

---

# **Requirements Specification for CMS**

**Version 2.0**

**Prepared by: Bikram Gill, 5938923**

**Wesley Nantais, 5903455**

**Tyler Applin, 5920616**

**Brock Jones, 5859723**

**Eduardo Galindo, 5480504**

**Emmanuel Guarino, 6038970**

**Ege Can Kaya, 6150817**

**4F00 CMS Analysis & Design**

**April 24th, 2020**

# Table of Contents

<b>Table of Contents</b>	<b>ii</b>
<b>Revision History</b>	<b>ii</b>
<b>1. Introduction</b>	<b>1</b>
1.1 Purpose	1
1.2 Document Conventions	1
1.3 Intended Audience and Reading Suggestions	3
1.4 Project Scope	4
1.5 References	5
<b>2. Overall Description</b>	<b>6</b>
2.1 Product Perspective	6
2.2 Product Features	7
2.3 User Classes and Characteristics	9
2.4 Operating Environment	10
2.5 Design and Implementation Constraints	11
2.6 User Documentation	12
2.7 Assumptions and Dependencies	13
<b>3. System Features</b>	<b>14</b>
3.1 Account Creation	14
3.2 Account Authentication	16
3.3 Website Entry	19
3.4 Site Management	22

3.5	Editor	24	
3.6	Editor Layout	27	
3.7	Widgets (Page Elements)	28	
3.8	CMS Administrative Portal	31	
3.9	Backend System	34	
3.10	Account Page	37	
3.11	Redirection Protocol	39	
3.12	Ticket Management	41	
<b>4.</b>	<b>External Interface Requirements</b>		<b>43</b>
4.1	User Interfaces	43	
4.2	Hardware Interfaces	49	
4.3	Software Interfaces	51	
4.4	Communications Interfaces	55	
<b>5.</b>	<b>Other Nonfunctional Requirements</b>		<b>57</b>
5.1	Performance Requirements	57	
5.2	Safety Requirements	58	
5.3	Security Requirements	59	
5.4	Software Quality Attributes	63	
5.5	Legal/Ethical Requirements	65	
5.6	Operational Requirements	66	
<b>6.</b>	<b>Other Requirements</b>		<b>67</b>
	<b>Appendix A: Glossary</b>		<b>67</b>
	<b>Appendix B: Analysis Models</b>		<b>77</b>

<b>Appendix C: Issues List</b>	<b>83</b>
<b>Appendix D: Design Requirements</b>	<b>84</b>
<b>Appendix E: Revision History</b>	<b>85</b>

# **1. Introduction**

## **1.1 Purpose**

The purpose of this product is to provide users with an intuitive and easy way to create their very own website. Our priority is to make an easy to use system that allows users to create a reasonable looking website with as little trouble as possible. Our system will allow users to customize their websites through the use of widgets. While we will aim to include additional features, our focus is for users who have no idea how to utilize these tools to still be able to create a functioning website.

## **1.2 Document Conventions**

This document contains certain words and phrases that are stylized with special fonts, sizes, and weights. The purpose of this is to highlight words that belong to a specific category. These include the following:

`command`

This represents command line entries. It is implied that one can enter this phrase and illicit a response.

e.g. the *SQL* commands `POST` and `GET`

file name

File names and paths are represented in this manner. This style should indicate that a file of the given name can be found in the directory mentioned.

e.g. /home/student/c4f00g05/public\_html/people/

## **application**

This style is used to denote a 3<sup>rd</sup> party application.

e.g. the browsers **Chrome** and **Firefox**

**button**

This style is used to denote a button that corresponds to a GUI button on the application. e.g.

the preview button: **preview**

## URL's

This style is used to highlight URL links. Digital versions of this document will support hypertext functionality

e.g. [https://www.w3schools.com/cssref/css3\\_browsersupport.asp](https://www.w3schools.com/cssref/css3_browsersupport.asp)

## *languages*

This style highlights programming and markup languages(and other technologies)

e.g. *HTML*, *CSS*, and *JS*

## **1.3 Intended Audience and Reading Suggestions**

This system requirements specification is intended for developers, project managers, marketing staff, users and testers. This document aims to inform the user of general and advanced features available in the software. This varies from general use of the program to documentation of how the project is built and the various layers of code that build up the program. The various sections of this document are labelled accordingly to help the user find the section of this document that pertains to their given situation. We recommend each interested party mentioned previously read the document in its entirety.

## **1.4 Project Scope**

This project is designed to create a Content Management System (CMS) that allows non-technical users to be able to create personalized websites without needing any coding knowledge. To engage as many users to use the system as possible, the system is designed to give users from all walks of computing experience the opportunity to create an elegant website of their own. It is our goal to remove barriers to entry in the field of website development for novice users as well as allow power users a place they can easily build and practice their skills. By focusing on the needs of the novice user we believe we can establish a respectable market share in an already saturated market. In short, this system aims to create a website development tool that is novice friendly so that first time developers can grow with our system integrated into their workflow.



## 1.5 References

<i>HTML</i>	<a href="https://developer.mozilla.org/en-US/docs/Web/HTML/Element">https://developer.mozilla.org/en-US/docs/Web/HTML/Element</a>
<i>CSS</i>	<a href="https://developer.mozilla.org/en-US/docs/Web/CSS/Reference">https://developer.mozilla.org/en-US/docs/Web/CSS/Reference</a>
<i>JavaScript</i>	<a href="https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference">https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference</a>
<i>jQuery</i>	<a href="https://api.jquery.com/">https://api.jquery.com/</a>
<i>jQuery UI</i>	<a href="https://api.jqueryui.com/">https://api.jqueryui.com/</a>
<i>feather icons</i>	<a href="https://feathericons.com/">https://feathericons.com/</a>
<i>Bootstrap</i>	<a href="https://getbootstrap.com/docs/4.3/getting-started/introduction/">https://getbootstrap.com/docs/4.3/getting-started/introduction/</a>
<i>redbean</i>	<a href="http://redbeanphp.com/manual.php">http://redbeanphp.com/manual.php</a>
<i>CKEditor</i>	<a href="https://ckeditor.com/docs/">https://ckeditor.com/docs/</a>
<i>password_compat</i>	<a href="https://github.com/iremaxell/password_compat">https://github.com/iremaxell/password_compat</a>
SRS template	<a href="https://www.processimpact.com/process_assets/srs_preview.pdf">https://www.processimpact.com/process_assets/srs_preview.pdf</a>

## **2. Overall Description**

### **2.1 Product Perspective**

This Content Management System is a follow-on member to other Content Management Systems. Most of the Content Management Systems already out there are complicated and seemingly provided so that experienced developers can sell their skills to novice users. This system is unlike traditional website editing software, this system emphasizes simplicity. By putting the user experience first, this product was designed so that the journey from novice to power user is as painless as possible.

## **2.2 Product Features**

### **2.2.1 Account Creation and Authentication**

This system allows for the creation of user accounts. User accounts are protected by user provided passwords that will be used to authenticate any login attempts. The website editing system will encrypt and protect user information from unauthorized access.

### **2.2.2 Create, Edit, and delete a Website**

This system allows for the creation, modification, and deletion of a website. A website can contain multiple web pages. Users can select from a variety of templates and styles for their website and then easily drag and drop widgets/elements from the editing interface to where they want that function to exist on their webpage. In addition to these widgets, users may supply their own images or files to include on their website.

### **2.2.3 Storing Multiple Websites**

Each user of this system can store multiple websites on the system. The upper bound in regards to how many websites a user can create and store is defined by their respective account level. The websites are stored and retrieved by the website editing system and are not kept on the users personal machine.

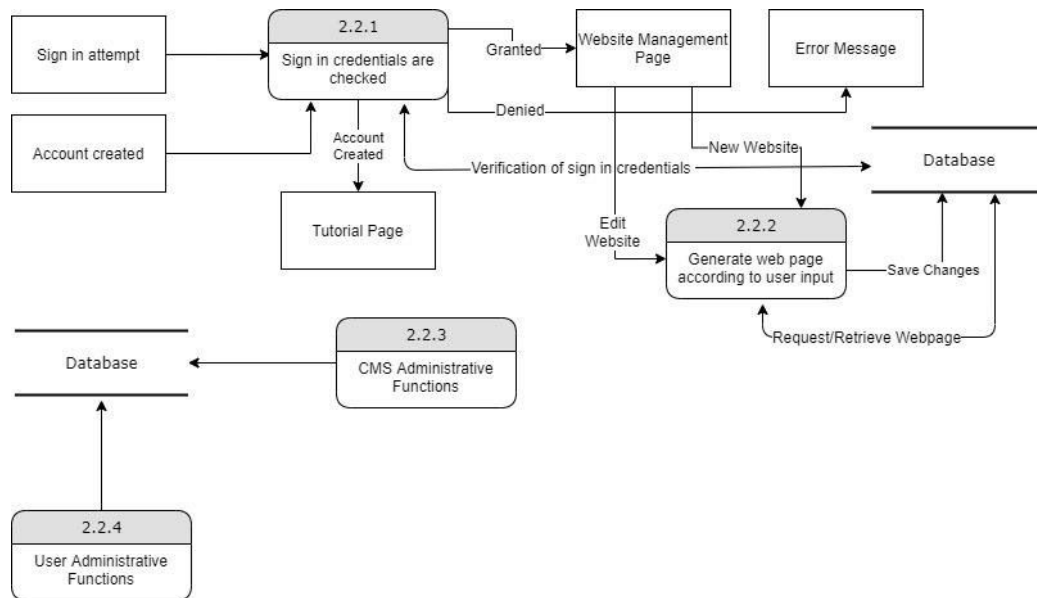
#### 2.2.4 CMS Administrative Functions

CMS administrators are able to do the following:

- Suspend or unsuspend user accounts
- Delete user accounts
- Message users via email
- Retrieve and resolve user tickets
- Edit and view user accounts.
- Generate accounting reports for the system

#### 2.2.5 User Administrative Functions

Users are able to change their account information, plan types, as well as add or delete websites of their own.



