Software Project Management Plan

Commerce Bank Web Portal

3/6/2020

Team Members

Tanvi Jain
Alec Shern
Jordan Tiu
Nathanael Goertzen

University of Missouri - Kansas City
5000 Holmes St, Kansas City, MO 64110

Document Control

Change History

Revision	Change Date	Description of changes
V1.0	3/6/2020	Initial release

Document Storage
The document is stored in the project's github repository at https://github.com/umkc-cs-451-2020-spring/semester-project-group-8/tree/master/Documentation

Document Owner
Nathanael Goertzen and Jordan Tiu are responsible for writing and maintaining this document.

Table of Contents

1 Overview

1.1 Purpose and Scope

This project document specifies the development and management for the Commerce Bank Web Portal application. The purpose for this web application is to provide bank account holders with an easy way to view transactions and receive alerts on suspicious transactions in order to reduce fraud.

The scope of this project is to develop the web application and deploy to a local environment fully featured. The features and use cases are specified within the system requirements document for this project. Any updates to the requirements document constitute a change in scope. A user guide will be provided at the end of the project's life cycle.

1.2 Goals and Objectives

Project goals:

1. Give banking customers a simple way to view banking transactions and to set and modify transaction alerts.
2. Create a web application that is intuitive and easy to navigate.

Project objectives:

1. Create a database that stores transaction details of banking customer's accounts.
2. Create an interface that allows users to login to view bank transaction details.
3. Create a homepage that utilizes a dashboard that displays transaction notification information.
4. Create a feature that allows users to add/edit/delete notification rules for transaction triggers.

1.3 Project Deliverables

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

1. Source code
2. User's Guide
3. Test Plan

4. System test Cases

1.4 Assumptions and Constraints

Assumptions:

1. The Commerce Bank team will be available to answer questions in a timely manner.
2. The project team will have a group meeting at least once per week.

Constraints:

1. The software must be in a newer web framework.
2. There must be database software running in conjunction with the app.
3. The password field must be masked on the login page.
4. The project must be completed and ready by 5/04/2020.

1.5 Schedule and Budget Summary

Schedule Information (Major milestones and deliverables):

- 02/17/2020 - Iteration #1 Planning Complete
- 02/23/2020 - Project Charter Complete
- 02/29/2020 - Requirements Document Complete
- 03/02/2020 - Iteration #1 Closeout
- 03/02/2020 - Iteration #2 Planning Complete
- 03/08/2020 - Project Plan Complete
- 03/13/2020 - Group Powerpoint/Video Presentation Complete
- 03/16/2020 - Iteration #2 Closeout
- 03/16/2020 - Iteration #3 Planning Complete
- 03/16/2020 - Technical Prototype Complete
- 04/03/2020 - Architecture Document Complete
- 04/06/2020 - Iteration #3 Closeout
- 04/06/2020 - Iteration #4 Planning Complete
- 04/20/2020 - Iteration #4 Closeout
- 04/20/2020 - Iteration #5 Planning Complete
- 04/26/2020 - Test Plan/Reports Complete
- 04/27/2020 - User and System Guide Complete
- 05/01/2020 - Stakeholder PowerPoint Presentation Complete
- 05/04/2020 - Product Go-live; Iteration #5 Closeout

Budget Summary:

Based on the team working a minimum of 3 hours per week on the project, the total estimated budget for the project is $5100. A full breakdown of budget expenses can be viewed in Section 3.4.

1.6 Success Criteria

- Project is feature complete by May 4th

- All test cases pass and have code coverage over 10%

1.7 Definitions

Dashboard - refers to portion of the web application that displays a snapshot of recent banking transactions as well as a summary of the number of times a notification rule is triggered.

Notification Rule - Refers to any set criteria that will alert the application user when said criteria matches any bank transaction.

Transaction - Refers to any deposit or withdrawal event that occurs within a bank account. Information within a transaction includes the account number, processing date, balance, amount, and description of the event.

Trigger - Alert directed to user whenever a banking transaction matches criteria set by any set of notification rules.

1.8 Evolution of the Project Plan

The project plan will be updated with identified risks at the start of each iteration. Mitigation plans and risk responses will also be recorded. These will be added to the risks section of this document.

The technical process is subject to change. Before any change in process will be made, the team will consult with the project manager and update the technical process section of this document.

2 Startup Plan

2.1 Team Organization

Project Manager:	The project manager is responsible for organizing the project and the rest of the team. The project manager is also tasked with assisting in programming.
Programmers (3):	The programmers are tasked with writing the code and unit tests associated with this project. The programmers will also assist in planning and design.

