**Software Project Management Plan**

**Commerce Bank Notifications**

3/7/2020

**Team Members**

Kory Overbay

Matthew Hopkins

Jimin Choi

Connor Marchand

Document Control

**Change History**

|  |  |  |
| --- | --- | --- |
| **Revision** | **Change Date** | **Description of changes** |
| V1.0 | 3/7/20 | Initial release |
|  |  |  |

**Document Storage**

This document is stored in the project’s SVN repository at: https://github.com/umkc-cs-451-2020-spring/semester-project-group-2

**Document Owner**

Matthew Hopkins is 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** 7

[**1.8**](#_heading=h.3rdcrjn) **Evolution of the Project Plan** 7

[**2**](#_heading=h.26in1rg) **STARTUP PLAN** 8

[**2.1**](#_heading=h.lnxbz9) **Team Organization** 8

[**2.2**](#_heading=h.35nkun2) **Project Communications** 8

[**2.3**](#_heading=h.1ksv4uv) **Technical Process** 8

[**2.4**](#_heading=h.44sinio) **Tools** 8

[**3**](#_heading=h.2jxsxqh) **WORK PLAN** 9

[**3.1**](#_heading=h.z337ya) **Activities and Tasks** 9

**[3.2](#_heading=h.3j2qqm3)****Release Plan** 10

[**4**](#_heading=h.2xcytpi) **CONTROL PLAN** 10

[**4.1**](#_heading=h.1ci93xb) **Monitoring and Control** 10

[**4.2**](#_heading=h.3whwml4) **Metrics Collection** 11

[**5**](#_heading=h.2bn6wsx) **SUPPORTING PROCESS PLANS** 12

[**5.1**](#_heading=h.qsh70q) **Risk Management Plan** 12

[**5.2**](#_heading=h.3as4poj) **Configuration Management Plan** 12

[**5.3**](#_heading=h.1pxezwc) **Verification and Validation Plan** 13 **[5.4](#_heading=h.1pxezwc)** [**Product Acceptance Plan**](#_heading=h.1pxezwc) 13

# Overview

## Purpose and Scope

This document defines the Software Management Plan for the Notifications web application being developed for Commerce Bank. This document aims to show the the clients and stakeholders our plan for startup, work, control and support

This document was made in mind with an existing project charter and software requirement specifications that includes project issues like schedule, cost, methodology, and the existence of other deliverables such as a testing plan.

The Commerce Bank Notifications web application is a tool for Commerce Bank customers to set up and receive notifications based upon rules they set for their transactions, namely potential fraudulent transactions.

The website application will (1) allow a user to securely log into their account, (2) see all of their notifications for triggered transactions and (3) be able to control how their notifications are set. (4) be able to see their transactions history. Time permitting our team will focus on the aesthetics and responsiveness of the website. The website will be developed with Firefox, Chrome, Safari, or Edge in mind, and will not guarantee support for Internet Explorer, NetScape, or Opera.

## Goals and Objectives

The overall goal is to allow Commerce Bank Account Holders to view their account in an easy to read format and give them the ability to make sure their account is safe.

The five main goals of this project are

1. Implement notifications when certain events happen to keep the account holder informed on possible suspicious activity
2. Show a history of all transactions for a user
3. User interface is arranged in a simple and intuitive manner aligned with Commerce’s styling.
4. Achieve 10% or more code test coverage
5. Effectively use source control throughout the project

## Project Deliverables

The following items will be delivered to the customer on or before 5/4/2020:

1. Source code for both the client and server portions of the system.
2. User’s Guide
3. System Administrators Manual
4. Test Plan
5. System test Cases
6. Suite of regression tests
7. Data conversion program for migrating existing data to new database format.

## Assumptions and Constraints

Assumptions:

1. The website application is being developed as a proof of concept
2. Users of the website are not maliciously attacking security vulnerabilities
3. Our website will not notify any government agencies of suspicious transactions
4. Our application does not need to follow Commerce Bank’s standard operating procedures on overdrafting or check clearing

Constraints:

1. Users can log on to the website through any modern day browser or smartphone
2. The styling of the application should look similar to other Commerce websites
3. Since we are using MySQL instead of Sql Server 2012, we understand that assistance from the Commerce team will be limited
4. Approval from Commerce Bank will be required to make this an official application.

## Schedule and Budget Summary

Schedule:

|  |  |  |  |
| --- | --- | --- | --- |
| Activity | Start Date | End Date | Deliverable |
| Iteration 1 | February 17th 2020 | March 2nd 2020 | Resource Planning  System Requirements  Minimal Functionality |
| Iteration 2 | March 2nd 2020 | March 16 2020 | Major Functionality |
| Iteration 3 | March 16th 2020 | April 6th 2020 | Alpha Application |
| Iteration 4 | April 6th 2020 | April 20th 2020 | Beta Application  Stretch Goals |
| Iteration 5 | April 20th 2020 | May 4th 2020 | Final release |

Budget:

In reality our budget is $0 however if our team was was working for Commerce we would estimate the following:

|  |  |
| --- | --- |
| Item | Cost |
| Salary | $120,000 project development |
| Server space | $10,000 per year |
| Maintenance | $20,000 per year |

## Success Criteria

* All high-priority use cases in the requirements specification are delivered before May 15.
* All project documentation is completed
* The Commerce Bank team is satisfied with our project
* Risk is minimized during the software development process

## Definitions

**Commerce Bank Notification Web-Application** - The product being described within this document

**Angular** - Javascript Framework for web development. Used to create the user interface of this application

**Spring** (Spring Boot) - Java Framework used to create web microservices. Used for back end code of the application

**MySQL** - Database Management System. Used to store data for the application

**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.

**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.

## Evolution of the Project Plan

Before the start of an iteration, the project plan will be updated to include

schedule of detailed tasks for the upcoming iteration. At the conclusion of an

iteration, the project plan will be updated to include the actual effort for each

completed task. Move tasks that don’t get completed to the product backlog at the

end of an iteration.

