description and Priority	12
4.2.2	Use Case: Set a budgeting goal	12
4.2.3	Use Case: See past goal performance	12

<b>4.3</b>	<b>Feature: Gamification</b> .....	13
<b>4.3.1</b>	<b>Description and Priority</b> .....	13
<b>4.3.2</b>	<b>Use Case:</b> Budget goal rewards .....	13

## Revision History

Version	Date	Name	Description
1	2/12/17	Jeremiah Trahern	Initial Document
1.1	2/12/17	Alexander Sweeney	Scope filled out
1.2	2/13/17	Jonathan Whipple	Section 2 - General Design Constraints filled out
1.3	2/14/17	Jeremiah Trahern	Definitions
1.4	2/14/17	Jeremy Troshynski	Nonfunctional Requirements
1.5	2/14/17	Colin Zeiler	System Features
1.6	2/14/17	Alexander Sweeney	System Features
1.7	2/15/17	Derek Wilson	Various formatting and additions
1.8	2/15/17	Daniel McNary	Updated Security Requirements, Safety Requirements, and Legal Requirements
1.9	2/15/17	Jonathan Whipple	Final design modifications for submission

# 1 Introduction

## 1.1 Overview

The Commerce Bank Budgeting Tool web application will be designed to help users of Commerce Bank budget, manage, and understand their money. With this budgeting tool, users will be able to monitor their spending patterns by pairing the budgeting tool with their Commerce bank account. Data from their bank account will be organized and categorized, showing each user a custom page intended to provide help in understanding their past transactions. To help users manage their spending, the budgeting tool will include the ability to set spending and savings goals, and will offer graphs and progress bars in order to help each user visualize their progress. The budgeting tool will also feature a gamification aspect to help motivate users to achieve their goals.

This document will layout the requirements for this product, starting with goals, scope, definitions, and assumptions in the introduction. The environment the budgeting tool will be developed in will then be discussed, according to constraints and functional/non-functional requirements of the project. The user characteristics, non-functional requirements, and features/potential features will all be described.

## 1.2 Goals and Objectives

The most important objectives of this project will be to:

1. Provide Commerce Bank users with a visual aid which accurately describes their spending habits based on their bank account transaction history.
2. Responsive on desktop computers and smaller screens, e.g. mobile phones, tablets.
3. Allow Commerce Bank users to freely set budgeting and saving goals and monitor the progress of their current goals.
4. Design the application to be visually appealing and concise, so the user can easily view the information they are looking for.
5. Feature a gamification aspect, rewarding users for saving and for reaching their goals.

## 1.3 Scope

This web application will gather account information from the user such as account statements, timestamps of the account statements, and whether the statements come from the checking or savings account. Using the information collected from the user's account, the application will display visualizations of the expenditures and deposits according to details about the account statements.

## 1.4 Definitions

**Client** - the organization for which this application is being built - Commerce Bank.

**User** - The people who will use the application.

**Stakeholder** - Anyone directly affected by the application and its market performance, which includes clients, customers, users, senior managers, project managers, and developers.

**Gamify/gamification** - Adding elements to an otherwise serious app, intended to provide incentives or a sense of achievement for users. This encourages users to continue using the budgeting tool by keeping users interested.

**Use case** - An interaction between an actor (user of the system or another system) and the system (our application). They are written to help understand how this system is intended to be used after it has been developed.

**Actor** - The person or part of the system that a use case is intended to serve.

**Visuals/visualize** - This term is used in reference to graphs and models which will show data in a more user-friendly way, instead of just text/numbers. Most likely, donut charts and bar graphs will be most of what this budgeting tool will use.

**Constraint** - A guideline which restricts how the budgeting tool can be implemented.

**Non-functional Requirements** - Properties the system should have, serving as a guideline for designing the product.

**Product** - The software which this document describes.

**Environment** - Refers to the language, framework, and coding tools in which the product will be implemented, as well as what other tools/products it will be compatible with.

## 1.5 Document Conventions

Portions of this document that are incomplete will be marked with TBD. Each TBD item will have an owner and estimated date for resolving the issue.

## 1.6 Assumptions

It is assumed that we as a team do not have to handle any security considerations for the web application.

