

***Software Engineering***

***Typical Nerd Stuff Team***

***Software Requirements Specifications (SRS) Document***

*Tom Bowker*

*Cassandra Brewer*

*Michael Pontius*

*Kelson Vella*

*Weber State University*

*September 16, 2017*

Table of Contents

[Table of Contents 0](#_Toc501212516)

[1. Introduction 0](#_Toc501212517)

[1.1 Purpose 0](#_Toc501212518)

[1.2 Scope of Project 0](#_Toc501212519)

[1.3 Overview 0](#_Toc501212520)

[1.4 Definitions, Acronyms, & Abbreviations 0](#_Toc501212521)

[2. General Description 0](#_Toc501212522)

[2.1 Product Perspective 0](#_Toc501212523)

[2.2 Product Functions: 0](#_Toc501212524)

[2.3 User Characteristics 0](#_Toc501212525)

[2.4 User Problem Statement 0](#_Toc501212526)

[2.5 User Objectives 0](#_Toc501212527)

[2.6. General Constraints 0](#_Toc501212528)

[3. Functional Requirements 0](#_Toc501212529)

[3.1 Critical Requirements 0](#_Toc501212530)

[4. Interface Requirements 0](#_Toc501212531)

[4.1 User Interfaces 0](#_Toc501212532)

[4.2 Hardware Interfaces 0](#_Toc501212533)

[4.3 Communication Interfaces 0](#_Toc501212534)

[4.4 Software Interfaces 0](#_Toc501212535)

[4.5 Performance Requirements 0](#_Toc501212536)

[5. Other Non-Functional Attributes 0](#_Toc501212537)

[5.1 Security 0](#_Toc501212538)

[5.2 Binary Compatibility 0](#_Toc501212539)

[5.3 Reliability 0](#_Toc501212540)

[5.4 Maintainability 0](#_Toc501212541)

[5.5 Portability 0](#_Toc501212542)

[5.6 Extensibility 0](#_Toc501212543)

[5.7 Reusability 0](#_Toc501212544)

[5.8 Resource Utilization 0](#_Toc501212545)

[5.9 Serviceability 0](#_Toc501212546)

[6. Analysis Models 0](#_Toc501212547)

[6.1 Use Case Model 0](#_Toc501212548)

[6.1.1 User Use Case Scenario: View Site Content 0](#_Toc501212549)

[6.1.2 Admin Use Case Scenario: View Site Content 0](#_Toc501212550)

[6.1.3 Admin Use Case Scenario: Login 0](#_Toc501212551)

[6.1.4 Admin Use Case Scenario: Logout 0](#_Toc501212552)

[6.1.5 Admin Use Case Scenario: Edit Page Content 0](#_Toc501212553)

[6.2 Sequence Diagrams 0](#_Toc501212554)

[6.2.1 User Sequence: 0](#_Toc501212555)

[6.2.2 Administrator/Employee Sequence: 0](#_Toc501212556)

[6.3 Class Diagram 0](#_Toc501212557)

[7. Database Schema 0](#_Toc501212558)

[8. Potential Features and Bugs 0](#_Toc501212559)

[9. Client Meetings and Documentation 0](#_Toc501212560)

[9.1 Group Meetings 0](#_Toc501212561)

[9.2 Client Meetings 0](#_Toc501212562)

1. Introduction

1.1 Purpose

The purpose of this document is to layout our goals and detailed information of the data management software for Youth Futures Shelter Homes. This will explain, in detail, features that will let YF edit and update their content. With that in mind, this document will provide every detail on YF website content manager. These details include features, constraints and schemas that lay out each process. This document is intended for the YF foundation administrators and the developers. This document is also intended for YF foundation as the main audience. Faith Satterwaite will be the final editor and will oversee this project.

1.2 Scope of Project

This document will outline each part and phase of the content management software. This software will allow Youth Futures Shelter Homes administrator update and edit their new website. Also, as stated, this will be a new website that will provide these rights of editing and updating. This editing process will be communicating to a database where the data will be stored and updated. This will require us to develop and build a database. By allowing the Youth Futures Shelter Homes administrators rights to update the content of the website, the organization will no longer need to obtain assistance to update information in their database or obtain assistance to update the website.

Overall, this content management system will be will be a component that will lead to a more modern website that will grant the power to edit and update their website in a more streamlined and sophisticated way.

The constraints of the project will include:

* Scheduling time to block off to achieve our goals.
* Creating a relational database that will store and update the content
* Obtaining Skill and research required to complete this project,

1.3 Overview

After the project is complete, Youth Futures will have a modern updated website, that can be edited, coupled with dynamic content population from the newly created relational database. The project will be considered complete after the following steps are accomplished:

1. Authorization through login page
2. Page editing ability by section for users
3. Database creation and population of existing data
4. Current code updated to match Model View Controller design pattern

1.4 Definitions, Acronyms, & Abbreviations

|  |  |
| --- | --- |
| **MVC** | Model View Controller - code architecture that separates a project into three parts that interact with each other |
| **MSSQL** | Microsoft SQL Server - Microsoft's relational database management system |
| **Database** | A collection of all relevant information stored in appropriate tables/locations used by the project |
| **Population** | The process of accessing and displaying information from the database on the webpage |
| **Authorization** | The process of establishing the identity of a user through their login information |
| **CSS** | Cascading Style Sheet - used by HTML to display page elements in a certain style (fonts, colors, etc.) |
| **C#** | A coding language from Microsoft that is essentially a combination of the languages C++ and Java |
| **HTML** | Hypertext Markup Language - the language used to develop web pages |
| **JavaScript** | A coding language used with HTML to create effects on pages |
| **YF** | Youth Futures |

2. General Description

2.1 Product Perspective

Youth Futures’ current site offers minimal features, difficult navigation, and no easily editable content. Many other similar programs’ web pages are fresh, with high ease-of-use. However, no other comparable site has a login feature for administrator editing capability. This will be a helpful and unique additional feature for YF’s website. The new YF website will also have greater dynamic capabilities than other offerings, more extensive program information, as well as a sleeker, more modern design.

