Software Project Management Plan

<<Group 1 Commerce Bank Project>>

<<May 01, 2021>>

Team Members

<<Sushant Acharya>>
<<Kole Keeney>>
<<Ozzie Loewen>>
<< Thomas Starr-Timberlake>>

Document Control

Change History

Revision	Change Date	Description of changes
V1.0	03/08/2021	Initial release
V2.0	05/01/2021	Updated Plan

Document Storage

This document is stored on the team's GitHub repository at:
https://github.com/UMKC-CS451R-Spring-2021/semester-project-group-1-commerce

Document Owner

Ozzie Loewen is responsible for developing and maintaining this document.

Table of Contents

1 Overview

1.1 *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 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.

The purpose of this project is to create an easy system for Commerce Bank members to view, edit, and track their transactions without needing to visit a location in person. This project will provide banking members of Commerce with the ability to login to their account and view their transaction history. A user should also be able to view notifications sent to them from the application. The user will be allowed to add transactions to their account themselves should they wish to make an update.

There will be a user guide and instillation guides about operating the application created, but personalized training is outside the scope of this project. While the user is able to view their account history, they will be unable to make significant changes to their account foundation, beyond being able to add transactions.

1.2 *Goals and Objectives*

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

Project goals generally related 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 an easy-to-use web application for Commerce Bank and their banking customers.
2. Create an application that is appealing to the user.

Project Objectives:

1. Create a database of the bank's customer's transactions.
2. Develop the application to be used on *both* a desktop and mobile devices (e.g. cellphones, tablets, iPads)
3. Make the application responsive and aesthetically pleasing.
    a. Front-end framework/libraries are up to you but must be included in project (aka no external resources), but you must use at least one CSS framework such as Bootstrap.
    b. Use Commerce Bank styling.
        Dark Green: #006747
        Green: #4FA800
        Blue: #007AA3
        Yellow Orange: #FFB300
        Red: #E30000
    c. Fonts: Poppins for Headings, Open Sans for Body Text.
4. User Experience should be easy enough for anyone to use.
    a. Text visible in font size and color.
    b. Should utilize everyday language (ex. 2:00p.m. instead of 14:00:00).
    c. Date should be readable instead of a timestamp format.
    d. Text in tables should be properly aligned.
        i. Numeric should be right aligned.
        ii. Alphanumeric should be left aligned.
    e. White space should be intentional and not sporadic.
5. Provide a reliable login page for the user to navigate to their personal account.
    a. Simple login and passwords fields.
        i. Mask the password field.
        ii. Password minimum requirements:
            1. 8 characters
            2. 1 upper case letter
            3. 1 symbol
            4. 1 number
            5. 1 lower case letter
    b. Login button
6. The notifications the user will get should be based upon their transaction type.
7. The notifications the user receives should also be configurable.
    a. Users should be able to Add/Edit/Delete notification rules without technical assistance.

b. These are examples (you have the liberty of coming up with what types of notification rules there are and how they are implemented):
    i. Minimum of 3 notification rules added to system.
        1. Out of state transactions
        2. Timeframe usage
        3. Categories

8. The user should have a summary of their transactions with a particular set up.
    a. Transaction list sorted by date.
        i. Don’t need to worry about searching/filtering.
    b. Users should have the ability to add transactions here, which should automatically trigger any associated notification rules.
9. The home page for the user should also be easy to navigate while also giving them the options for viewing their personal transaction information.
    a. Dashboard (Summary for triggered notification rules):
    b. Number of times each notification rule has been triggered over the past month and year.
    c. Daily screen should be easy to read, easy to use, and provide a snapshot of data.
    d. Ability to hide notification rules where the times tripped is zero.
    e. Ability to pull/compare notification rule different timeframes.
    f. Ability to export to spreadsheet.

1.3 Project Deliverables

The following items will be demonstratable for the company by: 03/15/2021:

1. Source code for the client.
2. A User’s guide.
3. Project Plan
4. Requirements Document
5. Viewable Login Page
6. Horizontal Prototype of specific user.

The following items will be demonstrated for the company by: 05/03/2021:

1. A user’s guide (this may come in the form of a video demonstration)
2. The items listed form the section above, dated to be available for demonstration on 03/15/2021
3. A functioning Login Page.
4. A functioning Home Page.
5. A Login Page that is synced with the user in the SQL database.
6. A Home Page that is synced with the user in the SQL database.