Data flow diagram of basic system functions

## **2.3 User Classes and Characteristics**

Power user:

A tech savvy user that has extensive experience with the website and is familiar with a majority of the products features. This user frequently uses the website to update and publish at least one website. It is important to keep this user satisfied since they are the most frequent users.

Knowledgeable intermittent user:

This user is comfortable with technology but uses the website less frequently. They are familiar with the semantics of the website but may have forgotten how to use certain features of the website.

Novice user:

The novice user has limited, if any exposure to the website or similar websites. They must possess basic technology skills in order to interact with the website. Uses a limited subset of the website's features. This user requires a shallow learning curve to perform the actions to meet their goals. It is important to satisfy this user and facilitate their development from a novice user to a power user.

## **2.4 Operating Environment**

This web editing system will be designed to operate primarily on one's internet browser opened on their personal computer. Although the website will be accessible via mobile device, the editor itself will not be accessible. As a result of providing this service through internet browsers such as **Chrome**, **Firefox**, and **Opera**, this system will be able to run on any operating system so long as it supports the previously mentioned browsers.

The database for this website edition system will be hosted on Sandcastle as provided by Brock University. It should be acknowledged that this third party system will provide development limitations (mentioned in the next section), and any changes to this system will also be outside the scope of developer and administrative authority in regards to this website editing system. This means if Sandcastle goes down, the site will be at the mercy of Sandcastle owners and administrators for its repair before website editing operations can resume. In addition to Sandcastle, this system will integrate *PostgreSQL* relational database management system.

## **2.5 Design and Implementation Constraints**

There are design and implementation constraints on the content management system. The primary design and implementation constraint is that the system must run and be hosted by Sandcastle. Sandcastle requires that front end developers only use *Javascript*, *HTML*, *CSS*, *Ajax*, and *PHP*. Sandcastle also does not allow the use of *React* or *Angular* libraries. Likewise and *PHP* frameworks will not be permitted to run on Sandcastle. Sandcastle also limits the relational database management system to *PostgreSQL*.

It follows that since front end developers are restricted to using only *Javascript*, *HTML*, *CSS*, *Ajax*, and *PHP*, that the web site editor system must run within an internet browser. As a result of running the system within an internet browser, designers will be restricted to using *TCP/IP* and *HTTPS* communication protocols. Certainly designers must keep user data safe during all stages of system use.

In the interest of keeping user data safe, user data will be stored on the Sandcastle server. While stored on the sandcastle server user data will be encrypted and kept separate from other user data items. Other users will be able to view stored files of other users only with the express permission of the user who owns that data.

Maintenance for this web editing system will be provided by the developers of the web editing system and funded by user subscriptions. Users must be able to contact the maintenance team regarding bugs or queries to aid in the development and maintenance cycles of the system. The maintenance of the user's personal computer and the security of any data stored on the user's own system is the user's sole responsibility.

## **2.6 User Documentation**

User documentation will be provided in the user manual. User documentation will provide users with a written and pictorial tutorial that describes system features, functions, and direct additional queries to the FAQ page or to contact the developers directly. In summary, User documentation will be provided for the following:

- Getting started.
  - Creating an account
  - Selecting a template
  - Selecting a style
  - Using widgets
  - Saving and publishing your first website
- Managing your websites
- Editing CSS through the editor



## **2.7 Assumptions and Dependencies**

### **2.7.1 Third Party Components**

- Brock University's Sandcastle server for website and database hosting services.
- Our software will rely on various open source libraries and software such as *Ajax*, *bootstrap*, *CKEditor*, and *Jquery*.

### **2.7.2 Development and Operating Constraints**

- Due to Sandcastle server limitations, the system's front end is constrained to only use *HTML*, *CSS*, *Javascript*, *Ajax*, and *PHP*.
- Sandcastle will only provide 4883 MB of space for system operations.
- Developers will use *Bootstrap* for a *CSS* framework, as well as *Jquery*, *Jquery UI*, and *CKEditor*. This should not be affected by Sandcastle's developer limitations.

### **2.7.3 External Dependencies**

- It is assumed that this project will be developed without any external dependencies. All code will be provided by development - not reuse.

## **3. System Features**

### **3.1 Account Creation**

#### 3.1.1 Description and Priority

- Users are able to create an account to use the CMS system. They must meet the minimum requirements and the account credentials will be created in the database.
- High priority

#### 3.1.1 Stimulus/Response Sequences

- New users wanting to use the service will be required to create an account. Users will have the option to sign up on the home page. If they attempt to access any of the features/resources of the CMS they will be redirected to the login account portal

#### 3.1.2 Functional Requirements

##### REQ-1: Enter Username

Users shall be able to enter a username. It must be provided to create an account.

#### REQ-2: Validate Username

Username provided by the user must be unique and meet minimum requirements. It must be between 4-32 characters and contain only letters, numbers, dashes(-), periods(.) and underscores(\_). If the username is in use the system shall ask the user for another username and also provide a link to the account recovery process.

#### REQ-3: Enter Password

Users shall be able to enter a password. It must be provided in order to create an account.

#### REQ-4 Recover Password

If a user forgets their password then said user will be able to enter the email associated with their account to allow them to change their password.

#### REQ-5: Enter Email

Users will be able to provide an email address. It is required to create an account.

#### REQ-6: Validate Email

The email address provided by the user must not be associated with an existing account.

## **3.2 Account Authentication**

### **3.2.1 Description and Priority**

- Users attempting to login to the CMS system will have their login credentials authenticated against the account details stored in the database. In order to access the site these must be consistent with the database
- High Priority

### **3.2.2 Stimulus/Response Sequences**

- Users will enter credentials to login and use the system. Username and password provided will be compared those stored in the database. Correct information will grant access. Incorrect username and/or password will not grant access.

### **3.2.3 Functional Requirements**

#### **REQ-7: Enter Username**

Users shall be able to enter a username. It must be provided to create an account.

REQ-8: Validate Username

Username provided by the user must be unique and meet minimum requirements. It must be between 4-32 characters and contain only letters, numbers, dashes(-), periods(.) and underscores(\_). If the username is in use the system shall ask the user for another username and also provide a link to the account recovery process.

REQ-9: Enter Password

Users shall be able to enter a password. It must be provided in order to create an account

REQ-10: Validate Password

Passwords must be at least 8 characters long, there are no other password requirements.

REQ-11: Enter Email

Users will be able to provide an email address. It is required to create an account.

REQ-12: Validate Email

The email address provided by the user must not be associated with an existing account. Any email entered must be validated by the *HTML* email input field.

## **3.3 Website Entry**

### **3.3.1 Description and Priority**

- The entry page to the CMS will provide the users with basic information regarding what functions the CMS will offer them. The entry page will also provide access to additional pages which will provide more in depth information regarding additional CMS services.
- High Priority

### **3.3.2 Stimulus/Response Sequences**

- Users navigating to the domain will land on the entry page. From this page they will be able to explore the features of the system. They will be prompted to login or create an account if they attempt to use any of the functions.

### **3.3.3 Functional Requirements**

#### **REQ-13: Login Portal**

Users shall have the option to login from the home page. A link shall be labelled 'Login' on the navigation bar, and lead to a form which will accept account details.

REQ-14: Sign-up Portal

Users shall have the option to sign up for a CMS account from the home page. and lead to a form which will accept account creation details.

REQ 15: Navigation Bar:

Users shall be provided on the main page with a navigation bar that will allow users to navigate to Features, Styles, Subscriptions, Templates, or Support pages.

REQ-16: About

Users will be provided an 'About' link to know more about the site. It will detail the team vision and goals for the system. It will also provide brief statements from select members of the team. Users will also be able to see how the CMS operates behind the scenes(ticket resolution,maintenance, etc).

REQ-17: Product/Plans

Users will be able to view different plans and products available for their site. Each plan shall have its respective features listed beside it.



#### REQ-18: Support

Users shall have an option to get support from the home page. A 'Support' link will list several options for users. It must contain an FAQ page. It shall also contain the option of reporting a bug and creating a ticket. It shall also contain an option to contact site administrators. And finally, users must be able to review privacy policies and terms of agreement.

