Software Requirements Specification

For

Group 2

February 29, 2020

Version 1

Prepared by:

Jimin Choi, Matthew Hopkins, Connor Marchand, and Kory Overbay

**Table of Contents**

[**Revision History 3**](#_30j0zll)

[**1**](#_1fob9te) **Introduction 4**

[1.1 Overview 4](#_3znysh7)

[1.2 Goals and Objectives 4](#_2et92p0)

[1.3 Scope](#_tyjcwt) 4

[1.4 Definitions](#_3dy6vkm) 4

[1.5 Assumptions](#_4d34og8) 5

[**2**](#_2s8eyo1) **General Design Constraints 5**

[2.1 Commerce Bank Notifications Environment](#_17dp8vu) 5

[2.2 User Characteristics](#_3rdcrjn) 5

[2.3 Mandated Constraints](#_26in1rg) 5

[2.4 Potential System Evolution](#_lnxbz9) 6

[**3**](#_35nkun2) **Nonfunctional Requirements 6**

[3.1 Usability Requirements](#_1ksv4uv) 6

[3.2 Operational Requirements](#_44sinio) 6

[3.3 Performance Requirements](#_2jxsxqh) 6

[3.4 Security Requirements](#_z337ya) 6

[3.5 Legal Requirements](#_1y810tw) 6

[3.6 Other Quality Attributes](#_4i7ojhp) 6

[3.7 Documentation and Training](#_2xcytpi) 7

[3.8 External Interface](#_1ci93xb) 7

[3.8.1 User Interface](#_3whwml4) 7

[3.8.2 Software Interface](#_2bn6wsx) 7

[**4**](#_qsh70q) **System Features 7**

[4.1 Required Features](#_3as4poj) 7

[4.1.1 Use Case 1](#_1pxezwc) 7

[4.1.2 Use Case 2](#_49x2ik5) 7

[4.1.3 Use Case 3](#_2p2csry) 8

[4.1.4 Use Case 4](#_1pxezwc) 9

[4.1.5 Use Case 5](#_1pxezwc) 9

[4.1.6 Use Case 6](#_1pxezwc) 10

[4.1.7 Use Case 7](#_1pxezwc) 10

[4.2 Optional Features](#_147n2zr) 10

[4.2.1 Optional Use Case 1](#_3o7alnk) 10

[4.2.2 Optional Use Case 2](#_23ckvvd) 11

[4.2.3 Optional Use Case 3 1](#_ihv636)1

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Name** | **Description** |
| 1 | 02/29/20 | Kory Overbay,  Matthew Hopkins, Jimin Choi,  Connor Marchand | Initial Document |
|  |  |  |  |

# **Introduction**

## ***Overview***

This document defines the requirements and functionality for the Notifications web application being developed for Commerce Bank. This document aims to show the system requirements for the clients and stakeholders to understand and agree to with detail for the programming team to develop with.

This document was made in mind with an existing project charter that includes project issues like schedule, cost, methodology, and the existence of other deliverables such as a testing plan.

The Commerce Bank Notifications web application is a tool for Commerce Bank customers to set up and receive notifications based upon rules they set for their transactions, namely potential fraudulent transactions.

## ***Goals and Objectives***

The overall goal is to allow Commerce Bank Account Holders to view their account in an easy to read format and give them the ability to make sure their account is safe.

The five main goals of this project are

1. Implement notifications when certain events happen to keep the account holder informed on possible suspicious activity
2. Show a history of all transactions for a user
3. User interface is arranged in a simple and intuitive manner aligned with Commerce’s styling.
4. Achieve 10% or more code test coverage
5. Effectively use source control throughout the project

## ***Scope***

The website application will (1) allow a user to securely log into their account, (2) see all of their notifications for triggered transactions and (3) be able to control how their notifications are set. (4) be able to see their transactions history. Time permitting our team will focus on the aesthetics and responsiveness of the website. The website will be developed with Firefox, Chrome, Safari, or Edge in mind, and will not guarantee support for Internet Explorer, NetScape, or Opera.

