Software Requirements Specification

For

Commerce Bank Web Portal - Group 8

3/1/2020

Version 1

Prepared by:

Nathanael Goertzen

**Table of Contents**

**REVISION HISTORY 3**

**1** **INTRODUCTION 4**

**1.1** **Overview 4**

**1.2** **Goals and Objectives 4**

**1.3** **Scope 4**

**1.4** **Definitions 4**

**1.5** **Document Conventions 5**

**1.6** **Assumptions 5**

**2** **GENERAL DESIGN CONSTRAINTS 5**

**2.1** **Product Environment 5**

**2.2** **User Characteristics 5**

**2.3** **Mandated Constraints 5**

**2.4** **Potential System Evolution 6**

**3** **NONFUNCTIONAL REQUIREMENTS 6**

**3.1** **Usability Requirements 6**

**3.2** **Operational Requirements 6**

**3.3** **Performance Requirements 6**

**3.4** **Security Requirements 6**

**3.5** **Safety Requirements 6**

**3.6** **Legal Requirements 6**

**3.7** **Other Quality Attributes 6**

**3.8** **Documentation and Training 6**

**3.9** **External Interface 7**

3.9.1 User Interface **7**

3.9.2 Software Interface **7**

**4** **SYSTEM FEATURES 7**

**4.1** **Feature: User login / Dashboard 7**

4.1.1 Description and Priority **7**

4.1.2 Use Case: Login to dashboard **7**

4.1.3 Use Case: Export to spreadsheet **7**

4.1.4 Additional Requirements **8**

**4.2** **Feature: Transaction flags 8**

4.2.1 Description and Priority **8**

4.2.2 Use Case: Set transaction trigger **8**

4.2.3 Additional Requirements **8**

**4.3** **Feature: Add Transaction 8**

4.3.1 Description and Priority **8**

4.3.2 Use Case: Add Transaction **9**

4.3.3 Additional Requirements **9**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Name** | **Description** |
| 1 | 3/1/2020 | Nathanael Goertzen | Initial Document |

# **Introduction**

## ***Overview***

This document specifies the overarching requirements for the Commerce Bank Web Portal. This document’s purpose is to present the requirements in a format that can be understood by nontechnical personnel while also providing sufficient context for the developers to make design decisions for the final application.

The Commerce Bank Web Portal will be a web application that can be accessed on any web-capable device. The application will contain account information relating to bank transactions, and will allow users to set up notification rules to flag specific transactions. The application will also allow users to manually input transactions and to export their transaction data into a spreadsheet format.

The first section of this document contains information on the project’s goals and scope and provides definitions for terms used within this document. The second section details the project’s Design Constraints. The third section will list all of the non-functional requirements. Finally, the fourth section will detail the functional requirements.

## ***Goals and Objectives***

The goals of the Commerce Bank Web Portal are as follows:

1. Provide a web application for bank users to view their transaction history.
2. Allow bank users to set flags for their account that will notify the users if a transaction occurs that trips one of the flags. The purpose of this feature is to reduce bank fraud risk by alerting our users early to suspicious transactions.
3. The system should be able to be accessed on any internet-capable device, and should be intuitive to use.

## ***Scope***

The Commerce Bank Web Portal will allow users to view their account’s transaction history. Users will be able to set transaction flags on a “Triggers” page. Any transaction on the account that fulfills the requirements for a flag will appear on the user’s home-page dashboard. Users will be able to sign up for an account with a username and password. Users will be able to add transactions to their account. Users will be able to export transaction history in a spreadsheet format.

## ***Definitions***

**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.

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

**Product** – The web application described by this requirements specification.

**Project** – activities that will lead to the production of the product described here. Project issues will be described in a separate project plan.

**Must** – adverb used to indicate importance; indicates the requirement is mandatory. “Shall” and “will” are synonyms for “must”.

**Should** – adverb used to indicate importance; indicates the requirement is desired but not mandatory.

**May** – adverb used to indicate an option. For example, “The system may be taken offline for up to one hour every evening for maintenance.” Not used to express a requirement, but rather to specifically allow an option.

## ***Document Conventions***

This document describes the requirements of the project as understood at the published date. This document is subject to be changed for minor section revisions or clarity of definitions.

## ***Assumptions***

It is assumed that the client has a SQL capable database software on the environment that this web application will run alongside.

# **General Design Constraints**

## ***Product Environment***

The Commerce Bank Web Portal will be an ASP.NET web application. The application will be run on a web server with a SQL capable database installed. Transaction data and user account information will be hosted in the database.

## ***User Characteristics***

**Commerce Bank Web Portal Users:** Any Commerce Bank customer that has an account with the application. Bank customers in the US are a large group with varying technical literacy.

