Software Project Management Plan

Commerce Budget Application

03/04/2019

Team Members

Michael Essmyer
Matthew Downs
Colby Chandler
Josh Reeves

Document Control

Change History

Revision	Change Date	Description of changes
V1.0	3/05/2019	Initial release

[Note change history doesn't have to be stored with the document. Most version control tools like SVN keep track of change history automatically. The comments entered when checking in documents become the change history for the document. However, even if your version control tool supports change history, you make want to track a more detailed version here.]

Document Storage

This document is stored in the project's git repository at:
https://github.com/Everest84/Commerce

Document Owner

Michael Essmyer is responsible for developing and maintaining this document.

Table of Contents

1 Overview

1.1 Purpose and Scope

The purpose of this project is to streamline account awareness in a fun way so that users have a greater success at saving money. The Commerce Budget Application will be a web application where users can set goals in regards to saving money in various ways. It will have an emphasis on user friendliness and provide gamification as an incentive to meet goals. The application will also have an emphasis on ease of use on mobile devices and smaller screens. The scope of interaction will precede with a database containing the information of a Commerce Bank account, including transactions, balance, etc. Using this information, the system will decide whether user set goals are being met, and provide feedback and tracking. User feedback will also be taken into account once the application is released for testing.

1.2 Goals and Objectives

The main objective of this project is to allow Commerce Bank users a fun way to save money. By utilizing an app that gamifies milestones the users will have an easier way to save money.

1. Establish SQL database for the user, their account, the transactions of the account and the specified goals of the user.
2. Restrict spending on certain items to allow users to save a set amount within a certain timeframe.
3. Provide animation and tracking for milestones.

1.3 Project Deliverables

The following items will be delivered to the customer on or before 5/06/2019:

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 analysis program for advising users on spending

1.4 Assumptions and Constraints

Assumptions:

1. The client has an ODBC compliant database installed and this database will be accessible from the machine where the system will run.
2. The client maintains an SSL certificate to encrypt sensitive financial information.
3. The client is running a .NET Core compatible operating system from the machine where the system will run.
4. The web hosting ISP will allow server-side scripts to access the file system.

Constraints:

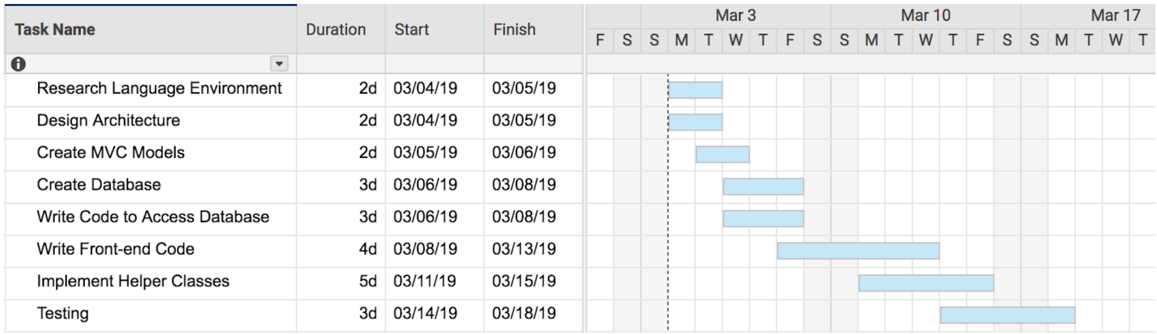
1. Accessible via mobile devices.
2. Able to support individual and family accounts.
3. Application must run off of an SQL server.
4. All client side libraries and frameworks have to be included in the source code.
5. The software must be ready by 5/06/2019.

1.5 Schedule and Budget Summary

Schedule:

02/08/2019	Gather requirements
02/22/2019	Project Charter Complete
03/01/2019	Project Release Plan Complete
03/01/2019	Requirements Complete
03/04/2019	Iteration #1 Complete
03/18/2019	Technical Prototype Complete
03/18/2019	Iteration #2 Complete
03/20/2019	Customer Approved UI Prototype Complete
04/01/2019	Iteration #3 Complete
04/05/2019	Architecture Document Complete
04/15/2019	Iteration #4 Complete
05/05/2019	User Guide and System Administration manual Complete
05/05/2019	System Test Complete
05/06/2019	Iteration #5 Complete

Gantt Chart (completed prototype):



Task Name	Duration	Start	Finish
Research Language Environment	2d	03/04/19	03/05/19
Design Architecture	2d	03/04/19	03/05/19
Create MVC Models	2d	03/05/19	03/06/19
Create Database	3d	03/06/19	03/08/19
Write Code to Access Database	3d	03/06/19	03/08/19
Write Front-end Code	4d	03/08/19	03/13/19
Implement Helper Classes	5d	03/11/19	03/15/19
Testing	3d	03/14/19	03/18/19

Budget:

1 project manager at 4 hours per week for 11 weeks 44 hr*$50/hr = $2200
1 requirements engineer at 4 hours per week for 11 weeks 44 hrs*$40/hr = $1760
2 software engineers at 4 hours per week each for 11 weeks 88 hr*$40/hr = $3520
176 hours total, $7480 total, avg, $42.50 per hour

1.6 Success Criteria

1. The team delivers an operational prototype at the end of the semester with the features mentioned in the goal section above.
2. 75% or more of the team members would be willing to work together on another software project in the future.