2.2 Product Functions:

The system will support an editable view, where the user has the ability to edit certain sections on the website, and the modification will be saved to the database directly when those changes are saved. It will also have the ability to update and populate certain blocks of information such as staffs and donors automatically from the database. The user will be required to login first before he/she can enter the editable mode, and the user will be provided an email to login to the system. All of these functions will be combined with a new, updated user interface for a finished, productive, user-friendly, and attractive content management system.

2.3 User Characteristics

The board of directors and certain authorized staffs at Youth Futures Shelter Home will be the users of this content management system. The user will be responsible for updating the information on the website in editable view as needed, and the changes will be saved to database directly when the administrator chooses to save the changes. The user does not need a lot of experience with computer software systems to update the information on the website, as the system will be designed to be user-friendly and straightforward. All other users without authorization are considered limited permission users of the system, as the content on the website is read-only to them. These non-authorized users’ primary objective is to navigate through the website, and obtain information regarding Youth Futures Shelter Home.

The user will generally have some web-surfing experience, is expected to be able to competently edit/update information in this content management system. All the main screens of the Youth Futures Shelter Home website will contain a login button. The user will then be able to click the “Login” button that will take them to the login page. The user is expected to know how to log in. Each editable section in the editable view will have an “Edit” button next to it. The user will be able to click the “Edit” button and then edit the information. The user will then be able to choose if they want to choose whether to save the changes by clicking the “Save” button or discard the changes by clicking the “Cancel” button.

2.4 User Problem Statement

The old system is outdated and the client could not update website content. Ineffective and inconsistent navigation structure throughout website and overall content required improved organization. The goal is to create a modern website layout to promote client-side information and intuitive navigation structure and to allow a login screen for easier system management.

2.5 User Objectives

This new system will provide a client-side application and an admin-side application which will support and interact with various admin-side features. The system project will be designed to maximize the admin productivity by providing tools to assist in editing the website content, which would otherwise have to be performed manually. The content management system will be database driven and admin users will have the ability to edit the database information without outside assistance. The project encompasses both client and admin-side functionalities and will provide an updated and more attractive user interface.

2.6. General Constraints

The following constraints may present possible problems:

* Cascading Style Sheets (CSS) and other functionality must be compatible across Internet Browsers.
* The system must be database driven.
* The system must be created in C# with the MVC framework
* The current HTML, CSS and JavaScript code must be updated to work with the system.

The following constraint are not deemed major problems:

* The system must be completed by December 10, 2017

3. Functional Requirements

3.1 Critical Requirements

**Critical Items Scale (1-5):**

1. **Very Low:** Items that can be eliminated easily if needed.
2. **Low:** Items that provide extra functionalities but can be evaluated for elimination if needed.
3. **Medium:** Items that are desired strongly by system users.
4. **High:** Items which are required for lower items to function correctly.
5. **Very High:** Items which are critical to the functionality of the system.
6. **The system will display a login page.**

* Description

After proper coding, the system will display a login page across multiple Internet browsers after the user click the login button on the websites.

* Critical Level

**5**

* Technical Issues

Pre-condition: website must be coded correctly.

Post-condition: the system will correctly display the login page.

* Risks

With many Internet browsers, the login page may not display exactly the same (i.e. the input field length, the fonts that the login page use)

* Dependencies with other requirements

The login button will need to be added

1. **The system will display an editable view of the website**

* Description

After proper coding, the system will display an editable view across multiple Internet browsers.

* Critical Level

**5**

* Technical Issues

Pre-condition: website must be coded correctly and the login page must be properly display and function

Post-condition: the system will correctly display the editable view

* Risks

With many Internet browsers, the editable view may not display properly.

* Dependencies with other requirements

N/A

1. **The system will validate the user's email/username**

* Description

After user has entered their email/username, the system will validate if this user has entered the correct email/username that matches the password in the database

* Critical Level

**5**

* Technical Issues

Pre-condition: website must be coded correctly and login page is displayed.

Post-condition: the system will either accept the user’s credentials, or reject them and prompt for reentry.

* Risks

System may accept the incorrect user email/username and password combination or may reject a correct one

* Dependencies with other requirements

1, 4

1. **The system will validate the user’s password**

* Description

After the user has entered their email/username and password, the system will validate if the user has entered the correct password that matches the email/username in the database

* Critical Level

**5**

* Technical Issues

Pre-condition: website must be coded correctly and login page is displayed

Post-condition: the system will either accept the user’s credentials, or reject them and prompt for reentry

* Risks

System may accept the incorrect email/username and password combination or may reject a correct one

* Dependencies with other requirements

1, 3

1. **The system will validate if the user who is logging in is a human via CAPTCHA**

* Description

After the user has entered their email/username and password combination, the system will prompt the user to solve a CAPTCHA as a way to verify if this user is not a robot

* Critical Level

**4**

* Technical Issues

Pre-condition: website must be coded correctly, login page is displayed, and the code for displaying the CAPTCHA and the verification for solution of the CAPTCHA from the user must be coded correctly

Post-condition: the user is able to see the display of the CAPTCHA and is able to solve the CAPTCHA

* Risks

System may approve the user to continue logging in even though they have not solved the CAPTCHA or have not solved the CAPTCHA in a correct way

* Dependencies with other requirements

1, 3, and 4

1. **The user will be able to log in to the system**

* Description

After the user access the Youth Future Shelter Home website, the user is able to click on a “Log In” button in an eye-catching position on the website to log in to the system

* Critical Level

**5**

* Technical Issues

Pre-condition: the website must be coded correctly with a visible log in button

Post-condition: the user will be able to update/edit the content of the website and the modification will be saved to the database

* Risks

The system may fail to authenticate the log in because of an incorrect username/password combination. The system will then prompt for credential reentry. Or the user may not solve the CAPTCHA correctly. The system will then prompt the user for solving the CAPTCHA again.