It is assumed that Commerce Bank uses SQL 2012 for their databases.

It is assumed said database will be accessible from the machine where the system will run.

It is assumed that Commerce Bank uses the .NET framework.

## 2 General Design Constraints

### 2.1 Product Environment

The gamified budgeting web application will be a standalone project that can be integrated easily into another environment, preferably in Commerce Bank's environment. The product must exist in a .NET framework which uses Microsoft SQL Server 2012 for the database. The usage of outside libraries is acceptable, but the source files must be included in the project. The product will need the ability to integrate easily with security, but this is not our priority during the creation of the tool.

### 2.2 User Characteristics

**Commerce Bank Users** - Customers of Commerce Bank who access online banking. They can access the budgeting tool through a computer browser or through their smart phone's browser.

### 2.3 Mandated Constraints

The budgeting tool must be built in the .NET framework. For the database the tool must use Microsoft SQL Server 2012. Any external resources used must be included in the project. These are requirements that our client, Commerce Bank, has given us.

### 2.4 Potential System Evolution

The potential of this system is to be integrated into Commerce Bank's online banking webpage. The project should be written in such a way that it is easy to read the code and follow the workflows. The project should also have an API that can be used.

## **3 Nonfunctional Requirements**

### **3.1 Usability Requirements**

The budgeting tool must be usable by basic users as well as more experienced users without task failures.

The budgeting tool should be subjectively appealing to all ages. An acceptable satisfaction percentage of those surveyed must be above seventy percent.

The budgeting tool should take on average less than two minutes to learn the most basic functionality.

Any and all tasks should be either intuitive for most users, or outlined within the user manual for reference.

Users should feel that the tool is easy to understand. An acceptable percentage of users who feel this way must be above sixty percent.

### **3.2 Operational Requirements**

Users will access the budgeting tool from a web browser and must be able to view the same information on any device. Smartphones and other similar screens must be taken into consideration.

### **3.3 Performance Requirements**

The system as a whole should not take too long to run. Any action that a user completes should take less than five seconds. Space is not a major concern of this project, but it's still important that we keep our database as tight as possible. The tool should be able to function quickly on any size screen.

### **3.4 Security Requirements**

The security requirements for this project are minimal, as the focus set forth by Commerce Bank is to create a more feature-rich application. It is not necessary for this application to provide any sort of secure login for customers. Security features will need to be implemented by Commerce Bank, if the project is chosen to proceed in their future releases. Although developers on this project will still follow certain standards of practice throughout the duration of the project. Users of the application should only be able to see details of their particular account and budgeting information. If a user opts to share information outside of the application to third parties, certain sensitive information should not be shared, such as monthly budget amounts. If an administrator role is provided, this account should be able to view any pertinent information for general users of the application.



### **3.5 Safety Requirements**

Personal account information should be kept within the web application and not shared with outside sources, except with expressed consent of the user. Any leaks of data should be treated as a safety concern for our users, although this is mostly outside of the scope of this project. User data security will need to be addressed by Commerce Bank.

### **3.6 Legal Requirements**

Commerce Bank account information should not be shared with any outside source or other online entity. Any and all sharing functionality must be void of any personal or transaction related data. Security features related to the legal requirements of a bank will need to be implemented by Commerce Bank, if the project is chosen to proceed in their future releases.

### **3.7 Other Quality Attributes**

Commerce Bank Users should be able to access the budgeting tool quickly and easily with the use of a device connected to the internet.

### **3.8 Documentation and Training**

A short help screen will be displayed within the web application to guide users on how to use the product. Functionality that isn't readily apparent will be outlined in the user manual. The web application will have descriptions of each of the pages and their modules. Any and all help should be available to users within the user manual accessed on any page of the application.

### **3.9 External Interface**

#### **3.9.1 User Interface**

The overview screen will provide a general sense of how the user is doing on their spending/saving goals. From this page, users should be able to navigate to other tabs including: budget, goals, and transactions.

The budget page should display graphs of the spending goals that a user has set for themselves for a particular month, as well as their current status of spending for each category.