This list is still in development and does not include the stretch goals of the project. For a fuller understanding of what will be accomplished by May 3rd, 2021 (05/03/2021), please refer to section 1.2 Goals and Objectives as well or the Requirements document.

1.4 Assumptions and Constraints

Assumptions:

1. The information provided by Commerce Bank is not real customer information that would be compromised should we look or use it for testing.
2. It is assumed that the contract and provisions made by both company and team members will be upheld.
3. It is assumed that the necessary steps taken to provide this learning experience have been approved by the proper entities.
4. It is assumed that all team members, instructors, clients, sponsors, and stakeholders will act in accordance with the standards set by 'The ACM Code of Ethics and Professional Conduct' to the best of their ability.

Constraints:

1. The software must be able to run on a phone and desktop using the Chrome browser.
2. The database must be comprised of test data provided by Commerce.
3. The project will have a horizontal prototype ready on 03/15/2021
4. The entire application will be ready for a demonstration on 05/03/2021

This section maybe subjection to revision once further into the development process.

1.5 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.

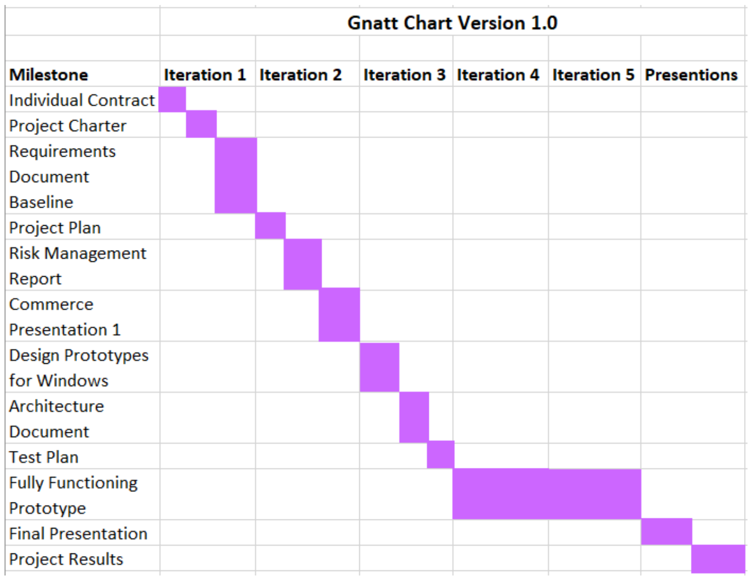
Dates that are highlighted in grey have already passed and have had the associated material delivered.

Schedule Information (Major milestones and deliverables):

03/02/2021 - Project Charter Complete
03/02/2021 – Requirements Document Baselined
03/01/2021 – Iteration #1 Close Out
03/08/2021 – Project Plan
03/14/2021 – Risk Management Report
03/15/2021 – Commerce Bank Presentation #1
03/15/2021 – Iteration #2 Close Out
03/26/2021 – Mid Semester Presentation

04/04/2021 – Architecture Document Due
04/05/2021 – Iteration #3 Close Out
04/19/2021 – Iteration #4 Close Out
04/30/2021 – Test Plan
05/03/2021 – Iteration #5 Close Out
05/XX/2021 – Presentation
05/07/2021 – Project Results
05/08/2021 – Team Evaluation

Gantt chart for visible representation of the material being worked on at a high concept level. This chart has held accurate through Iteration 5 Close Out.



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

Financial Information (Cost estimate and budget information):

Worker Cost:

Type	**Hourly Rate**	**Hours per Week**	**Weeks Working**	**Rate**	**Grand Total**
Project manager	$50	4 Hours	14 Weeks	$2800	
Developer	$40	4 Hours	14 Weeks	$2240	
Developer	$40	4 Hours	14 Weeks	$2240	
Developer	$40	4 Hours	14 Weeks	$2240	
Part Time	$20	2	14 Weeks	$560	
					$10080

Equipment Cost:
Note that Computer lab access is provided as a resource to University Members, who are the only members being asked to work on this project. Meaning that equipment cost could be reduced to zero but would greatly reduce the number of workable hours for the employee.

