

Software Requirements Specification For Commerce Bank System

[02/29/2020]
[Version 6]

Prepared by:
Linden Stirk, Brian Roden, Debbie Kirchner, etc.

Table of Contents

REVISION HISTORY	3
1 INTRODUCTION	4
1.1 Overview	4
1.2 Goals and Objectives	4
1.3 Scope	3
1.4 Definitions	4
2 GENERAL DESIGN CONSTRAINTS	4
2.1 Product Environment	4
2.2 User Characteristics	4
2.3 Mandated Constraints	4
2.4 Potential System Evolution	5
3 NONFUNCTIONAL REQUIREMENTS	6
3.1 Usability Requirements	6
3.2 Performance Requirements	6
3.3 Security Requirements	7
3.4 Documentation and Training	7
3.5 External Interface	7
3.5.1 User Interface	7
3.5.2 Software Interface	7
4 SYSTEM FEATURES	8
4.1 Required Features	8
4.1.1 Use Case: View transaction history	8
4.1.2 Use Case: View notifications	8
4.1.3 Use Case: Add, edit, and delete notification rules	9
4.1.4 Use Case: Log in	10
4.1.5 Use Case: Easy-to-use, responsive user interface	10
4.1.6 Use Case: Export transactions	11

Revision History

Version	Date	Name	Description
1	02/24/18	Linden Stirk	Initial Document
2	02/22/20	Brian Roden	Completed Introduction (Section 1)
3	02/24/20	Debbie Kirchner	Completed Section 2; added definitions to Section 1.4
4	02/24/20	Linden Stirk	Added required features & use cases to Section 4; added some nonfunctional requirements to Section 3
5	02/25/20	Linden Stirk	Add usability requirements in Section 3
6	02/29/20	Linden Stirk	Final Document

1 Introduction

1.1 Overview

The Commerce Bank System is a web application that will allow customers who bank with Commerce to access information about their accounts online. This system will provide a home page, a transaction history, the ability to create and apply notification triggers specific to the customer's preference, and the ability for the customer to view their transactions in a spreadsheet.

This document will outline requirements specific to the Commerce Bank System. The goals and objectives, along with the project scope and definitions, are found following this section. The constraints of the Commerce Bank System will be described afterward. Non-functional requirements, system features, and use cases will be outlined later in the document.

1.2 Goals and Objectives

The four main goals of the Commerce Bank System are as follows:

1. Provide a web application to allow users to access different information about their accounts.
2. The system should function in a way that is easy to understand.
3. Allow users to create notification triggers specific to their preferences.
4. Allow users to view their transaction history and export the information to a spreadsheet.

1.3 Scope

The Commerce Bank System will allow users to view information about their account, specifically transaction history and notifications. Users of the Commerce Bank System will be able to apply specific triggers to their notifications, so they can be notified about events that they are concerned with. Users should also be able to export their transaction history in the form of a spreadsheet. The Commerce Bank System will not let people transfer funds, make payments, or apply for any line of credit to their account.

1.4 Definitions

Commerce Bank System – the web-based application that is being described within this document.

Transaction – a record of some amount of money that has moved either in or out of a person's bank account

Notification – a visual indication that a target event has happened

Trigger – a specific event that causes a notification to appear

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 another 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 Commerce Bank System.

Client – the person or organization for which the Commerce Bank System is being developed.

User – the person or people who will actually interact with the Commerce Bank System.

Developer – the person or people who are developing the Commerce Bank System.

2 General Design Constraints

2.1 *Product Environment*

The Commerce Online Banking Transaction Viewer is predominantly a standalone web application, according to the current scope of the project. It will use sample data instead of connecting to a larger transaction database through the bank.

2.2 *User Characteristics*

Commerce Bank customers who own a computer, tablet, or smartphone. These customers include people from many different backgrounds, and some may not be adept at using technology. Therefore, the application should be easy and intuitive to use. It must also be responsive and able to fit on a wide variety of screen sizes.

2.3 *Mandated Constraints*

The Commerce Online Banking Transaction Viewer will be a responsive web application built in a .Net framework or a similar modern web development environment. The database should be in a SQL server from 2012 or more recent. At least one CSS framework must also be used.

2.4 *Potential System Evolution*

In the future, the Commerce Online Banking Transaction Viewer could be linked to the real transaction data for the customers of Commerce Bank. Increased security measures for users of this application could also be implemented in the future, especially if this application would be used for real transaction data from real customers.

3 Nonfunctional Requirements

3.1 Usability Requirements

For the following usability requirements, user satisfaction surveys can be used to measure user satisfaction. A benchmark of three out of four, or 75%, would indicate an overall positive experience.

Tasks 1, 2, 3, and 5 will test the ease of learning and measure the results in terms of task time. It should not take the user longer than one minute to view their transaction history, longer than one minute to view notifications, longer than three minutes to use the notification rule editor, or longer than 1 minute to export transaction. If it takes any longer, then it is an indication that the website is difficult to learn.

Tasks 3 and 4 will test task efficiency and measure the results with a problem count. Problems can slow down the user considerably, thus measuring the task efficiency in terms of a problem count is a reasonable metric.

