Software Project Management Plan

Commerce Bank Web Application

March 8th, 2021

Team Members

Zach Gharst

William Keke

Benaiah Kilen

Atticus Parris

Andrew Poitras

Document Control

Change History

Revision	Change Date	Description of changes
V1.0	3/8/2021	Initial release

Document Storage

This document is stored in the project's Git repository at:
https://github.com/UMKC-CS451R-Spring-2021/semester-project-group-3-commerce.

Document Owner

William Keke is responsible for developing and maintaining this document.

Table of Contents

1 Overview 4
1.1 Purpose and Scope 4
1.2 Goals and Objectives 4
1.3 Project Deliverables 5
1.5 Schedule 6
1.6 Success Criteria 7
1.7 Definitions 7
1.8 Evolution of the Project Plan 8
2 Startup Plan 9
2.1 Team Organization 9
2.2 Project Communications 10
2.3 Technical Process 10
2.4 Tools 10
3 Work Plan 11
3.1 Activities and Tasks 11
3.2 Iteration Plans 12
3.3 Budget 13
4 Control Plan 14
4.1 Monitoring and Control 14
4.2 Project Measurements 14
5 Supporting Process Plans 15
5.1 Risk Management Plan 15
5.2 Configuration Management Plan 15
5.3 Verification and Validation Plan 15

1 Overview

1.1 Purpose and Scope

The project described within this document is a Web Application designed with the specific goal of creating an online banking alternative. The application is designed to fill the gap left by those who are unable to visit Commerce Bank in person. The application is targeted at those who own computers and/or cellular devices. More so, the prospective end users will find this application more convenient than traditional in-person banking methods because they will be able to log in at any time. The aim of this project is twofold. First, Commerce Bank customers will be able to log on to the Web Application and view their personalized list of transactions. Secondly, customers will be able to set triggers to receive notifications. This document gives a preliminary plan for how we aim to achieve the above stated aims. This section will provide an overview of goals,objectives,deliverables and constraints. The next four sections give an overview on plans(Startup,Work,Control,Process).

1.2 Goals and Objectives

Project Goals:

1. Give customers an online alternative to in-person interactions and give them the ability to stay informed about their banking accounts.
2. Provide Commerce Bank with a web application that will improve the user experience for their customers and increase product sales.

Project Objectives:

1. Be designed so that anyone, regardless of technical level should be able to use and understand.
2. Have an intuitive login form.
3. Provide users with the ability to Add/Edit/Delete notification rules about their bank accounts.
4. Have a detailed dashboard with a summary of triggered notification rules that is easy to read and use.
5. Display a Transaction list sorted to the user’s preference that can be modified to add additional transactions.
6. Give the customer functionality to export a spreadsheet of their transactions.
7. Accomplish at least two further feature requests (“stretch goals”) beyond the mandatory features.

1.3 Project Deliverables

The following items will be delivered to the customer (Commerce Bank) on or before 5/15/2021:

1. Project Charter
2. Software Requirements Specification Document
3. Source code
4. System test cases
5. A presentation about the finalized web application

1.4 Assumptions and Constraints

The project estimates depend on these assumptions being true:

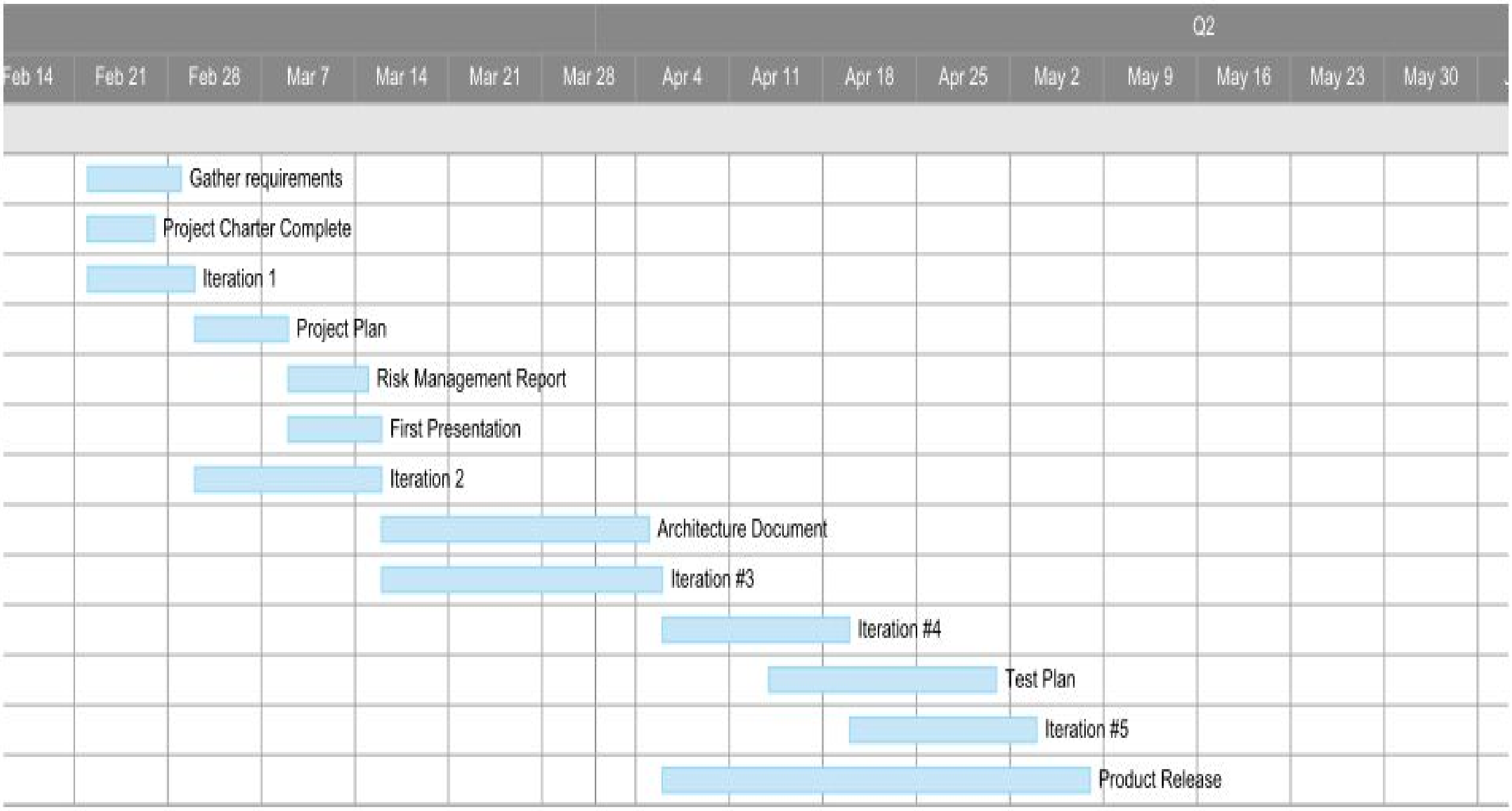
1. Team members will be able to work individually on their own machines. The database will be distributed in a way such that everyone has their own database to test on without any software issues.
2. Team members all have access to the github repository.
3. Because the project will not involve registering users or administering new products to customers, there must be an external system that Commerce has in place that does this already that reflects in our web application’s database.

The constraints of the project that must be considered:

1. Must be a web application built in a newer web development framework.
2. Support from Commerce Bank representatives is limited to .NET framework.
3. Integration of a persistent database is required - Commerce Bank representatives support SQL Server.
4. At least 10% of the app’s code must have unit test coverage. Explicit standards for the user experience and front-end development must be followed.

1.5 Schedule

This chart shows start and end dates for high-level activities ending in major milestones or deliverables.



1.6 Success Criteria

In order for the project and product to be considered “complete,” the following criteria must be met:

Success Criteria	**Success Value**
Project Success Criteria	
The project and its respective deliverables are completed and turned in punctually.	All deliverables part of the release plan (Section 3.2) are completed on or before May 7th, 2021.
The priority features found in Section 4 of the Requirements Document are integrated into the final iteration.	The features are fully functional and work as expected.
Enough of the stretch goals provided by Commerce Bank can be found in the final deliverables of the project.	≥ 2 stretch goals met by end of project
The cost of the project does not exceed the budget.	≤ $11,760
Product Success Criteria	
The web application is easily traversable and understandable by both desktop and mobile users.	95% of users can navigate and understand the application without the need to refer to external resources
Transaction notifications successfully reach their target users without too much delay.	< 2 minutes: *Acceptable* < 30 seconds: *Desired*
The product is seen favorably by users	≥ 80% of user reviews are positive

1.7 Definitions

User – An individual who is expected to interact with or use the web application.