A standard Lenovo - ThinkPad E14 14" Laptop - Intel Core i5 - 8GB Memory - 256GB Solid State Drive – Black costs: $712.99 without tax. Should this model be provided for *all* team members, while they use free coding software and documentation sites, like Google Drive, Google Sheets, Google Docs., then the price would be for the computer alone without factoring in electricity. Bringing the total cost, without tax to **$3564.95**

Travel:
There will be no travel involved in this project, everything can be demonstrated and presented online via zoom or Google Meet.

It would be prudent to factor in tax cost and shipping efforts for the computers however, and it is suggested to add a 'buffer' to the costs, in case there are any incidents that constitute an emergency or even bonuses for exemplary work. **The budget for this project should be: $20000.** A more in-depth breakdown of the budget is listed in section **3.4 Budget**.

1.6 Success Criteria

Success criteria listed for the project for this team. These items are referenced through our Test Plan document as well as the Requirements document.

- The project could be considered a success if the team is able to deliver an operational prototype by the end of iteration 5 (05/03/2021, as seen in section 1.5).
- The items deemed top priority have had their requirements fulfilled.
- At least 2 stretch goals provided by the stakeholders have been fulfilled.
- 80% or more of the team members would be willing to work on another software project in the future.
- The initial requests from the business have been addressed.

This list is still developing and will likely expand as the team continues to work on the project. It is also noted that success is dependent upon passing the Verification and Validation Plan as well as the Product Acceptance Plan, listed in sections 5.3 and 5.4, respectively.

1.7 Definitions

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

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

Extension – a term used while expanding a use case and its various outcome, but still relating the initial topic.

Actor – user or other software system that receives value from a use case.

Role – category of users that share similar characteristics.

Product – what is being described here; the software system specified in this document.

Project – activities that will lead to the production of the product described here. Project issues are described in a separate project plan.

Shall – adverb used to indicate importance; indicates the requirement is mandatory. “Must” and “will” are synonyms for “shall”.

Should – adverb used to indicate importance; indicates the requirement is desired but not mandatory.

May – adverb used to indicate an option. For example, “The system may be taken offline for up to one hour every evening for maintenance.” Not used to express a requirement, but rather to specifically allow an option.

Controls – the individual elements of a user interface such as buttons and check boxes.

ELMO- Enough Lets Move On. Used during meeting times to stop from going on tangents that will take away from other team member’s opportunity and time to speak during Stand-Up meetings.

Stand Ups- A stand up meeting or check in is a time where everyone can address what they worked on last time, what they are working on now, any blockers or help they might need from another team member.

Sprint- the time period in which a set number of tasks will be completed before the sprint’s end.

Feature- Aspect of the code or chunk of code that is being worked on by the developer.

It should be noted that most of these terms were taken as examples from the Project Charter document provided by Professor Bingham. This list has the potential to be expanded.

1.8 Evolution of the Project Plan

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

Specific options to extend the transactions application are other stretch goals:

1. Deploy the project into a windows server/cloud instance.
    a. Practice configuration properties for different environments

2. Use an open-source reporting tool/business intelligence suite for all the reporting and its data visualization.

3. Use pull requests/code review approved by a group member within your source control.

4. Security scan your application and fix Critical issues at a minimum. OWASP ZAP is a good open-source option.

5. Session for remembering user if they close their browser and then log in again.

6. Notifications via messaging center in the app

7. Notifications via email or text

Other options like creating a mobile app, rather than mobile compliant application is an option. Being able to update security as more advancements in that area are developed. Or developing using a more advanced framework. Going back through and making the existing code more efficient to use or take up less memory.

Creating an application that is able to be functionable for users who have vision impairment would also be a natural next step. It should also be noted that 'green' is often a color that is hard for people with visual color impairment to see. Creating a black and white, or grey scale version of the site would be prudent for creating a site that would be more easily viewed for Commerce customers.

It is understood that there are more ways to create extensible code and there are more and various options to extend. More to this section will be added should more ideas be presented through the development process.

2 Startup Plan

2.1 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 org chart. If development team is known, actual names can be associated with roles.