* Dependencies with other requirements

1, 3, 4, and 5

1. **The user will be able to log out of the system**

* Description

After the user has successfully logged in, he/she will be able to log back out of the system.

* Critical Level

**5**

* Technical Issues

Pre-condition: the user must have logged in to the system, the website must be coded correctly with a visible “Log out” button

Post-condition: the user will no longer be able to update/edit the content on the website until he/she logs in again

* Risks

The system may fail to logout the user.

* Dependencies with other requirements

1, 3, 4, and 5

1. **The user will be able to update/edit the information through the user interface**

* Description

After the user has logged in to the system, he/she will be able to update/edit the content, and any modifications that the user made via UI will be saved to the database.

* Critical Level

**5**

* Technical Issues

Pre-condition: the user must be logged in and navigate to sections that are editable, and the user will choose to save the changes.

Post-condition: the changes will be saved to database and presenting on the website.

* Risks

The system may not reflect the user’s changes.

* Dependencies with other requirements

1, 2, 3, 4, and 5

1. **The user will be able to save the changes**

* Description

After the user makes changes to the section, the user will be able to save the changes by clicking a “Save” button, and the changes will be saved to the database and presenting on the website.

* Critical Level

**5**

* Technical Issues

Pre-condition: the user must be logged in and navigate to a section that is editable, and the website must be coded correctly with a visible “Save” button.

Post-condition: the user will be able save the changes that he/she made and the changes will be saved to the database and presenting on the website.

* Risks

The code may not be coded correctly resulting that the “Save” button is not visible. The changes that the user made may not be saved to the database successfully.

* Dependencies with other requirements

1, 2, 3, 4, and 5

1. **The user will be able to cancel/discard the changes**

* Description

After the user makes changes to the section, the user will be able to discard the changes by clicking a “Cancel” or “Discard” button, and the changes will not be saved to the database and neither presenting on the website.

* Critical Level

**5**

* Technical Issues

Pre-condition: the user must be logged in and navigate to a section that is editable, and the website must be coded correctly with a visible “Cancel” or “Discard” button.

Post-condition: the user will be able to discard any changes that he/she made and the changes will not be saved to the database neither presenting on the website.

* Risks

The code may not be coded correctly resulting that the “Cancel” or “Discard” button is not visible. The changes that the user made may get saved to the database successfully and present on the website.

* Dependencies with other requirements

1, 2, 3, 4, and 5

1. **The system will be able to update the database based on the changes that the user made**

* Description

After the user chooses to save his/her changes to the content of a section, the corresponding data in the database will be updated as well.

* Critical Level

**5**

* Technical Issues

Pre-condition: Set up a MS SQL database to store the system’s information, and the system is setup to utilize the new database

Post-condition: any changes that the user made on the website will be updated to the database directly

* Risks

The relational database may not be designed correctly to hold the information in the system

* Dependencies with other requirements

N/A

1. **The user will be able to navigate to certain section that they want to make changes quickly**

* Description

After the user logged in to the system, he/she is able to navigate to certain section that he/she wants to edit quickly

* Critical Level

**4**

* Technical Issues

Pre-condition: the user will need to log in to the system

Post-condition: the user is able to navigate to a specific section quickly to do the modification

* Risks

The user may not be navigated to the correct section

* Dependencies with other requirements

1, 3, 4, and 5

4. Interface Requirements

4.1 User Interfaces

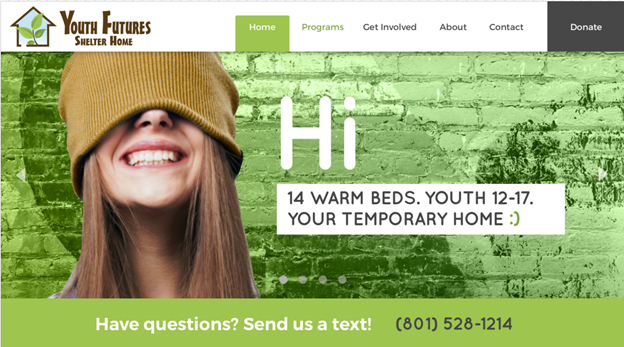
* **4.1.1 GUI**

The user interface will follow a new color scheme; however, content of the user interface will be like the old website. The new pages will be an updated in both functionality and styling. This includes a new style, graphics, colors, logos and menu bar as well as better navigation.

