Software Project Management Plan

<<Commerce Bank App>>

<<March 09,2020>>

Team Members

<<Tayler Singleton>>
<<Reed Heinrich>>
<<Samuel Feye>>
<<Connor Larson>>
<<Landon Volkmann>>

<<Publishing organization name>>
<<Publishing organization address>>

Document Control

Change History

Revision	Change Date	Description of changes
V1.0	03/08/2020	Initial release

[Note change history doesn't have to be stored with the document. Most version control tools like SVN 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 changing history, you want to track a more detailed version here.]

Document Storage
This document is stored in the project's SVN repository at:
http://company.com/svn/project-name/docs/spmp.doc.

Document Owner
Tayler Singleton and Reed Heinrich are responsible for developing and maintaining this document.

Table of Contents

1 Overview

1.1 Purpose and Scope

The purpose of this project is to create a web application where users can go to understand their transactions and to have a better sense of how, where, and when they are spending their money.

A user's guide will be provided. The user should not need technical assistance to Add/Delete/Edit notification rules.

1.2 Goals and Objectives

Project goals:

1. Use version control throughout the project.
2. Use the Commerce Bank color scheme will be used for styling of the application.

Project objectives:

1. Create an web application using a newer web development framework
2. Create an responsive web application
3. Must use a database to create recurring reports.
4. On the homepage the user must have the ability to pull/compare notification rules and export transactions.
5. The dashboard must include the number of times an notification has been triggered and the user must have the ability to hide notifications if the time trips to zero.
6. The login page must mask the password and have an login button
7. Users should be able to Add/Delete/Edit notifications rules without any technical assistance.

1.3 Project Deliverables

- Project Charter
- Requirements Documentation
- Project Plan
- Technical Prototype
- Video Presentation

- Architecture Documentation
- Source code for both the client and server portions of the system.
- User's Guide
- System Administrators Manual
- Test Plan
- Unit Test Cases

1.4 Assumptions and Constraints

Assumptions:

1. The client will be able to operate and understand the web application using React and Redux frontend and MariaDB Database as the backend instead of .Net framework and SQLServer database.

Constraints:

1. Web application built on a newer web development framework.
2. Application must be responsive
3. Client side framework/libraries must be included in the project

1.5 Schedule and Budget Summary

02/23/20 - Project Charter Approved

02/24/20 - Preliminary Requirements Complete

03/02/20 - Iteration #1 Complete

03/06/20- Preliminary Project Plan Complete

03/16/20 - Technical Prototype

03/16/20 - Iteration #2 Complete

04/02/20 - Architecture Document Written

04/06/20 - Iteration #3 Complete

04/20/20 - Iteration #4 Complete

04/27/20 - User Guide and System Administration Manual Complete

04/26/20 - System Test Plan

05/04/20 - Iteration #5 Complete

10 weeks

1 Program Manager - 50/hr - 4 hours a week x 10 weeks = $2000

1 Requirements Designer - 40/hr - 4 hours a week x 10 weeks = $1600

3 Engineers - 40/hr - 12 hours a week x 10 weeks = $4800

Project total = $8400

Project hours 200. AVG per hour $28.09

1.6 Success Criteria

- All requirements specified by Commerce Bank have been completed and two stretch goals have been successfully implemented into the project before May 4, 2020.

1.7 Definitions

Trigger - Tool should allow for configurable notification rules to be created to notify user when transactions fit into a set of criteria

Notification Rule – Determine which notifications (alerts) will be sent to the user.

JSX - JavaScript XML, which is used with REACT as an extension for XML-like code for the elements and components inside an REACT app.

1.8 Evolution of the Project Plan

Before the start of an iteration, the project plan will be updated to include the current tasks being given out and what tasks have been finished from the previous iteration.

Risk mitigation efforts will be evaluated at weekly standup meetings on Wednesday. If a severe risk appears, it will be analyzed and added to the project plan as soon as possible.