#### REQ-19: Features

Users shall have an option to navigate to the features page which will inform them of the many features provided to them from the website editor.

#### REQ-20: Styles

Users shall have an option to navigate to a styles page which will inform them of any available styles to base their website off of.

#### REQ-21: Templates

Users shall have an option to navigate to a templates page which will inform them of any available templates to base their website off of.

## **3.4 Site Management**

### **3.4.1 Description and Priority**

- The site management portal acts as a control hub for existing users. From this page users can edit existing sites and create new ones. It also permits users to delete sites.
- High priority

### **3.4.2 Stimulus/Response Sequences**

- Users will arrive at this page after signing in or navigating to it from the menu. It will display all the existing sites the user has and the option to create a new one. Upon selecting one of these options they will be directed to the editing tool for the website chosen.

### **3.4.3 Functional Requirements**

#### **REQ-22: Select Site**

Users will be shown all existing sites. They must be able to select a website and enter editing mode directly.

#### **REQ-23: New Website**

Users shall be able to create a new site from this panel. They may choose a name that they would want, but cannot have a name similar to one they have already created..

REQ-24: URL Validation

Users must be able to provide a name for the site. This name must be unique. It must be compared to existing names hosted by the CMS. If the name is already being used the user will be given an error with the appropriate message to let the user know that the name has been taken

REQ-25: Delete Site

Users shall have the option to delete a website. Users will be warned that all content and data associated with the site will be lost before a delete can be made. They will be required to enter another confirmation to allow the deletion.

## **3.5 Editor**

### **3.5.1 Description and Priority**

- Users with existing websites are able to edit those websites. They are able to add and remove components as desired. They can preview the changes and publish them if desired.
- High priority

### **3.5.2 Stimulus/Response Sequences**

- Users will be able to access the editing mode for a site by selecting it from the site management panel. Then they can make the desired changes and publish if desired.

#### **Functional Requirements**

##### **REQ-26: Add/Remove Widgets**

The user should be able to drag and drop predefined widgets onto the page and re-arrange pre-existing elements

##### **REQ-27: User content**

Users must be able to display personal content. This content can be directly embedded onto the page or linked from elsewhere.

##### **REQ-28: Preview**

Users must be able to preview the outcome of pending changes without going live. A fully functional preview must be generated in a new tab.

REQ-29: Publish

Users should be able to save changes and implement them to the live site.

Upon publishing users should be linked to the site.

REQ-30: Save Changes

Users shall be able to save changes they have made while in editing mode.

Once a web page is published, the button will change and allow the user to unpublish the whole website.

REQ-31: Tool Panel

The system must provide the user a panel which contains all content and tools required for the editing process

REQ-32: Link Pages

The user shall be able to link pages(internal and external) by attaching them to widgets, media, and text

REQ-33: Adding and Removing Web Pages

Once a website is selected, users will be able to add and remove web pages from within that website.

#### REQ-34: Publishing, and Saving the Website

Once a website is selected, users will be able to publish, unpublish, and save the changes made to their website. Unpublished pages are only viewable by the author. Published pages are viewable by anyone. Saving a website will overwrite the old website and not modify a website's published/unpublished state.

## **3.6 Editor Layout**

### **3.6.1 Description and Priority**

- The editing mode will have a layout that will contain designated areas for navigation footer, and body. The user can fully edit the body. location of the footer can not be changed. The navigation bar will automatically add created web pages but can not be manually edited.
- High priority

### **3.6.2 Stimulus/Response Sequences**

- When the user starts the process of creating a new site they will have a layout consisting of navigation bar,a footer and body sections. The user can add and remove items from the body and footer but can't move these sections around. The navigation bar can not be manually edited and will automatically update when a new page is added or deleted.

### **3.6.3 Functional Requirements**

#### **REQ-35: Footer Region**

The user shall be able to edit the fixed footer region but not add items that are not compatible with this component

## **3.7 Widgets (Page Elements)**

### **3.7.1 Description and Priority**

- The editor mode will support drag and drop widgets. Each widget will provide a unique feature the user can edit and implement on their webpage.
- High priority

### **3.7.2 Stimulus/Response Sequences**

- Users in editing mode will be able to drag and drop these elements. They will select a widget in the tools panel and drag it into the location they desire. Once placed, the user will have the ability to click the element and edit some of its attributes.

### **3.7.3 Functional Requirements**

#### **REQ-36: Text box**

Users shall be able to place a text box on the page.. The user will be able to edit its content as well as customize some of its styling from some system defined presets.

#### **REQ-37: Pictures**

Users shall be able to place images on the page. This includes a file upload or through an external link..



REQ-38: Embeddable content

Users shall be able to embed external content. This may include social media posts and videos/media hosted on external sites. By clicking on the element, users should be able to modify the location of external contents as well as change the size of the component.

REQ-39: Grid Layout

Users shall be able to place contents next to each other. By clicking on this element, users can specify how many columns to include and which widget will go in each column.

REQ-40: Rich Text

Users shall be able to have flexibility on adding text. If users need more than a simple text box, they should be able to include lists, linked text, and text within tables to their webpage.

REQ-41: HTML Block

Users shall be able to add custom *HTML* blocks to their page. If users would like customized elements, they should be able to add their own *HTML* to the page as a widget to the extent that such content does not compromise the security of the system.

REQ-42: Button

Users shall be able to add buttons to their page. Buttons should allow users to be directed to external content on click.

REQ-43: Spacer

Users shall be able to space widgets out on their website.

REQ-44: Divider

Users shall be able to divide page content into various sections

## **3.8 CMS Administrative Portal**

### **3.8.1 Description and Priority**

- Site administrators will have the ability to supervise user activity. They will also be able to modify user accounts and websites
- High priority

### **3.8.2 Stimulus/Response Sequences**

- Administrators when logged in will be automatically redirected into a specialized portal which grants them all required functionality. From this portal they are able to address tickets and other concerns which users are unable to themselves

### **3.8.3 Functional Requirements**

#### **REQ-45: Delete User Accounts**

System administrators shall be able to delete user accounts. This may be to satisfy a user request or to shut down their account hosted on the system. A deleted user will cause all their associated websites to be deleted as well. The system administrator must not be able to delete themselves

REQ-46: View User Accounts

System administrators shall be able to see current accounts profiles and status of their subscription and if they are an administrator or member of the website

REQ-47: Delete User websites

System administrators shall be able to delete user websites, when selecting the delete button, the admin will be prompted a second time to make sure they want to delete that specific website. This may be used to satisfy a user's request to delete a website that might be harmful to the reputation of the CMS.

REQ-48: Toggle users as administrators or members

System administrators can add or remove users as administrator, they will be prompted once a change has been made and can see what user is an admin or not

REQ-49: Solve tickets/bugs

System administrators should be able to fix user issues and address bugs that the user has found. The ticket can then later be deleted once resolved. Each ticket shall be able to display the time, date, email, name, and the message to the administrator, and sort it by oldest to new.

REQ-50: Delete all websites

System administrators must be able to completely delete all websites, clicking on this option shall display a warning where the admin must be able to click delete again.

REQ 51: Delete all Users

System administrators must be able to delete all users from a website, this might be helpful if there has been a breach or if there needs to be a change of database. This can be used for temporary testing purposes. Clicking delete must prompt a warning where the administrator will be able to confirm the deletion. When deleting the database a basic administrator will be created so that there will always be one administrator as a user.

## **3.9 Backend System**

### **3.9.1 Description and Priority**

- The CMS will have to take user input and store that into the database. It must also be able to pull the required data from the database and bring it to the user end.
- High priority

### **3.9.2 Stimulus/Response Sequences**

- When the user enters details regarding the account or setting for their site those updates will be sent to the database. Vice versa when users request a template or to pull up account details, those are located and sent from the database to the client side terminal

### **3.9.3 Functional Requirements**

#### **REQ-52: Query Handler**

The backend must be able to handle *SQL* type queries

#### **REQ-53: Delete Data**

The backend system must be able to delete data stored in the database

#### **REQ-54: Store Data**

The backend system must be able to save data to the database

REQ-55: Verify permissions

The backend system must be able to confirm the user requesting a sensitive operation is authorized to do so.

REQ-56: Update data

The backend system must have the functionality to update existing data/records

REQ-57: Store images

The backend system will have the ability to store image files so that the editor can include them

REQ-58: Store Websites

The backend will be able to store and retrieve any changes you make to any of your websites, as well as retrieve a list of all your websites and the data associated with each website.

REQ-59: Pass messages

The backend will be able to pass temporary messages for the front end to display that aren't stored in the database in order to let users know any information about the request they made (i.e error with request).

REQ-60: Cascade Delete

The backend must be able to do a cascading delete when deleting a user or website.

REQ-61: Pull Data

The backend must be able to pull data from tables.



