**Software Project Management Plan**

**Commerce Bank Project – Group 4**

3/8/2021

**Team Members**

Brian Bui

Connor Bensyl

Ruby Rios

Shelby Mohar

Thomas Tran

Document Control

**Change History**

|  |  |  |
| --- | --- | --- |
| **Revision** | **Change Date** | **Description of changes** |
| V1.0 | 3/8/2021 | Initial release |
| V2.0 | 4/3/2021 | Changed to reflect changes in tools |

**Document Storage**

This document is stored in the project’s Github repository at: <https://github.com/UMKC-CS451R-Spring-2021/semester-project-group-4-commerce>

**Document Owner**

Ruby Rios is responsible for developing and maintaining this document.**Table of Contents**

[**Overview**](#_heading=h.1dtajcby24vm) **4**

[Purpose and Scope](#_heading=h.1fob9te) 4

[Goals and Objectives](#_heading=h.3znysh7) 4

[Project Deliverables](#_heading=h.2et92p0) 4

[Assumptions and Constraints](#_heading=h.tyjcwt) 5

[Schedule and Budget Summary](#_heading=h.3dy6vkm) 5

[Success Criteria](#_heading=h.1t3h5sf) 6

[Definitions](#_heading=h.4d34og8) 6

[Evolution of the Project Plan](#_heading=h.2s8eyo1) 6

[**Startup Plan**](#_heading=h.17dp8vu) **7**

[Team Organization](#_heading=h.3rdcrjn) 7

[Project Communications](#_heading=h.26in1rg) 7

[Technical Process](#_heading=h.lnxbz9) 8

[Tools](#_heading=h.35nkun2) 8

[**Work Plan**](#_heading=h.1ksv4uv) **8**

[Activities and Tasks](#_heading=h.44sinio) 8

[Release Plan](#_heading=h.z337ya) 11

[Iteration Plans](#_heading=h.3j2qqm3) 12

[Budget](#_heading=h.1y810tw) 12

[**Control Plan**](#_heading=h.4i7ojhp) **13**

[Monitoring and Control](#_heading=h.2xcytpi) 13

[Project Measurements](#_heading=h.1ci93xb) 13

[**Supporting Process Plans**](#_heading=h.3whwml4) **13**

[Risk Management Plan](#_heading=h.2bn6wsx) 13

[Configuration Management Plan](#_heading=h.qsh70q) 14

[Verification and Validation Plan](#_heading=h.3as4poj) 15

[Product Acceptance Plan](#_heading=h.bq8z4bg7mbu2) 15

# 

# Overview

## Purpose and Scope

The purpose of this project is to create a web application that allows Commerce Bank customers to get updated on their current bank account activity based on their preferences for notifications. The application will help Commerce Bank customers that have access to the web application to be able to more easily obtain important financial information regarding their account balance.

The web application will have the following features/functionality:

* A login screen that allows the user to log into their Commerce Bank account
* A home screen that shows notifications through the web application about recent account activity
* A home screen that allows the user to manually add transactions
* Screens to set notifications based on user input
* Screens to allow the user to change notification preferences
* User guides and system documentation will be provided to project stakeholders.

## Goals and Objectives

Project goals:

1. Help Commerce Bank customers that have computer and internet access to obtain useful financial information about their bank account.
2. Create an accessible, user-friendly web application that customers want to use.

Project objectives:

1. Create a database that stores transaction information.
2. Create a database that stores login information.
3. Create a database that stores notification information.
4. Connect these databases and allow users to interact with this information using .NET.
5. Create an interface that Commerce Bank customers can use to input transactions and notification specifications.

## Project Deliverables

The following items will be delivered to the customer on or before 5/7/2021:

1. Source code for both the client and server portions of the system.
2. User’s Guide
3. System Administration Manual
4. Test Plan
5. System Test Cases

## Assumptions and Constraints

Assumptions:

1. Commerce Bank will host and maintain the product past the product release.
2. Transaction information and login information will be updated and integrated into Commerce systems by Commerce Bank.
3. Performing full security scans and adding security enhancements to the product will also be performed by Commerce Bank.

Constraints:

1. Any Commerce branding used in the application must adhere to style guidelines provided.
2. The product will be a web application (preferably with .NET)
3. The product will preferably use SQL Server (2012 or above) as the database engine.