2 Startup Plan

2.1 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 providing monthly status reports to senior management.
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.
Software Tester:	Software Tester is responsible for identifying test conditions and creating test designs. These include test cases and the test plan.
Technical Writer:	Technical Writer is a professional writer that creates technical documentation that includes things like instruction manuals, user guide, and system administrators manual

2.2 Project Communications

Team standup meetings are to be weekly on Wednesday. The goal is to discuss what has been completed over the past week and what needs to be completed. Slack is the used for teammates to communicate about availability for impromptu standup meetings, tutorials giving new information about React and Redux, and current progress of tasks. GitKraken will be used to make new cards giving out roles and tasks for each team member.

2.3 Technical Process

Code will be formatted using the REACT/JSX style guide and the recommended Redux Style guide. One approval before a pull request is merged into the base branch.

2.4 Tools

- Programming Language – JavaScript
- Library - React
- State Management Tool - Redux
- Source Server Environment - Node.js
- Database - MariaDB Database
- GUI Git Client - GitKraken
- Source Code Editor - Visual Studio Code
- Version Control – source code and written artifacts will be stored in a umkc-cs-451-2020-spring/semester-project-group-3 repository.

- Automated testing – unit tests will be implemented with the Jest testing framework.

3 Work Plan

3.1 Activities and Tasks

We have an estimated effort spreadsheet here that describes how long we take for each task, and which iteration it will fall under. We plan this out accordingly so nothing is done without considering what might need to be done before completing that task. Using this iteration-based planning lets us get a good idea of who is doing what, and which tasks need to be done by a certain date or iteration before moving on with the project. However, this only currently covers tasks being done through iteration 2, so the rest are listed below.

Home page

- Task description: An easy to use and read home page is up and running, complete with notification reception and modification
- Owner: Everyone
- Effort estimate: 8
- Actual effort: TBD
- Planned start and stop dates: Begin by iteration 3, fully finish by iteration 5
- Actual start and stop dates: TBD
- Dependencies among other tasks: dependent on notification rule logic

Dashboard

- Task Description: User can view summary for triggered notification rules
- Owner: Everyone
- Effort estimate: 10
- Actual effort: TBD
- Planned start and stop dates: Begin by iteration 3 or 4, finish by iteration 5
- Actual start and stop dates: TBD
- Dependencies among other tasks: Dependent on notification rule logic

Transaction Summary

- Task Description: Can view, sort, and add transactions
- Owner: Everyone
- Effort estimate: 8
- Actual effort: TBD
- Planned start and stop dates: Begin by iteration 3 or 4, finish by iteration 5
- Actual start and stop dates: TBD
- Dependencies among other tasks: Connect to sample data

Triggers

- Task Description: Configurable notification rules.

- Owner: Everyone
- Effort estimate: 10
- Actual effort: TBD
- Planned start and stop dates: Begin by iteration 4, finish by iteration 5
- Actual start and stop dates: TBD
- Dependencies among other tasks: Dependent on notification logic, trigger logic

3.2 Release Plan

Prototype/Demo ready by: March 13th, 2020

Product release: May 4th, 2020

3.3 Iteration Plans

Iteration One - March 2nd (Completed):

- Research new technology and languages
- Begin forming iterative plan for project
- Start login page
- Paperwork & documentation

Iteration Two - March 16th (In progress):

- Tune up, finish login page (Commerce Bank color scheme)
- Develop testing plan
- Research DB, connecting front end and back end
- Set up dashboard
- Add notification feature
- Paperwork & Documentation

Iteration Three through Five (Three - April 6th; Four - April 20th; Five - May 4th):

- For iteration three and onwards, we as a team need to discuss our priorities for the early iteration and what all we can push back to the later ones. We also need to see what all we get done in iteration two before we start planning out the next few weeks, but listed below are the tasks we plan to get done in iterations three and onwards.
- Finish home page
- Fully connect front end and back in
- Trigger, notification logic
- Unit testing, full testing plan (10% code coverage)
- Paperwork & documentation where necessary