**Home Page: Before**

****

**Home Page: After**

****

**About Page: Before**

****

**About Page: After**



****

**Our Purpose Page: Before**

****

**Our Purpose Page: After**

****

**Donation Page: Before**

****

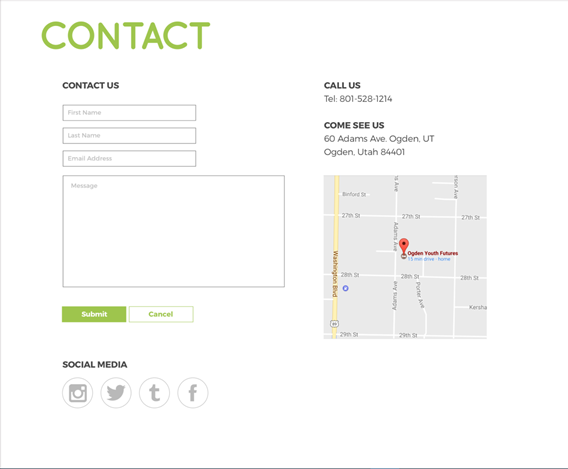
**Donation Page: After**



**Contact Page: Before**

****

**Contact Page: After**



**New pages that will be added:**

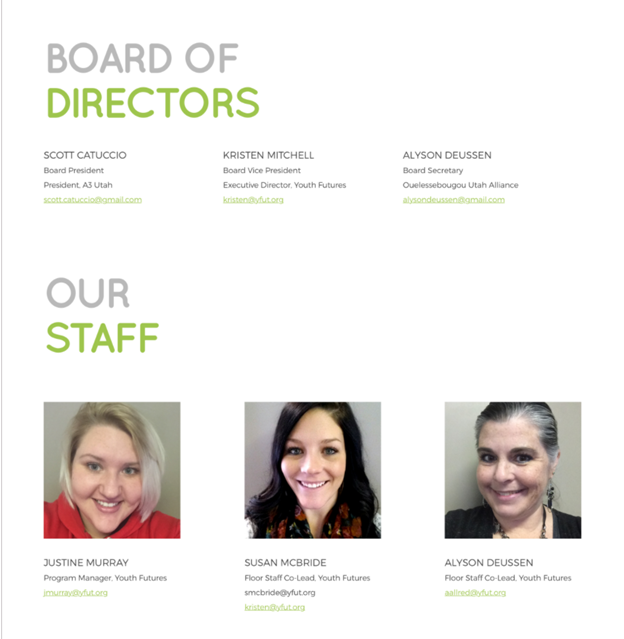
**Interactive Calendar**

****

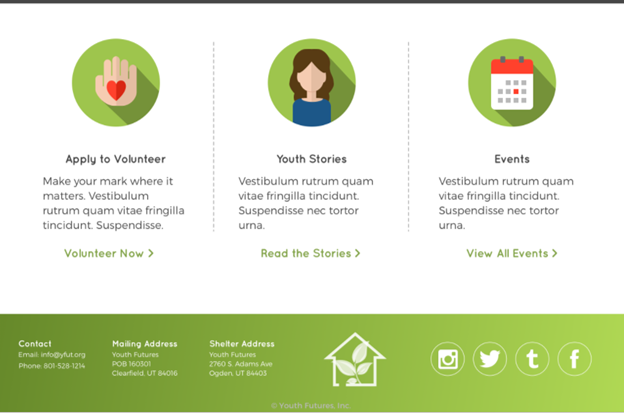
**Donors List**

****

**Staffs and Board of Directors List**

****

**Additional Front-Page Material**

****

* **4.1.2 CLI**

The server will not provide any specific type of command line interface.

* **4.1.3 API**

N/A

* **4.1.4 Diagnostics**

The system will have error-handling built in. When an error is displayed it will follow with a short description of the error. The system’s admin may be contacted for further information on the error.

4.2 Hardware Interfaces

The system may be used with any computer with mouse and keyboard internet. The system may also be used with a smartphone or tablet, although the functionality may be limited by doing so. The computer or other device the user chooses to access the system with must also have the capability of accessing the internet via a web browser.

4.3 Communication Interfaces

Users must have an active internet connection to connect to the website and use the system.

4.4 Software Interfaces

The system will require an operating system capable of running a modern up-to-date web browser as well as said browser to view properly.

4.5 Performance Requirements

The system will be coded to work with most web browsers but 100% compatibility cannot be guaranteed. For optimal functionality and viewing capabilities users should have one of the following browsers installed and updated:

* Firefox 4.0 and up
* Google Chrome 10.0 and up
* Internet Explorer 8 and up
* Safari 5.03 and up

5. Other Non-Functional Attributes

5.1 Security

The Website will have its own security to prevent unauthorized *editing* and *deleting* access. There is no restriction on *read* access; however, users – with exception of the administrator – will not have access to modify any content within the system. There will be only one administrator permission level in this system. Administrators will have the ability to log in with validation to prove the user is valid (e.g. CAPTCHA to prove that this user is not a robot).

5.2 Binary Compatibility

The system will be written in C# with the MVC framework. It is a web-based application and should communicate between Windows and Mac operating systems. The system will be accessible from any computer device with an Internet connection. The following browsers are preferred for optimal viewing and functionality: Internet Explorer, Mozilla Firefox, or Google Chrome.

5.3 Reliability

The system will reside on one of Youth Futures Homeless Shelter servers and will be accessible at all times, with exception to infrastructure failure, or system maintenance.

5.4 Maintainability

The system will be maintained by the administrator. There is very little maintenance that will need to be supplied. Most of the maintenance involved will be the updating and editing of the database, which the administrator can do through the admin page.

5.5 Portability

The system is highly convenient, as it will be available to any user who possesses a functioning Internet connection and with an updated browser.

5.6 Extensibility

The code shall become the property of Youth Futures Homeless Shelter, and may be extended as necessary for future use.

5.7 Reusability

The system will be well-documented and developed with N-Tier architecture, allowing for reuse of code as needed.

5.8 Resource Utilization

In order to run, the system will require an active server with adequate memory and hard drive space, which will be supplied through Youth Futures Homeless Shelter. With these resources in place, the system will be accessible from any type of computer with a functioning Internet connection.

5.9 Serviceability

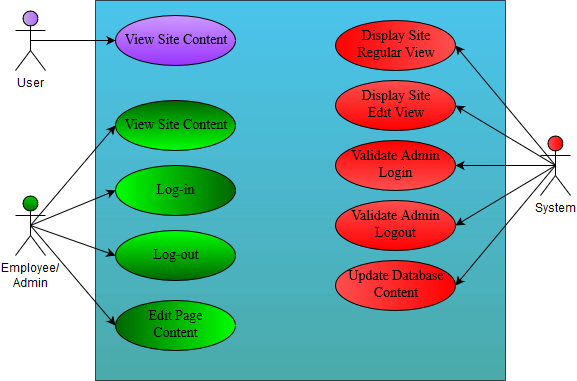
The system will be well-documented, allowing for easy service of the code as needed.