## Schedule and Budget Summary

Schedule Summary:

02/26/2021 - Project Charter Complete  
02/28/2021 – Requirement Document Complete

03/01/2021 – Iteration 1 Complete  
03/07/2021 - Project Plan Complete  
03/15/2021 – Risk Management Report

03/15/2021 – Iteration 2 Complete

04/04/2021 - Architecture Document Complete

04/06/2021 - Iteration 3 Complete

04/19/2021 – Iteration 4 Complete

04/30/2021 - Test Plan Complete

05/03/2021 - Iteration 5 Complete  
05/03/2021 - User Guide and System Administration Manual Complete  
05/07/2021 - Product Released

Budget Summary:

1 project manager at 5 hours per week for 12 weeks:

60 hours \* $50/hr = $3000

4 software engineers at 5 hours per week each for 12 weeks:

240 hours \* $40/hr = $9600

300 hours total, $12600 total, average $42 per hour

## Success Criteria

* A functional product that meets at least the basic requirements provided by the Client shall be delivered on May 7.
* All team members working on this project rate their experience as at least “satisfactory”.

## Definitions

**Commerce Bank Project**- the product that is being described. This is the web application specified in this document.

**User** – the person or persons who will actually interact with the Commerce Bank Project.

**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.

**Client** – the person or organization for which the Commerce Bank Project web application is being built. For the product being described, this would be Commerce Bank.

**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.

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

**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.

**Developer** – the person or organization developing the system.

**Controls** – the individual elements of a user interface such as buttons and checkboxes.

## Evolution of the Project Plan

At the start of an iteration, the project plan will be updated by the Project Manager to include tasks that the team intends to complete during the iteration. At the end of the iteration, the project plan will be updated to include which tasks got completed and the actual effort for each completed task.

Possible risks will be evaluated at the start of each iteration. Severe risks will be analyzed and added to the project plan as soon as they materialize.

# Startup Plan

## Team Organization

Project Manager: The project manager is responsible for creating project documentation (with input from all team members), running the weekly meeting, and resolving technical disputes. Works with Testing and Integration lead in particular, but with the rest of the team on requirements analysis and feature implementation decisions. This role will be fulfilled by Ruby Rios.

Testing and Integration Lead: The Testing and Integration Lead is primarily responsible for writing test cases and ensuring the code meets testing goals. In addition, the Testing and Integration Lead is responsible for helping other team members integrate their portions of code into the larger product. This role will be fulfilled by Shelby Mohar.

Database Designer: The Database Designer is responsible for creating ER diagrams, implementing the necessary databases for the project, and, if required, managing other individuals in the development of databases. Co-responsible for system architecture with Web API Lead. This role will be fulfilled by Thomas Tran.

UI/UX Designer: The UI/UX Designer is responsible for creating wireframes, implementing front-end components, and, if required, managing other individuals in the development of front-end components. This role will be fulfilled by Brian Bui.

Web API Lead: The Web API Lead is responsible for developing tasks related to and coding back-end components, and managing other individuals in the development of back-end components. Co-responsible for system architecture with Database Designer. This role will be fulfilled by Connor Bensyl.

## Project Communications

Communication during this project will be mainly handled internally via Discord, and external communication with the client will be through email. Any communication through these channels directed towards you (either a sent email or an @ message on Discord) should be responded to within 24 hours. All finalized documentation will be posted to GitHub.

There will be bi-weekly stand up meetings hosted via Discord and led by the Project Manager. Extra meetings will be scheduled as necessary.

If/when issues arise, they will be reported to the lead individual in charge of that part of the project via Discord direct message when possible, or to the whole group. Issues will also be discussed during the weekly meetings, and larger issues will be put into GitHub for review.

## Technical Process

For this project, we are using an adaptive Agile work methodology, and we are implementing Scrum practices. We will use GitHub for version control and storing all relevant project items. Each individual working on coding products will maintain a Kanban board on the GitHub page with tasks to work on regarding their chosen expertise for this project. Scrum practices being implemented include a bi-weekly “standing” meetings to review the current status of the project, each individuals’ progress with their tasks, and the assignment of new tasks. The documentation of these action items will also go into a Discord channel, as to ensure everyone knows what current projects are being worked on. Each iteration will be 2 weeks, and following each iteration, the team will do a retrospective to determine what went well and what could have gone better, so that we can update our work processes accordingly.

## Tools

* Programming Language – C#, using a Blazor Server
* Front End Framework - Blazor Pages with Razor and Bootstrap
* Databases – Microsoft SQL Server (SSMS & SSCM). Implemented with a Hamachi VPN. Integrated with Dapper.
* Version Control – source code and written artifacts will be stored in a GitHub repository.
* Automated Testing – unit tests will be implemented with the xUnit testing framework.