The goals page should display saving goals that the user has defined for a period of time. The goals should have a form of progress tracking. A user should be able to add goals or edit existing ones to reflect their budgeting decisions. While adding or editing a goal, a popup screen should appear, which will take the user through the process of constraints that the goal will impose.

The transactions page will show the user all transactions that have occurred within their account. Users will be able to edit transactions for their account, to better reflect categorization or to add cash transactions to the list of transactions. From this page, a user should be able to add or edit existing categories of transactions. Users may define their categories as they see fit.

Generally, the interface should appear to be inviting, professional, and clean. It's important for the interface to not feel too much like the users are completing tasks, but are instead exploring their budgeting possibilities. The color scheme will match that of Commerce Bank's green and white theme.

The general user of the system will be a wide range of ages including young adults as well as seniors. The interface must be somewhat intuitive to use, by taking advantage of common conventions. Not all users should be expected to know how to use the system, so a user manual will be provided on each page of the interface. A user will simply have to click a help button to be guided to their destination.

### **3.9.2 Software Interface**

All libraries used must be included internally within the final submission of the project. The web application cannot rely on any external resources. After completion, the budgeting tool will be implemented into Commerce Bank's existing ecosystem. Our project will be linked to an SQL Server 2012 database for testing purposes only. Commerce Bank would then integrate the system into their own database.

## 4 System Features

### 4.1 Feature: Budgeting Overview

#### 4.1.1 Description and Priority

The Budgeting Overview screen will be where user can see an overall view of their spending and current budgeting goals. This section discusses the features available to users on this screen.

Cost: low

Risk: low

Value: high

#### 4.1.2 Use Case: See Account Information

**Primary Actor:** User

**Trigger:** User wants to see their account information

**Main Success Scenario:**

1. User logs into website
2. Data is posted on the first screen after logging in

**Extensions:**

- 1a. User fails to log into website

#### 4.1.3 Use Case: Manipulate Data View

**Primary Actor:** User

**Trigger:** User manipulates the way data is viewed

**Main Success Scenario:**

1. User clicks on drop-down box
2. User selects which view they would like to switch to

**Extensions:**

- 2a. User doesn't select an option. No changes made

#### 4.1.4 Use Case: Transaction Categorization

**Primary Actor:** User

**Trigger:** User wants to see their transactions by category

**Main Success Scenario:**

1. Transactions will be categorized automatically

**Extensions:**

- 1a. User does not select a category. No changes made.

#### 4.1.5 Use Case: Set a Budget

**Primary Actor:** User

**Trigger:** User wants to set a budget, choosing a repeating or a long term budget.

**Main Success Scenario:**

1. User selects what kind of budget needed.
2. User enters in basic information about the budget.
3. Budget is shown in the budgeting section.

**Extensions:**

- 3a. User does not select a budget. No changes made

### 4.2 Feature: Goals

#### 4.2.1 Description and Priority

The Goals screen will be where users can view their current and past goals, create new budgeting goals, or edit existing goals.

Cost: Medium

Risk: Low

Value: High

#### 4.2.2 Use Case: Set a budgeting goal

**Primary Actor:** User

**Trigger:** Set goal button

**Main Success Scenario:**

1. User clicks on button to set a goal
2. System redirects user to another page/pop-up box
3. User types in goal information
4. System saves goal to their profile

**Extensions:**

- 2a. System loads wrong page/pop-up box
- 3a. User enters invalid data
- 4a. System fails to save information

#### 4.2.3 Use Case: See past goal performance

**Primary Actor:** User

**Trigger:** Past performance button

**Main Success Scenario:**

1. User clicks on past performance button
2. System redirects to past-performance page

**Extensions:**

- 2a. System loads wrong page

## **4.3 Feature: Gamification**

### **4.3.1 Description and Priority**

This feature will allow users to be rewarded for meeting their budgeting goals. Exact rewards TBD.

### **4.3.2 Use Case:**Budget goal rewards

**Primary Actor:** User

**Trigger:** User wants to be rewarded for meeting budget goal rewards

**Main Success Scenario:**

1. System finds that the user has met their budget goal(s)
2. System rewards user

**Extensions:**

- 2a. System fails to reward user properly