6. Analysis Models

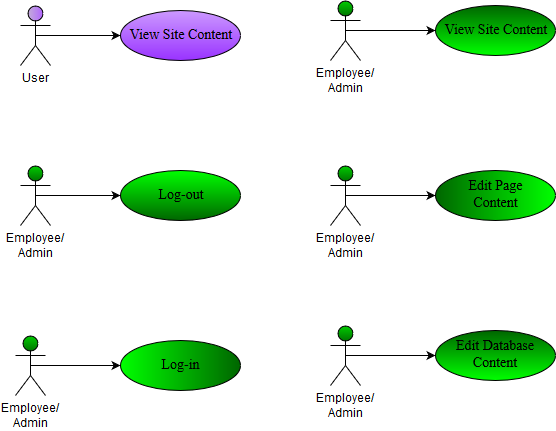
6.1 Use Case Model

There are two different types of users for the new Youth Futures website and associated data management system: (1) a typical non-employee user, (2) an employee/content administrator. The outside user will be able to view all the pages of the site and their contents, but will not be able to log-in or access the editing capabilities. An employee/admin will also be able to view the pages and contents, but will also be able to log-in, log-out, edit pages, and edit data. The system itself will display regular site pages and content, display editable site pages and content, validate employee logins/logouts, and update the database itself if given new information.

***Use Case Model Diagram:***



6.1.1 User Use Case Scenario: View Site Content



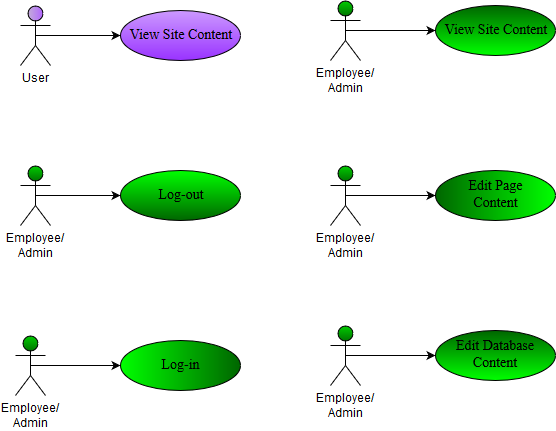
**Summary:**

User navigates to the Youth Futures webpage, views Home page information, navigates to relevant content using page tools/controls.

**Step-By-Step Description**

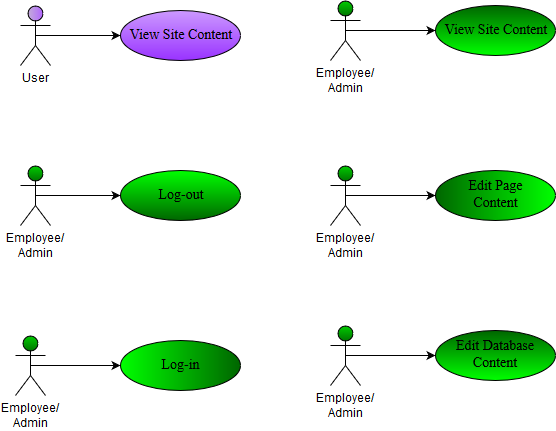
1. User types URL into address bar or clicks on result from a search.
2. YF homepage is displayed to user.
3. The user processes the content’s information.
4. The user may navigate to a secondary page of the site.

6.1.2 Admin Use Case Scenario: View Site Content



*Please see “User Use Case Scenario: View Site Content” (above)*

6.1.3 Admin Use Case Scenario: Login



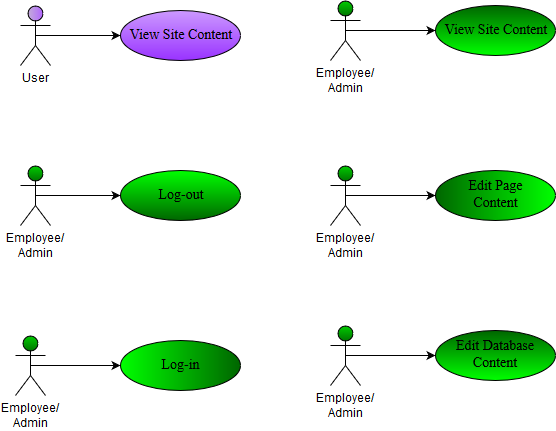
**Summary:**

The employee/administrator logs in with valid credentials to utilize page editing functions by clicking the “Login” button.

**Step-By-Step Description**

1. The employee clicks the “Login” button at the top-right of the page.
2. A window appears asking for username and password.
3. Employee enters and submits their information.
   1. The username and password are correct:
      1. The window closes and the employee is redirected to the edit version of the site.
   2. The username and/or password are not correct:
      1. The window displays an error message and the employee is prompted to enter their credentials again.
      2. On third failure, the employee must pass a CAPTCHA test in addition to re-entering their credentials.

