Software Requirements Specification

For

Commerce Project

February 29, 2020

Version 1

Prepared by:

Alexa Summers

**Table of Contents**

[**REVISION HISTORY 3**](#_heading=h.32hioqz)

[**1**](#_heading=h.1hmsyys) **INTRODUCTION 4**

[**1.1**](#_heading=h.41mghml) **Overview 4**

[**1.2**](#_heading=h.2grqrue) **Goals and Objectives 4**

[**1.3**](#_heading=h.vx1227) **Scope 5**

[**1.4**](#_heading=h.3fwokq0) **Definitions 5**

[**1.5**](#_heading=h.1v1yuxt) **Document Conventions 6**

[**1.6**](#_heading=h.4f1mdlm) **Assumptions 6**

[**2**](#_heading=h.2u6wntf) **GENERAL DESIGN CONSTRAINTS 7**

[**2.1**](#_heading=h.19c6y18) **Product Environment 7**

[**2.2**](#_heading=h.3tbugp1) **User Characteristics 7**

[**2.3**](#_heading=h.28h4qwu) **Mandated Constraints 7**

[**2.4**](#_heading=h.nmf14n) **Potential System Evolution 8**

[**3**](#_heading=h.37m2jsg) **NONFUNCTIONAL REQUIREMENTS 8**

[**3.1**](#_heading=h.1mrcu09) **Usability Requirements 8**

[**3.2**](#_heading=h.44sinio) **Operational Requirements 8**

[**3.3**](#_heading=h.46r0co2) **Performance Requirements 9**

[**3.4**](#_heading=h.2lwamvv) **Security Requirements 9**

[**3.5**](#_heading=h.111kx3o) **Safety Requirements 9**

[**3.6**](#_heading=h.3l18frh) **Legal Requirements 9**

[**3.7**](#_heading=h.206ipza) **Other Quality Attributes 9**

[**3.8**](#_heading=h.4k668n3) **Documentation and Training 9**

[**3.9**](#_heading=h.2zbgiuw) **External Interface 9**

[3.9.1 User Interface 10](#_heading=h.1egqt2p)

[3.9.2 Software Interface 10](#_heading=h.3ygebqi)

[**4**](#_heading=h.2dlolyb) **SYSTEM FEATURES 10**

[**4.1**](#_heading=h.sqyw64) **Feature: Triggers 11**

[4.1.1 Description and Priority 11](#_heading=h.3cqmetx)

[4.1.2 Use Case:](#_heading=h.1rvwp1q) Triggers 11

[4.1.3 Additional Requirements 11](#_heading=h.4bvk7pj)

[**4.2**](#_heading=h.2r0uhxc) **Feature: Login Page 12**

[4.2.1 Description and Priority 12](#_heading=h.1664s55)

[4.2.2 Use Case:](#_heading=h.3q5sasy) Login 12

[4.2.3 Additional Requirements 12](#_heading=h.25b2l0r)

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Name** | **Description** |
| 1.0 | 02/29/2020 | Alexa Summers | Initial Documentation |
| 1.1 | 3/1/2020 | Tarsus Arciga | Extra specifications |

# Introduction

## Overview

This Commerce Bank web application will provide clients of the bank access to all of their banking needs right in one place, at any time day or night. This application will be available to both computer and mobile users. Users will be able to log into their account, view past transactions, and set triggers for transaction notifications. Triggers will include things such as out-of-state purchases, purchases outside of a certain time frame, and purchases over a set amount. Past triggers will also be viewable from the dashboard, and are editable.

This document solely provides information on the point of the project, goals, the scope, assumptions, constraints, system features, and system requirements. Everything else, including budget and scheduling, will be addressed in documentation at a later time.

## Goals and Objectives

The main objective of this project is to provide Commerce Bank clients with a place to keep track of transactions, and set triggers for transactions that might have been fraudulent. The Commerce application is expected to:

1. Provide a place for Commerce bank users to keep track of transactions on their account.
2. Set triggers for potential fraudulent activity on their account.
3. Make the client’s banking experience more enjoyable by giving them access to their banking information whenever and wherever they are.

## Scope

The Commerce Bank application will allow clients to log into their banking account, see past transactions, edit and create transactions, as well as set triggers to notify them of potential fraudulent activity. Fraudulent activity triggers will be set by creating state purchasing restrictions, price restrictions, and time-of-day restrictions. Information included when viewing transactions will be account type/number, processing date, balance, deposit/withdrawal, amount, and location.