## ***Definitions***

**Commerce Bank Notification Web-Application** - The product being described within this document

**Angular** - Javascript Framework for web development. Used to create the user interface of this application

**Spring** (Spring Boot) - Java Framework used to create web microservices. Used for back end code of the application

**MySQL** - Database Management System. Used to store data for the application

**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.

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

## ***Assumptions***

We assume that this application is, for the most part, a proof of concept. We will do our best to work towards a secure and efficient application, but with the limited time frames and expertise we understand perfection is not expected.

# **General Design Constraints**

## ***Commerce Bank Notifications Environment***

The Commerce Bank Notifications application will include a web application, developed with the Angular Javascript framework, with the development intended for modern web browsers. The application will connect to a SpringBoot microservice application using HTTP calls and REST services. The SpringBoot application will have a connection to a MySQL database that will store all the user data necessary for the application.

## ***User Characteristics***

**Commerce Bank Customers**: Bank members that use the internet, whether with a smartphone or a PC, to track their banking transactions. Users who use existing Commerce Banking applications should be able to recognize the applications interface layout.

## ***Mandated Constraints***

Approval will be required from Commerce Bank to make this an official application. Since many other teams will be asking the Commerce representatives questions as well, we understand their input and approval may be delayed in most situations.

Since we are treating this project as a proof of concept we are not following Commerce Bank’s requirements of utilizing sql server 2012 or higher. We understand that they will be limited in helping our team overcome obstacles.

## ***Potential System Evolution***

The system is designed as a microservice so that in the future, it could be adopted into a larger webpage while remaining contained within itself. If more notification requirements are desired for the application, the design of the notification triggers and saving of notifications on the database can adapt to them with relative ease.

# **Nonfunctional Requirements**

## ***Usability Requirements***

* 90% of users should be able to navigate the page and add/edit their notification rules without assistance.

## ***Operational Requirements***

* The user should have their data secure. If the application is left idle, there should be a timeout after 1 minute of inactivity.

## ***Performance Requirements***

* The application should be quick/responsive, taking no more than 3 seconds when performing user actions.
* The user should be able to login and view their notifications within 5 seconds with a stable internet connection

## ***Security Requirements***

Users will only be able to log onto their accounts. Our login page will require users to enter a specific user name and the correct password. Passwords will be protected through hashing. Since there are financial records with personally identifiable information, all data must be protected and secured.

## ***Legal Requirements***

Since these are financial transactions our website will need to be compliance with the Federal Reserve, IRS, FBI and other government agencies.

## ***Other Quality Attributes***

* The web application should be able to be accessed on modern mobile and PC web browsers, with page sizing and orientation being automatic between devices.

## ***3.7 Documentation and Training***

The Notification application will be available on a web server. A user guide will be provided on the page itself, and system documentation will be provided to project stakeholders.

## ***3.8 External Interface***

### **3.8.1 User Interface**

The user interface will be professional, minimalist, and straightforward so that the site is consistent with existing Commerce Bank pages. The page will be simple to use across many web enabled devices, namely smartphones and PCs, with a goal of 90% of its users to be able to use it without instruction.

### **3.8.2 Software Interface**

The spring boot server acting as back end code with primarily serve as an interface between the web application and the database holding the information the user needs to view.

# **System Features**

## ***Required Features***

### **Use Case: 1**

**Description: User Checks Notifications**

Actors: Commerce Bank Customers

Value = high

Cost = high

Basic Path

1. User logs in to application
2. The Notifications dashboard is loaded
3. User can view the number of times a notification can be triggered
4. User logs out
5. System exits

Alternate Path

1. User enters incorrect login information
2. System prompts user “Login information is incorrect” and asks user to re enter information
3. User enters correct information and follows basic path

### **Use Case: 2**

**Description: User Login / Check Transactions**

Actors: Commerce Bank Customers

Value = high

Cost = medium

Basic Path

1. User logs into application
2. User clicks “Transactions” tab in the navigation bar
3. System loads the users transactions
4. User can view the transactions made on their account
5. User logs out
6. System exits

Alternate Path

1. User enters incorrect login information
2. System prompts user “Login information is incorrect” and asks user to re enter information
3. User enters correct information and follows basic path