2.2 Project Communications

The team will communicate through Slack for general scheduling and questions. The team will meet twice each iteration at the minimum with additional meetings scheduled when unforeseen situations arise.

If there are technical disputes, the team members shall bring the dispute to a team meeting and argue each side. If a consensus can not be reached, the project manager has the final say on the technical implementation.

2.3 Technical Process

At the start of an iteration, each team member will either self-assign or be assigned to issues. When developing code for an issue, the team member will branch off of the master branch in the project's Github repo. Once code is ready to be reviewed, the team member will open a pull request to merge their proposed change into the master branch. When an approving review is given, the branch can be merged.

At the end of every iteration, a release tag will be attached to the most recent commit on master, signifying the end of that iteration. The team will then evaluate needs for the next iteration and start the iteration process once again.

2.4 Tools

- Programming Language – C#, HTTP
- Framework - ASP.NET
- Database software - SQL Server
- Version Control – Version control will be handled through Github
- Defect tracking – Defects will be reported to the Github issues page for the project
- Build tools – local and main builds will be done using CodeMake.
- Automated testing – Unit testing will be accomplished using an xUnit testing framework.

3 Work Plan

3.1 Activities and Tasks

The iteration plan will be updated with greater depth at the beginning of each iteration.

Iteration 1:	Role	Owners	Estimated Effort	Actual Effort
Determine technology needs	Developer	**Tanvi, Nate, Alec, Jordan**	1	2
Learn the ASP.NET and C# Fundamentals	Developer	**Tanvi, Nate, Alec, Jordan**	4	8
Set up development environment and IDE	Developer	**Tanvi, Nate, Alec, Jordan**	1	1
Set up local servers	Developer	**Tanvi, Nate, Alec, Jordan**	1	1

Set up database table format	Developer	**Alec, Nate**	1.5	1.5
Generate test datasets	Developer	**Jordan**	1	1
Create sample title page with login	Developer	**Tanvi**	4	2.5
Add styling to title page	Developer	**Tanvi**	2	0.5
Create Standard Notification Rules	Developer	**Nate**	1	1
Unit testing	Tester	**Nate**	2	0
System Testing	Tester	**Jordan**	0.5	0
Fix any bugs	Developer	**Jordan**	2	0
Evaluate needs for next iteration	Project Manager	**Tanvi, Nate, Alec, Jordan**	2	4
Iteration 2:				
Design basic CSS styles for project	Developer	**Tanvi, Nate, Alec, Jordan**	1	
Integrate hex codes for designing background	Developer	**Tanvi, Nate, Alec, Jordan**	1	
Set up database for username and password	Developer	**Alec**	2	
Set up database for transactions	Developer	**Alec**	2	
Create a dashboard on the user homepage	Developer	**Tanvi, Nate, Alec**	4	
Create a log-in screen,home page	Developer	**Tanvi**	4	
Create a transaction flags page	Developer	**Jordan**	4	
Create a transaction history page	Developer	**Nate**	3	
Add registration form page for new user	Developer	**Tanvi**	3	
Set up basic	Developer	**Alec, Nate, Tanvi,**	8	

database integration		**Jordan**		
System Testing	Tester	Tanvi, Nate, Alec, **Jordan**	2	
Fix found bugs	Developer	Tanvi, Nate, Alec, **Jordan**	4	
Evaluate needs for next iteration	Project Manager	**Tanvi, Nate, Alec, Jordan**	4	
Iteration 3:				
Create logic to add transactions	Developer		3	
Develop logic to add users	Developer		3	
Developer logic for transaction triggers	Developer		3	
System Testing	Tester		6	
Fix found bugs	Developer		2	
Evaluate needs for next iteration	Project Manager		4	
Iteration 4:				
Style web pages consistently	Developer		4	
Add functionality to transaction triggers	Developer		3	
Update homepage to display flagged transactions	Developer		3	
System Testing	Tester		6	
Fix found bugs	Developer		2	
Evaluate needs for next iteration	Project Manager		4	
Iteration 5:				
Ensure testing code coverage	Tester		5	
Perform usability tests	Tester		4	
Finalize UI design implementation	Developer		4	
System Testing	Tester		6	

Fix found bugs	Developer		2	
Evaluate needs for next iteration	Project Manager		4	

3.2 Release Plan

03/13/2020 - **MILESTONE** - Basic web pages are developed as a horizontal prototype.