Role – A category of users with like characteristics. (*i.e. customers and administrators*)

Product – The project being described within this document; the web application to be created for Commerce Bank regarding transaction notifications.

Project – Activities that will serve as a means to produce the product described here.

Shall – Adverb; indicates that requirements are mandatory. *Must* and *will* are synonymous, therefore they may be used in place of *shall*.

Should – Adverb; indicates that requirements are desired but not mandatory.

May – Adverb; indicates options. Can be used to express an option instead of a requirement. (*i.e. “The system may be taken offline for up to one hour every evening for maintenance”*)

Controls – Individual elements of a user interface, such as checkbox/radio buttons and forms.

Customer Account – The digital representation of a person who is a Commerce Bank customer.

Bank Account – The digital representation of a checking/savings account owned by one or more customer accounts. A customer may own more than one bank account.

1.8 Evolution of the Project Plan

As the project evolves the nature of the work to be done will be better understood and plans will become more detailed. Thus, each version of the plan should be placed under configuration management, and each version should contain a schedule for subsequent updates to the plan before the start of each iteration.This project management plan is a living document and as such will be subject to change as the term of the project moves forward.

2 Startup Plan

2.1 Team Organization

Project Manager (Will Keke): The project manager is responsible for creating the project plan (with help from the rest of the team), managing risks, running the weekly team meeting, and accomplishing some of the team's development tasks.

Developers (Benaiah Kilen, Atticus Parris) : Developers are primarily responsible for coding and helping test modules. They are also expected to take part in architecture planning and review meetings along with helping the Project Manager with developing and updating the project plan.

Database Administrator (Zach Gharst): The Database Administrator is responsible for creating, updating, and researching the needs of the database. They will also be responsible for making sure that the rest of the team is all up to date on relevant changes made to the database and will also help the Project Manager with developing and updating the project plan.

Tester (Andrew Poitras) : The Tester is primarily responsible for writing unit tests and doing manual testing of the product. The tester will also help with the development and updating of the project plan and will help the Developers with coding modules.

2.2 Project Communications

The full development team will communicate regularly through Discord with planned meetings at least once a week and constant text communication through the team’s Discord chat channel. Impromptu meetings are expected in the case of unforeseen circumstances.

The team will communicate with the class professor at the end of each project iteration. This will be done through a canvas submission of an iteration report that includes logged hours, an iteration reflection, and a tag on github of the all work done up to the end of the specified iteration.

The team may communicate with Commerce Bank through email if any issues or questions arise regarding the technology stack being used or any other clarifying questions. The team will communicate the progress of the project halfway through the semester and the results of the project at the end of the semester to Commerce Bank through presentations done over Zoom.

2.3 Technical Process

All developers are required to participate in the agile methodology to minimize risk (such as bugs, cost overruns, and changing requirements) and release software in iterations. These iterations will allow the team to realize software benefits earlier, with frequent incremental improvements.

2.4 Tools

Programming Language(s)	C#, JavaScript, HTML, CSS, SQL
Framework(s)	ASP.NET MVC
Environment(s)	Visual Studio 2019, Microsoft SQL Server
Version Control	Source code and database snapshots will be stored using Git (remote repo: GitHub); documentation is completed through Google Docs which automates the version history (with named versions stored in the Git repo)
Defect Tracking	GitHub Issues
Build Tools	.NET CLI, Ubuntu Make
Automated Testing	xUnit Testing
Communication	Outlook, Canvas, Discord, GitHub

