Software Requirements Specification

For

Team 9

February 17, 2020

Version 1

Prepared by:

Michael Cu, Joshua Daniels, Mao Zheng, Toni Tan

**Table of Contents**

[**REVISION HISTORY 3**](#_32hioqz)

[**1**](#_1hmsyys) **INTRODUCTION 4**

[**1.1**](#_41mghml) **Overview 4**

[**1.2**](#_2grqrue) **Goals and Objectives 4**

[**1.3**](#_vx1227) **Scope 5**

[**1.4**](#_3fwokq0) **Definitions 5**

[**1.5**](#_1v1yuxt) **Document Conventions 6**

[**1.6**](#_4f1mdlm) **Assumptions 6**

[**2**](#_2u6wntf) **GENERAL DESIGN CONSTRAINTS 7**

[**2.1**](#_19c6y18) **Product Environment 7**

[**2.2**](#_3tbugp1) **User Characteristics 7**

[**2.3**](#_28h4qwu) **Mandated Constraints 7**

[**2.4**](#_nmf14n) **Potential System Evolution 8**

[**3**](#_37m2jsg) **NONFUNCTIONAL REQUIREMENTS 8**

[**3.1**](#_1mrcu09) **Usability Requirements 8**

[**3.2**](#_44sinio) **Operational Requirements 8**

[**3.3**](#_46r0co2) **Performance Requirements 9**

[**3.4**](#_2lwamvv) **Security Requirements 9**

[**3.5**](#_111kx3o) **Safety Requirements 9**

[**3.6**](#_3l18frh) **Legal Requirements 9**

[**3.7**](#_206ipza) **Other Quality Attributes 9**

[**3.8**](#_4k668n3) **Documentation and Training 9**

[**3.9**](#_2zbgiuw) **External Interface 9**

[3.9.1 User Interface 10](#_1egqt2p)

[3.9.2 Software Interface 10](#_3ygebqi)

[**4**](#_2dlolyb) **SYSTEM FEATURES 10**

[**4.1**](#_sqyw64) **Feature: Required Features 11**

[4.1.1](#_3cqmetx) Use Case: 1 [11](#_3cqmetx)

[4.1.2 Use Case:](#_1rvwp1q) 2 11

[4.1.3 Use Case: 3 11](#_1rvwp1q)

[4.1.4 Use Case: 4 11](#_1rvwp1q)

[4.1.5 Additional Requirements 11](#_4bvk7pj)

[**4.2**](#_2r0uhxc) **Feature: Optional Features** 11

[4.2.1](#_1664s55) Use Case: 5 11

[4.2.2 Use Case:](#_3q5sasy) 6 11-12

4.2.3 Use Case:7 12

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Name** | **Description** |
| 1 | 02/24/2020 | Toni Tan | Initial Documentation; Introduction and Design |
| 2 | 2/29/2020 | Joshua Daniels | Introduction and Design (Revision) |
| 3 | 3/1/2020 | Mao Zheng | Revision on User case and initial documentation |
| 4 | 3/1/2020 | Michael Cu | Proof-Reading and Revision of Full document |

# Introduction

## Overview

The Commerce Bank project is a web application that is created in the .Net framework.

The application will allow users to log in to their bank’s account, check their transaction history, and add transactions as needed. Triggers for notification rules will also be set up, allowing the user to track their spending.

This document provides information on the requirements for the “Commerce Bank” website application. Project goals, scope, and definitions are given in the introduction, while design constraints and application environment are described in the following section. Functional requirements are given to show the system features and expected user interaction, and Non-functional requirements are outlined for later verification.

## Goals and Objectives

The main goal of this project is to give users the ability to check transactions and set trigger alerts. The web application is expected to:

* Have a user-friendly interface
* Allow users to turn on/off notification
* Export data to a spreadsheet
* Allow saving of data to a database

## Scope

Bare minimum, the software will allow users to check their bank account’s balance and transactions and set up notifications. Time permitting, the scope could be extended to include features such as getting notifications via email or text, or remembering users if they close their browser and log in again.