Partial Example

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. They are also in charge of trying to document the process as well as any specific coding styles that are being used for the benefit of the other team members. It is also the Project Manger's responsibility to present the prototyping with help from the programmers should specific functionality need to be further explained. They also have the final judgement on the necessity of a requirement as well as the planned schedule of when requirements are completed. This scheduling ability also allows them to determine which features should be done in what iteration. (Ozzie Loewen)
Programmers (3):	Programmers are primary responsible for coding and unit testing modules. They are also expected to take part in architecture planning and review meetings, determining the architecture of the system and the overall project design through delegation. Each developer will be given the authority to determine how each of their specifically assigned features will be implemented into the overall project, communicating with other members to ensure optimal integration. It is there responsibility to explain the functionality so the feature can be properly demonstrated to the client and senior management. The programmers are also responsible for merging code and making sure their chunk of code works with their counterparts, as well has logically connects with the potential for future development or enhancements. (Sushant Acharya, Kole Kenney, Thomas Starr-Timberlake, Justin Thomas*)
Build Coordinator:	The build coordinator is responsible for setting up, running and distributing the results of the nightly build. The build coordinator in this case will be a rotation between programmers so everyone will gain the experience of having this role. (See above developers)

It should be noted that these have been expanded upon but were originally part of a guide given by Professor Bingham. This section has the potential to be expanded should there be any updates throughout the coding process.

2.2 Project Communications

The plan for communicating with fellow team members begins with using our Discord server. We have a group on there that allows for us to communicate openly and conveniently. The first base of our understanding for communicating is also within our calendar and schedule that was assigned at the beginning of the year by Professor Bingham. This schedule was also reviewed in our weekly stand-up meeting which happens every Tuesday night at 8:15 p.m. This meeting is a standard and was agreed upon using our Discord channel. In addition, we also used our meeting time to cover the Requirements document so we all were agreed as to what direction we would be taking with the project.

When we reach the point where we need more meetings to occur, it was also decided we would continue to meet at 8:15 p.m. other nights of the week (to be decided upon depending on the week) to accommodate everyone's busy day schedules.

On days we do not meet, members communicate a daily stand up in the Discord channel so the Project Manger, Ozzie, is able to see progress with the work. It is additionally noted when major merges or code changes have been made. If any technical difficulties occur, it is up to the group member to inform the rest of the team in order to get assistance to resolve their technical issues. If the issue cannot be solved in an expedited manner, then the issue will likely be the main subject of the subsequent zoom meeting, where the other team members will form an agreement of how to tackle the issue. Discord will also be the form used when an emergency occurs and needs to be addressed. Further details about how Configuration Management will be handled in the section labeled as such (5.2).

The team also uses GitHub for merging and committing code for the project. There has been a repository created for this project. This repository also holds the information we submit for the end of our iterations and the documentation that relates to it.

The Project Manger has also created a Google Drive Folder that holds information relating to the team and work being submitted for group grades, such as the Requirements Document or even this, Project Plan. This is our way of being able to check what, exactly, is being submitted and is agreeable for all team members.

Notes about when things are uploaded to Canvas (for submitting assignments), the Google Drive folder, or GitHub are communicated through the Discord Channel specific to our team.

In rare cases, UMKC emails/Canvas notifications are used if a team member is not responding in the Discord Channel.

Zoom is how the team meets virtually both for stand-up meetings and class time.

2.3 Technical Process

Our planned phases are in accordance with the schedule listed in section 1.5, with the plan of our team being done a day earlier to allow time to fail and recover should it be necessary. Feedback from the previous iteration will be used and applied in the development of the to the next iteration. The first iteration will focus on requirements gathering and setting up the project process. This iteration has already been completed. The following iteration will be the primary foundation for beginning the applications UI setup. Subsequent iterations will build upon that and incorporate the SQL database; this is subject to change as time and workload allows. The following will be determined by the projection of the project.

2.4 Tools

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

Tools:

- Bootstrap CSS framework
- ASP.NET 5.0.3 will be used for the web application development.
- SQL Server 2019 will be used for the database.
- Development will be done using visual studio Community 2019.
- Testing will be done on a Windows system
    - The web frontend will be manually tested using Google Chrome.
- GitHub will be used for a point of merging and combining code from the various team members.
    - GitHub is also used to document end of iterations.
- Discord will be used for communication (see section 2.2 for full details)
- Specified and shared Google Drive Folder will be used for work that is being submitted in document format for a group grade.
- Canvas is being used to submit work for this team.
- Zoom is being used for weekly stand ups.

