**Software Project Management Plan**

**Commerce Bank System**

03/07/2020

**Team Members**

Michelle Frost

Debbie Kirchner

Brian Roden

Linden Stirk

Issac Zeilinger

Document Control

**Change History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision** | **Change Date** | **Change By** | **Description of changes** |
| V1.0 | 02/27/2020 | Linden Stirk | Initial release |
| V1.1 | 03/01/2020 | Debbie Kirchner | Edited Section 1, 2.1, and 2.2 |
| V1.2 | 03/01/2020 | Brian Roden; Issac Zeilinger | Section 5.1; 5.2 |
| V1.3 | 03/02/2020 | Linden Stirk | Updated Section 4 |
| V1.4 | 03/07/2020 | Linden Stirk | Final release |

**Table of Contents**

[**1**](#_3znysh7) **OVERVIEW 4**

[**1.1**](#_49x2ik5) **Purpose and Scope 4**

[**1.2**](#_2p2csry) **Goals and Objectives 4**

[**1.3**](#_3dy6vkm) **Project Deliverables 5**

[**1.4**](#_1t3h5sf) **Assumptions and Constraints 5**

[**1.5**](#_4d34og8) **Schedule and Budget Summary 6**

[**1.6**](#_2s8eyo1) **Success Criteria 6**

[**1.7**](#_17dp8vu) **Definitions 6**

[**1.8**](#_3rdcrjn) **Evolution of the Project Plan 7**

[**2**](#_26in1rg) **STARTUP PLAN 7**

[**2.1**](#_lnxbz9) **Team Organization 7**

[**2.2**](#_35nkun2) **Project Communications 7**

[**2.3**](#_1ksv4uv) **Technical Process 7**

[**2.4**](#_44sinio) **Tools 8**

[**3**](#_2jxsxqh) **WORK PLAN 8**

[**3.1**](#_z337ya) **Activities and Tasks 8**

**[3.2](#_3j2qqm3)****Release Plan 8**

[**3.3**](#_1y810tw) **Iteration Plans 9**

**[3.4](#_4i7ojhp)****Budget 9**

[**4**](#_2xcytpi) **CONTROL PLAN 10**

[**4.1**](#_1ci93xb) **Monitoring and Control 10**

[**4.2**](#_3whwml4) **Project Measurements 10**

[**5**](#_2bn6wsx) **SUPPORTING PROCESS PLANS 11**

[**5.1**](#_qsh70q) **Risk Management Plan 11**

[**5.2**](#_3as4poj) **Configuration Management Plan 12**

[**5.3**](#_1pxezwc) **Verification and Validation Plan 12**

# Overview

## **Purpose and Scope**

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 export their transactions in a spreadsheet for uses outside of the application. This application will allow customers to view their transactions and receive alerts when unusual or potentially dangerous transactions are made.

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 afterwards. Non-functional requirements, system features, and use cases will be outlined later in the document.

The scope of the project is the requirements given by Commerce Bank. The application is only required to use a preset list of transactions, which we are allowed to modify to fit our needs. Users must be able to view a list of their transactions, but searching and filtering these transactions are not required. The team can decide which notification rules to make available for use in the application. Based on these rules, users can add, edit, or delete notifications triggers. The notifications should be accessible from inside the application, but do not need to be accessible from outside applications such as email or text messaging. Features that are beyond the scope of this project may be addressed with stretch goals, but only if the team completes the necessary product requirements first.

## **Goals and Objectives**

The goal is to create a web-based application that allows users to keep track of transaction history and receive notifications when specific triggers are activated, e.g., unusual transactions. The system needs to save the data to a database in order to create recurring reports. The application needs to be easy to use, responsive, and aesthetically pleasing. Furthermore, any client-side framework or libraries must be included in the application, not externally referenced.

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.

The main objectives of the Commerce Bank System are as follows:

1. Create a database to store transaction data and generate reports from the data.
2. Create a user interface that displays transaction history and notification triggers.
3. Create a means to display alerts on the application when a transaction triggers a notification rule in real-time.

## **Project Deliverables**

The following items will be delivered to the customer on or before 05/04/2020:

1. Source code for both the user interface and back-end portions of the web application
2. User’s Guide
3. System Administrators Manual
4. Technical Architecture
5. Test Plan
6. System Test Cases
7. Suite of regression tests

## **Assumptions and Constraints**

Assumptions:

1. Test data is accurate and sufficient.
2. Google Cloud hosting will be available.
3. Consulting will be provided by members of the Commerce Bank team.

Constraints:

1. The software must run on versions of Firefox, Google Chrome, and Internet Explorer for Windows 7-10.
2. The deliverable source code must be open source.
3. Must be in a .Net environment.
4. The software must be ready by 05/04/2020.