# Work Plan

## Activities and Tasks

|  |  |
| --- | --- |
| Task Name | Database Setup |
| Description | Creation of a “localhost” server using Microsoft SQL Server. Creation of schema and tables using data provided and generated “dummy” data. |
| Owner | Thomas Tran |
| Effort (estimate) | 8 hours |
| Start/Stop date | 3/3/2021 - 3/12/2021 |
| Dependencies | N/A |

|  |  |
| --- | --- |
| Task Name | Database-Blazor Application Integration |
| Description | Connecting Microsoft SQL Server to the Blazor application using Dapper. Create a page and function that displays information from the database. |
| Owner | Thomas Tran |
| Effort (estimate) | 25 hours |
| Start/Stop date | 3/3/2021 - 4/20/2021 |
| Dependencies | Task “Database Setup”, Blazor Application initialized |

|  |  |
| --- | --- |
| Task Name | Database Security |
| Description | Creation of login credentials. Implement basic database securities (Roles, System privileges, Object privileges). Password security (Salting) and Connection security (hashing). |
| Owner | Thomas Tran |
| Effort (estimate) | 8 hours |
| Start/Stop date | 3/29/2021 - 5/1/2021 |
| Dependencies | Task “Database Setup” |

|  |  |
| --- | --- |
| Task Name | Front-end integration |
| Description | The front-end will sync with the Database and Web API. The user will input data which will be saved into the database. The user will be able to see data from the database on the UI. |
| Owner | Brian Bui |
| Effort (estimate) | 10 |
| Start/Stop date | 3/15/2021 - 5/7/2021 |
| Dependencies | Database, Web API |

|  |  |
| --- | --- |
| Task Name | UI |
| Description | The front-end will consist of the Login page, Home page, Settings page, and reports page. The login page will allow the user to login by inputting a username and password. The Home page will display the user’s most recent transactions, a notification tray, and navigation buttons to go to settings.  The settings page will allow the user to edit their account and notification preferences. |
| Owner | Brian Bui |
| Effort (estimate) | 40 |
| Start/Stop date | 2/26/2021 - 5/7/2021 |
| Dependencies | None |

|  |  |
| --- | --- |
| Task Name | Product Testing |
| Description | Unit tests will be created for the front-end, back-end, and database to test each segment individually for proper functionality. Integration tests will be created to test the interactive behavior between the front-end and back-end, and between the back-end and database. End-to-end (E2E) tests will be created to test the product’s workflow and integration of all components. |
| Owner | Shelby Mohar |
| Effort (estimate) | 40 hours |
| Start/Stop date | 3/1/2021 - 5/7/2021 |
| Dependencies | Testing is contingent upon the completion of the front-end, back-end, and database components. |

|  |  |
| --- | --- |
| Task Name | Login Screen |
| Description | Login screen logic in the back-end should integrate with database and UI. |
| Owner | Connor Bensyl |
| Effort(estimate) | 10 hours |
| Start/Stop date | 3/8/2021-5/7/2021 |
| Dependencies | None |

|  |  |
| --- | --- |
| Task Name | Web API |
| Description | Web API should fetch data from database and respond to UI requests. |
| Owner | Connor Bensyl |
| Effort(estimate) | 50 hours |
| Start/Stop date | 3/9/2021 - 5/7/2021 |
| Dependencies | Web API functionality is contingent on the back-end and front-end communicating properly. |

## Release Plan

Key Dates:

02/26/2021 - Project Charter Complete  
02/28/2021 – Requirement Document Complete  
03/07/2021 - Project Plan Complete  
03/15/2021 – Risk Management Report

03/26/2021 - Mid-Semester Presentation

04/04/2021 - Architecture Document Complete

04/30/2021 - Test Plan Complete  
05/03/2021 - User Guide and System Administration Manual Complete  
05/07/2021 - Product Released

Other Milestones:

* Milestone 1: Prototype Completed (Estimated to be completed March 16th)
* Milestone 2: Minimal Viable Product Completed (Estimated to be completed April 19th)
* Milestone 3: Documentation Completed (Estimated to be completed May 3rd)
* Milestone 4: Stretch Goals Completed (Estimated to be completed May 3rd)
* Milestone 5: At least 10% unit test coverage (Estimated to be completed May 3rd)

## Iteration Plans

Iteration Dates and Goals:

03/01/2021 – Iteration 1 Complete

* Complete Project Charter, Requirements Documentation, preliminary research for product

03/15/2021 – Iteration 2 Complete

* Complete Software Project Management Plan, Risk Assessment, establish a prototype to present to project client

