**Software Project Management Plan**

**Commerce Project**

03/03/2020

**Team Members**

Dalton Gilmore

Tarsus Arciga

Ekrem Kalabak

Long Dang

Alexa Summers

Document Control

**Change History**

|  |  |  |
| --- | --- | --- |
| **Revision** | **Change Date** | **Description of changes** |
| V1.0 | 03/03/2020 | Initial release |
|  |  |  |

**Document Storage**

This document is stored in the project’s repository at: <https://github.com/umkc-cs-451-2020-spring/Transaction-Trigger-Tracker>

**Document Owner**

Alexa Summers 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 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 Commerce Bank web application will provide clients of the bank access to all of their banking needs right in one place, at any time day or night. This application will be available to both computer and mobile users. Users will be able to log into their account, view past transactions, and set triggers for transaction notifications. Triggers will include things such as out-of-state purchases, purchases outside of a certain time frame, and purchases over a set amount. Past triggers will also be viewable from the dashboard, and are editable.

There will be a user guide provided. No further training will be given.

## *Goals and Objectives*

Project goals:

1. Create a web application for Commerce bank customers.
2. Allow customers to customize their preferences pertaining to their transactions.
3. Application should be responsive.
4. Elements should be clearly defined and easy to navigate.

Project objectives:

1. Create a database containing information on accounts, triggers and preferences.
2. Create web application that allows customers of Commerce to access banking needs any time of day.
3. Create an interface that allows users to log into their account and navigate their dashboard with previously set preferences with ease.
4. Allow users to set triggers pertaining to dollar amount, time frame, and location of purchases.

## *Project Deliverables*

The following items will be delivered to the customer on or before 13/05/2020:

1. Source code for both the web application and database portions of the system.
2. User’s Guide
3. System Guide
4. Test Plan
5. Test cases

## *Assumptions and Constraints*

Assumptions:

1. The user will not need to create any custom trigger types outside of location, monetary value, and timeframe.
2. Transaction data will have columns representing triggers.
3. The user will only need to access the application from a web browser on a computer or mobile device.

Constraints:

1. Final solution should not rely on third-party licensed software beyond the operating system—creates distribution flexibility and keeps the cost low.
2. All documentation and the final solution must be submitted by 13/05/2020.

## *Schedule and Budget Summary*

02/23/2020 - Project Charter Approved  
02/29/2020 - Preliminary Requirements Complete  
03/06/2020 - Preliminary Project Plan Complete  
03/02/2020 - Iteration #1 Complete  
03/16/2020 - Iteration #2 Complete  
04/03/2020 - Architecture Complete  
04/07/2020 - Iteration #3 Complete  
04/20/2020 - Iteration #4 Complete  
04/26/2020 - System Test Complete  
04/27/2020 - User Guide and System Administration Manual Complete  
05/04/2020 - Iteration #5 Complete  
05/04/2020 - Product Released

Budget Summary:

Salary: $0.00

Equipment: $0.00

Travel: $0.00

Overhead: $0.00

Total budget: $0.00

## *Success Criteria*

* A user can successfully log into their account
* A user can be prompted to re-enter login information if incorrect
* A user can navigate to the dashboard to view all their options
* A user can create triggers for various purposes pertaining to transactions
* A user can access past transactions and create new transactions
* Total project cost does not exceed $0.00
* All deliverables and the final product is submitted before 13/05/2020.

## *Definitions*

**Client**—Customers of Commerce Bank whom are projected to utilize the application.

**Fraudulent activity** – Activity pertaining to one of the following: out-of-state transactions, out-of-timeframe transactions, and out-of-set-budget transactions.

## *Evolution of the Project Plan*

The project plan will begin as a broad overview until we progress to further iterations and have more specific things to document. While we are still getting things set up and learned on our part, our plans are up in the air past the immediate event horizon.

Risks will be evaluated on a continuous basis, and added to the project plan as needed. The schedule will begin broad, and then narrow down to that specific iteration as we get into further iterations. If more goals and objectives become apparent, or we begin to make any other assumptions, we will update those as well on a continuous basis.

# **Startup Plan**

## *Team Organization*

Project Manager (Alexa Summers): The project manager is responsible for the completion of paperwork, monitoring risks, and keeping everything running according to schedule. The PM might also assist with certain programming implementations.

Programmers (4): (Dalton Gilmore, Tarsus Arciga, Ekrem Kalabak, Long Dang): The programmers are responsible for developing and testing code, as well as partaking in the documentation.

## *Project Communications*

Documentation is initially stored in Google Docs so everyone can view and edit, and the final copies are pushed to GitHub. Code is stored in GitHub as well—there is a final repository, and two branches being pushed to for initial commits until programs are finalized to ensure no one else’s code gets overwritten.

## *Technical Process*

We are using the Kanban process, which helps assign and output deliverables in a timely manner without burning out the programmers. No deviations from this process are planned at this time.

