Project: Saphira (Asset Management)

Web Application Development Project

Student: Devin Campbell

Course: CIS-355, Winter 2017 Section: 90

This document is confidential, to be viewed only by team members.

Contents

[Introduction 2](#_Toc475519107)

[Project Team 2](#_Toc475519108)

[Hard Deadline: April 18 2](#_Toc475519109)

[1. Project Overview 2](#_Toc475519110)

[2. Technical Specifications and Diagrams 2](#_Toc475519111)

[Entity Relationship Diagram 2](#_Toc475519112)

[Screen Flow Diagram 2](#_Toc475519113)

[Wireframe Diagrams 3](#_Toc475519114)

[Use Case Diagrams 3](#_Toc475519115)

[Source Code on Github 3](#_Toc475519116)

[3. Programmer System Test Plan and Results 3](#_Toc475519117)

[4. User System Test Plan and Results 4](#_Toc475519118)

[5. Project Plan and Status 5](#_Toc475519119)

[6. Document Update Log 6](#_Toc475519120)

# Introduction

The purpose of Project **Saphira** is to implement a website, which accomplishes the following:

Manage users to log entries, Manage Equipment inventories, Schedule and view equipment maintenance records. **This document describes a one-semester project.**

## Project Team

The personnel involved in the project are as follows.

* Devin Campbell, Lead Programmer. The Lead Programmer will design and code the system in C#, ASP.Net, Entity Framework, and SQL code. The Lead Programmer will maintain this document throughout the development process.
* Ashley Ames, Code Reviewer. This project must have at least one fellow student code reviewer.
* User Tester. None. This project may or may not have user testers.

## Hard Deadline: April 18

This project will be complete by Tuesday, April 18. The Lead Programmer will provide an in-person demonstration or YouTube video demonstration on or before that date.

# 1. Project Overview

This project is a test environment for what will become a larger, commercial quality asset management software to be used at the Bay County Fire Department. The main use of this software will be to manage employees in charge of asset management, the equipment to be managed, and the logs of which equipment had been maintained and how.

This software is to be a custom software requested by the client and its target audience is the staff of the client

# 2. Technical Specifications and Diagrams

Write HOW the system will work. You must explain what the final system will look like when it is done.

## Entity Relationship Diagram



## Screen Flow Diagram



## Wireframe Diagrams

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

## Use Case Diagrams

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Source Code on Github

\*\*\* <https://github.com/Campbellds1991/Saphira> \*\*\*

# 3. Programmer System Test Plan and Results

Enumerate what was tested and what was not tested.

Use a table like this.

| System Function: Condition Tested | Tested By | Date | Results |
| --- | --- | --- | --- |
| Login: Successful |  |  | Not Implemented |
| Login: Unsuccessful |  |  | Not Implemented |
| New Registration: Successful |  |  | Not Implemented |
| New Registration: Unsuccessful |  |  | Not Implemented |
| Display Main Menu | Devin C. | April 5, 2017 | Passed |
| Display Table1 List: No Records in DB | Devin C. | April 5, 2017 | Passed |
| Display Table2 List: No Records in DB | Devin C. | April 5, 2017 | Passed |
| Display Table3 List: No Records in DB | Devin C. | April 5, 2017 | Passed |
| Add 2 Records to Table1 | Devin C. | April 5, 2017 | Passed |
| Add 2 Records to Table2 | Devin C. | April 5, 2017 | Passed |
| Add 2 Records to Table3 | Devin C. | April 5, 2017 | Passed |
| Edit Record in Table 1 | Devin C. | April 5, 2017 | Passed |
| Edit Record in Table 2 | Devin C. | April 5, 2017 | Passed |
| Edit Record in Table 3 | Devin C. | April 5, 2017 | Passed |
| Delete Table 2 Record (Cascades to Table 3) |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 4. User System Test Plan and Results

Enumerate what was tested and what was not tested.

Use a table like this.

| System Function: Condition Tested | Tested By | Date | Results |
| --- | --- | --- | --- |
| Login: Successful |  |  |  |
| Login: Unsuccessful |  |  |  |
| New Registration: Successful |  |  |  |
| New Registration: Unsuccessful |  |  |  |
| Display Main Menu |  |  |  |
| Display Table1 List: No Records in DB |  |  |  |
| … |  |  |  |

# 5. Project Plan and Status

The following table shows the activities and deliverables of the project.

| Task | Performed By | Status | Date |
| --- | --- | --- | --- |
| Determine scope |  |  |  |
| Write Project Plan |  |  |  |
| Establish web hosting  *Files at* ***saphira.azurewebsites.net*** |  |  |  |
| Establish private github repository  *Private repo:* ***https://github.com*/*Campbellds1991/Saphira*** |  |  |  |
| Write sections 1 and 2 of this document |  |  |  |
| Design data tables |  |  |  |
| Implement data tables in MySQL |  |  |  |
| Design screen flow |  |  |  |
| Design screens (wireframes) |  |  |  |
| Implement screens/flows in PHP: Table1 (CRUD\*) |  |  |  |
| Code review #1 |  |  |  |
| Implement screens/flows in PHP: Table2 (CRUD\*) |  |  |  |
| Implement screens/flows in PHP: Table3 (CRUD\*) |  |  |  |
| Implement file upload/download on Table3 (invoice) |  |  |  |
| Implement login/registration |  |  |  |
| Implement dropdown list boxes on Table3 |  |  |  |
| Code review #2 |  |  |  |
| Make all PHP code object oriented |  |  |  |
| Make all database calls using AJAX |  |  |  |
| Implement API that returns JSON object |  |  |  |
| Prepare system test plan |  |  |  |
| Execute system test plan |  |  |  |
| Code/test review #3 |  |  |  |
| Prepare user test plan |  |  |  |
| Execute user test plan |  |  |  |
| Implement CodeIgniter mini-version of system   * ***Cannot implement – ASP.Net application*** | N/A | Rejected | 04-05 |
| **Present completed system** |  |  | 04-18 |
| Review/report status using this document |  |  |  |

\* CRUD stands for create, read, update, delete

# 6. Document Update Log

The following table logs changes made to this document.

| Version (yyyymmdd#) | Editor | Description of change |
| --- | --- | --- |
| 20170222a | George | Prepared Template |
| 20170405a | Devin | Initial Status Update |
|  |  |  |
|  |  |  |
|  |  |  |