User Tasks

Task 1: View transaction history (See 4.1.1 for more details)

Task 2: View notifications (See 4.1.2)

Task 3: Add, edit, and delete notification rules (See 4.1.3)

Task 4: Login (See 4.1.4)

Task 5: Export transactions (See 4.1.6)

Req 1.1: At least 15 out of 20 users shall complete Task 1 within 1 minute

Req 1.2: At least 75% of users shall have a positive experience with viewing the transaction history

Req 2.1: At least 15 out of 20 users shall complete Task 2 within 1 minute

Req 3.1: At least 15 out of 20 users shall complete Task 3 within 3 minutes

Req 3.2: At least 75% of users shall have a positive experience with adding, editing, and deleting notification rules

Req 3.3: The user shall encounter no more than 1 medium problem in total

Req 4.1: A maximum of 1 out of 20 users shall encounter a critical problem

Req 5.1: The user shall complete Task 5 within 1 minute

3.2 Performance Requirements

When a user logs into the system, the first page of transactions should load within two seconds. Scrolling down the list of transactions should not result in any lag time. Using React to render only elements that change will result in faster load time than other frameworks. Furthermore, the system uptime needs to be at least 99%.

3.3 Security Requirements

Security is extremely important to Commerce Bank and its online users. User passwords will be hashed, and during login, the password will be masked. After the user logs in, the full account number will not be visible on the homepage/dashboard nor on the csv file when the user exports the transactions.

3.4 Documentation and Training

The web page will be easy to navigate without training. However, for notification rules, users can click a *Help* icon to explain how the notification triggers work and how to add, edit, or delete them. System documentation can be provided to stakeholders.

3.5 External Interface

3.5.1 User Interface

The user interface will be minimal and aesthetically pleasing using the Commerce Bank color scheme, easy to read, and responsive. The homepage displays a dashboard with clear buttons indicating interactive actions. The transaction history is in the center of the screen as a list. At the top right of the transaction list, is an Export button, which will export a csv file of the transaction history. At the top of the page, there is a notification window showing the most recent notifications. When the user clicks the notification settings icon, a new window opens where the user can add, edit, or delete notification rules. The interface will be intuitive, with 9 out of 10 users requiring no training to navigate the web page.

3.5.2 Software Interface

The system will use a React.js framework to pull information programmatically from the database and display HTML/CSS on the webpage. This applies to the operations defined by the uses cases in section 4.

4 System Features

4.1 Required Features

4.1.1 Use Case: View transaction history

Description

- As a user, I need to be able to view transaction history
- Transaction history should be part of the homepage/dashboard

Actor

Commerce Bank online user

Value = high

Cost = high

Risk = high

Basic Flow

1. Upon logging in, the user views the dashboard on the homepage, which displays transaction history
2. The system displays the transaction date, the balance at the time of the transactions, the type of transaction (deposit or withdrawal), the amount, and a description of the transaction
3. The user scrolls down to see all transactions.
4. The system dynamically loads older transactions as the user scrolls

4.1.2 Use Case: View notifications

Description

As a user, I need to be able to view notifications

Actor

Commerce Bank online user

Value = high

Cost = high

Risk = high

Basic Flow

1. On the homepage/dashboard, the user clicks a notification icon to open a list of notifications

2. The system displays the number of times each notification rule has been triggered over the past month and year

4.1.3 Use Case: Add, edit, and delete notification rules

Description

As a user, I need to be able to add, edit, and delete notification rules.

Actor

Commerce Bank online user

Value = high

Cost = high

Risk = high

Basic Flow

1. The user clicks on the notification settings icon to enter the notification settings screen
2. The system displays the current notification rules associated with the account
3. The user clicks on a notification rule to edit the rule
4. The system displays options to edit or delete the rule
5. The user edits or deletes the rule
6. The system prompts the user to confirm the edit or delete
7. The user confirms and returns to the notification

Alternate Flow

3. The user clicks the add icon to create a new notification rule
4. The system displays options to edit or delete the rule
5. The user edits or deletes the rule
6. The system prompts the user to confirm the edit or delete
7. The user confirms and returns to the notification

4.1.4 Use Case: Log in

Description

- As a user, I need to be able to log in
- As a user, I want my username to be remembered when I close the browser and return to the login page (Stretch Goal)

Actor

Commerce Bank online user

Value = high

Cost = low

Risk = medium

Basic Flow

1. Upon entering the website, the system prompts the user to login with a username and a password
2. The user enters a username and password
3. The system validates the username and password
4. The user accesses the account

Alternate Flow 1

4. The username and password cannot be validated
5. The system prompts the user to re-enter the username and password

Alternate Flow 2

1. Upon entering the website, the system prompts the user to login with a username and a password. The system auto-fills the username from the previous session if the user has allowed session remembering
2. The user enters a username and password
3. The system validates the username and password
4. The user accesses the account

4.1.5 Use Case: Easy-to-use, responsive user interface**Description**

As a user, the UI needs to be responsive and easy-to-use.

Actor

Commerce Bank online user

Value = medium

Cost = medium

Risk = low

Basic Flow

1. The user can see the contents of the homepage/dashboard on any screen size
2. The system automatically scales and positions the graphical elements dynamically based on the size of the window or screen
3. The user can view transactions, view or edit notifications, and export a csv with clear buttons indicating an interactive action is available

4.1.6 Use Case: Export transactions

Description

As a user, I need to be able to export transactions to a csv file to view my transactions in a spreadsheet or import them into another application.

Actor

Commerce Bank online user

Value = low

Cost = low

Risk = low

Basic Flow

4. On the homepage/dashboard, the user clicks a download icon
5. A prompt asks the user to confirm the action
6. The entire transaction history is downloaded to a csv file
7. A dialogue box tells the user that the cvs was successfully exported

Alternate Flow

4. A dialogue box tells the user that the cvs was not successfully exported and to try again