## *Tools*

* Database || MariaDB Instance
* Frontend host & database connection || Spring Boot Application
* Reactive UI || Angular Application

# **Work Plan**

## *Activities and Tasks*

Activities and tasks located in GitHub. We are using an adaptive planning method, so tasks planned further in advance have a greater amount of abstraction when it comes to their completion date and exactly what they include. The near-term tasks are outlined fully in our GitHub and in our iteration plans. Look there for this iteration’s tasks.

## *Release Plan*

Project Plan release: March 8th, 2020  
Iteration 2 closeout: March 16th,   
Risk management report release: March 16th, 2020  
Technical Prototype release: March 16th, 2020  
Architecture documentation release: April 3th, 2020  
Iteration 3 closeout: April 7th, 2020  
Iteration 4 closeout: April 20th, 2020  
Test plan: April 26th, 2020  
User & System guide: April 27th, 2020  
Final Release: May 1st, 2020  
Iteration 5 closeout: May 4th, 2020

## *Iteration Plans*

Iteration plans are located in the GitHub. Future iterations will be based on the amount of work accomplished in past iterations, so those will be completed at a later date.

## *Budget*



# **Control Plan**

## *Monitoring and Control*

Weekly – Team meeting. Determine what needs to be done next, what is accomplished, and any potential risks or problems in upcoming assignments.

3/15/2020 – Risk management review. Making sure all potential risks are accounted for and make sure none have grown since discovered.

3/16/2020 – Iteration 2 closeout.

4/7/2020 – Iteration 3 closeout.

4/12/2020 – Software review. Make sure all software is working properly and all platforms are connected to each other correctly.

4/20/2020 – Iteration 4 closeout.

4/29/2020 – Product review. Review all finishing touches and make sure the final product is ready for release.

5/4/2020 – Iteration 5 closeout.

## *Project Measurements*

|  |  |  |
| --- | --- | --- |
| **Phase** | **Measurement** | **Source** |
| Release Planning | Estimation on how long features will take to implement and debug. | PM |
| Iteration Planning | Estimation on amount of effort for new tasks, tasks from previous iterations that were not completed, and testing. | PM |
| Iteration Closeout | Relate estimation to actual effort for new tasks from that iteration, tasks from previous iterations that got moved to the new iteration, and tasks from current iteration that were not completed and need to be moved. | PM/Coder |
| System Test | Determine where errors exist, and how well the product works. | PM/Coder(QA) |
| Project Closeout | Performance of project work assessed in accordance to the work completed. | PM |
| Ongoing | Keep track of issues in code and fix where necessary. Categorize all issues into different subtypes. | PM/Coder(QA) |

# **Supporting Process Plans**

## *Risk Management Plan*

The monitoring and managing of risk management involves constantly referring back to the potential risks and making sure they either still exist, or have not changed in magnitude. To better determine what needs the most attention, a list of the top ten risks should be kept at all times. Our biggest risk at this point in time is that our programmers are still trying to learn everything they need to know, and this might put us behind our initially planned schedule. Either things will not get done, or they might get done significantly slower than we intended. To combat this, we are all participating in various tutorials to practice the knowledge before trying to implement it within this project right off the bat, which could be really overwhelming to the programmers.

## *Configuration Management Plan*

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

*Partial Example*

1. All documentations will be stored in the project GitHub.
2. All documentations will be named their title, and inside will have information regarding the version.
3. All documentations will be stored with their prior versions.
4. Should something need to be changed, the member wishing to implement the change will reach out to the rest of the group and state the reason why it needs to be changed, how it might impact the rest of the project if we do or do not change it, and the risks involved with changing. This is the only way a member might go about changing something already written.
5. All changes will be documented in the GitHub, and information about versions will be kept inside the document.
6. Risks will be reassessed at the beginning and end of every change to make sure a change does not become detrimental.

## *Verification and Validation Plan*

The Verification and Validation plan will be outlined in our documentation for every iteration. This is to ensure that as we work on new tasks, the verification and validation become tailored to what will be the most efficient for that specific set of tasks. Since we are only on the second iteration and our adaptive planning method only plans in detail for the nearest tasks, our verification and validation plans are still somewhat in the air. For this current iteration, the verification and validation included making sure the code ran for ourselves, making sure the code ran for everyone else, making minor changes and building often, and then once things are connected together, we would rebuild and make sure everyone could run again. This ensures that the quality of development is high because it gives everyone in the group a chance to look at the code and make suggestions, and it ensures the resulting products are quality based on the functionality across multiple machines.

## *Product Acceptance Plan*

* Keep budget and cost to $0.00.
* All members of the team have contributions recognized.
* Planned schedule is followed with minimal deviations.
* Have all deliverables turned in by May 1st, 2020.
* 90% of user feedback should state the application is easy to use.
* Triggers should pass all functionality tests.
* Login screen should pass all functionality tests.
* Application should be responsive.