6.1.4 Admin Use Case Scenario: Logout



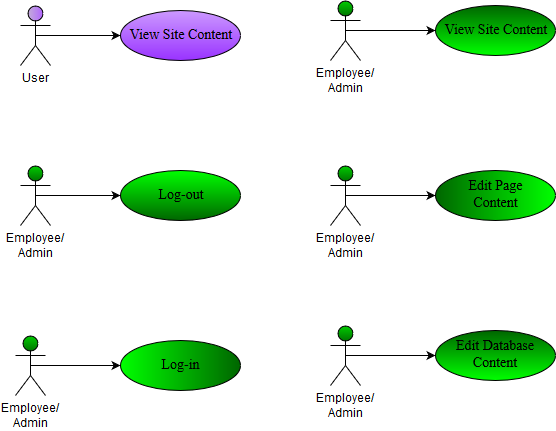
**Summary:**

The employee/administrator logs out of the system by clicking the “Logout” button, disabling any editing features.

**Step-By-Step Description**

1. The employee clicks the “Logout” button at the top-right of the page.
2. A window appears asking for confirmation of the logout.
3. The employee clicks the “Yes” button.
   1. The system successfully logs the employee out.
      1. The window displays a success message.
      2. The employee clicks the “Ok” button.
   2. The system unsuccessfully logs the employee out.
      1. The window displays an error message.
      2. The employee clicks the “Try Again” button.
      3. Repeat until 3a is reached.

6.1.5 Admin Use Case Scenario: Edit Page Content



**Summary:**

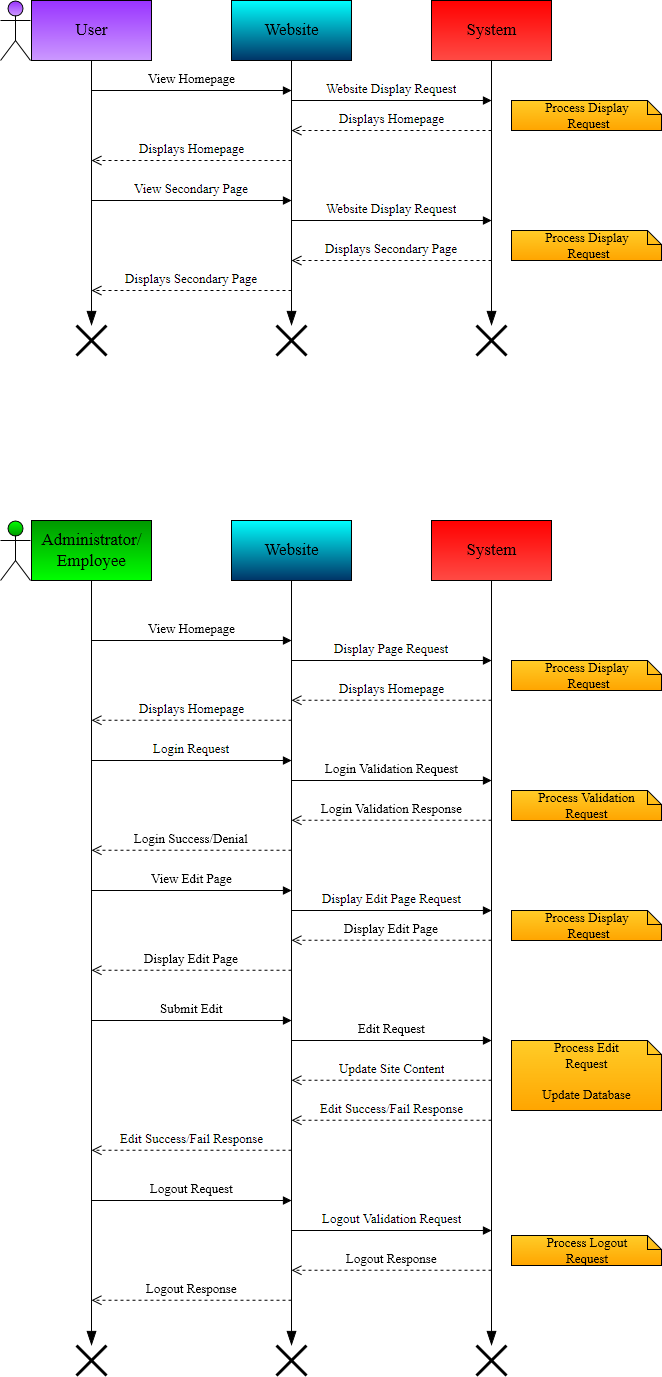
After logging into the site, an employee/administrator edits the content displayed on the current page, then saves their changes.