1.7 Definitions

Use case – user selects a time and amount to save along with a template of milestones. The app will track spending in all accounts and make suggestions when needed.

Scenario – user connects their checking account and selects that they want to save 100$ within a month. The app calculates the milestones to save 20$ a week. The app analyses and displays their spending. The app tracks the transactions on the account and rewards the user when the milestone of 20$ a week is achieved. Finally the app will classify spending and attempt to advise the user to reduce unwarranted spending.

Account User – the person that will be interacting with the application directly.

Account Owner – the person who will be allowed to set goals.

Product – app that tracks spending and rewards saving.

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

1.8 Evolution of the Project Plan

Our goal is to have a completely functional prototype in time for customer approval. We know this is a bit insane, since three of our four developers have never worked with a project like this we want to get all the bugs out of the way quickly as well as better future estimates. We are also hoping this gives us a good chance to receive the most constructive feedback from the customer.

Our biggest risk resides in our lack of experience, but we are smart and motivated. Under the leadership of our experienced developer we should not run into any significant problems since we plan to keep the project simple.

2 Startup Plan

2.1 Team Organization

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.

Programmers (2): Programmers are primary responsible for coding and unit testing modules. They are also expected to take part in architecture planning and review meetings.

Build Coordinator: The build coordinator is responsible for setting up, running and distributing the results of the nightly build.

2.2 Project Communications

Tasks will be defined and distributed through Trello. Communication and the exchanging of helpful files will occur through slack. Code will be built, modified and exchanged through github.

2.3 Technical Process

This project will use an agile methodology with 2 week iterations. There are no planned deviations from this schedule. Changes to the source code will be added to the GitHub repository, and each iteration will have an internal release until the final iteration.

2.4 Tools

Programming Language – C#, .NET Core
Version Control – Source code and documentation will be stored on a Github repository.
Build tools – Builds will be locally made using Visual Studio and Microsoft SQL Server.
Automated testing – Unit tests will be implemented using Microsoft unit test framework.

3 Work Plan

3.1 Activities and Tasks

Create Database Schema

- Estimated Effort: 4 Hours
- Actual Effort: TBD

- Planned Start: 3/5/2019
- Planned Stop: 3/5/2019
- Actual Start: TBD
- Actual Stop: TBD
- Dependencies: None
- Owner: Colby Chandler

Insert Customer Data Into Tables

- Estimated Effort: 2 Hours
- Actual Effort: TBD
- Planned Start: 3/6/2019
- Planned Stop: 3/6/2019
- Actual Start: TBD
- Actual Stop: TBD
- Dependencies: Must begin after database creation
- Owner: Michael Essmyer

Create View for User Login

- Estimated Effort: 1 Hour
- Actual Effort: TBD
- Planned Start: 3/7/2019
- Planned Stop: 3/7/2019
- Actual Start: TBD
- Actual Stop: TBD
- Dependencies: None
- Owner: Matthew Downs

Create View for Transaction History

- Estimated Effort: 1 Hour
- Actual Effort: TBD
- Planned Start: 3/7/2019
- Planned Stop: 3/7/2019
- Actual Start: TBD
- Actual Stop: TBD
- Dependencies: None
- Owner: Matthew Downs

Create View for Adding New Goal

- Estimated Effort: 1 Hour
- Actual Effort: TBD
- Planned Start: 3/7/2019
- Planned Stop: 3/7/2019
- Actual Start: TBD
- Actual Stop: TBD
- Dependencies: None

- Owner: Matthew Downs

Create View for Progress Towards Goals

- Estimated Effort: 1 Hour
- Planned Start: 3/7/2019
- Planned Stop: 3/7/2019
- Actual Start: TBD
- Actual Stop: TBD
- Dependencies: None
- Owner: Matthew Downs

Add Gamification For Goal Completions

- Estimated Effort: 5 Hours
- Planned Start: 3/19/2019
- Planned Stop: 3/26/2019
- Actual Start: TBD
- Actual Stop: TBD
- Dependencies: Begins after the goal system has been established
- Owner: Michael

Create View for Account Management

- Estimated Effort: 3 Hours
- Planned Start: 3/9/2019
- Planned Stop: 3/10/2019
- Actual Start: TBD
- Actual Stop: TBD
- Dependencies: Account model must be created first
- Owner: Colby

Create Models that match Database Tables

- Estimated Effort: 3 Hours
- Planned Start: 3/7/2019
- Planned Stop: 3/7/2019
- Actual Start: TBD
- Actual Stop: TBD
- Dependencies: None
- Owner: Josh