## **3.10 Account Page**

### **3.10.1 Description and Priority**

- The account page is a centre for all the users settings and details. From this page the user can update their info and also change settings
- High priority

### **3.10.2 Stimulus/Response Sequences**

- The users can access their account settings anytime once logged in. They do so by clicking the account link located on all pages in the header bar. The user will be able view all the settings and details they can change from the main settings page.

### **3.10.3 Functional Requirements**

#### **REQ-62: Add/RemoveFull Name**

Users shall have the option to add/remove their full name. This will also change the greeting in the accounts page to ensure they are greeted by their first name, if available. Otherwise to only greet the users by their username.

#### **REQ-63: Change email**

Users must be able to change the email address associated with the account. This will require the new email address to be validated and checked to make sure there is no other account with the same email address.

REQ-64: Change plan type

Users shall have the option to change the plan they have set for the account.

The option will be available in the following three tiers: Starter, Basic and Pro.

REQ-65: Change date of birth

Users shall have the option to change the date of birth for the account holder.

REQ-66: Change Payment Method

Users will be able to change the payment source if they are paying for any services. This will provide fields such as credit card number, CCV, card type, and expiration date.

REQ-67: Delete account and associated websites

Users must be able to delete an account for the CMS with their associated websites. They will be warned that all content will be lost. Once deleted their websites and associated account information will be wiped permanently for privacy reasons.

## **3.11 Redirection Protocol**

### **3.11.1 Description and Priority**

- The Redirection Protocol is a dedicated process to deal with redirecting the users. The user will be navigated to the appropriate page based upon their activity and requests.
- High priority

### **3.11.2 Stimulus/Response Sequences**

- If the user attempts to access a page they do not have access to or does not exist they will be sent to the corresponding page.

### **3.11.3 Functional Requirements**

#### **REQ-68: Not authorized**

This protocol will respond to users attempting to access an administrative page or an unpublished site. Users will automatically be redirected to the login page if they don't have permissions.

#### **REQ-69: Page doesn't exist**

The user must be notified if they are attempting to access a page that does not exist. This may be an incorrect domain or a page that has been deleted.

REQ-70: Error (404, 500, ect)

The protocol shall return http status codes.

REQ-71: Exception

The protocol must be able to detail an exception and provide the stack trace

## **3.12 Ticket Management**

### **3.12.1 Description and Priority**

- Users should be able to create ticket for issues related to the service
- High Priority

### **3.12.2 Stimulus/Response Sequences**

- Users/visitors that have a concern/issue will be able to create a ticket by selecting the option support on the navigation bar. The user will be notified by banner once the ticket was successfully made, or if there was an issue with sending it with an appropriate error. Administrators will be forwarded the issue and address it.

### **3.12.3 Functional Requirements**

#### **REQ-72: Create Ticket**

Users/visitors can create a ticket for an inquiry. This will include a name, the message, and their email.

#### **REQ-73: FAQ**

Users/Visitors must be able to see a list of frequently asked questions and shall be organized in a way that when selecting the question, the window will have an anchor that will focus the scroll to where the answer is.

REQ-74: Contact support

Users/Visitors must be able to edit a list of possible support contacts where they can contact the stakeholders/administrators that will be able to assist them with their preferred contact method

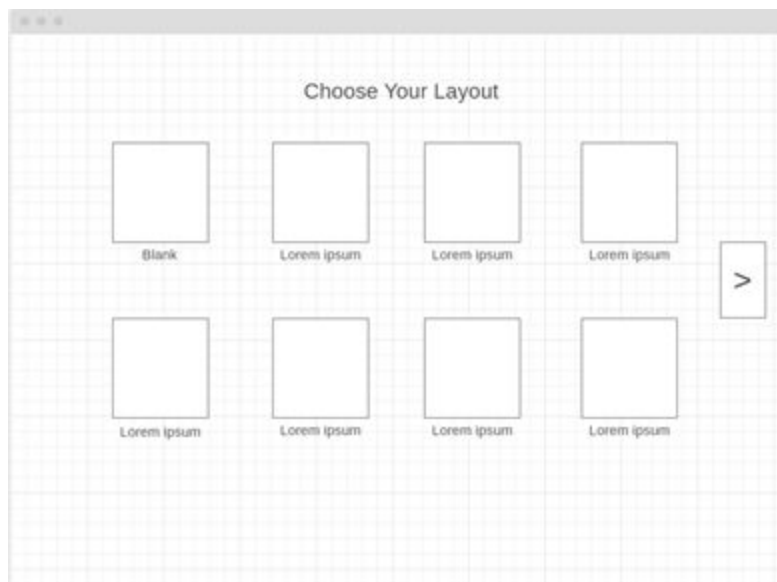
## 4. External Interface Requirements

### 4.1 User Interfaces

#### Screen Images

Each page of the web application will have a user interface to facilitate the interaction between the users and the system. The standards, styles, and libraries used for the user interface will be consistent throughout the system. However, the placement and content of user interface components may vary based on where in the application it is used.

Examples of user interface design for the main pages of the application can be found below:

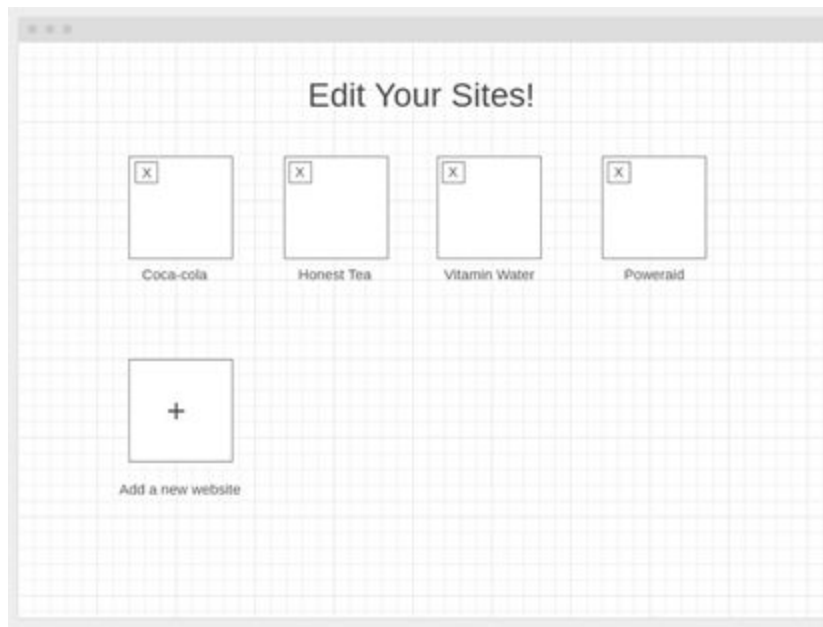


Screen Image 1 - Layout/Themes Page



A wireframe of a 'Create Your Account' form. The form is titled 'Create Your Account' and is divided into two sections: 'Required' and 'Optional'. The 'Required' section includes four input fields: 'Account Name\*', 'Email\*', 'Password\*', and 'Confirm Password\*'. The 'Optional' section includes four input fields: 'Name:', 'Phone Number:', 'Birthday:', and 'Biography:'. A 'Create Account' button is located at the bottom right of the form.

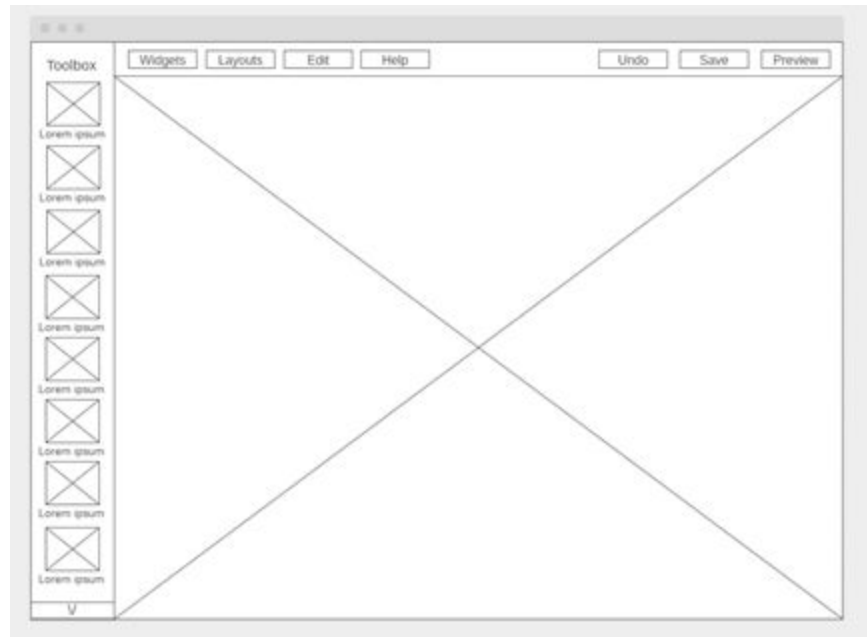
Screen Image 2 – Create Your Account Page



