**Software Project Management Plan**

**Commerce Bank App**

10/10/2021

**Team Members**

Andrew Aeilts

Blake Simpson

Daanyaal Tariq

Jacob Anderson

Lila Shelton

Renny Soto

Document Control

**Change History**

| **Revision** | **Change Date** | **Description of changes** |
| --- | --- | --- |
| V1.0 | 10/10/2021 | Initial release |
|  |  |  |

[Note change history doesn’t have to be stored with the document. Most version control tools like Git keep track of change history automatically. The comments entered when checking in documents become the change history for the document. However, even if your version control tool supports change history, you may want to track a more detailed version here.]

**Document Storage**

This document is stored in the project’s Git repository at: https://github.com/Software-Engineering-Capstone/Commerce-Banking-App

**Document Owner**

The Red Eyes Black Dragons are responsible for developing and maintaining this document.**Table of Contents**

[**1**](#_heading=h.3znysh7) **OVERVIEW 4**

[**1.1**](#_heading=h.49x2ik5) **Purpose and Scope 4**

[**1.2**](#_heading=h.2p2csry) **Goals and Objectives 4**

[**1.3**](#_heading=h.3dy6vkm) **Project Deliverables 5**

[**1.4**](#_heading=h.1t3h5sf) **Assumptions and Constraints 5**

[**1.5**](#_heading=h.4d34og8) **Schedule and Budget Summary 6**

[**1.6**](#_heading=h.2s8eyo1) **Success Criteria 6**

[**1.7**](#_heading=h.17dp8vu) **Definitions 6**

[**1.8**](#_heading=h.3rdcrjn) **Evolution of the Project Plan 6**

[**2**](#_heading=h.26in1rg) **STARTUP PLAN 6**

[**2.1**](#_heading=h.lnxbz9) **Team Organization 6**

[**2.2**](#_heading=h.35nkun2) **Project Communications 7**

[**2.3**](#_heading=h.1ksv4uv) **Technical Process 7**

[**2.4**](#_heading=h.44sinio) **Tools 7**

[**3**](#_heading=h.2jxsxqh) **WORK PLAN 7**

[**3.1**](#_heading=h.z337ya) **Activities and Tasks 7**

[**3.2**](#_heading=h.3j2qqm3) **Release Plan 8**

[**3.3**](#_heading=h.1y810tw) **Iteration Plans 8**

[**3.4**](#_heading=h.4i7ojhp) **Budget 8**

[**4**](#_heading=h.2xcytpi) **CONTROL PLAN 8**

[**4.1**](#_heading=h.1ci93xb) **Monitoring and Control 8**

[**4.2**](#_heading=h.3whwml4) **Metrics Collection 8**

[**5**](#_heading=h.2bn6wsx) **SUPPORTING PROCESS PLANS 8**

[**5.1**](#_heading=h.qsh70q) **Risk Management Plan 8**

[**5.2**](#_heading=h.3as4poj) **Configuration Management Plan 8**

[**5.3**](#_heading=h.1pxezwc) **Verification and Validation Plan 9**

# **Overview**

## *Purpose and Scope*

This section provides an executive overview of the project. It explains why the project is being initiated and what can and cannot be expected from the project. It may also include any background or contextual information necessary for understanding the project.

The purpose for the project explains the problem or opportunity the project will address. The statement of purpose isn’t a statement of what you are doing (“we plan to automate billing”), but rather why you are doing it (“The purpose of this project is to streamline billing in order to save time, money and resources.”).

Project scope defines the boundaries of the project—what will and won’t be included in the project. Defining project scope helps set expectations regarding what can be expected from the project. The scope definition may also play a role in evaluating requests for changes or new features. Project plans and estimates are based on the scope definition. A request for a change that is outside the current scope of the project can’t be accepted without a change in project scope.

Purpose:

This document provides information on the full-stack Commerce Banking software application.

The problem is we are given the data of a customer's transaction history. They want us to create a web application that allows them to view and handle their data.

This project aims to pull that data in and display it to the user in order for them to read it in a clear and understandable way. The application also aims to notify the user when a new transaction appears in order to let the customer notice any abnormal account activity. OWASP ZAP security will be applied so we can safely secure our application from malicious attacks in order to protect customer’s information.

Scope:

Users should expect to login into their Commerce Bank account that will grant them the ability to access and track finances. When they login into their home page they will have a variety of options like: view their recent transactions, view and edit notifications, receive notifications by email, and export transaction summaries into a spreadsheet. Also, they’ll be able to add transactions and all transactions will be sorted by date.

