Mercy Clinic

Volunteer Solution
Software Development Plan
Version 1.6

Revision History

Date	Version	Description	Author
10/24/19	1.0	Initial Draft of Introduction section	Kenzie Clarke
10/24/19	1.1	Product Overview section	Cole Weber
10/24/19	1.2	Filled in Project Estimates (4.1)	Wynn Pho
10/27/19	1.3	Project Plan section (4.2)	Minh Nguyen
1/13/20	1.4	Resolved 2 comments from Dr. Wei Updated Evolution of Software Development Plan (2.4) Completed Section 3.1, 3.2 Updated & Completed Section 3.3	Kenzie Clarke
1/16/20	1.5	Updated & Completed 4.3	Wynn Pho
1/16/20	1.5.1	Resolved use case names	Cole Weber
03/30/20	1.6	Review grammar, format, & content Updated date & hosting server accordingly Updated table of contents page numbers	Minh Nguyen

Mercy Clinic

Table of Contents

Software Development Plan

1. Introduction

This Software Development Plan is used and maintained for the purpose of the volunteer solution for the non-profit clinic called Mercy Clinic, in Fort Worth, TX. This project is designed to provide a web-based platform and database solution for automating the volunteer scheduling and recording process. Mercy Clinic receives a range of over 150 volunteers per month of all ages, technological skills, and medical specialties.

This Software Development Plan is used by the Volunteer Solution team as a unified reference to the purpose and scope of the project, as well as containing important definitions, references, and acronyms. This document also detailed the overview of development, risk management, and the organizational structure of the team's project plans for all the software to be created as a solution for Mercy clinic.

1.1 Purpose

The purpose of the *Software Development Plan* is to gather all information necessary to control the project. It describes the approach to the development of the software and is the top-level plan generated and used by project managers to direct the development effort.

The following people use the *Software Development Plan*:

- The **project manager** uses it to plan the project schedule and resource needs, and to track progress against the schedule.

- **Project team members** use it to understand what they need to do, when they need to do it, and what other activities they are dependent upon.

- The **Professor** uses it to understand the team process and discern who is completing certain tasks. They use it to access the abilities of each team member.

1.2 Scope

This *Software Development Plan* describes the overall plan to be used by the Volunteer Solution project, including deployment of the product. The details of the individual iterations will be described in the Iteration Plans document.
The plans as outlined in this document are based upon the product requirements as defined in the *Vision Document* and Use Cases.

1.3 Definitions, Acronyms, and Abbreviations

See the Project Glossary.

1.4 References

For the *Software Development Plan,* the list of referenced artifacts includes:
- The *Vision Document* - Updated Oct 21th by Mercy Team
- The *Glossary* - Updated Oct 21th by Mercy Team

1.5 Overview

This *Software Development Plan* contains the following information:

Project Overview — provides a description of the project's purpose, scope, and objectives. It also defines the deliverables that the project is expected to deliver.

Project Organization — describes the organizational structure of the project team.

Management Process — explains the estimated cost and schedule, defines the major phases and milestones for the project, and describes how the project will be monitored.

Applicable Plans and Guidelines — provides an overview of the software development process, including methods, tools and techniques to be followed.

2. Project Overview

2.1 Project Purpose, Scope, and Objectives

The purpose of this project is to provide Mercy Clinic with an online and digital solution to their current paper scheduling and check in/out systems, which will reduce manual workload required to keep track of volunteer shifts and hours by hand. Our team will deliver a web based scheduling system that will allow volunteers to view and schedule appointments online and generate automatic email reminders, as well as a digital check in/out system to keep track of hours worked, as well as provide the ability to generate reports of these hours. In addition to these two systems, all required documentation to use and maintain them will be provided.

2.2 Assumptions and Constraints

2.2.1 We assume that any costs associated in the development and maintenance of our solution will be approved and covered by Mercy Clinic.

2.2.2 We assume that any further maintenance on the system beyond the complete release will be the responsibility of the Mercy Clinic staff.

2.2.3 We assume that any equipment necessary to support our system will be maintained by the Mercy Clinic staff.

2.2.4 We assume that volunteer information will be input into the system by the Mercy Clinic staff

2.3 Project Deliverables

Deliverables are delivered towards the end of the iteration, as specified in section *4.2.4 Project Schedule*.

2.4 Evolution of the Software Development Plan

The *Software Development Plan* will be revised prior to the start of each Iteration phase. The target dates for the end of each phase and iteration are shown below.