## Definitions

**Web application** - the product that is being described here; the software system specified in this document.

**Use case** – describes a goal-oriented interaction between the system and an actor. A use case may define several variants called scenarios that result in different paths through the use case and usually different outcomes.

**Scenario** – one path through a use case.

**Actor** – user or other software system that receives value from a use case.

**Role** – category of users that share similar characteristics.

**Developer** – the person or organization developing the system, also sometimes called the supplier.

**Stakeholder** – anyone with an interest in the project and its outcomes. This includes clients, customers, users, developers, testers, managers and executives.

**Project** – activities that will lead to the production of the Commerce Bank web application.

**Controls** – the individual elements of a user interface such as buttons and checkboxes.

## Document Conventions

People who work on this document will be shown on the revision history table.

## Assumptions

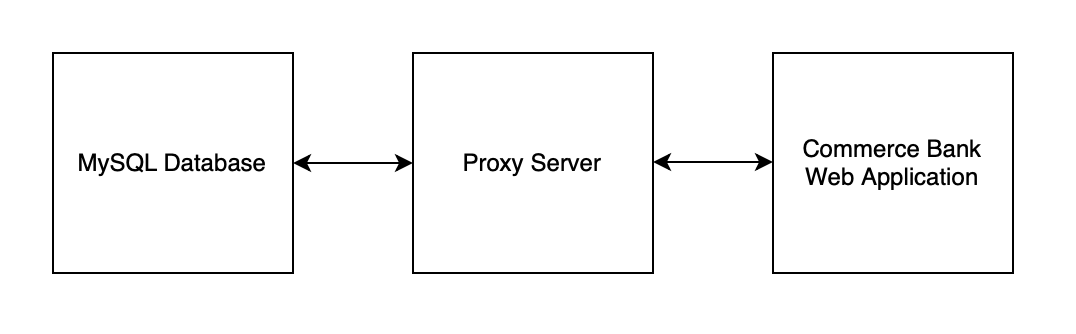
It is assumed that all users should already have an account, one user has one account, and all users have a device that connects to the internet and allows them to access the web application.

# General Design Constraints

## Commerce Bank web application Environment

The Commerce Bank project will include a web application to work on browser on any device. The web application will interface with a proxy server of our design.

* Database: MySQL Database
* Platform: .Net, Heroku
* Operating system: Windows/MacOS/Linux



## User Characteristics

**Commerce Bank web application:** All bank users who have a bank account must have a device such as a computer or mobile instrument with an internet browser that is able to connect.

## Mandated Constraints

The web application will run on the .Net framework, and it’s data will be stored in the Microsoft SQL database. This framework was chosen based on C# on team consensus, and data will be stored in a MySQL database.

## Potential System Evolution

This software can be extended for multiple account management when users have more than one account. Users can switch to another account without typing in account number and password back and forth.

# Nonfunctional Requirements

Nonfunctional requirements of the system could remember user login account number if they close their browser and log in again. The system could automatically log out users after no action on screen in 2 minutes.

## Usability Requirements

At least 80% users will not have problems in the login page to transaction page. Users will be able to learn and remember how to use the functions of the software in at most 30 minutes. Users should not need a manual. The data should update immediately after adding a transaction.

## Operational Requirements

The users should be in a safe environment and connect to web applications through a secure network. The application will require users to be able to operate by hand.

## Performance Requirements

Website should take less than 3 seconds to respond after action on the login button, either login success or fail. Every action on each page should take less than 2 seconds to respond to the users.

## Security Requirements

The Commerce Bank web application will mask the password field while the user is typing in. Password will be required to meet the password requirement. Password stored in the database will be hashed. User can not see full text of their personal information such as phone numbers, and email addresses to prevent masquerading and hacking. Users will receive notifications based on if they log in to their account, when they log in, and where they log in.

## Legal Requirements

To prevent fraud, users’ social security number will not be visible anywhere on the web application and database will only allow store the last four digit of social security number.