## **Schedule and Budget Summary**

Schedule Summary:

1. Technical prototype:

Start by 03/02/2020

Finished by 03/16/2020

2. Draft of Web Application:

Start by 03/16/2020

Finished by 04/26/2020

3. Testing of Application:

Start by 04/26/2020

Finished by 05/04/2020

## **Success Criteria**

* All features required by Commerce Bank plus at least two stretch goals are delivered by 05/04/2020.
* Unit tests covering at least ten percent of the code are completed and passed by 05/04/2020.
* The user interface is determined to be aesthetically pleasing and easy to use by human testers.

## **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.

**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 interact with the Commerce Bank System.

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

## **Evolution of the Project Plan**

Before the start of each iteration, the project plan will be updated to include a detailed schedule of the tasks and the owners and expected effort for each task for the upcoming iteration. Risks and risk mitigation will also be evaluated and discussed before the beginning of each iteration. Any severe risks will be dealt with and added to the project plan as soon as possible.

At the end of each iteration, the project plan will be updated to include the actual effort exerted for each task for the iteration.

# Startup Plan

## **Team Organization**

Project Manager: The project manager is responsible for creating the project plan (with input from those doing the work), managing risks, running the weekly team meeting, and creating team documents.

Technical Lead: The lead programmer is responsible for setting up the coding environments and guiding the programmers as they handle the coding and testing.

Programmers (3): Programmers are primarily responsible for coding and unit testing modules. They are also expected to take part in architecture planning and review meetings.

## **Project Communications**

In-person team meetings will be held at least once a week. The team will meet at the beginning of each iteration to plan the tasks for the upcoming iterations and update the project plan with details from the previous iteration. The team will also meet once during each iteration to discuss any problems that may arise during the iteration. The team will use the designated Slack channel and the Github repository to communicate between meetings and share documents related to the project.

## **Technical Process**

The team will use an iterative development methodology. Iterations span two weeks and cannot be extended. There are a total of five iterations. During each iteration, there is a milestone related to the state of the overall software program. These include the Hello World app, which is a simple test of the software connections from the server, api, and the website, the vertical prototype which tests the full-stack with sample data, the login page, which adds functionality for multiple users, and finally the user interface, which adds aesthetic formatting and functionality to the frontend.

## **Tools**

The following development tools and languages will be used by the team.

* Programming Languages – C#, React
* Backend – .NET Framework
* Database – Microsoft SQL Server
* Frontend – React, Sass, HTML/CSS
* Web hosting – Google Cloud
* Version Control – source code and written artifacts will be stored in a Github repository.
* Defect tracking – defects and issues will be tracked using Github.

# Work Plan

## **Activities and Tasks**

For each task, the following attributes will be recorded.

* Task name
* Task Description
* Owner
* Effort estimate
* Actual effort
* Planned start and stop dates
* Actual start and stop dates
* Dependencies among other tasks

## **Release Plan**

The following release plan lists the expected completion dates for iterations and milestones, as well as the final product delivery date.

02/17/2020 – Iteration 1 Begins

02/23/2020 – Project Charter Complete

02/24/2020 – Requirements Document Complete

03/02/2020 – Iteration 1 Closeout

Milestone 1: Hello World app

Iteration 2 Begins

03/06/2020 – Project Plan Complete

Milestone 2: Vertical Prototype Complete

03/16/2020 – Iteration 2 Closeout

Technical Prototype Complete

Milestone 3: Login page

Iteration 3 Begins

04/03/2020 – Architecture Document Complete

04/06/2020 – Iteration 3 Closeout

Milestone 4: UI Prototype

Iteration 4 Begins

04/20/2020 – Iteration 4 Closeout

Iteration 5 Begins

04/27/2020 – User and System Guide Complete  
05/04/2020 – Iteration 5 Closeout  
 Application Tests Complete  
 Product Released

05/08/2020 – Product Demonstrated to Customer

## **Iteration Plans**

The following is the iteration schedule. For a detailed breakdown of current and past iterations, see this spreadsheet: <https://docs.google.com/spreadsheets/d/1eKz609CmGscxZm_oEJyS6ATNjpntzbZu9PPzk51sBrI/edit#gid=0>

02/17/2020 – Iteration 1 Begins

03/02/2020 – Iteration 1 Closeout

Iteration 2 Begins

03/06/2020 – Iteration 2 Closeout

Iteration 3 Begins

04/06/2020 – Iteration 3 Closeout

Iteration 4 Begins

04/20/2020 – Iteration 4 Closeout

Iteration 5 Begins  
05/04/2020 – Iteration 5 Closeout

## **Budget**

The total estimated budget is as follows:

1 project manager/developer at 5 hours per week for 11 weeks:  
55 hours \* $50/hour = $2750

1 technical lead programmer at 5 hours per week for 11 weeks:  
55 hours \* $50/hour = $2750

3 software developers at 5 hours per week for 11 weeks:  
165 hours \* $40/hour = $6600

275 hours total, $12,100 total, average $44 per hour

# Control Plan

## **Monitoring and Control**

The following is a plan for tracking progress and controlling performance. Included here will be the approximate dates of iteration and technical reviews.