Inception	September 23rd – October 7th
Elaboration	October 14th – October 28th
Construction Iteration 1	November 4th – December 9th
Construction Iteration 2	December 23rd – January 6th
Construction Iteration 3	January 13th – February 10th
Deployment & Testing	February 17th – March 31st

3.   Project Organization

3.1   Organizational Structure

This team is comprised of a core of 5 full stack developers, each with expertise in certain areas that allows them to assist other members of the team. One full stack developer is also given the responsibility of the Team Lead, which requires them to monitor team progress in correlation to the project deadlines and organize tasks among the developers. The class professor assumes the role of the Project Manager, and is required to help the developer team stay on schedule and coordinate presentation and meeting dates.

3.2   External Interfaces

Deployment is handled by the developer team, and further maintenance is handled by the Mercy Clinic Staff. These individuals include:

Lorena Montalvo, email: lorenamontalvo@mercy-clinic.org
Alice Gee Lui, email: agliu@att.net

3.3   Roles and Responsibilities

Person	Rational Unified Process Role
Bingyang Wei, Project Manager	Project Manager Architecture Reviewer Requirements Reviewer Tester
Kenzie Clarke, Team Lead & Full Stack Developer	Program Manager Deployment Manager Configuration Manager Change Control Manager Front-end Developer Nodejs Developer Prototype Designer Code Implementer Code Reviewer Tester Presentation Speaker
Cole Weber, Full Stack Developer	Riogrande Website Manager Front-End Developer and to a lesser extent the following roles: Nodejs Developer Technical Writer Tester Code Reviewer
Wynn Pho, Full Stack Developer	Front-end Developer Nodejs Developer QA Tester User Interface Designer Prototype Designer Design Reviewer and to a lesser extent the following roles: Implementer Code Reviewer Test Designer Technical Writer
Nestor Guererro, Full Stack Developer	Front-end Developer

	Nodejs Developer Database Designer Presentation Speaker and to a lesser extent the following roles: Architecture Designer Code Reviewer Tester Technical Writer
Minh Nguyen, Full Stack Developer	Database Designer Database Manager, Front-end Developer Nodejs Developer and to a lesser extent the following roles: Code Reviewer Presentation Speaker Tester Technical Writer

4.   Management Process

4.1   Project Estimates

Estimated Cost

Currently, Mercy Clinic is using Springs Hosting to host their website, which comes at a cost of $30 every 3 months.

In the future, Text/Email Messaging feature will be implemented, which will require a third-party API. The cost to send a message from a virtual phone number using a messaging API averages at $0.007. Mercy Clinic may need a volume of 500 messages/month, making the cost $3.5/month.

A server is going to be needed to host the internal application. Heroku is the hosting server for the project and the pricing is based on multiple factors such as the user traffic and how big the data storage is. The estimated cost is $7/month.

Therefore, we estimate the cost for the application to be up and running to be approximately $10.5/month (not including the price Mercy Clinic currently pays for Springs Hosting).

However, as we have not finalized on what third-party services we want to use, re-estimation will occur when

1. We find a better pricing service
2. The website traffic increases/decreases
3. The number of volunteers changes significantly
   a. More storage is needed if there are more volunteers

Estimated Schedule

We plan to finish documentation and designing architecture by the end of November. We will start coding, testing and deploying starting from December 2019. By the end of March 2020, our product will be demoed to the clients. In April 2020, we will have the poster and the presentation ready to present to COSC faculty.

4.2   Project Plan

4.2.1   Phase Plan

Phase	Iteration/Deliverable	Description
Inception	Pre-iteration planning	The team met over the course of several weeks to discuss the overall schedule and divide tasks for each members
Elaboration	Uses Cases, Software Development Plan	The team came up with a number of suitable use cases together and each members picked two use cases to work on
Development	Iteration 1/Prototype	Two members of the team deliver a prototype and the rest give feedback
	Iteration 2	Develop UC01 Volunteer Schedules Appointment and UC02 Volunteer Cancels Appointment
	Iteration 3	Develop UC03 Admin Schedules Appointment and UC04 Admin Cancels Appointment
	Iteration 4	Develop UC09 User Selects Appointment and UC012 Admin Logs In
	Iteration 5	Develop the rest of the use cases, deploy the project
Testing	Iteration 6	Create and run tests for all use cases
Evaluation	Final Product	Deliver and present the product to the clients

4.2.2   Iteration Objectives:

4.2.2.1   Iteration 1 will focusing on developing the prototype

4.2.2.2   Iteration 2 will focus on developing UC01 Volunteer Schedules Appointment and UC02 Volunteer Cancels Appointment at low fidelity

4.2.2.3   Iteration 3 will focus on developing UC03 Admin Schedules Appointment and UC04 Admin Cancels Appointment at low fidelity and increasing the fidelity of UC01 Volunteer Schedules Appointment and UC02 Volunteer Cancels Appointment to high at iteration 2