3 Work Plan

3.1 Activities and Tasks

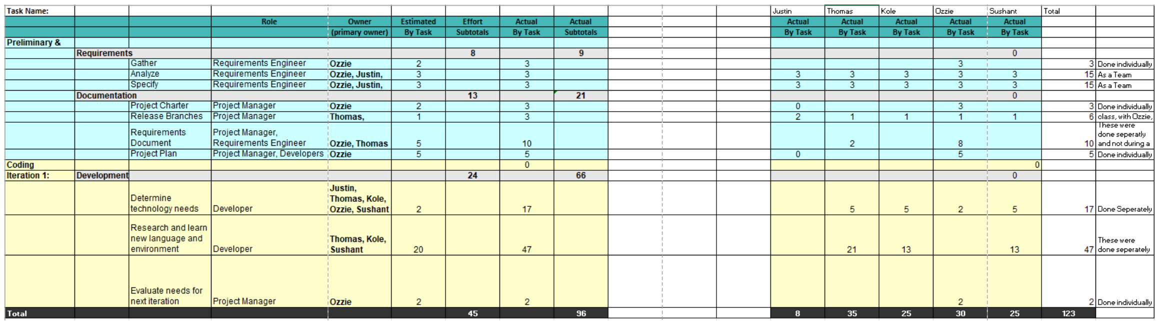
A work breakdown structure is an excellent tool for identifying a complete list of tasks.

Depending on the needs of the project, some or all of the following attributes will be recorded for each task:

- Task name
- Task Description
- Owner
- Effort estimate
- Actual effort
- Planned start and stop date

- Actual start and stop dates
- Dependencies among other tasks

For the sake of condensing this document, our assigned tasks and their subsequent information is stored in a separate spreadsheet. This information is broken down iteration and has the above information listed with it. An example of this is displayed below:



Task Name:									Justin	Thomas	Kole	Ozzie	Sushant	Total	
			Role	Owner	Estimated	Effort	Actual	Actual	Actual	Actual	Actual	Actual	Actual		
				(primary owner)	By Task	Subtotals	By Task	Subtotals	By Task	By Task	By Task	By Task	By Task		
Preliminary &															
	Requirements					**8**		**9**					0		
		Gather	Requirements Engineer	**Ozzie**	2		3					3		3	Done individually
		Analyze	Requirements Engineer	**Ozzie, Justin,**	3		3		3	3	3	3	3	15	As a Team
		Specify	Requirements Engineer	**Ozzie, Justin,**	3		3		3	3	3	3	3	15	As a Team
	Documentation					**13**		**21**					0		
		Project Charter	Project Manager	**Ozzie**	2		3		0			3		3	Done individually
		Release Branches	Project Manager	**Thomas,**	1		3		2	1	1	1	1	6	class, with Ozzie,
		Requirements Document	Project Manager, Requirements Engineer	**Ozzie, Thomas**	5		10			2		8		10	These were done seperatly and not during a
		Project Plan	Project Manager, Developers	**Ozzie**	5		5		0			5		5	Done individually
Coding							0						0		
Iteration 1:	**Development**					**24**		**66**					0		
		Determine technology needs	Developer	**Justin, Thomas, Kole, Ozzie, Sushant**	2		17			5	5	2	5	17	Done Seperately
		Research and learn new language and environment	Developer	**Thomas, Kole, Sushant**	20		47			21	13		13	47	These were done seperately
		Evaluate needs for next iteration	Project Manager	**Ozzie**	2		2					2		2	Done individually
Total						**45**		**96**	**8**	**35**	**25**	**30**	**25**	**123**	

3.2 Release Plan

The process being used by this group is known as the 'Agile Unified Process'[1]. Focusing on the business model, the implementation, testing, deployment, configuration management, project management, and the environment of the following phases: inception, elaboration, construction and transition.

This is meant to be an adaptive approach to project development, with milestones being set dates, and goals for those dates. These dates aren't being pushed back for deployments. Instead, there would be an acknowledgement of what didn't get done and how to approach it for the next sprint. Referencing the milestone and important dates from section 1.5 and how it works from the point of an iteration are listed in the next section.