### **Use Case: 3**

**Description: User Adds Transaction/Triggers Notification**

Actors: Commerce Bank Customers

Value = high

Cost = high

Basic Path

1. User logs into application
2. Clicks Transaction tab
3. User clicks “Add Transaction” button
4. Form generates for user:
   1. User checks box for deposit or withdrawal
   2. User enters amount
   3. User enters description
5. User clicks “Add” button within popup
6. System takes user inputted values and combines with
   1. User’s account number
   2. The date of the transaction
7. System send new transaction to SQL through server
8. SQL database enters new transaction into table
   1. Balance is incremented if transaction is deposit
   2. Balance is decremented if transaction is withdrawal
9. User will see transaction added to their transaction log
10. If added transaction triggers notification user will be able to see notification in notification tab
11. User Clicks Log Out
12. System Exits

Alternative Path

User logs into application

1. User logs into application
2. Clicks Transaction tab
3. User clicks “Add Transaction” button
4. Form generates for user:
   1. User checks box for deposit or withdrawal
   2. User enters amount
   3. User enters description
5. User clicks “Cancel” button within popup
6. User Clicks Log Out
7. System Exits

### **Use Case: 4**

**Description: User Adds Trigger Notification**

Actors: Commerce Bank Customers

Value = high

Cost = high

Basic Path

1. User logs in to application
2. User loads into Notifications page
3. User clicks “Add Notification” button
4. System loads the Add Notification popup
5. User enters desired notification constraints
6. User clicks “Add” button within popup
7. System sends the new notification request to the server
8. System creates new SQL trigger with given notification rules
9. New notification rule is shown in notifications table
10. User clicks log out
11. System Exits

### **Use Case: 5**

**Description: User Edits Trigger Notification**

Actors: Commerce Bank Customers

Value = high

Cost = high

Basic Path

1. User logs in to application
2. User loads into Notifications page
3. User clicks “Edit Notification” button
4. System loads the Edit Notification popup
5. User alters desired notification constraints
6. User clicks “Done” button within popup
7. System sends the edited notification request to the server
8. System modifies SQL trigger with given notification rules
9. Modified notification rule is shown in notifications table
10. User clicks log out
11. System Exits

### **Use Case: 6**

**Description: User Removes Trigger Notification**

Actors: Commerce Bank Customers

Value = high

Cost = high

Basic Path

1. User logs in to application
2. User loads into Notifications page
3. User clicks “Delete Notification” button
4. System asks user “Are you sure”
5. User clicks “Yes” button
6. System sends request to backend to remove the trigger from the database
7. The notification rule is removed from the table
8. User clicks log out
9. System Exits

### **Use Case: 7**

**Description: User Exports Notifications**

Actors: Commerce Bank Customers

Value = high

Cost = medium

Basic Path

1. User login to application.
2. User clicks the Export Notifications on the page of Notifications.
3. User selects which notifications to export
4. User selects format of export
5. User confirms export
6. User clicks log out
7. System Exits

## ***Optional Features***

### **Optional Use Case: 1**

**Description: Responsiveness**

Actors: Commerce Bank Customers

1. User logons to website using a recent Android or iOS phone/tablet
2. Website will scale accordingly to viewport in either portrait or landscape mode
3. User will log out
4. System Exits

### **Optional Use Case: 2**

**Description: User Checks Transactions that caused notifications**

Actors: Commerce Bank Customers

Value = high

Cost = high

Basic Path

1. User logs in to application
2. The Notifications dashboard is loaded
3. User clicks a notification row in table
4. Row expands to show the transactions that triggered it
5. User logs out
6. System exits

### **Optional Use Case: 3**

**Description: System paginates transactions page**

Actors: Commerce Bank Customers

Value = high

Cost = medium

Basic Path

1. User logs into application
2. User clicks “Transactions” tab in the navigation bar
3. System loads the users 10 most recent transactions
   1. System has a “load more” link at the bottom of the table when there are more than 10 records for the user’s account
4. When the user clicks “load more”, 10 more records are loaded to the table, and the system determines if there are still more records again
5. User logs out
6. System exits