## *Goals and Objectives*

Goals and objectives define expected project outcomes. Goals are broad and inspirational. Objectives are narrow and measurable.

Project goals generally relate project outcomes to business objectives (reduced cost, increased revenue, improved quality, etc).

A well-worded objective is SMART: Specific, Measurable, Attainable/Achievable, Realistic and Time-bound.

Project Goals

1. Create a Bug-Free Experience for the customer

2. Satisfy \*more than\* two Stretch Goals

3. Build the web application to handle mobile platforms

Project Objectives

1. Follow Commerce Bank styling

2. Create a safe Login page

3. Create a Homepage

4. Create a Notification system

5. Allow user to create specific notification rules

6. Allow user to export transaction to spreadsheet

## *Project Deliverables*

This section lists the outputs of the project that are delivered to the customer.

The following items will be delivered to the customer on or before 11/01/2021:

1. Demo showcasing project basics

2. Midterm Report

The following items will be delivered to the customer on or before 12/17/2021:

1. Final Project Report

2. Live Presentation and Product Demo for Client

3. Source Code for the Client

4. User’s Guide

5. System Guide

6. Team Evaluation

## *Assumptions and Constraints*

Assumptions are conditions, usually outside the control of the project team, that are taken for granted. Project plans (i.e. estimates) typically depend on certain assumptions being true. Assumptions that turn out to be false, may jeopardize project success. In order to reduce project risk, the project manager may elect to validate certain assumptions as part of the risk management process.

This is also a good place to document verbal promises or assurances given to you.

Constraints are limits or restrictions on freedom. Projects may have technical as well as non-technical constraints. Priorities for schedule and budget can impose non-technical constraints on a project. Restrictions on programming language or delivery platform are examples of technical constraints that limit design and implementation options.

Note, the following is not a reasonable assumption for inclusion in this section: “We assume that our group has the necessary skills and knowledge needed to complete the project.” This might be something you are taking for granted, but it is not something worth documenting in the project plan. The assumptions you want to list here are those that are outside your control. Once the development team is established, it is their responsibility to possess or develop the skills and knowledge needed to complete the project. If there is a concern that the existing team doesn’t have the skills and knowledge needed to complete the project successfully, add it as a risk and develop a plan for mitigating the risk.

Assumptions:

1. Customer Data will be given to use for implementation
2. The web application will only be used by UMKC and not the public

Constraints:

1. The project must be a full-stack web application
2. The project must be ready by 12/6/2021
3. The project must pull in given data from a database and be displayed

## *Schedule and Budget Summary*

The schedule summary shows start and end dates for high-level activities ending in major milestones or deliverables. Milestones are major events in the project life cycle that are used to measure progress.

A Gantt chart is an excellent tool for visualizing the start and stop dates of major scheduled activities.

The budget summary shows total project cost, possibly broken down into separate categories for such things as salaries, equipment, travel, overhead, etc.

schedule:

**Iteration 1:**

* start: 9/25/2021
  + **Project Plan Document**
    - due: 10/10/2021
* end: 10/10/2021

**Iteration 2**:

* start: 10/10/2021
  + **Architecture Document**:
    - start: 10/20/2021
    - due: 10/22/2021
  + **Create login page and transaction summary**
    - start: 10/10/2021
    - due: 10/25/2021
* end: 10/25/2021

**Iteration 3**:

* start: 10/27/2021
  + **Notification Functionality and pages**
    - start: 10/27/2021
    - end: 11/25/2021
* end: 11/25/2021

**Iteration 4**:

* start: 11/8/2021
  + **Test Plan**:
    - start: 11/12/2021
    - due: 11/15/2021
  + **Notification via email**
    - start: 11/8/2021
    - end: 11/20/2021
  + **OWASP security scan**
    - start: 11/8/2021
    - end: 11/22/2021
* end: 11/22/2021

**Iteration 5**:

* start: 11/22/2021
  + **Unit testing**
    - start: 11/22/2021
    - end: 12/6/2021
  + **User and System Guide:**
    - due: 11/29/2021
  + **Team Evaluation:**
    - start: 12/1/2021
    - due: 12/3/2021
* end: 12/6/2021

Budget Summary:

Since this is a University project, we estimate the total project cost by calculating how much it cost for each student and adding them up.

total cost = [ $313.90 (cost/credit hour) \* 3 (this classes total hours) ] \* 6(team size)

total cost = $941.70 \* 6

**total cost = $5646.00**

A big thing we have to budget is our **time**. All of us are full time students so finding a proper time to contribute to our project is crucial to the success of the application.