3.3 Iteration Plans

An iteration plan is a short-term fine-grained plan that shows the tasks to be completed during an iteration. For the first iteration, which ended March 1st, 2021, the team's goal was finish our preliminary work to be able to begin work on the project and create a horizontal prototype for the end of the second iteration.
Iteration 2

- There will be a demonstration or presentation of this prototype, which is to be ready, March 15st, 2021, while the presentation will be March 19th, 2021.
- This presentation and end of iteration will focus mainly on the visual appeal of the site.

[1] This was found at Agile Software Development Methods: What is the Agile Unified Process https://blog.bydrec.com/agile-software-development-methods-what-is-the-agile-unified-process

    - Focus on the frontend and creating logic for the login page, as well as a design for the home page dashboard detailed in section 1.2 Goals and Objectives.

Iteration 3: April 5th, 2021

- Iteration 3 will be closed out on April 4th, 2021 and will be taking what wasn't accomplished in the previous sprint or things that didn't work in the previous type to be fixed, as well as adding more backend logic for incorporating the SQL database.

Iteration 4: April 19th, 2021

- Transactions Page Development
    - Creation Page and backend development
    - Editable Transactions for the transactions
    - Automatic update to the Balance in the user's account
    - Anything that wasn't done in the previous sprint will worked and implement in this sprint.
    - Frontend coordination for page

Iteration 5: May 3rd, 2021

- Notifications Page
    - Set rules for the page
    - Implementation of rules for the page
    - Frontend coordination with page
- Dashboard Finalization
    - Implementation of the CSS framework with the backend work for the dashboard.
- Final touches will be done. Any make up work can be resolved.
    - Coordination for the frontend work and the backend

Schedule Information (Major milestones and deliverables):
03/02/2021 - Project Charter Complete
03/02/2021 – Requirements Document Baselined
03/01/2021 – Iteration #1 Close Out
03/07/2021 – Project Plan
03/14/2021 – Risk Management Report
03/15/2021 – Iteration #2 Close Out
03/19/2021 – Mid Semester Presentation
04/04/2021 – Architecture Document Due
04/05/2021 – Iteration #3 Close Out
04/19/2021 – Iteration #4 Close Out
04/30/2021 – Test Plan
05/03/2021 – Iteration #5 Close Out
05/XX/2021 – Presentation
05/07/2021 – Project Results
05/08/2021 – Team Evaluation

3.4 Budget

The budget was outlined in section 1.5 but will be further broken down in this section. Currently, the projected budget for this project is $20000. And is outlined here:

Worker Cost:

Type	Hourly Rate	Hours per Week	Weeks Working	Rate	Grand Total
Project manager	$50	4 Hours	14 Weeks	$2800	
Developer	$40	4 Hours	14 Weeks	$2240	
Developer	$40	4 Hours	14 Weeks	$2240	
Developer	$40	4 Hours	14 Weeks	$2240	
Part Time	$20	2 Hours	14 Weeks	$560	
					$10080

Equipment Cost:
Note that Computer lab access is provided as a resource to University Members, who are the only members being asked to work on this project. Meaning that equipment cost could be reduced to zero but would greatly reduce the number of workable hours for the employee.

A standard Lenovo - ThinkPad E14 14" Laptop - Intel Core i5 - 8GB Memory - 256GB Solid State Drive – Black costs: $712.99 without tax. Should this model be provided for *all* team members, while they use free coding software and documentation sites, like Google Drive, Google Sheets, Google Docs., then the price would be for the computer alone without factoring in electricity. Bringing the total cost, without tax to $3564.95

Travel:
There will be no travel involved in this project, everything can be demonstrated and presented online via zoom or Google Meet.

The total cost of the project then being $13644.95. It would be prudent to factor in tax cost and shipping efforts for the computers however, and it is suggested to add a 'buffer' to the costs, incase there are any incidents that constitute an emergency or even bonuses for exemplary work. **The budget for this project should be: $20000.**

The price for a developer in the first week, including the cost of their equipment: $872.99. $160 will go to the developer, the remaining $712.99 the price of their computer.
The cost of all three developers for their starting week: $2618.97
The cost for the Project Manger's starting week: $912.99

The price of a Project Manger sees a more immediate result, since they should be able to generate some documentation that the business would like to see fairly quickly into their start on the project. For a developer, it would really take a full iteration, or two weeks to see some results.
In the case of a full iteration, from the start with the inclusion of the machine specified above that would be: $1032.99.
With $320 going to the developer and the remainder being the cost of the computer being provided.
An iteration without the initial startup cost of the computer would be: $320 for one developer and $960 for the three that are on this team.
The price for one iteration with three (3) developers and one (1) project manager: $1360