3 Work Plan

3.1 Activities and Tasks

		Role	**Owner**	**Estimated**	**Effort**
			(primary) **(secondary)**	**By Task**	**Subtotals**
	Meetings and miscellaneous				
Architecture					**12**
	Design architecture	Project Manager	**Will**	12	
Requirements					**9**
	Gather	Requirements Engineer	**All**	2	
	Analyze	Requirements Engineer	**All**	4	
	Specify	Requirements Engineer	**All**	3	
Documentation					**14**
	Project Charter	Project Manager	**Will, Zach**	2	
	Release Plan	Project Manager	**Will**	2	
	Requirements Document	Project Manager, Requirements Engineer	**All**	5	
	Project Plan	Project Manager, Developers	**Will**, All	5	
	Architecture Document	Architect	*TBD*	*TBD*	
	Test Report	Tester	*TBD*	*TBD*	
	User Guide and System Admin Doc	Developers, Requirements Engineer	*TBD*	*TBD*	

3.2 Iteration Plans

		Role	Owner	Estimated	Effort	
			(primary) (secondary)	By Task	Subtotals	
Iteration 1:	**Design**					**27**
		Determine technology needs	Developers	**All**	5	
		Research and learn new language and environment	Developers	**All**	20	
		Initialize project and set up proxy server software	Production Engineer	**Zach**	2	
	Analysis					**6**
		System testing	Tester	**Drew**	2	
		Bug fix as required	Tester, Developer	**All**	2	
		Evaluate needs for next iteration	Project Manager	**Will**	2	
Iteration 2:	**Design**					**5**
		Site map	Project Manager	**Will, Zach**	1	
		Design prototype	UX Engineer	**Atticus**	2	
		Database design and schema	Database administrator	**Zach**, Benaiah	2	
	Development					**14**
		Create database and implement schema	Database administrator	**Zach**, Benaiah	3	
		Page path URLs and page skeletons	Developers	**Atticus**	2	
		Non-functional login form	Developers	**Benaiah**	1	
		Mock transaction table	Developers	**Will, Benaiah**	6	
		Header and navigation menu	Developers	**Drew**	2	
	Analysis					**6**
		System testing	Tester	**Drew**	2	

3.3 Budget

The total budget estimates of the project are broken down below as a product of estimated man-hours and cost per hour.

1 project manager at 4 hours per week for 14 weeks
56 hours * $50/hr = $2800

2 software engineers at 4 hours per week each for 14 weeks
112 hours * $40/hr = $4480

1 testing engineer at 4 hours per week for 14 weeks
56 hours * $40/hr = $2240

1 database administrator at 4 hours per week for 14 weeks
56 hours * $40/hr = $2240

Grand Total: 280 hours, $11760, avg: $42.00 per hour.

4 Control Plan

4.1 Monitoring and Control

Weekly	–	Team meeting. Project participants report status, progress and potential problems.
Bi-Weekly	–	Iteration Review. Project Participants go over what was accomplished, what went right, what went wrong, and suggestions for next iteration.
3/26/2021	–	Mid Semester Review. Project Team presents status and accomplishments to the sponsor (Commerce Bank).
5/05/2021	–	Release Review. The project team presents the final product to the sponsors (Commerce Bank).

4.2 Project Measurements

Phase	Measurement	Source
Initial planning	Record effort estimates for requirement analysis Team reflection on progress	Mgr
Iteration planning	Record/Update effort estimates for current iteration Team reflection on previous progress	Mgr
Iteration Closeout	Record/Update effort estimates for current iteration Team reflection on previous progress	Mgr

5 Supporting Process Plans

5.1 Risk Management Plan

No.	Risk	Monitoring and Controlling
1	Inappropriate version of the tools and components.	Select specific versions of tools and components to use and every member will adhere to the choice throughout the entire project.
2	Failure to meet deadlines for deliverable.	Setup milestones in advance of the final due date for each deliverable
3	Requirements change	Clients will be made aware, in advance, of the amount of change that can be accommodated within the term of the project.
4	Accidental loss of valuable information	Copies of work not stored online will be kept by all team members. The configuration management plan policy shall help prevent valuable information from being shared.

5.2 Configuration Management Plan

1. All documents will be stored in a “Resource” section available to all team member.Any record, track, control, or audit configurations of said document will need approval from project manager.Documents may also be stored at the team repository site for reference.
2. A change history with all documents is encouraged .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.

5.3 Verification and Validation Plan

Any major changes to the main (master) branch of the repository will require two approvals from other team members by means of a code review (specifically, GitHub pull requests).