3.4 Budget

10 weeks

1 Program Manager - 50/hr - 4 hours a week x 10 weeks = $2000

1 Requirements Designer - 40/hr - 4 hours a week x 10 weeks = $1600

3 Engineers - 40/hr - 12 hours a week x 10 weeks = $4800

Project total = $8400

Project hours 200. AVG per hour $28.09

For rough weekly estimates, we could multiply by that week's factor. For example, if we are in week 4, we can assume we have spent roughly 8400 x 0.4 = $3360

4 Control Plan

4.1 Monitoring and Control

We have been conducting weekly standup meetings on Wednesdays right before class where our Project Manager has been checking in on our progress. We also have had a few impromptu meetings a day or two before something is due to make sure we have met all the requirements and that everything is up to our desired standard. By March 13th, we need to create a demo video to send to Commerce to show our progress, so we will likely have a meeting for that. We have done a nice job of checking in on each other's progress and staying up to date on relevant information, and this will surely continue to and through the project's release date, May 4th.

4.2 Project Measurements

Phase	Measurement	Source
Release Planning	Record estimated efforts for each task by iteration. Estimate release dates and budget.	Product Manager
Iteration Planning	Change or push back necessary tasks, assign new tasks based on previous iteration's success or shortcomings.	Product Manager
Iteration Closeout	Record actual effort for tasks that were completed, push back or modify tasks that were not completed or are still in progress.	Product Manager, Developers
System Test	Record the rate at which errors are found.	Tester
Project Closeout	Archive project performance data in process database. Calculate final actual efforts for the project. Reflect on successes and shortcomings.	Product Manager

Ongoing	The product will be "released" at the end of iteration five, but beyond that it is out of our hands. Perhaps we could reflect on how the project went and describe what all we would do if we actually released the project to the public and had to improve it in the future.	Project Manager, Developers, Tester

5 Supporting Process Plans

5.1 Risk Management Plan

Risk: Not turning in on time
Level: Low
Solution: Communication. Let each other know if we're struggling to meet a deadline, and we can either push the deadline back (depending on the task) or offer help.

Risk: Bugs
Level: Potentially high
Solution: Be thorough in our development and testing, ask for help when needed.

Risk: Unfamiliar with language
Level: Low
Solution: Research necessary frameworks, ask questions when you have them.

Risk: Sensitive information is collected within the app for the user - this information could be accessed by someone who shouldn't have access to it.
Level: High
Solution: Again, be thorough in testing and be sure that something like this couldn't happen.

Risk: Wrong information displayed (transaction amount wrong, wrong account balance, etc).
Level: High
Solution: Be sure that the transaction list is pulling from the right location and test for any faulty information being displayed.

Risk: A user logging in to the wrong account
Level: High
Solution: Be absolutely sure that each username and password is unique, and that it is not possible for someone to log in without both the right username and password.

5.2 Configuration Management Plan

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

1. All project versions will be available to be viewed on Github.
2. Any items or tasks subject to change will be reviewed upon a pull request from one of the developers of the project.
3. All documentation will be linked on our Github page.
4. Items that are subject to change control will be considered baselined after a group review at the end of the life cycle phase during which they are created. Baselined here means that the product has undergone a formal review and can only be changed through the prescribed change control procedures.
5. Major changes are always discussed in our Slack channel, but it is ultimately up to the developer's to consider if these changes are for the best for the project, and then it is approved upon pull request. Any issues are discussed as well, or marked as an issue on Github.

5.3 Verification and Validation Plan

All verification and validation are done through quality testing throughout the development process. This will be done through unit testing by using the Jest framework, and also through communication in our Slack channel by cross-checking each other's work. If something is wrong or there is an error in testing, this will be communicated and we will work to find a fix.

5.4 Product Acceptance Plan

- The product will be released on time on May 4th, and the product for each iteration will be turned in on time with no exceptions.
- Each documentation such as the project plan will be turned in on time.
- The final product will pass all unit tests with 10% code coverage.
- The product may go over budget (group members work overtime), but this will be limited as much as possible through teamwork and support.