**Step-By-Step Description**

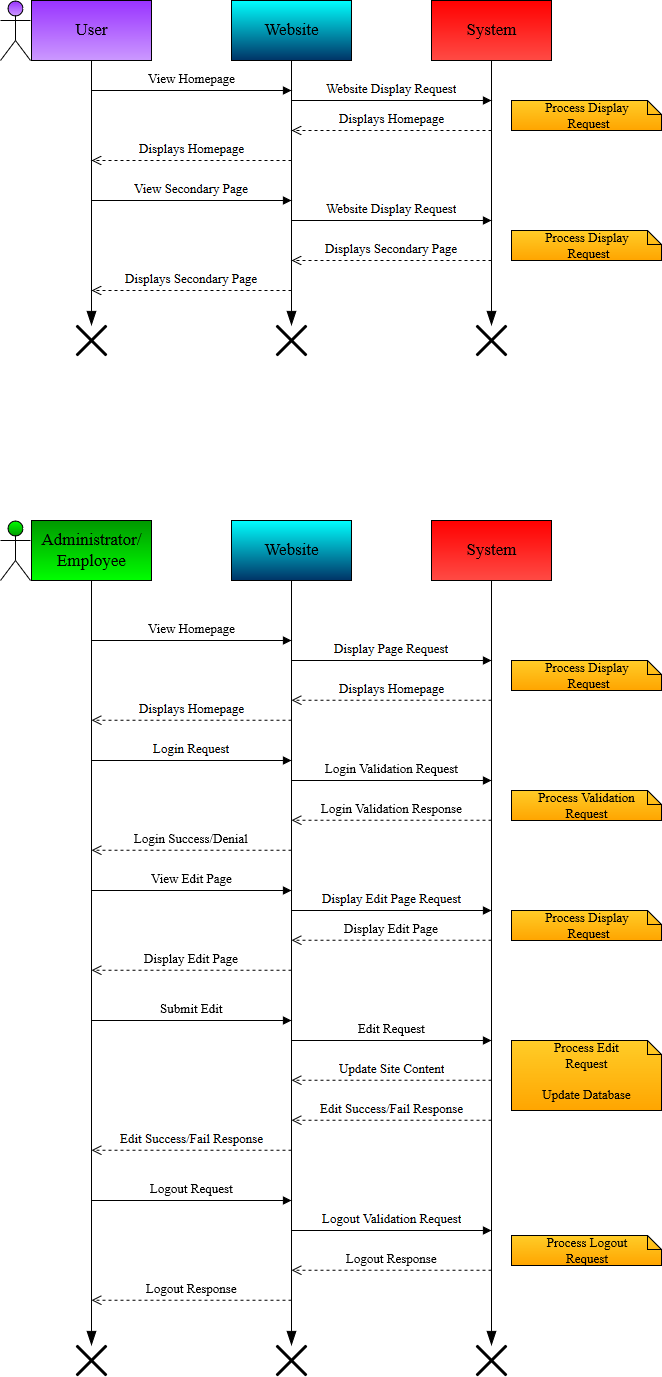
1. The employee clicks the “Edit” button next to the section of the page they would like to edit the content of.
2. The employee chooses/adds new content to be displayed on the page.
   1. The employee clicks the “Save Changes” button at the bottom of the section they are currently editing.
      1. Move to step 3.
   2. The employee clicks the “Cancel” button at the bottom of the section they are currently editing.
      1. Back to step 1.
3. The changes are submitted to the system.
   1. The system updates the content accordingly.
   2. The system does not update the content and displays an error message.

6.2 Sequence Diagrams

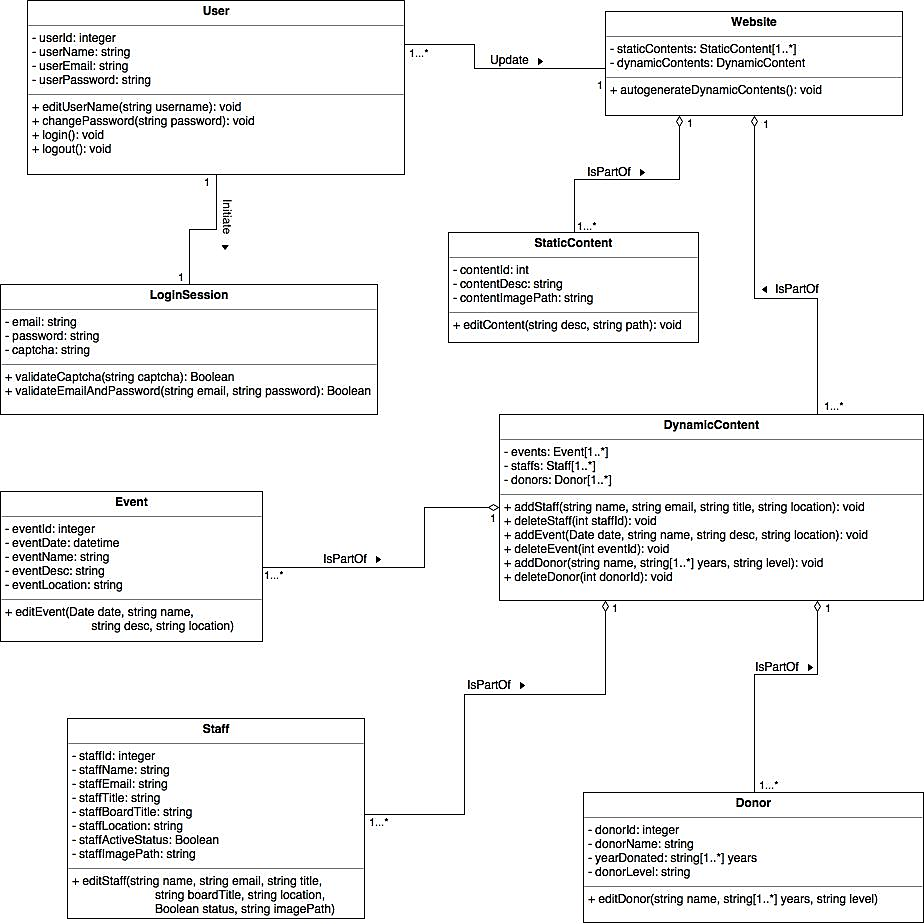
6.2.1 User Sequence:



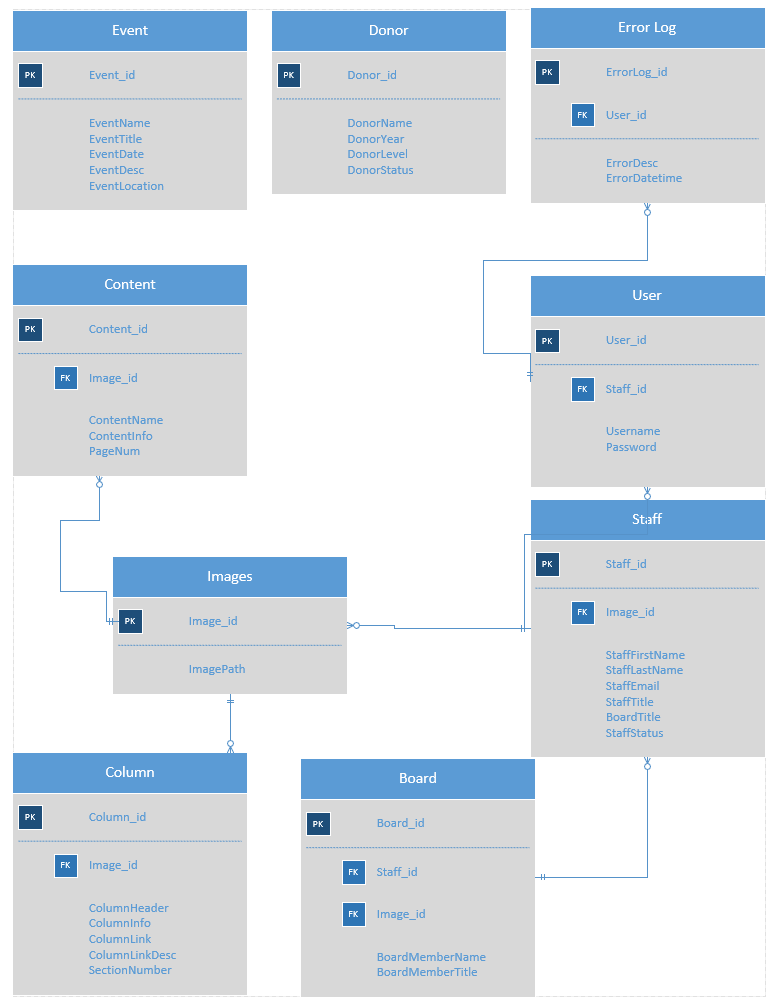
6.2.2 Administrator/Employee Sequence:



6.3 Class Diagram

****

7. Database Schema



8. Potential Features and Bugs

**8.1 Potential Features and Ideas**

***Fonts***

Adding a font editor would be our next top priority. This would consist of various fonts style that would be modern and professional. This would also include features to include bold, italics, and underline features. We will also add the ability to add color to the font, which we will only restrict it to the color theme of the website. This will give the admin of the site the ability to add variety to their content.

***Greying out the Submit button***

This will give the contact form the ability to prevent premature submitting. This is where we collect information from potential volunteering patrons using a Simple Mail Transfer Protocol Server (SMTP for short). As it stands now we can submit data any time we want. This would prevent the email server from getting spammed and it also will help out as a friendly reminder to fill the form out. The grey coloring would symbolize as a disabled button, and when the correct information is filled and passes the forms conditions, it will enable and turn blue. It’s a clever way to control the process of collecting the data.

***Youth Stories Blog Page***

This is very simple, a feature that would add a blog for stories of successful youth overcoming difficult odds. It would be a feature that would add new pages and delete old one.

**8.2 Bugs**

***Clean up CSS files***

To help with uniformity and control any potential CSS bugs, it would be best to move it to one uniform file and refactor any styling bugs.

***Linking bugs***

There are a few links that do not go to their proper sites, such as Facebook does not go to their Facebook page. This is a simple fix, however if we are to proceed we would need the proper information to link to the proper sites.

9. Client Meetings and Documentation

9.1 Group Meetings

***8/31/17 CS3750 - Typical Nerd Stuffs***

Met together for first time. We created a GitHub project, and assigned different documentational responsibilities for sections 1 to 3. Set up meeting with client and prep questions to ask client for upcoming discussion.

**Goals:** Complete document assignments by the following Tuesday 9/5/17 in preparation for our meeting in class.

**9/5/17 *CS3750 - Typical Nerd Stuffs***

Assigned out the rest of the document work minus the database diagram. Went over more requirements for the upcoming website.

**Goals:** Compete document assignments minus the database diagram by Tuesday 9/12/17

**9/14/17 *CS3750 - Typical Nerd Stuffs***

Discussed our database schema. Hashed out what is dynamic and what is not, and which data will be edited. The results will be included in the database schema section

* Features (a Possible idea that can be carried out later)
  + Inserting a fact or a tidbit in the page to create a cause and effect. Example is the “5,000 youth…” block. Can we add in different blocks for the admin to edit?
  + Populating a year calendar of all the events that will and could happen in the year.
  + Adding tab titles to the event tabs
  + Adding in a new section, with the ability to choose from different block styles
  + Rollback feature so if they screw up they can go back to the previous view.
* Questions
  + Donors (but there are other) tab below the “donors” acknowledge, how will it be populated?

9.2 Client Meetings

***9/1/17- Initial Client Meeting***

Went over the expectations for the website, discussed what data is being processed and what needs to updated and controlled for organizations administer. Here are the notes recorded in the following meeting:

* What are the functional requirements for this system?
  + Certain sections are updatable/editable and the change will be update to the database (additional screen? and have edit button for sections for the user to edit the content) (e.g. WordPress and some content management system site) and when user logs in then he/she is able to edit the web page
  + Some sections you can choose to pull the data from the database and populate the blocks on the web page (e.g. staffs, donors, etc.)
  + No need to worry about donation form and contact form, they are already taken care of
  + Users should have the ability to log in (we will need create the login page and log in button)
    - Log in page should have some sort of validation (e.g. CAPTCHA to prove that this user is not a robot)
  + We will give them the email to log in and edit the page (only one permission level in this system), which means that we don't need to worry about user registration
* User Problem Statements for the old system
  + The old system does not have anything cool and updated that they can use
  + And they really don’t have a good way to update the content themselves
* Project overall scope:
  + Some sections are editable
  + Some sections/blocks can be auto generated from database
  + Authorized users can log in and edit the web page

* User cases will be what the user will do with the web page
* Project outline:
  + The understanding of the project
  + The way we are going to accomplish this project (provides users the ability to edit each section instead of editing the whole page directly, and we won't have them keep scrolling down to find the section to edit after entering the editable mode)
  + Good indication of where the project is going