## ***Mandated Constraints***

The application must be a web application built in a newer, stable framework. The application must be responsive to whichever device requests access. The application must use a CSS framework.

## ***Potential System Evolution***

The resulting software system should be maintainable and extensible. Knowing the types of anticipated changes aids significantly in establishing an architecture that will accommodate the types of expected changes. This section suggests ways the system is likely to be extended or modified in the future.

# **Nonfunctional Requirements**

## ***Usability Requirements***

All users familiar with web applications should have no issues navigating the web application. 95% of users with minimal web application interaction experience should be able to view basic transactions and set basic transaction flags.

## ***Operational Requirements***

The application must be able to be used on any device with ease. The webpages should be responsive to any device that access the application.

## ***Performance Requirements***

Barring connection issues, webpages should load in under 10 seconds for the user.

## ***Security Requirements***

User account information should only be visible and accessible to the user who owns said account. Users should verify themselves as the owners of their account by logging into the web application with their account number and password. The password must have a capital letter, a lower case letter, a number, a symbol, and contain a minimum of 8 characters. The webpage should mask the password field.

## ***Safety Requirements***

The web application necessitates no special requirements pertaining to safety of use.

## ***Legal Requirements***

Transactions on an account must only be performed by the account holder.

## ***Other Quality Attributes***

The web application should be available during daytime and evening hours. Application maintenance should be performed weekly on a consistent schedule.

## ***Documentation and Training***

The Commerce Bank Web Portal will be available to users and require no documentation or training in order to access it. System documentation and a user guide will be provided to stakeholders.

## ***External Interface***

### **User Interface**

The user interface will be simple and organized. There shall be a menu bar leading to each important page of the web application. When a user clicks on a tab in the menu bar, the web application should render the selected section.

The interface will be minimal. In order to provide the users with a simple interface, there should be no obfuscations and pages should be clearly and concisely named. 95% of users should be able to operate the web application without prior training.

### **Software Interface**

There are no current plans to allow the software to be accessed via API endpoints. All interaction must be performed through the user interface specified above.

# **System Features**

## ***Feature:*** User login / Dashboard

### **Description and Priority**

Upon accessing the web application, first the user must login in order to see transactions. The user will be able to log in with a provided account ID and password. Once this is accomplished, the application will redirect them to a dashboard with the account’s most recent transactions.

Cost: medium

Risk: low

Value: high

### **Use Case:** Login to dashboard

Actors: Bank account holders

Basic Path

1. User clicks the menu option to log in.
2. The account login webpage loads, and prompts user to enter their account ID and password
3. User enters their ID and password and clicks the login button
4. The dashboard webpage loads with the user’s recent transactions shown.

Alternate Path

1. User clicks the menu option to log in.
2. The account login webpage loads, and prompts user to enter their account ID and password
3. User enters an incorrect ID and password combination and clicks login.
4. The web application displays a login error message: “Invalid ID and/or password.”
5. The user may attempt to login again, returning to step 1

### 4.1.3 Use Case: Export to spreadsheet

Actors: Bank account holders

Basic Path

1. User selects “Export to spreadsheet” from the dashboard.
2. The web application issues a download of the current dashboard information into a spreadsheet readable format

### 4.1.4 **Additional Requirements**

The dashboard shall not load unless correct credentials are entered.

## ***Feature:*** Transaction flags

### **Description and Priority**

The web application will have a page dedicated to setting up transaction triggers. These will flag transactions that meet the set criteria for the account holder’s review.

Cost: medium

Risk: low

Value: high

### **Use Case:** Set transaction trigger

Actors: Bank account holders

Basic Path

1. User selects the “Transaction Flags” option from the menu and clicks.
2. The web application renders the transaction flag settings web page
3. User clicks “Create new transaction flag”.
4. The web application loads a page where the user can create a new transaction flag rule.
5. User selects a flag type from the drop down menu and enters further information into a textbox if required (dollar amount, date, etc.)
6. User clicks “Submit”.
7. The transaction flags page renders and displays the new transaction flag.

### **Additional Requirements**

## 4.3 Feature: Add Transaction

### 4.3.1 Description and Priority

The web application will offer users the ability to add transactions to their account.

Cost: low

Risk: low

Value: high

### 4.3.2 Use Case: Add Transaction

Actors: Bank account holders

Basic Path

1. User selects “Transactions” from the menu bar and clicks.
2. Web application renders the transactions page.
3. User clicks “Add new transaction”.
4. Web application renders a page where the user can add transaction information.
5. User enters the transaction description and dollar amount and then clicks “Submit”.
6. If the new transaction triggers a transaction flag, the user will be notified on their dashboard.
7. The web page redirects the user to the dashboard.

### 4.3.3 Additional Requirements

Duplicate rules should not trigger twice.