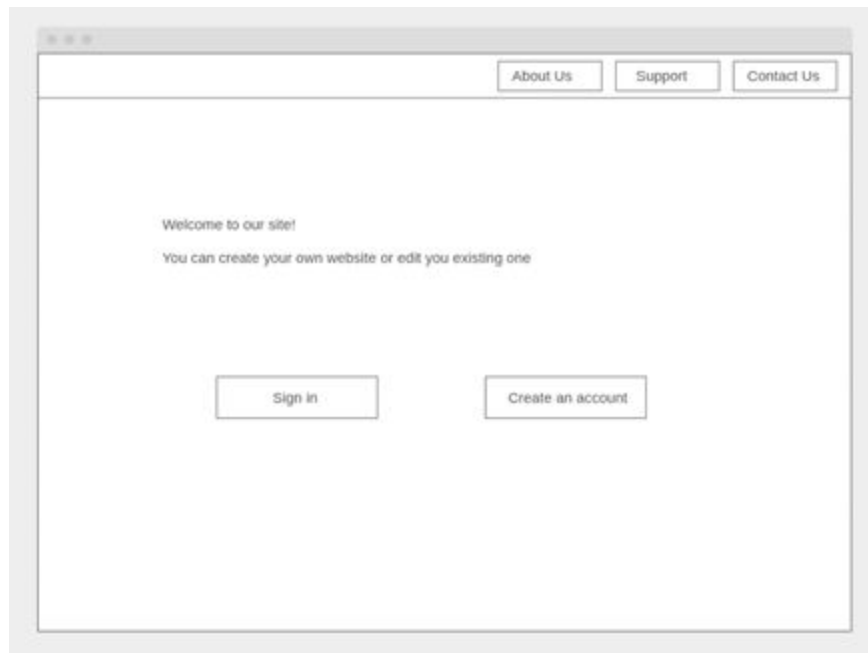
A wireframe of an 'Edit Your Sites!' form. The form is titled 'Edit Your Sites!' and displays four existing sites, each with a small 'X' icon in the top left corner and a label below it: 'Coca-cola', 'Honest Tea', 'Vitamin Water', and 'Poweraid'. Below these sites is a large square button with a '+' icon and the text 'Add a new website'.

Screen Image 3 – View All Sites Page





Screen Image 4 – Editor Page



Screen Image 5 – Login and Register Page

## **GUI standards and style guidelines**

For the user interface components, we will be using the *Bootstrap* front-end component library, which is an open source toolkit for developing with *HTML*, *CSS* and *JS*. For more information about the specifications of this library and the standards of the front-end components, visit:

<https://getbootstrap.com/>

Additionally, some components will be screen friendly so that users with vision impairment can access and view them. This will be achieved by adding label elements and following the *HTML* accessibility guidelines. The Bootstrap library provides additional support for accessibility.

## **Screen layout constraints**

The editor will not be available when the user is accessing from devices that have small screens (e.g. phones, tablets, etc). This is because the editor functionality designed for desktop screens would not be compatible when accessing the website through a phone.

However, the user will be able to access all other parts of the application through a device with a small screen, and all parts of the application will be displayed in a mobile-friendly way. For example, the system will detect when the user is accessing the website from a mobile device and resize the components accordingly so that some components can fit within the screen boundaries of the user.

In accordance with the *Bootstrap* library guidelines, the breakpoints at which the system will mostly resize the components are as follows:

Extra small devices	Less than 768 pixels
Small devices	Greater or equal to 768 pixels
Medium devices	Greater or equal to 992 pixels
Large devices	Greater or equal to 1200 pixels

## **Standard buttons & functions that will appear on every screen**

The main navigation bar will appear on every screen of the application so as to provide the user with an easy way to reach the resources they need. The navigation bar will include a logo/title, buttons such as [home](#) , [about us](#) , [support](#) , [contact us](#) , and also a dropdown menu where the user can access various items.

## **Error message display standards**

Based on what type of alert we want to display to the user, three distinct message alert components will be used. A red message alert component will be used for errors, security alerts, etc. A yellow message alert component will be used for warning the user about potential errors. And a blue message alert component will be used for informational alerts that can help the user.

## 4.2 Hardware Interfaces

### Supported device types

On the client-side, the application will be fully supported on the **Google Chrome** and **Firefox** browsers. Nothing specific has to be changed to ensure **Google Chrome** and **Firefox** compatibility, besides adhering to the common *HTML*, *CSS* and *JS* development patterns. We will ensure that the application uses *CSS* elements that guarantee the browser version compatibility for **Google Chrome** and **Firefox**, as can be determined by the 'CSS Browser Support' table at:

[https://www.w3schools.com/cssref/css3\\_browsersupport.asp](https://www.w3schools.com/cssref/css3_browsersupport.asp)

Below you can find the link to *HTML*, *CSS*, *JS* references which will be used as a guideline in the development of this application:

HTML	<a href="https://developer.mozilla.org/en-US/docs/Web/HTML/Element">https://developer.mozilla.org/en-US/docs/Web/HTML/Element</a>
CSS	<a href="https://developer.mozilla.org/en-US/docs/Web/CSS/Reference">https://developer.mozilla.org/en-US/docs/Web/CSS/Reference</a>
JavaScript	<a href="https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference">https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference</a>

## **Data and control interaction between the software and hardware**

For the database, the *SQL* software will interact with the hardware when storing data in the hard drive. Along with the data stored, it will also store secondary items such as relational information about the data, indexing, table structure, etc. There will also be networking connections created and managed so as to allow external queries, and for the purpose of allowing data to come in and out of the database system.

The web server (*PHP*) software will also interact with the hardware when serving the web content to incoming requests from the clients. The *PHP* code will be executed to run as a web server process on the operating system of the system where it is deployed (in our case Sandcastle). When the users type a URL in their browser, it will send a message to the web server through the network connection of Sandcastle, and the web server will respond by sending the requested files (*HTML*, *CSS*, etc).

The client will also have limited interaction with the underlying hardware for saving cookies, sessions, cache, etc. The client will first make a request to the web server, which will usually be done utilizing a web browser process on the client operating system. Then the response for that request will pass through the networking port of the client, which will then be processed to display the *HTML*, *CSS* and *JS* files appropriately on the web browser. These files will be stored temporarily and deleted after the user closes the application, with the exception of cookies and cache.

## 4.3 Software Interfaces

The data will be passed from the server to the client using the *GET HTML* method. Actually, both *POST* and *GET* can be used, but *GET* is faster and is the standard convention in web development. *AJAX* will be used to make server calls from the web application after the page has loaded. It should be noted that in most cases a *GET* request to the server will also result in the web server making a call to the database to fetch some sort data, which then will be returned to the client. As an example of making a *GET* call to the server, this is the code that would be used:

```
xhttp.open("GET", "demo_get.asp", true);  
xhttp.send();
```

Code – *HTML GET* example.

The *POST HTML* method will be used to send data to the server. Again, both *GET* and *POST* calls can be used to send data to the server, but *POST* can append the entire form element to the request and is the standard convention among web developers. Therefore, we will use *HTML POST* in the development of this project when posting/sending data to the server. The use of *HTML POST* is usually facilitated by using a *HTML* form, which is a user-friendly way to configure a *HTML POST* to the server. Below is a code example of how this would look:

```
<form action="https://www.foo.com" method="POST">
```

Code – *HTML* POST example.

The ‘action’ attribute in the above example defines where the data gets sent. Its value can be a relative or absolute URL. In our case, this will point to one of the endpoints of the *PHP* server so that when the user submits some input, it gets sent to the server, and the server can then process the input (i.e. send a response, save something to the database, etc). The ‘method’ attribute defines the type of *HTML* method used when sending the data to the server, and this field can be ‘POST’, ‘GET’ and several others.

Below is a diagram that illustrates the relationship of the GET and POST calls with the server and the response back to the client:



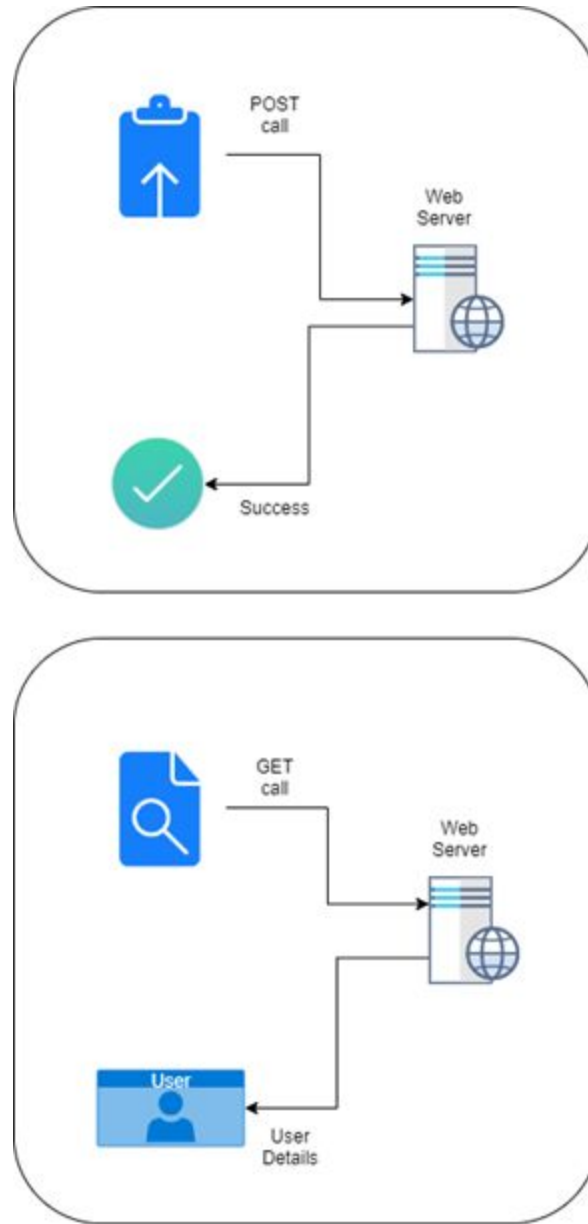
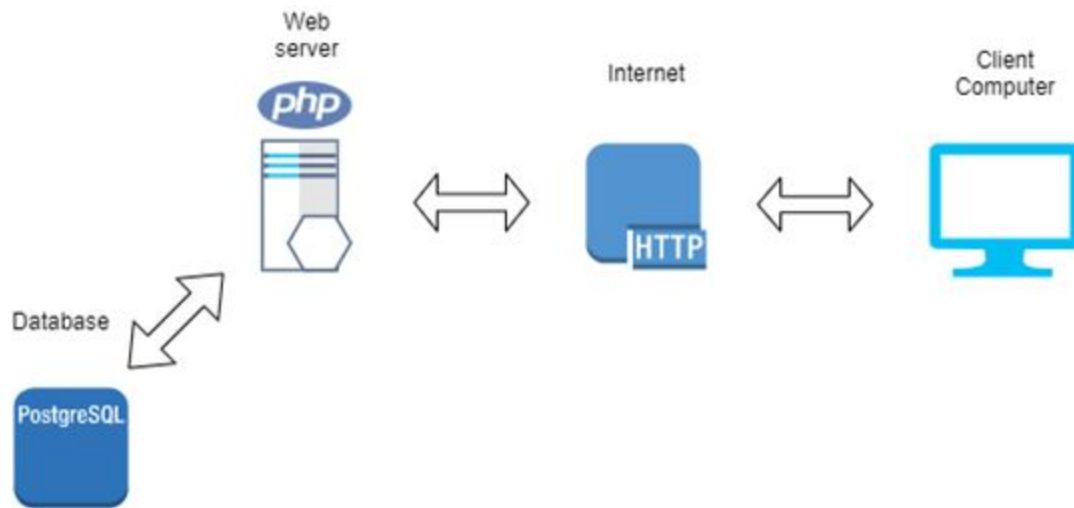


Diagram – *HTML* GET and POST methods example.

The diagram below illustrates how the database will also be communicating with the *PHP* web server through a software interface.



## **4.4 Communications Interfaces**

The main communication interface utilized will be the *TCP/IP* (Transmission Control Protocol/Internet Protocol) which is commonly used with web applications. This protocol specifies how data is exchanged over the internet by providing end-to-end communications that specify how it should be broken into packets. *TCP/IP* is also developed in a way that ensures reliable networks, with the ability to recover automatically from the failure of any device on the network. Channels of communication across a network are created and defined using the *TCP* protocol, and it also specifies how a message is assembled into smaller packets before they are then transmitted over the internet and reassembled in the correct order at the destination address. Addressing and routing each packet is defined and managed by *IP* protocol. This is done to ensure that each packet reaches the right destination.

The *TCP/IP* model is broken into layers. The Application Layer sends and receives data for application processes. The Transport Layer handles services related to ensuring reliable communications and inter-network tasks. The Network Layer handles the routing of packets across networks. And the Data Link Layer handles communication on the physical network components (e.g. Ethernet protocol). Since the *TCP/IP* model is layered, security controls exist separately for each layer of the system. It should also be remembered that data is passed from the highest to the lowest layer, and more information is added at each subsequent layer. As a consequence to this, a

security control at the highest level cannot ensure that the lower levels are protected, because the higher levels will not be aware of some functions performed at the lower levels.

## **5. Other Nonfunctional Requirements**

### **5.1 Performance Requirements**

- The server and the site shall not allow operations that allow the system to slow down or cause deadlocks.
  - The system shall remain steady and quick to load. If the system does need time, a loading screen must be present.
  - It shall take no more than one second to load a web page.
  - It shall take no more than 0.5 seconds from the time an event happens (mouse click, keyboard shortcut) to the time something on the client changes, even if that change is a loading screen.
  - The server should be able to handle multiple users without affecting its performance
  - The system shall be able to work on many different environments, such as operating systems, and varying screen sizes.
  - If the site was to go down, all of the saved data shall be recoverable
  - If you click on something then it should take no more than half a second or some result to appear, even if it is a loading
  - When the server makes queries to the database, it should not take more than half a second to return query results.
  - When the front-end makes server calls (GET, POST, etc.), it should not take more than half a second to return data from the server.
  - The site shall have 0.99 availability

- The site shall run on **Chrome** and **Firefox**. Tests will be conducted on all browsers to verify it works.
- Should be able to service multiple users at the same time
- Multiple users shall be able to access the service simultaneously

## **5.2 Safety Requirements**

- Our service shall not allow for any quickly flashing content that could trigger epilepsy except for users video content which won't play by default.
- The system will allow you to save your content and will have a way of showing the user when they have created changes that need saving.

## **5.3 Security Requirements**

### **Physical security**

The data and service shall be stored on the sandcastle server, which will satisfy all of our requirements for physical security. Sandcastle will ensure that no one can physically access our servers including people working on this service except for trusted technicians.

### **Database security**

In order to properly secure this database the model of least privilege shall be employed. This means that none of our developers shall have access to System Privileges in order to prevent major damage happening to the database. Instead a trusted 3rd party shall handle the administration of the server, in this case Cale. Object Privileges shall be limited to only team members who work in the backend, not the full development team. A strong and random database password shall be employed for accessing the database, along with a non-standard username. In order to reduce the impact of an attack on the database regular backups shall be performed on the database by sandcastle. In order to decrease the attack area of the database it shall only be accessible through sandcastle and not exposed to the wider internet.

In order to prevent DDoS attacks on the database the database will be designed to handle much more load than it will be put under normal circumstances.

In order to minimize the effects of a database leak it is required that all passwords shall be hashed within the database, and optionally the password will be salted and stretched to make it even more difficult to figure out the passwords in case of a database leak.

## **System security**

The system shall be designed in a way to prevent users from running any code on our server. The system shall not run any user created executable content, this means no custom php will be run, no user created javascript will be run. To do this XSS shall be prevented through code reviews. *SQL* injection shall be prevented by avoiding code that is prone to *SQL* injecting through code reviews. Users will be able to upload their own *HTML* content, but no javascript and *HTML* will only be able to go into sandboxed components. Users shall not be allowed to directly edit the *HTML* the service produces.

Information about the underlying system shall be obscured in order to prevent hackers from gaining information about the system. That means you won't be able to see what operating system or web server the system runs on from the web. Sandcastle provides this obfuscation for us.

The system shall require new users to sign up for an account in order to use this service. This will require creating a unique username as well as a password in order to authenticate your account. The



only requirement for the password will be that it be at least 8 *ASCII* characters, there must not be any other requirements, as too many password requirements can irritate users and lead to worse passwords. If a user forgets their password then it will need to be reset, so it will be required that our system allows the user to reset their password either by sending them a reset email or entering the answer to a security question. Emailing them a reset link will be preferable as it is more secure than a security question but may not be feasible. A user will only be able to enter the wrong password 5 times before you're locked out of the system and will need to confirm your account via email.

The authentication system shall prevent other users from accessing your unpublished content. In order to request some content the front end will need to send the request along with the session id or else the system will reject it.

This system has limited storage space available, so in order to avoid a denial of service attack users shall only be allowed to upload 8MB images and will be limited as to how many images they will be allowed to upload depending on their account.

There shall be different levels of access to the service, there will be website administrators who have the ability to manage the way the entire CMS works and manage other users. Then there will be users who will only be able to edit their website and website people have granted them permission to edit.

## **Privacy**

In order to protect users' privacy this website shall not sell any user data to advertisers or any other 3rd parties. It is required that no tracking cookies will be used although cookies may be used for functionality reasons.

This service shall collect as little information as possible in order to minimize privacy issues if there is a data leak.

A user shall be able to grant access to other users to edit their website, this will allow them to share access to their website without sharing a password.