## *Success Criteria*

Success criteria spell out what has to happen before the project can be considered a success. Having explicit success criteria serves two purposes. First, during a project success criteria help to focus attention on what is important. Second, at the conclusion of a project (project closure) success criteria are used to assess whether or not the goals and objectives of the project have been achieved.

To be effective in both of these endeavors, success criteria must be defined in a way that is both quantifiable and verifiable.

For more advice on how to define the success criteria for a project, I recommend: *Success Criteria Breed Success*, by Karl Wiegers. It is available on the web.

In order for the application to be considered a success, these goals must be met:

* MUST be a web application
* A database must be utilized
* Unit Testing must be used and cover 10% of code
* User Experience standards must be met (easy to use, proper language, etc)
* Front-End development standards must be met (proper styling, etc)
* A Login Page must be used with proper user/password field requirements
* A notification system must be created
* A customer must be allowed to create new rules that trigger a notification
* User must be shown a transaction summary
* When a user logs into their account, they must be taken to a home page
* Our application must satisfy two stretch goals

## *Definitions*

This section should define potentially unfamiliar or ambiguous words, acronyms and abbreviations.

**Commerce Bank Application** – the product that is being described here; the software system specified in this document.

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

**Application** - software that performs specific tasks for an end-user

**Client** – the person or organization for which this Commerce Bank application is being built.

**Sprint** - A specific period of time where a development team works to complete distinct tasks, milestones, or deliverables.

## *Evolution of the Project Plan*

This section describes plans for updating the project plan throughout the project.

Before the start of an iteration, the project plan will be updated to include a schedule of detailed tasks for the upcoming iteration. At the conclusion of an iteration, our team’s estimated effort spreadsheet will be updated to include the actual effort for each completed task.

Risk mitigation efforts will be evaluated at the start of each iteration. When a risk is encountered in the middle of a sprint we will address it in our mid-iteration meetings. Severe risks will be analyzed and added to the project plan as soon as they materialize.

# **Startup Plan**

## *Team Organization*

This section explains project roles and the authorities and responsibilities associated with these roles. Lines of communication, authority and reporting relationships are often shown with an organization chart. If the development team is known, actual names can be associated with roles.

Project Manager: **Blake Simpson** - 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 providing monthly status reports to senior management.

Programmers (6): **Andrew Aeilts, Blake Simpson, Daanyaal Tariq, Jacob Anderson, Lila Shelton, Renny Soto** - Programmers are primarily responsible for coding and unit testing modules. They are also expected to take part in architecture planning and review meetings.

Build Coordinator: **Andrew Aeilts, Blake Simpson, Daanyaal Tariq, Jacob Anderson, Lila Shelton, Renny Soto** - The build coordinator is responsible for setting up, running and distributing the results of the nightly build.

## *Project Communications*

This section contains the project communications plan. The communications plan describes how information is gathered and distributed.

Our current communication plan is to meet every Tuesday and Saturday @ 11:30 to discuss tasks for a given sprint. Our meetings are mostly held on the discord app but sometimes we’ll have in-person meetings.

## *Technical Process*

This section describes the software development methodology or conventions the team agrees to live by. When following an organization standard process, this section will refer to the standard process and state any deviations that are planned for this project. In the absence of an organization standard process, this section will define planned phases, entry and exit criteria for each phase, major milestones, workflows, and other aspects of the proposed development process.

We will have 5 iterations to work on this project.

The 1st iteration will see a website created that will have very little actual functionality but will be uploaded to github as well as a database created that we can store our pseudo-data on. The major milestone is the website and database being created.

The 2nd iteration will see the creation of a login page and the transaction summary page. The database will also be linked to the website with the pseudo-data being pulled from the database and displayed on the transaction summary page. The login information will also be stored in the database and retrieved when signing in or registering. The major milestone is the login page and transaction summary being created.

The 3rd iteration will see the creation of the notification configuration page and notification viewer page. The user will be able to add a notification, specify the criteria for the notification, enable the notification, and trigger the notification with a new transaction or by pressing a button to trigger the notification if the criteria is met by an entry in the transaction history. The major milestone is the notification functionality being created.

The 4th iteration will focus on the required stretch goals. The user will receive notifications via email and we will apply the OWASP security scan to our code. The major milestone is the stretch goals being added.