Risk mitigation efforts 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: Kory Overbay

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 (2): Jimin Choi and Matthew Hopkins

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

DBA: Connor Marchland

The Database Administrator will create, design, and maintain the database. The DBA will integrate the database with the web application.

## Project Communications

Information is gathered through communication with the team using Slack and Discord calls. The meetings are where most of the information that is gathered is analyzed and distributed to each team member.

## Technical Process

The software development process that our team uses is the Agile methodology. We use two week sprints to complete a variety of different stories, or features, that are assigned to each team member and are expected to be completed by the end of the sprint.

## Tools

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

* Programming Language – Java, SQL, and Angular Javascript
* Version Control – Git/Github
* Defect tracking – defects and issues will be tracked using Github.
* Build tools – local IDE to build.
* Automated testing – unit tests will be implemented with the JUnit testing framework.

# Work Plan

## Activities and Tasks

Activities:

* Design page mockups
  + Iterations 1 and 2
* Determine technology needs
* Research and learn new languages and environments
  + Iteration 1 and 2
* Environment Setup
  + Iteration 1
* Evaluate designs with team
* Testing
* Bug Fixes
* Iteration planning

Tasks:

* Setup Springboot application
  + Iteration 1
* Create login page in Angular
  + Iteration 1
* Create transactions page in Angular
  + Iteration 1
* Define Schema/ER diagram
  + Iteration 1
* Create DDL and set up SQL Server
  + Iteration 1
* Create Notifications popup
  + Iteration 2
* Create Dashboard page
  + Iteration 2
* Send query information from front end to backend
  + Iteration 2
  + Dependent on Springboot server setup and connection to MySQL server
* Send received query information from backend to MySQL
* Set up live MySQL Server for development
  + Iteration 2
* Learn how to set up project on server
  + Iteration 2
* Configure Angular proxy settings to hit Springboot server
  + Iteration 2
  + Dependent on the setup of the Springboot and mysql servers
* Write code to load transactions table with data from MySQL Server
* Iteration 2
* Dependent on Springboot application and MySQL Setup
  + Implement basic login functionality
* Iteration 2
  + Dependent on Springboot application and MySQL Setup

## Release Plan

3/13 Initial prototype done

Minimal functionality, shows the expected page layouts

* 3/27 Second prototype done
  + Most required functionality is working.
* 4/17 Alpha application
  + All required functionality is accounted for and is being tested
* 4/24 Beta Application
  + Stretch goals completed and being tested with required functionality
* 5/1 Final Release

# Control Plan

## Monitoring and Control

Include in this section 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.

*Partial Example*

Weekly – Team meeting. Project participants report status, progress and potential problems.

Bi-weekly - Iteration retrospective and planning

2/21/2020 - Evaluate project charter

2/28/2020 - Create Baseline for requirements

3/7/2020 - Review Project plan

3/9/2020 - Analyze needs for initial prototype

3/14/2020 - Review Risk Management

4/1/2020 - Review Architecture document

4/24/2020 - Review Test plan

4/24/2020 - Review User and system guide

## 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** |
| Release Planning | Record effort estimates for product features | Mgr |
| Iteration Planning | Record effort estimates for scheduled tasks  Update effort estimates for product features  Update estimated dates in release plan | Mgr |
| Iteration Closeout | Record actual effort for scheduled tasks  Record actual effort for product features  Record LOC count for modules written | Mgr/Pgr |
| System Test | Record the rate at which errors are found. | QA |
| Project Closeout | Archive project performance data in process database. (See process database definition for a list of measures to record.) | Mgr |
| Ongoing | Record defects found from integration testing through 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 |

# Supporting Process Plans

## Risk Management Plan

|  |  |  |  |
| --- | --- | --- | --- |
|  | Likelihood  Low/ Med/High | Impact  Low/Med/High | Mitigation Strategies |
| Inexperience | Med | Med | Learn necessary techniques for project projects and communicate with the team. |
| Losing Team Members | Low | High | Someone else on the team can perform their duties. |
| Resource Allocation  (hosting) | Med | Med | Spend time researching hosting techniques and practices |
| Web Security | High | High | encrypting usernames and passwords on the database. Don’t store sensitive information as plain text. |
| Time Constraints | High | High | Team staying on project plan and communicating when time is constrained for other responsibilities |

## Configuration Management Plan

1. All work products will be stored in the github repository in the classroom’s organization
2. All project (work products) items (documents, source code, test cases, program data, test data, etc) will be stored in the repository.
3. 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 if agreed upon by the team.
4. The change control procedure once a product is baselined is: (1) anyone wanting to make a change to a baselined item sends a message to the rest of the team describing the change, reason for the change, expected impact, and timeline for integrating the change. (2) if no one responds to the group within a day with a reason for why the change request shouldn't be permitted, it will be considered accepted and the person proposing the change may proceed with the change. 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 on whether or not the change should be allowed.

## Verification and Validation Plan

Verification and validation of task completion will be performed by the team at the end of iterations.

* Checks for functional completeness
* Asserts test cases were made and ran

## Product Acceptance Plan

The team will determine the product being Acceptable when all the required functionality of the project is working, the defined goals and dates are met, and when the project’s documents and code are clean, clear, and definitive.