## **5.4 Software Quality Attributes**

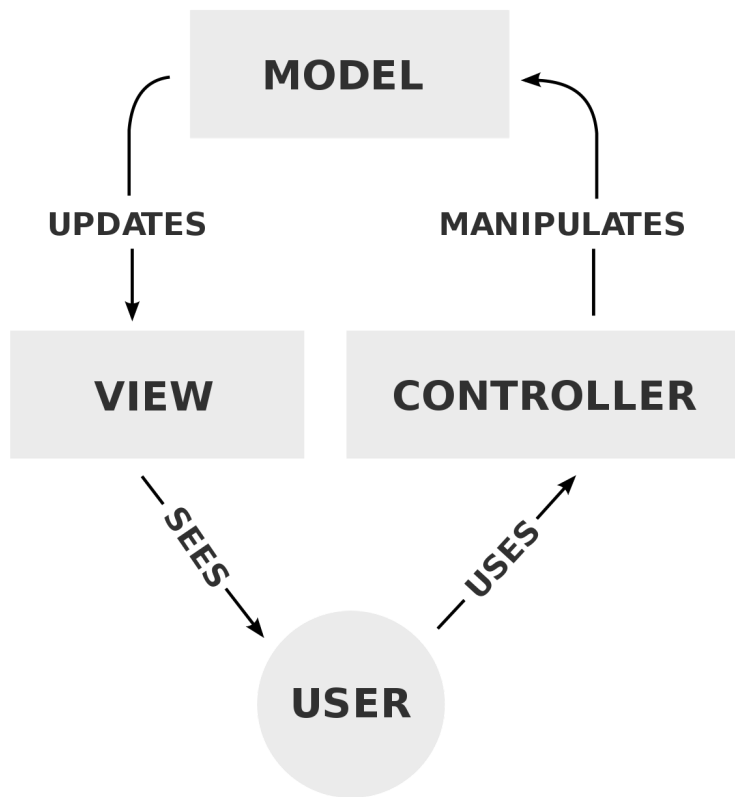
There shall be integration tests performed on all browsers we support. That includes **Firefox** and **Google Chrome**.

Error prone constructs shall be avoided; there will be no goto statements in the code, no pointers in the code, and concurrency will be limited so that it will only be handled by the web server.

Static analysis of the code for errors or undesirable code shall be performed. This will involve both code reviews for all commits as well as static analysis tools. Every commit will require one (1) other person to look it over. This code analysis will be used to prevent problems such as XSS and SQL injection.

In order to avoid a whole host of issues such as buffer overflows, unchecked arrays, pointer errors php shall be the only back end language employed here.

A MVC architectural pattern shall be used to organize the code and make it easy to distribute work.



## **5.5 Legal/Ethical Requirements**

- Multiple users shall be able to access the service simultaneously
- Admins shall remove any uploaded content that is illegal in Canada.
- Admins shall remove any content in order to comply with DMCA requests.
- Users shall not use this website to publish pornography or solicit/advertise other adult services.
- Users shall not use this service to bully individuals, post hate speech, incite violence, or post other content with gratuitous violence.
- Admins shall consider terminating any accounts that violate these rules.

## **5.6 Operational Requirements**

- Multiple users shall be able to access the service simultaneously
- Novice users shall be able to use the website without training beyond provided tutorial.T
- The system shall not have more than 95% downtime.
- There shall be at least one backup of the entire server in case the drive fails or becomes corrupted.
- The website shall be documented and any changes shall be noted, the code will also be written such that if any changes in the future is to be done that it will be simple and quick to understand.
- It is required that there be a function to roll back to an older stable version in case the new version has issues.
- Tickets should be closed within 5 business days.

## 6. Other Requirements

## Appendix A: Glossary

---

<b>1NF (First Normal Form)</b>	the fundamental rules for database normalization and relates to a single table within a relational database system
<hr/>	
<b>2NF (Second Normal Form)</b>	the second step in normalizing a database. 2NF builds on the first normal form (1NF)
<hr/>	
<b>Agile</b>	a lightweight software engineering framework that promotes iterative development throughout the lifecycle of the project, close collaboration between the development team and business side, constant communication, and tightly knit teams
<hr/>	
<b>Ajax</b>	(Asynchronous JavaScript and XML) client-side web development technique that is used to produce interactive Web applications

---

<b>ASCII</b>	(American Standard for Information Interchange) a method of encoding characters that is based on the order of alphabetic characters in the English language.
--------------	--

---

<b>Back-End System</b>	any system that supports back-office applications.  These systems are used as part of corporate management and they work by obtaining user input and gathering input from other systems to provide responsive output. Refers to indirectly linked devices that respond to end user activities or requests
------------------------	---

---

<b>Bootstrap Library</b>	is a library of code for GUI
--------------------------	------------------------------

---

<b>Buffer Overflow</b>	A buffer overflow occurs when more data is written to a buffer than it can hold. The excess data is written to the adjacent memory, overwriting the contents of that location and causing unpredictable results in a program
------------------------	--

---

<b>Cache</b>	a data storing technique that provides the ability to access data or files at a higher speed.
--------------	---

---

<b>CMS (Content Management System)</b>	an interface that allows users to publish content directly to the Web. The process of adding content
--	--



pages directly to the Web is one step ahead of creating and uploading pages from a local machine because it allows many people to add and share the data remotely.

---

**Cookies**

a text file that a Web browser stores on a user's machine. Cookies are a way for Web applications to maintain application state. They are used by websites for authentication, storing website information/preferences, other browsing information and anything else that can help the Web browser while accessing Web servers.

---

**CSS (Cascading Style Sheet)**

standard (or language) that describes the formatting of markup language pages

---

**Database (DB)**

an organized collection of data. More specifically, a database is an electronic system that allows data to be easily accessed, manipulated and updated.

---

**DDoS attack**

a malicious attempt to disrupt normal traffic of a targeted server, service or network by overwhelming the target or its surrounding infrastructure with a flood of Internet traffic.

---

<b>Deadlock</b>	a situation where a set of processes are blocked because each process is holding a resource and waiting for another resource acquired by some other process
-----------------	---

---

<b>Dropdown Menu</b>	a list of items that appear whenever a piece of text or a button is clicked. This is a graphical approach presented to users from which they can choose a value from the list presented.
----------------------	--

---

<b>Encrypt (Encryption)</b>	Encryption is the process of using an algorithm to transform information to make it unreadable for unauthorized users.
-----------------------------	--

---

<b>Epilepsy</b>	a neurological disorder marked by sudden recurrent episodes of sensory disturbance, loss of consciousness, or convulsions, associated with abnormal electrical activity in the brain.
-----------------	---

---

<b>Font</b>	the combination of typeface and other qualities, such as size, pitch, and spacing
-------------	---

---

<b>Front-End</b>	the top layer and interacts directly with end users
------------------	---

**GUI (Graphical User Interface)**

an interface through which a user interacts with electronic devices such as computers, hand-held devices and other appliances. This interface uses icons, menus and other visual indicator (graphics) representations to display information and related user controls, unlike text-based interfaces, where data and commands are in text. GUI representations are manipulated by a pointing device such as a mouse, trackball, stylus, or a finger on a touch screen.

---

**Hash/Hashing**

way to enable security during the process of message transmission when the message is intended for a recipient only. A formula generates the hash, which helps to protect the security of the transmission against tampering.

---

**HTML (Hypertext markup language)**

the major markup language used to display Web pages on the Internet. In other words, Web pages are composed of HTML, which is used to display text, images or other resources through a Web browser.

---

**Indexing**

the organization of data according to a specific schema or plan. In IT, the term has various similar

uses including, among other things, making information more presentable and accessible.

---

**JavaScript (JS)**

a scripting language, primarily used on the Web. It is used to enhance HTML pages and is commonly found embedded in HTML code.

---

**Libraries**

a suite of data and programming code that is used to develop software programs and applications. It is designed to assist both the programmer and the programming language compiler in building and executing software.

---

**Mashup**

a technique by which a website or Web application uses data, presentation or functionality from two or more sources to create a new service. To a user, a mashup should provide a richer, more interactive experience. A mashup is also beneficial to developers because it requires less code, allowing for a quicker development cycle.

---

**MySQL(My Structured Query Language)** MySQL is an open-source relational database management system.

---

**Operating system (OS)** software that allows a user to run other applications on a computing device. E.g. Windows, Linux, and MacOS.

---

**Packet** a packet is a container or box that carries data over a TCP/IP network and internetworks. A packet is the most fundamental logical arbitration of data that is passed over a network.

---

**PCI DSS Compliance** Payment Card Industry Data Security Standard is a set of security standards designed to ensure that all companies accept, process, store, or transmit credit card information to maintain a secure environment.

---

**PHP (Hypertext Preprocessor)** a scripting language used to create dynamic and interactive HTML Web pages. A server processes PHP commands when a website visitor opens a page, then sends results to the visitor's browser.

---

**Pixel** the smallest unit of a digital image or graphic that can be displayed and represented on a digital display device.

---

**Ports (Networking)**

A network port is a process-specific or an application-specific software construct serving as a communication endpoint, which is used by the Transport Layer protocols of Internet Protocol suite, such as User Datagram Protocol (UDP) and Transmission Control Protocol (TCP).

---

**Protocol**

a set of rules and guidelines for communicating data. Rules are defined for each step and process during communication between two or more computers. Networks must follow these rules to successfully transmit data.