By the end of iteration 1: $4211.96 (inclusion of the computer with no tax or shipping cost). It would be easy to round this price up to $5100, a little over 20% increase to account for any of those additional costs.
End of iteration 2: $1360+iteration 1 ($5100) = $6460
End of iteration 3: $1360+iteration 2 ($6460) = $7820
End of iteration 4: $1360+iteration 3 ($7820) = $9180
End of iteration 5: $1360+iteration 4 ($9180) = $10540

It is also factored in that an additional, part time developer might be used on this project. This could be factored in a few different ways. It would be essential that this worker is provided a computer to be working from, bringing his first weekly total up to: $752.99 at $20 an hour plus the cost of his computer ($712.99)
1st iteration: $792.99. Applying the same approximate 20% initial increase for iteration 1, we get: $1003.
2nd Iteration: $1083
3rd Iteration: $1163
4th Iteration: $1243
5th Iteration: $1323

Which added to the amount of the full-time workers about would be $10540 + $1323 = $11863, well within budget. This allows for some wiggle room as far as further shipping costs, emergency new equipment, or possible bonuses as appreciation for the hard work the workers have put into the project.

It should also be noted that the cost of the laptops is not a total loss, in most cases. This equipment can be returned and used by another employee should it be in good condition.

4 Control Plan

4.1 Monitoring and Control

Important Dates and what is to be accomplished at those times:

Weekly –	Weekly team meetings, standard Tuesday at 8:15 p.m. other days of the week at 8:15 p.m. when they apply. Project participants report status: what was worked on last time, what is being worked on now, any blockers or potential blocker, is there another team member that can help.
03/08/2021 –	Mid sprint review and what should be worked on in this upcoming week.
03/15/2021 –	Iteration 2 close out. This would include Iteration 2 Task Documentation to be uploaded to the GitHub and submitted to Canvas. Preparations for the Mid Semester presentations (to take place on 03/19/2021).
03/19/2021 -	Mid Semester presentations. A horizonal prototype, focused on the UI of the project to be presented.
04/04/2021 –	Architecture Document Due. This will have further explanation later.
04/05/2021 –	Iteration 3 close out. This would include Iteration 3 Task Documentation to be uploaded to the GitHub and submitted to Canvas.
04/19/2021 –	Iteration 4 close 0ut. This would include Iteration 4 Task Documentation to be uploaded to the GitHub and submitted to Canvas.
04/30/2021 –	Test Plan. This will have further explanation.
05/03/2021 –	Iteration 5 close out. This would include Iteration 5 Task Documentation to be uploaded to the GitHub and submitted to Canvas.
05/XX/2021 –	Presentation. This date is undetermined as of right now, but will take place before the 7th of May, when we will do results of the projects, but after the close out of the fifth iteration on May 3rd.
05/07/2021 –	Project Results. These will be the results of the project, which will include evaluation of the product.
05/08/2021 –	Team Evaluation. This will evaluating team members and what they have accomplished as well as how well each team member worked with each other.

4.2 Project Measurements

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

This example has been left as the initial example until further changes need to be made. As of right now, these seem like a great baseline to start off with and expand upon as more implementations are made.

5 Supporting Process Plans

5.1 Risk Management Plan

The biggest risk of the project is lack of experience. While everyone on this team has had previous experience coding and some of us have even had experience working for more individualized clients (smaller businesses or side hustles for people). However, none of us have worked with the materials being specified in the project make up, such as Bootstrap or .Net, and with minimal experience in SQL. This risk is mitigated by the fact that this is still a learning environment with the expectation to learn from the experience and not punish for it. The way the team plans on mitigating the risk is to focus on what we know, and not on what we don't; find work arounds to still deliver the requested material. This may mean the code isn't as efficient as if the team member were working in their chosen language, but it will still be operable (done is better than perfect in some instances).

The next risk is the shortened timeline. We don't have a lot of time to create a project that is client demonstration ready. Ideally, a major demonstration would be presented with top and medium priorities done and many low priorities in the works. At this time, the way to mitigate that concern is to work on what we can, develop in a sensical way and focus on the iteration and task at hand.