The 5th iteration will focus on unit testing. The major milestone is the unit testing.

Our workflow will consist of sprint planning meetings on the first Tuesday of the sprint where we will create tasks as Issues in our Github repository and assign those issues to a developer. We will then meet on Saturdays and Tuesdays to discuss progress on the tasks and offer help for anyone who needs it or to work together on a task during these meetings.

## *Tools*

This section specifies the development tools the team will be using to perform their work.

* Programming Language – React, Javascript
* Version Control – source code and written artifacts will be stored in a Github repository.
* Defect tracking – defects and issues will be tracked using Github Issues.
* Build tools – local and main builds will be done using Javascript and React.
* Automated testing – unit tests will be implemented with the Jest testing framework.

# **Work Plan**

## *Activities and Tasks*

| Name | Description | Owner | Status |
| --- | --- | --- | --- |
| Create website | Create initial website and upload it to github | Danny Tariq | Completed |
| Set Up Database | Creating a working database for the web app to read and write to | Lila Shelton | In progress |
| Host the site | Find reasonable place for our site to be hosted to allow access | Renny Soto | In progress |
| Create home page | Create a home page the customer will see upon arrival to the web app | Blake Simpson | Pending |
| Create login page | Create a page that allows users to sign in | Renny Soto | Pending |
| Create sign up page | Create a page that allows users to sign up for a new account | Andrew Aeilts | Pending |
| Create notification rules page | Create a page allowing customers to set custom notification rules for transactions | Andrew Aeilts | Pending |
| Create transaction summary page | Create a page allowing customers to view and sort their transaction history | Jacob Anderson | Pending |

## *Release Plan*

After each iteration a demo version with updated/completed features will be available on release:

Iteration 1: Oct 10, 2021

Iteration 2: Oct 23, 2021

Iteration 3: Nov 8, 2021

Iteration 4: Nov 22, 2021

Iteration 5: Dec 6, 2021

## *Iteration Plans*

An iteration plan is a short-term fine-grained plan that shows the tasks to be completed during an iteration.

Iteration 1: Basic website created

Iteration 2: Linked database and page creation 1

Iteration 3: Page creation 2 and page functionality 1

Iteration 4: Page functionality 2

Iteration 5: Full app testing

## *Budget*

5 Software engineers at $25 an hour for 12 hours a week for 16 weeks puts the budget at $24,000.

# **Control Plan - Jacob**

## *Monitoring and Control*

Included in this section are plans and procedures for tracking progress and controlling performance. Included here will be the approximate dates of technical as well as managerial reviews. Typically each major milestone or project phase will end in a review.

For projects that don’t have a separate communication plan, this section may also include information on the timing and content of status reports and other project review documentation.

Bi-Weekly - Team meeting, Project participants report progress, and we discuss the state of the project and future steps.

9 / 18 / 2021 Project Charter review: Our group leader will submit our project charter to our sponsors.

10 / 4 / 2021 Sponsor meeting: The group will present the project in its current form to our sponsors and gather feedback on methods to improve upon it.

10 / 9 / 2021 First phase review: The team will submit a progress report to our sponsors as well as conduct a meeting on the progress made during the first phase.

10 / 10 /2021 Project Plan review: The group leader will submit our project plan document to our sponsors

10 / 22 / 2021 Architecture review: Our group leader will submit our project architecture document to our sponsors.

10 / 25 / 2021 Second phase review: The team will submit a progress report to our sponsors as well as conduct a meeting on the progress made in the second phase.

11 / 8 / 2021 third phase review: The team will submit a progress report to our sponsors as well as conduct a meeting on the progress made in the third phase.

11 / 15 / 2021 Test Plan review: Our group leader will submit our testplanto our sponsors.

11 / 22 / 2021 Fourth phase review: The team will submit a progress report to our sponsors as well as conduct a meeting on the progress made in the fourth phase.

12 / 6 / 2021 Fifth phase review: The team will submit a progress report to our sponsors as well as conduct a meeting on the progress made in the fifth phase.

## *Project Measurements*

Product and process measures support project management and estimation by analogy. At the beginning of a project, estimates are made for product size, project cost and delivery dates. During a project, progress is tracked with measures of actual effort, integrated lines of code and actual expenditures. Keeping track of estimates and actuals during a project helps to calibrate whatever technique is being used to make estimates. Storing project performance data on completed projects provides a rich source of data for estimating future projects.