04/06/2020 - Basic functionality for adding transactions and transaction triggers.

04/20/2020 - **MILESTONE** - All features implemented at a minimum viable working condition.

05/04/2020 - **MILESTONE** - Release of final product

3.3 Iteration Plans

Each iteration will last 2 weeks. Planning will begin on the first day of each new iteration. At this planning meeting, the team will evaluate the current state of the project and the goal dates for milestones. Tasks will be defined and given estimated effort totals in this meeting. Then, the developers will be assigned tasks to work on for the iteration.

Code should be developed throughout the weeks and any testing must be completed before the end of the iteration. Any changes not submitted to the master branch on the Github repo by the end of the iteration will be carried into the next iteration.

Alongside regular iteration development and planning, ongoing documents and project plans will be maintained. Team members will decide who owns the ongoing responsibilities for that iteration term at the beginning of each iteration.

3.4 Budget

1 Project Manager working 3 hours per week --- 30 hrs * $50/hr = $1500
3 Software Engineers working 3 hours per week --- 90 hours * $40/hr = $3600
Total hours = 120
Average rate (per hour)= $42.50
Total Amount (for entire team)= $1500+ $3600= $5100

All tools the team uses will be free to use for this project's purpose, therefore not adding to the cost.

4 Control Plan

4.1 Monitoring and Control

Weekly – Team meeting for project updates, iteration planning and cooperative problem solving.

3/13/2020 – Technical prototype will be finished and submitted to Commerce for initial design review.

4.2 Project Measurements

Phase	Measurement	Source
Release Planning	Record effort estimates for product features	Project Manager
Iteration Planning	Record effort estimates for scheduled tasks Update effort estimates for product features Update estimated dates in release plan. Identify risks and form plans to manage found risks.	Project Manager
Iteration Closeout	Record actual effort for scheduled tasks Record actual effort for product features	Project Manager/ Programmers
System Test	Record the rate at which errors are found. Fix errors recorded.	Project Manager/ Programmers
Project Closeout	Archive project performance data in process database. Compile report on successes and failures of the project.	Project Manager
Ongoing	Record defects found from integration testing. Assign each defect to one of the following categories: major, minor, unimportant. Keep track of the state of each defect: open, assigned, fixed, closed.	Project Manager/ Programmers

5 Supporting Process Plans

5.1 Risk Management Plan

Risk	Probability	Loss	Risk Exposure	Mitigation Plan
Highest risk is the team's unfamiliarity with ASP.NET and Microsoft SQLServer.	80%	4 Weeks	3.2 weeks	Open communication among developers to minimize potential time stuck

Risk of under-scheduling time necessary to complete iterations	30%	1 week	0.3 week	Weekly meetings to evaluate time spent on iteration tasks. Dedicated time during iteration start dates and closeouts to evaluate/modify time scheduling predictions.
Local configurations might not accurately reflect how the web app runs in a server environment	20%	1 week	0.2 week	Research implementation of hosting in a server environment.
Hardware and software failures (developers, dependencies, and client)	20%	2 weeks	0.4 week	Set dedicated time for testing new feature implementations for issues and bugs.
Developers are using different platforms for development	10%	0.5 weeks	0.05 weeks	Determine technologies and their cross-platform capabilities early before development cycle
Scope creep	10%	2 Weeks	0.2 weeks	Set project deliverables will adhere to requirements and specifications clearly stated on Commerce Bank Project Requirements document

5.2 Configuration Management Plan

1. All work products will be stored in the project's Github repo. Documents will be stored in a directory entitled "Documents" once a complete version has been produced.
2. Documents will have a fully qualified name attached to them in order to remove any ambiguity that may arise from abbreviations. Github contains

versioning natively, so no version numbers will be attached to the document names themselves.

3. Only the system requirements, project plan and source code will be baselined and under configuration control. All other artifacts can be updated as needed by the team.
4. Any items that follow change control procedures are considered baselined on the item's initial release.
5. When a baselined document needs change, the change control process is as follows: (1) A team member proposes a change to a specific document with detailed information on the subject and context of the change. (2) The team democratically votes on whether the change is necessary. (3) The team member who had envisioned the change will then own the change workload and must update and notate the changes made in the respective documents within the next 48 hours.
6. All documents subject to the change control process must have a change history section. Once a change is made, the following must be added to the change history section: Revision number, Date the change was made, Description detailing change and why it was necessary, and the name of the person who introduced the change.