The way to monitor these risks is for checking in, being honest, and owning material. It is much better to be up front about issues and things that cannot be accomplished early on rather than waiting until the last minute to proclaim issues. By being honest while working on the material and asking for help in the area where the team member needs assistance, we can minimize the 'catch up' period and focus on the creation.

In working towards accomplishing iteration goals, it is possible technical debt may be built up. The team has factored in time for the fourth and fifth sprint to address these issues and hopefully work through them before the final presentation.

The other item that should be considered is that this isn't the only project (or class) being worked on by these team members. For many of the employees at Commerce Bank, that is their only job. For these team members, many of us are full time students as well as working part time or full time on top of that. Creating a workable schedule that isn't fraught with meetings and instead as much code time as possible is the idea. However, if there needs to be a meeting to address issues, then a meeting will be made.

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 work products will be stored in the class_group GitHub repository: https://github.com/UMKC-CS451R-Spring-2021/semester-project-group-1-commerce
2. The naming convention for documents will be: MM/DD_with the function of the code and version number, and 'suffix' is the standard/normal suffix for the document type. For example, the second version of the requirements document created as a Microsoft Word document might be labeled: 03/07_REQ-002.doc.
3. All project (work products) items (documents, source code, test cases, program data, test data, etc) will be stored in the GitHub repository above but not all will be under change control (subject to formal change control procedures.) Only the system requirements, project plan and source code will be baselined and under configuration control will be under change control.
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. It should also be noted that a Baselined matter may require change should it be an easier fix to the other overall code, rather than working around it.

5. The change control procedure once a product is baselined is: (1) anyone wanting to make a change to a baselined item sends an email to the rest of the group describing the change, reason for the change, expected impact, and timeline for integrating the change. (2) message the group within the Discord channel to notify the members more directly of the wished update. (3) if no one responds to the group within a day with a reason for why the change request shouldn't be permitted, then a direct message should be made to the member who originally worked on the material to be discussed. (4) If after those steps have been taken and there is not response, 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. At the end of the meeting a democratic vote will be held to decide whether or not the change should be allowed.
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.

It should be noted that the base of this section was provided by Professor Bingham and was updated with more specific material for our group. This section will likely grow as we develop our best practices through the development process.

5.3 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.

The verification and validation plan are still a work in progress and will be more fully detailed in a separate document.
The quality of development is set a fairly standard level. Programmers will first develop with the goal of creating and getting it done, before working on enhancing the item for efficiency's sake.

The first development of the item will not be the only working model. The feature will be updated before integrating it into the existing code. The update will undergo Peer Review (PR) from the fellow team members before going into Quality Assurance (QA). The Peer Review process will involve getting the approval of the other two developers and then having the QA testing done by a member who did not write the code. The QA for this team will be rotational to begin with but may become a more permanent position depending on someone's skill set or desire to work in that role.

After the code has been PRed and QAed, it will then be deployed into the rest of the functioning code. Depending on how it is being integrated, the entirety of the code may need to be tested with the new feature being implemented. Should the process yield a successful implementation, nothing outside of the expected results were produced, then the code can be merged into the main branch and set as the new baseline for the work source code.

5.4 Product Acceptance Plan

What defines success for this team in terms of product production is based on a few criteria. The feature being worked on by a team member should pass the Verification and Validation plan summarized in section 5.3 of this document (to be expanded upon once practices are more defined). Within that plan, the practice of review a peer's code is for the benefit of the programmer and the other members. Seeing what someone else is working on is a good way to understand their portion of the project, as well as identifying where the individual would need to take the next step in development. It gives the programmers the opportunity to view the material and identify issues early on rather than just plugging the feature into the main branch and hoping it will work. This is giving everyone the opportunity to (1) better understand what everyone is working on. (2) See how someone else interpreted the user story. (3) Gage where they should be heading with their code. (4) Validate their team member's hard work. (5) Allows team members to catch errors or faulty logic before being implemented into the code, saving us time.

For the product to be accepted it also needs to go through and pass the QA process. This is a series of unit testing that may be done to catch any outstanding data that might have been implemented in the test data.

Once the material has gone through both of these processes, and it has been successfully deployed and merged into the main branch without any issues, it is then within the acceptance plan. It is also noted that just because a section of code works with one version of the code, it may need to be updated later should it pose too much of a hinderance to the momentum of the project development.