04/06/2021 - Iteration 3 Complete

* Complete Architecture document, get approximately halfway through product development

04/19/2021 – Iteration 4 Complete

* Complete minimal viable product

05/03/2021 - Iteration 5 Complete

* Fix any bugs or issues with the product, complete stretch goals, create User Guide, System Administration Manual, and Test Plan

## Budget

Budget Estimate 1:

1 project manager at 5 hours per week for 12 weeks:

60 hours \* $50/hr = $3000

4 software engineers at 5 hours per week each for 12 weeks:

240 hours \* $40/hr = $9600

300 hours total, $12600 total, average $42 per hour

Total cost per iteration: $12600/5 = $2520 per iteration

Budget Estimate 2 (based on times given for tasks above):

1 project manager at 5 hours per week for 12 weeks:

60 hours \* $50/hr = $3000

1 Database Designer/Administrator for 41 hours:

41 hours \* $40/hr = $1640

1 UI/UX Designer for 40 hours:

40 hours \* $40/hr = $1600

1 Testing and Integration Engineer for 40 hours:

40 hours \* $40/hr = $1600

1 Backend Designer for 60 hours:

60 hours \* $50/hr = $3000

241 hours total, $10840 total, average $45 per hour

Total cost per iteration: $10840/5 = $2168 per iteration

# Control Plan

## Monitoring and Control

Bi-Weekly – Team meeting. Project participants report status, progress and potential problems. Review GitHub issues with the team.

3/26/2021 – Mid-Semester Presentation. We will use this opportunity to elicit feedback from our Client, and to do an internal review on product architecture.

5/7/2021 – Final Presentation. We will get an executive review of our project. The project manager, with help from the team, presents the current project status to project stakeholders.

## 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 | Project Manager |
| Iteration Closeout | Record actual effort for scheduled tasks  Record actual effort for product features | All team members |
| System Test | Record the rate at which errors are found. | Testing and Integration Lead, Project Manager |
| Project Closeout | Record team satisfaction and project retrospective  Archive other project documentation | Project Manager |
| Ongoing | Record defects found from integration testing. | All team members |

# Supporting Process Plans

## Risk Management Plan

Risk: Misunderstanding user requirements

Priority: High

Action to Minimize: Using project requirements sheet, keeping in constant contact with clients through the project, creating prototypes to present to the client to ensure expectations are met

Risk: First time working together as a team

Priority: High

Action to Minimize: Plan for slack time as team learns to work together, set ground rules and reasonable expectations, have multiple open communication channels, and plan regular check ins to try to avoid individuals from “going dark” or feeling like their voice is not being heard

Risk: Lack of experience as a team in developing web applications

Priority: High

Action to Minimize: Share helpful resources, such as YouTube tutorials and online articles, in Discord. Document coding processes for repeatability and to troubleshoot more effectively.

Risk: Lack of knowledge on Security/Accessibility requirements

Priority: Medium

Action to Minimize: Get more client feedback in regard to expectations in these areas

Risk: Underestimating amount of work

Priority: Low

Action to Minimize: Careful planning and scheduling with estimation techniques, regular check ins with all team members to ensure no one “goes dark”, if this does become a problem, contingency plan is to add more unpaid hours of work on to ensure completion

## Configuration Management Plan

1. All work products and documentation will be stored in the GitHub repository.
2. The system requirements, project plan, and source code will be baselined and under configuration control.
3. An item will be considered baselined after it has been both completed and formally reviewed by all team members.
4. Change history is required for all baselined items, but encouraged for all other work products and documentation.
5. Once a project item has been baselined, the change control procedure is:
   1. Anyone wanting to make a change to a baselined item either brings it up in a group meeting, sends an @everyone Discord notification in the group server, or emails the entire group. This message should include the change details, reasoning for change, expected impact, and timeline for integrating the change.
   2. If no response is given within 2 days with a reason the change should not be implemented as requested, it will be considered accepted and the change may be implemented. If anyone does object to a change, it will then be discussed at a team meeting. Everyone will be allowed to voice their opinions and the change will be put up to a democratic vote.

## Verification and Validation Plan

Project stakeholders will provide validation at the end of the project. Throughout the project, unit tests will be performed using xUnit. Further documentation on testing will be provided 4/30/2021 in the Test Plan. More information TBD.

## Product Acceptance Plan

Product acceptance criteria is detailed in the UMKC College Project Requirements document. (<https://umkc.instructure.com/courses/64472/files/3105423?module_item_id=907441>). To have an acceptable product, our team will meet the goals listed here, as well as 2 additional stretch goals, by May 7, 2021.