Users Manual

- Estimated Effort: 4 Hours
- Planned Start: 3/15/2019
- Planned Stop: 5/05/2019
- Begins once prototype is accepted and build plan is completed
- Owner: Michael Essmyer

Systems Test Document

- Estimated Effort: 5 Hours
- Planned Start: 3/5/2019
- Planned Stop: 5/05/2019
- Begins once UI is initiated
- Owner: Michael Essmyer

3.2 Release Plan

02/08/2019 - Gather requirements
02/22/2019 - Project Charter Complete
03/01/2019 - Project Release Plan Complete
03/01/2019 - Requirements Complete
03/04/2019 - Iteration #1 Complete
03/18/2019 - Technical Prototype Complete
03/18/2019 - Iteration #2 Complete
03/20/2019 - Customer Approved UI Prototype Complete
04/01/2019 - Iteration #3 Complete
04/05/2019 - Architecture Document Complete
04/15/2019 - Iteration #4 Complete
05/05/2019 - User Guide and System Administration Manual Complete
05/05/2019 - System Test Complete
05/06/2019 - Iteration #5 Complete
05/06/2019 - Product Release

3.3 Iteration Plans X

Iteration 1:

- Gather Requirements
- Create Project Plan
- Create Requirements Specification
- Research Language and Environment

Iteration 2:

- Create Database With Provided Sample Data
- Insert Customer Data Into Tables
- Create Models that match Database Tables
- Create View for User Login
- Create View for Transaction History
- Create View for Adding New Goal

Iteration 3:

- Create View for Account Management
- Add Gamification For Goal Completions
- Users Manual

3.4 Budget

1 project manager at 4 hours per week for 11 weeks 44 hr*$50/hr = $2200
1 requirements engineer at 4 hours per week for 11 weeks 44 hrs*$40/hr = $1760
2 software engineers at 4 hours per week each for 11 weeks 88 hr*$40/hr = $3520

176 hours total, $7480 total, avg, $42.50 per hour

4 Control Plan

4.1 Monitoring and Control

Weekly – Team meeting. Project participants report status, progress and potential problems.
3/5/2019 – Critical Design Review. Formal inspection of product architecture.
3/18/2019 – Prototype Demonstration. Informal inspection of the application’s progress, usability and gamification.
3/25/2019 – Executive Review. The project manager presents current project status to project sponsor and senior executives.
4/1/2019 – Critical Design Review. Formal inspection of product architecture.
4/15/2019 – Final Design Review. Formal inspection over the entire product from database implementation to user interface.
5/5/2019 – Project Deadline. Final product submission.

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	Mgr
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

5 Supporting Process Plans

5.1 Risk Management Plan

The most impending and immediate risk to this project is our groups overall lack of experience. Most of us have never used SQL Server, we have never written in html and are new to the .Net environment. Our mitigation plan for this is to study these subjects on our own and then collectively work together through the first iteration. This way no one gets left behind and everyone will know the skillset of the other members, that way everyone has help. Another common risk is the failure to effectively implement classes and objects, which we plan to discover through rapid deadlines and thorough testing.
Another likely risk is additional complexity with further requirements that may be added in later iterations, after presenting progress to the customer. We will mitigate this by keeping the system modular and having low coupling.
There is also a risk in our team structure, as we will likely have to all fill multiple roles instead of having a few dedicated roles. This will lead to less single role efficiency, but more experience and overall efficiency. If it comes to be an issue within the first few iterations, we will likely fall back on using dedicated roles.

5.2 Team Configuration

Management Plan:

1. All work products will be stored in a centralized git repository running on a central server.
2. All project (work products) items (documents, source code, test cases, program data, test data, etc) will be stored in a google repository 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.
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 through the prescribed change control procedures.
4. 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) if no one responds to the group within 2 days 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. At the end of the meeting a democratic vote will be held to decide whether or not the change should be allowed.

5. 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.

5.3 Verification and Validation Plan

The Verification and Validation plan is specified as a separate documented located in the version control system at: https://docs.google.com/spreadsheets/d/1Zsd4MVHymhwio_KB5GtIWTeE_PiBo5NMl5ujsZUjZII/edit#gid=2028977404

5.4 Product Acceptance Plan

The Commerce Budget Application has to be able to support individual and family accounts. Validation of individual accounts will occur by 3/18/2019 and family accounts by 4/1/2019. User accounts will be protected by a username and password. This will be completed by 4/1/2019. The user interface will be eye-catching and visually appealing and will be completed by 4/15/2019. When users access their accounts, the interface will provide a smooth transition with their Commerce account and have a straightforward, understated look and feel by the deadline of 5/5/2019. The application will offer more visual aid, rewards and offer different ways to track money by the validation date of 5/5/2019. The application will have at least 4 users by 5/5/2019.