Biweekly – There will be a team meeting at least twice a week for group members to report the status of their tasks, and for the team to review overall progress and address potential problems.

03/02/2020 – Iteration 1 Review. A discussion on what went right and wrong and how to improve the next iteration.

03/06/2020 – Vertical Prototype Review. A discussion on the full stack and whether it is a viable solution to the problem.

03/16/2020 – Iteration 2 Review. Technical Prototype Review. A discussion on what went right and wrong and how to improve the next iteration.

04/06/2020 – Iteration 3 Review. A discussion on what went right and wrong and how to improve the next iteration.

04/20/2020 – Iteration 4 Review. A review of application tests and any fixes that need to be completed by the end of the next iteration.

05/04/2020 – Iteration 5 Review. Final product reviewed.

05/08/2020 – Executive Review. The project manager presents the current project status to project sponsor and senior executives.

## **Project Measurements**

|  |  |  |
| --- | --- | --- |
| **Phase** | **Measurement** | **Source** |
| Release Planning | Record effort estimates for product features | Project Manager |
| Iteration Planning | Record effort estimates for scheduled tasks  Update effort estimates for product features  Update estimated dates in release plan | Project Manager; Programmers |
| Iteration Closeout | Record actual effort for scheduled tasks and product features | Project Manager; Programmers |
| System Test | Record the rate at which errors are found. | Team |
| Project Closeout | Archive project performance data in process spreadsheets and upload to GitHub. | Project Manager |

# Supporting Process Plans

## **Risk Management Plan**

The following risks have been identified as the most likely to occur.

*managerial risks*

1. Relying too much on one person.
   1. creates a bottleneck for workflow
   2. risk leaving a project at a dead end if the employee needs to leave.
2. miss-communication
   1. high possibility of missing deadlines
   2. high chance of exacerbating unequal workload
   3. moderate chance of strain on team morale
   4. moderate chance of creating duplicate work
   5. slight possibility of causing inter-team grudges
3. project time estimation
   1. high possibility of causing risk one
   2. wastes reorganization time
   3. encourages social loafing
4. procrastination and crunch
   1. high chance of lowering quality
   2. high chance of increased time need for future dependencies
   3. high chance of increases chance for bugs

*Technical risk*

1. security vulnerabilities
   1. high chance of data not remaining encapsulated
   2. high chance of Authentication and Session Management abuse
   3. slight chance of SQL injection if stored procedures are not maintained
   4. passwords being vulnerable if not hashed
2. Version control failures
   1. high possibility of duplicate work
   2. high possibility of losing work
   3. moderate chance of overwriting others works
   4. slight possibility of missing a submission deadline
3. Specification Breakdown
   1. high chance of causing wasted time and resources
   2. moderate chance of causing more miss-communication

## **Configuration Management Plan**

Configuration management plans for this document and other baselined work products, including review procedures and change management procedures.

1. All work product documentation will be stored in a centralized Github repository, and resources in a google cloud environment.
2. Weekly 20-minute checking meetings will be held to try and ensure quick resolutions to miss-communications and answer Specification questions.
3. All project items will be subject to review as part of iteration control to determine if they should be baselined. After which they may only be changed with group agreement.
4. Changes to code should be done through Github to track changes.
5. Any member who wishes to change a baselined item should send the team a message on Slack. Any compromises or objections can be voiced before the changes are made or abandoned. If there is no response after two days, the change can be considered accepted and pushed through.

## **Verification and Validation Plan**

In order to ensure the success and quality of the final product, the team must follow the requirements documents and the criteria listed in the acceptance plan below. Being present at meetings and available on Slack is also important, as well as reaching iteration and milestone requirements. In the event that a milestone is missed, the team will evaluate how to split up the task to other members in order to avoid further delays.

## **Product Acceptance Plan**

The success of the project is described in part by the requirements document released by Commerce Bank titled, *UMKC College Project Requirements 02-03-2020*. The document lists necessary features and stretch goals of which two are required. In addition to this document, there are specific performance, security, and usability requirements that need to be addressed.

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%.

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.

Viewing transaction history and notifications, editing notification rules, and exporting transactions are measurable by task time and with user satisfaction surveys. 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. Notification rules and logging in to the system are measured by task efficiency and problem counts. For detailed information in regards to usability requirements, see page 6 of the Requirements Document available here: <https://docs.google.com/document/d/1QilyeYwJh1gkFxWYIZ91KTpUZSd3grOnAlZLi-eeKao/edit#heading=h.otlu06rkow58>

A working application adhering to these requirements, in addition to a presentation successfully demonstrating the product, should result in a successful project. The project sponsors ultimately decide on the quality of the product, and thus, the success of the project.