4.2.2.4   Iteration 4 will focus on developing UC09 User Select Appointment and UC12 Admin Logs In at low fidelity and increasing the fidelity of UC03 Admin Schedules Appointment and UC04 Admin Cancels Appointment to high at iteration 3

4.2.2.5   Iteration 5 will focus on developing the rest of use cases to low fidelity and increasing the fidelity of UC09 User Selects Appointment and UC12 Admin Logs In to high at iteration 3

4.2.2.6   Iteration 6 will focus on increasing to the rest of use cases to high fidelity, creating and running test along will deploying to the web server

4.2.3   Releases

4.2.3.1   Demonstration 1 will involve a presentation to the Advisory Board members
4.2.3.2   Demonstration 2 will be a prototype (iteration 1) to be present to the clients
4.2.3.3   Beta 1.0 will involve a minimum viable product for the clients to test
4.2.3.4   Beta 2.0 will involve a near complete product with add-on features
4.2.3.5   Demonstration 3 will be a final product (presenting to clients as well as the Advisory Board members)

4.2.4   Project Schedule

Date	Name	Description
10/20/2019	Iteration 1	Prototype completed
10/25/2019	Demonstration 1	Present the project, process, plans and solutions to the Advisory Board members
10/31//2019	Demonstration 2	Present to the prototype to the clients and the instructor
11/6/2019	Iteration 2	Develop UC01 Volunteer Schedules Appointment and UC02 Volunteer Cancels Appointment
1/13/2019	Iteration 3	Develop UC03 Admin Schedules Appointment and UC04 Admin Cancels Appointment
1/13/2019	Iteration 4	Develop UC09 User Selects Appointment and UC012 Admin Logs In
2/10/2019	Iteration 5	Develop the rest of the use cases, deploy the project
02/17/2019	Iteration 6	Create and run tests for all use cases
02/27/2020	Beta 1.0	Deliver the minimum viable products to clients to test and receive feedback from them
03/1/2020	Beta 2.0	Deliver a near complete product with add-on features

03/15/2020		Test any changes and finalize the product
04/01/2020	Demonstration 3	Deliver a final product (presenting to clients as well as the Advisory Board members)

4.2.5 Project Resourcing

The team members and staff will remain the same through the course of development. Training for JavaScript framework is compulsory for every team member. Then, the team is required in training to creating, deploying the code to the server and maintaining the server

4.3 Project Monitoring and Control

Requirements Management

For controlling changes, after each iteration, we demo our working software to the clients and ask for feedback. Based on the feedback we gain from them, we have a better understanding of how the software should work, then we will tailor our solution if needed. We will detail the changes in our use cases and update other documents.

Schedule and Budget Control

Since we will host our solution on the clinic's website, there should be no cost from our end as of right now. If there is, we will update section 4.1 and 4.3 in this document.

Quality Control

All issues with the system will be recorded and addressed during the next iteration.

All deliverables are required to go through the appropriate review process, as described in the Development Case. The review is required to ensure that each deliverable is of acceptable quality. Each team member is responsible for testing and providing feedback. The review process will occur during team meetings when all members are present and able to contribute to the discussion.

Reporting and Measurement

After each iteration, we will update our schedule estimates and our reports. Reviewing the iteration plans after each iteration is essential as we have a better idea of what features are easy/hard to implement and what should be implemented before the others so that the coding procedure could go more smoothly.

Risk Management

Risks will be identified in Inception Phase using the steps identified in the RUP for Small Projects activity "Identify and Assess Risks". Project risk is evaluated at least once per iteration and documented in this table. The risks of the greatest magnitude are listed first in the table.

Risk Ranking	Risk Description	Mitigation Strategies
High	Our solution cannot be deployed on the hosting server that Mercy Clinic uses.	We host our solution on Heroku and the clinic may need to take charge of the monthly fees, which has been described in section 4.1.

Medium	A team member can no longer commit their time.	The missing team member's work is distributed among the remaining members.
Low	A team member's personal laptop is broken.	Senior Design's computers can be used.

Configuration Management

Appropriate tools will be selected which provide a database of Change Requests and a controlled versioned repository of project artifacts.

All source code, test scripts, and data files are included in baselines. Documentation related to the source code is also included in the baseline, such as design documentation. All customer deliverable artifacts are included in the final baseline of the iteration, including executables.

The Change Requests are reviewed and approved by one member of the project, the Change Control Manager role.

Full backups are performed monthly and increments are performed nightly.

5.   Annexes

The project will follow the RUP for Small Projects process, as tailored by the project Development Case.

Other applicable process plans are listed in the references section, including Programming Guidelines.