## Other Quality Attributes

Viewing the website on any devices’ browser regardless of the size of the screen should provide no issue in displaying information onto the website. Due to this, the user can use the application at any location.

## Documentation and Training

The Commerce Bank web application will be delivered to users with a simple user manual. The user manual should include screenshot of each web page and a small amount of text and a short demo video no longer than 2 minutes.

## External Interface

### User Interface

The user interface will be high contrast between white and green based on the Commerce Bank color scheme. Each web page should provide a smooth transition. Color used and alignment should stay constant. Users can easily understand just by looking at the interface and feel no pressure in learning or using. Expected users will be adults.

### Software Interface

The proxy server will serve as an interface between MySQL database and Commerce bank web application. It will provide a request, check, and send information between the user and the website.

# System Features

## Required Features

### Use Case 1

Description: User Login their accounts

Actors: Commerce bank clients

Value: High

Risk: Low

Cost: Medium

Basic Path

1. User go to web application page
2. System prompts user to enter account number and password
3. User enters correct user account number and password and clicks login
4. System display transaction summary
5. User clicks Log out
6. System exits

Alternate path

1. User go to web application page
2. System prompts user to enter account number and password
3. User enters incorrect user account number or password and clicks login
4. System displays error message: “Invalid Email Address and / or Password for user@umkc.com ..... Or you may have exceeded the number of consecutive attempts allowed. Please try again later.”
5. User may choose to login again, returning to step 1, or exit.
6. System exits

### Use Case: 2

Description: User add transactions

Actors: Commerce bank clients

Value: High

Risk: Low

Cost: Medium

Basic Path

1. User go to web application page,
2. User log in their account successfully
3. User click on add transactions
4. User enter transactions information
5. Automatically trigger any associated notification rules
6. System display transaction added successfully

Error Path

1. User go to web application page,
2. User log in their account successfully
3. User click on add transactions
4. User enter incorrect transactions information
5. System displays error message: “Invalid Transaction information entered or didn’t enter information with correct format. Please try again later.”
6. Users may choose to add transactions again, returning to step 1, or exit.

### Use Case: 3

Description: User receive, add/edit/delete notifications

Actors: Commerce bank clients

Value: High

Risk: Medium

Cost: High

Basic Path

1. User go to web application page,
2. User log in their account successfully
3. User receive notifications with different notifications rules
4. User click on notifications and be able to review it
5. User can add/edit/delete/hide/pull/compare notifications
6. System display modifying successfully

### 4.1.4 Use Case: 4

Description: User is able to export spreadsheet

Actors: Commerce bank clients

Value: High

Risk: Low

Cost: Medium

Basic Path

1. User go to web application page
2. User log in their account successfully
3. User click on the button to export spreadsheet
4. System exports the spreadsheet successfully
5. User downloads the spreadsheet.
6. User can review exported spreadsheet

### 4.1.5 Use Case: 5

Description: Notification rule set up

Actors: Commerce bank clients

Value: High

Cost: High

Basic Path

1. User login and go to dashboard
2. Click on notification rule setting
3. Type in number or make selection from giving states and categories
4. Click on done button after finish
5. System go back to dashboard page

### 4.1.6 Additional Requirements

Include in this section additional functional and non-functional requirement not specified in the use case(s) above.

## 4.2 Optional Feature

### 4.2.1 Use case:6

Description: User owns multiple accounts

Actors: Commerce bank clients

Value: High

Risk: Medium

Cost: High

Basic Path

1. User go to web application page,
2. User log in their account successfully
3. User click on the accounts they want to access
4. User enter the specific account successfully
5. User can manage and view any information associated with this account
6. User can switch between different accounts
7. User clicks logoff
8. System exits

**4.2.1 Use case:7**

Description: User phone number and email set up/change

Actors: Commerce bank clients

Value: High

Risk: Medium

Cost: High

Basic Path

1. User go to setting page
2. User click on change profile information
3. Type in phone number or email address for receiving notifications
4. User clicks on finish
5. System display double confirmed on contact information
6. User click on confirmed
7. System exits