## Definitions

**Client** – Customers of Commerce Bank that use the application.

**Fraudulent Activity** – Activity pertaining to one of three options: out-of-state, out-of-time-frame, or out-of-budget.

## Document Conventions

No conventions were used in this documentation.

## Assumptions

It is assumed that the client will have a computer or a mobile phone with internet to access this application from.

# General Design Constraints

## Transaction Trigger Tracker Environment

This Commerce Bank web application will be accessible via computer and mobile devices. In order to accomplish this, a three tier architecture will be used, constituting of the following:

**MariaDB Instance**: Stores user transactions and custom trigger criteria.

**Spring Boot Application:** Backend RESTful services for the frontend to interact with the trigger and transaction tables as well as enable user authentication and account management. The Spring Boot application will also be hosting the frontend web application itself.

**Angular Application:** Reactive frontend for users to log in and view transactions, export excel files, and create custom triggers for transactions.

## User Characteristics

Users of this application will be customers of Commerce Bank. The vast majority will have an average understanding of the way that web applications work, however there might be many customers that are not computer savvy. Those who are not well-versed in computers need to be the target audience, as it will increase the usability and clarity across the board for all users.

## Mandated Constraints

This web application will only be available on computer and mobile devices with internet access.

## Potential System Evolution

The Commerce Bank application might be improved in the future by allowing the user to perform actions pertaining to funds (for example, mailing checks, transferring money between accounts, etc.). Software-wise, it could become a mobile application rather than just a web application.

# Nonfunctional Requirements

## Usability Requirements

Usability: 90% of those using the application will be able to set triggers and notifications without any additional support. 95% of those using the application will be able to look up and create transactions without additional support.

## Operational Requirements

The user’s environment will vary between customers based on location and time of day, however it is an application that needs to be completely secure. Therefore, nothing should be audible, and sensitive information needs to be protected visually.

## Performance Requirements

Changes made to transactions should be automatically reflected in the transactions report.

## Security Requirements

Users will be required to log in with their username and password to their Commerce banking account. Past that point, there are no other security hurdles.

## Safety Requirements

There are no safety requirements beyond a personal information leak if security is faulty.

## Legal Requirements

No one else will have access to account information besides the bank itself.

## Other Quality Attributes

This web application is available on a variety of screens as long as it is internet accessible. The application is also available 24/7, 365 days a year.

## Documentation and Training

There will be no training provided. As for documentation, no other documentation will be provided other than what is required by stakeholders.

## External Interface

### User Interface

The User Interface will be professional and related to Commerce Bank (specifically, using the “Commerce Green”. Users will first come to a login page, and after signing in, they will see their dashboard which will give them a variety of options to get to where they want to be. Everything will be relatively straightforward, and therefore no training will be provided ahead of time. It is expected that the majority of users will be able to navigate on their own without any issues.

### Software Interface

Transactions will be stored in a database, as well as trigger preferences. A self contained instance of the Spring Boot application will host both the Angular web application and the microservices the web application invokes to interact with the database.

# System Features

## Feature: Triggers

### Description and Priority

The triggers feature is able to be set by the user, and alert them to things such as out-of-state purchases, out-of-country purchases, purchases after a certain time-frame, and purchases over a certain dollar amount.

Cost: none  
Risk: low  
Value: high

### Use Case: Triggers

1. User signs into web application.

2. System automatically brings user to dashboard (front page).

3. User can view account preferences from dashboard.

4. User can select triggers function.

5. User can activate desired triggers.

6. User can set constraints on desired triggers.

7. User saves preferences.

8. User returns to dashboard.

9. User can continue to use application or can log out.

### Additional Requirements

Everything should be relatively easy for the common user to navigate in order to find their account preferences.

## Feature: Login Page

### Description and Priority

This is the login page when the web application is first brought up. This will allow users to sign into their own accounts securely.

Cost: Low  
Risk: Low  
Value: High

### Use Case: Login

1. The user will navigate to the web application.

2. User will type in username.

3. User will type in password.

4. User will submit their credentials

5. Credentials are correct. User will securely sign into their account.

6. Once signed in, users will see their dashboard.

Alternate path:

1. The user will navigate to the web application.

2. User will type in username.

3. User will type in password.

4. User will submit their credentials.

5. Credentials are incorrect—user will be prompted for re-entry.

### Additional Requirements

All logins will have to be done securely and injection-less. This will also need to be easy enough for the common person to navigate.