

Software Project Management Plan

Bank Fundraiser Web Application

Team Members

Audrey Tsafack
Jessi Bautista Espino
Kyra Haake
Ricardo Alvarado

Document Control

Document Storage

This document is stored in the project's SVN repository at:

<https://github.com/baaau/Fundraiser-Website/tree/version-3.1>

Document Owner

Ricardo Alvarado 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.....	4
1.4	Assumptions and Constraints.....	5
1.5	Schedule and Budget Summary.....	5
1.6	Success Criteria.....	5
1.7	Definitions.....	6
1.8	Evolution of the Project Plan.....	6
2	STARTUP PLAN.....	7
2.1	Team Organization.....	7
2.2	Project Communications.....	7
2.3	Technical Process.....	7
2.4	Tools.....	7
3	WORK PLAN.....	7
3.1	Activities and Tasks.....	7
3.2	Release Plan.....	7
3.3	Iteration Plans.....	8
3.4	Budget.....	8
4	CONTROL PLAN.....	8
4.1	Monitoring and Control.....	8
4.2	Metrics Collection.....	8
5	SUPPORTING PROCESS PLANS.....	9

5.1	Risk Management Plan.....	9
5.2	Configuration Management Plan.....	9
5.3	Verification and Validation Plan.....	9

1 Overview

1.1 Purpose and Scope

The purpose of this project is to develop a web application in collaboration with Commerce Bank that enables users to create and manage their own fundraisers. This platform will also enable users to track their donations, as well as view and donate to other users' fundraisers. The application aims to provide a user-friendly experience that encourages users to engage in charitable causes and supports Commerce Bank's mission of giving back to the community.

The web application titled “Fund Your Mission” will give a chance to users to donate or create a fundraiser online. Users will be able to create an account, log in, or even donate without having to be registered. The platform will track donations, and money, increase a money bar, and will be able to customize donations after creation. The application will need a real email in order to reset a password if needed and fake card numbers will be used to ensure security and privacy of users. There will also be topics and categories based on the type of fundraiser.

The scope of the project is not to process real transactions, but rather to develop a platform that offers a smooth and secure user experience resembling the donation process.

1.2 Goals and Objectives

Project goals:

1. Establish a web application to create and donate to fundraisers.

Project objectives:

1. Create a database storing users, money, and fundraiser information.
2. Allow users to create a fundraiser with fake money.
3. Create an interface that is easy to use.

1.3 Project Deliverables

The following items will be delivered to the customer on or before 05/01/2023:

1. Requirement Docs
2. Project Plan Doc
3. Architecture
4. Test Documents:
 - a. Test plan
 - b. Test Cases
 - c. Test Report
5. Team Effort Estimation
6. User Guide
7. Project PowerPoint (With Video Demo)
9. GitHub link (source files, HTML, etc.)

1.4 Assumptions and Constraints

Assumptions:

1. Firebase will connect to UI.
2. Developers will start working after brainstorming.
3. Testing will be done after iteration.
- 4.

Constraints:

1. The software must run on a computer.
2. The database must be open source.
3. The software must be ready by 05/01/2023.

1.5 Schedule and Budget Summary

As this project is associated with the University of Missouri-Kansas City and utilizes entirely open-source resources, there is no need for a budget.

- Back-end infrastructure and database deployed – 03/14/2023
- Front-end complete UX design – 04/05/2023
- Back-end API application deployed, and methods created for web app information – 04/10/2023
- Front-end prototype completed – 04/29/2023
- Front-end enhancements completed – 04/29/2023

1.6 Success Criteria

- The web application uses Java and does not use a CMS.
- The database uses SQL Server 2012 or above
- Unit testing has 10% code coverage with the xUnit framework.
- Front-end uses 1 CSS framework and is included in the project.
- UX is usable by anyone, text is readable and properly aligned.