*Example*

| **Phase** | **Measurement** | **Source** | **Estimate** |
| --- | --- | --- | --- |
| Release Planning | Record effort estimates for product features | Mgr | 1 Day |
| Iteration Planning | Record effort estimates for scheduled tasks  Update effort estimates for product features  Update estimated dates in release plan | Mgr | 2 days |
| Iteration Closeout | Record actual effort for scheduled tasks  Record actual effort for product features  Record LOC count for modules written | Mgr/Pgr | 2 days |
| System Test | Record the rate at which errors are found. | QA | 5 days |
| Ongoing | Record defects found from integration testing through the first year of release.  Assign each defect to one of the following categories: blocker, critical, major, minor or trivial. Keep track of the state of each defect: open, assigned, fixed, closed. | Mgr/Pgr/QA | 2 hours per day |

# **Supporting Process Plans - Blake**

## *Risk Management Plan*

Identify technical and managerial risks. Prioritize risks. Consider the probability of each risk turning into a problem and the likely consequences. For the highest priority risks, what actions will be taken to minimize the probability of the risk turning into a problem and the resulting consequences? What are the contingency plans for selected risks that do become a problem? Identify processes for monitoring risks and updating the risk management plan.

Our risk management plan will start with our sprint planning meetings where we will be discussing potential risks that we may encounter as a preemptive assessment of our tasks. Throughout the project we may encounter risks. It is a developer's responsibility to bring up those risks during our mid-iteration meetings so that we can assess the risk and discuss the best course of action for mitigation.

A technical risk could be that the base react library does not contain a function required for database communication. To mitigate the risk a developer could find and import a library that did. This risk would be of higher priority because of how critical that line of communication is to the website's functionality.

A managerial risk could be that a task is not being completed on time. This would be of higher priority due to a possible task dependency that could block the progression of the entire project. An action that could be taken to minimize the probability of this risk turning into a problem would be that the team reaches out to the assignee of the task and asks them if they will have time to complete before the deadline. Developers should let the team know during the beginning of the sprint their availability to work on tasks so that we can assign tasks accordingly and not have task dependency be a problem.

## *Configuration Management Plan*

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

1. All of the project code will be stored in a github repository running on a cloud server.
2. The naming convention for documents will reflect the name of the assignment in canvas.
3. All project (work products) items (documents, source code, test cases, program data, test data, etc) will be stored in the github repository but not all will be under change control (subject to formal change control procedures.) Only the source code will be baselined and under configuration control while the project description documents and other text documents will be stored in a shared google drive but will be subject to review.
4. Items that are subject to change control will be considered baselined after a group review at the end of the iteration during which they are created. Baselined here means that the product has undergone a formal review and can only be changed through a github pull request.
5. The change control procedure once a product is baselined is: (1) anyone wanting to make a change to a baselined item opens a pull request on github, assigns the rest of the group as reviewers and describes the change, reason for the change, expected impact, and timeline for integrating the change. (2) We will require 2 people to approve the pull request before merging the new updates into the master branch. If anyone does object to the change, the reason for objecting will be discussed at a meeting where everyone is invited to attend and voice their opinion or discussed in the pull request comment section.
6. Including a change history with all documents is encouraged but only required for baselined documents. The change history should be at the front of the work item and include: (1) the name of the person making the change, (2) brief description of what has changed, (3) reason for the change, and (4) the date the change was integrated.

## *Verification and Validation Plan*

The verification and validation plan defines what actions are being taken to assure the quality of the development process and resulting software products.

To ensure we are following the development process we are submitting semi-regular updates to our sponsors on the status of the project, as well as using Github for our version control.

To ensure the quality of the product we will be using unit testing to test various parts of the product when they are completed each iteration. We will be reviewing each other's code through a github pull request.

We are also going to be running the app through a security test and updating it based on the results.

## *Product Acceptance Plan*

The product acceptance plan defines what is acceptable in terms of product quality and product functionality. Acceptance criteria should be objective and measurable. Note product success is one aspect of project success. Teams wanting to establish a clear understanding of what will be considered acceptable project performance may want to define a more general plan for project success that includes quantitative goals for delivery date, cost, etc.

To be accepted the application must have a login page

To be accepted the application must have a transaction summary page

To be accepted the application must The application must possess a home page

To be accepted the application must send out notifications by email

To be accepted the application must undergo an OWASP security scan

To be accepted the application must use Commerce Banking styling

To be accepted the application must have user configurable notification rules

To be accepted the application must be completed by the deadline

To be accepted the application must be completed at or under budget.