- There is a login page, home page/dashboard, donation form, and user profile and settings.
- Have the option for user if they forgot their username or password
- Have a session for remembering user if they close their browser and then log in again.

1.7 Definitions

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

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.

1.8 Evolution of the Project Plan

Before the start of an iteration, the project plan will be updated to include a schedule of detailed tasks for the upcoming iteration. At the conclusion of an iteration, the project plan will be updated to include the actual effort for each completed task.

Risk mitigation efforts will be evaluated at the start of each iteration. Severe risks will be analyzed and added to the project plan as soon as they materialize.

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.
Quality Assurance	
Analyst (2):	A professional who is responsible for monitoring, inspecting, and proposing measures to correct or improve an organization's final products in order to meet established quality standards. As well as help with documentation.

2.2 Project Communications

- Communication plans were made in Discord and in-person meet-ups.

2.3 Technical Process

The team plans to use a sprint-based development approach to create the application. Each sprint will last for two weeks, during which the team will allocate tasks and set specific goals to achieve by the end of the sprint. In addition, a meeting will be held halfway through each sprint to address any obstacles, queries, or concerns that may arise.

2.4 Tools

- Programming Language – Java, HTML, CSS, JavaScript
- Version Control – source code and written artifacts will be stored in a GitHub repository.
- Defect tracking – defects and issues will be tracked using GitHub.
- Automated testing – unit tests will be implemented with the JUnit testing framework.

3 Work Plan

3.1 Activities and Tasks

- Link to Team Effort Tasks
+ Team Effort Estimation

3.2 Release Plan

For day-to-day project management the release and iteration plans (described in the next section) are probably the two most important project management artifacts.

The release plan lists expected completion dates for major milestones and delivery dates of key work products. The project's technical development process to a certain extent will dictate the choice and timing of milestones and deliverables. For example, projects following the Rational Unified Process will have four major milestones: life-cycle objectives, life-cycle architecture, initial operation capability, and product release.

3.3 Iteration Plans

Team Effort Estimation

3.4 Budget

The budget is not applicable as it is a capstone university project.

4 Control Plan

4.1 Monitoring and Control

Partial Example

- Weekly – Team meeting. Project participants report status, progress and potential problems.
- 02/16/2023 – Critical Design Review. Formal inspection of product architecture.
- 04/28/2023 – Executive Review. The project manager presents current project status to project sponsor and senior executives.

4.2 Project Measurements

Phase	Measurement	Source
Release Planning	Record effort estimates for product features	Project MGR
Iteration Planning	Record effort estimates for scheduled tasks Update effort estimates for product features Update estimated dates in release plan	QA Tester
Iteration Closeout	Record actual effort for scheduled tasks Record actual effort for product features Record LOC count for modules written	Mgr/QA Tester
System Test	Record the rate at which errors are found.	DEV
Project Closeout	Archive project performance data in process database. (See process database definition for a list of measures to record.)	DEV

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
---------	---	------------

5 Supporting Process Plans

5.1 Risk Management Plan

Identify technical and managerial risks. Prioritize risks. Consider the probability of each risk turning into a problem and the likely consequences. For the highest priority risks, what actions will be taken to minimize the probability of the risk turning into a problem and the resulting consequences? What are the contingency plans for selected risks that do become a problem? Identify processes for monitoring risks and updating the risk management plan.

5.2 Configuration Management Plan

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

- 1) A centralized Github repository will serve as the storage location for all work products.
- 2) All project items, including documents, source code, test cases, program data, and test data, will be stored in Github and must undergo review by another team member before being added to the production code.
- 3) Items that are subject to change control will be considered baselined once they undergo a group review at the end of the life cycle phase in which they are created. Baselined items have undergone a formal review and can only be modified through the prescribed change control procedures.

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://github.com/baaau/Fundraiser-Website/tree/version-3.1>

5.4 Product Acceptance Plan

The product acceptance plan will be overviewed by the professor and Commerce Bank Team.