---

**Query**

a request for data or information from a database table or combination of tables.

---

**Sandbox**

a type of software testing environment that enables the isolated execution of software or programs for independent evaluation, monitoring or testing.

---

**Sandcastle**

This is the server that the website will be using to host this project

---

**SQL(Structured Query Language)**

SQL is a domain-specific language used in programming and designed for managing data held in a relational database management system, or for stream processing in a relational data stream management system

---

**SQL Injection**

a computer attack in which malicious code is embedded in a poorly designed application and then passed to the backend database

---

**Tab**

A new window within a web-browser

---

**TCP/IP**

(Transmission Control/Internet Protocol) the language a computer uses to access the internet. It consists of a suite of protocols designed to establish a network of networks to provide a host with access to the internet.

---

**Tooltip**

a graphical user interface (GUI) element used in conjunction with the cursor or mouse pointer to display information about an item without needing to click on it

---

**Unchecked Arrays**

Any array where the system allows you to reference a location in memory that has not been allocated

---

**Web Browser (Browser)**

a software program that allows a user to locate, access, and display web pages.

---

**Widgets**

Widget is a broad term that can refer to either any GUI (graphical user interface) element or a tiny application that can display information and/or interact with the user

---

**XSS (Cross Site Scripting)**

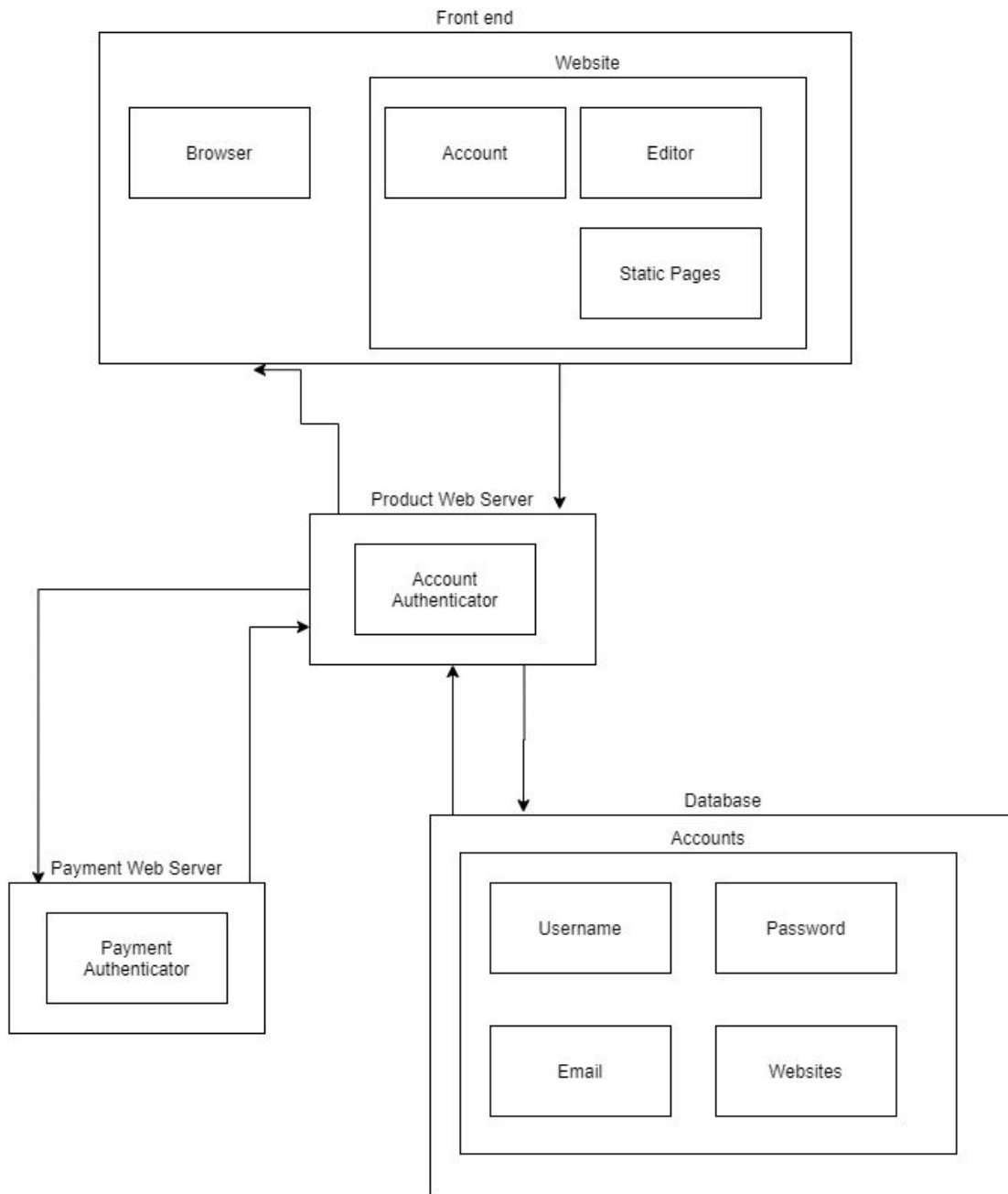
the process of addition of malicious code to a genuine website to gather user's information with a malicious intent

---

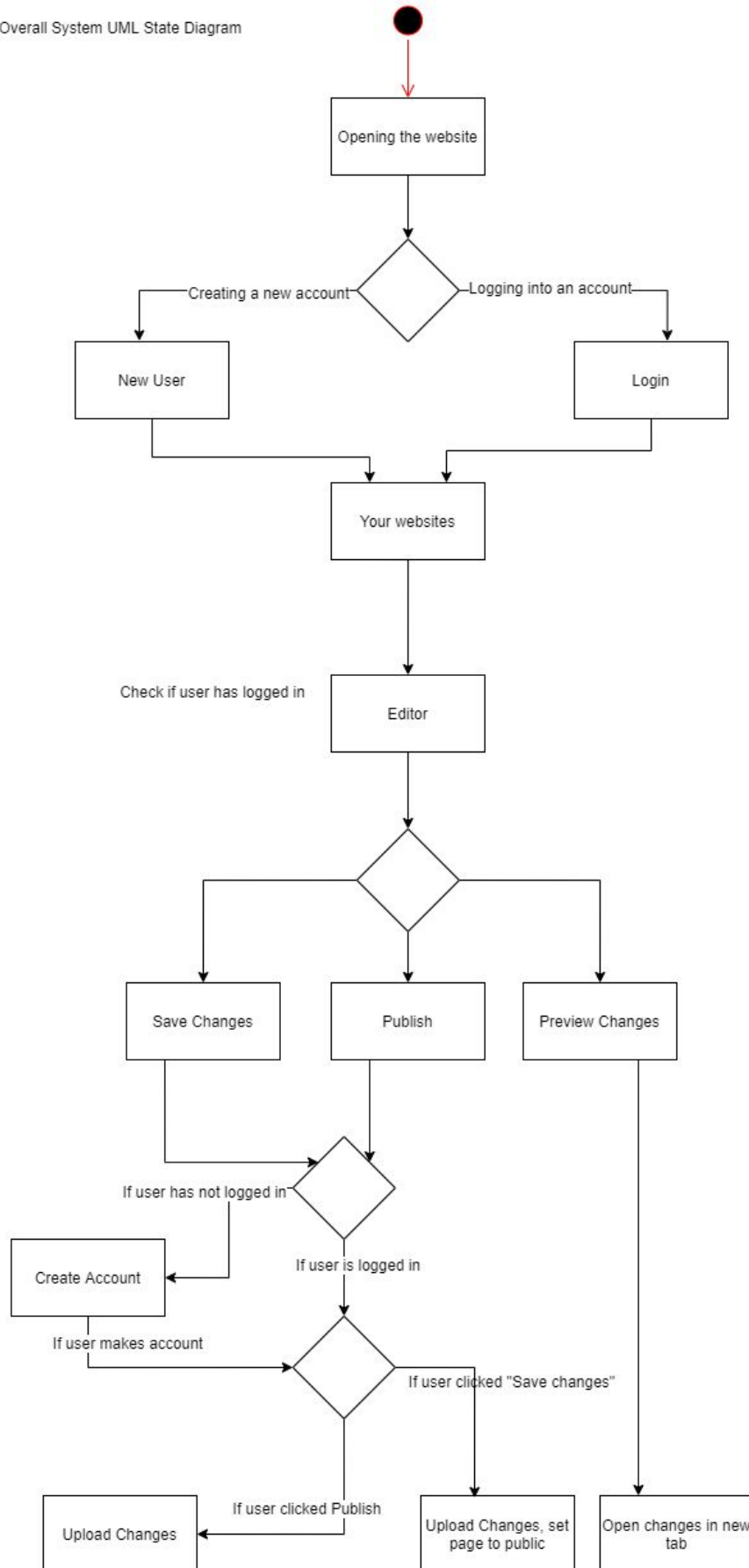


## Appendix B: Analysis Models

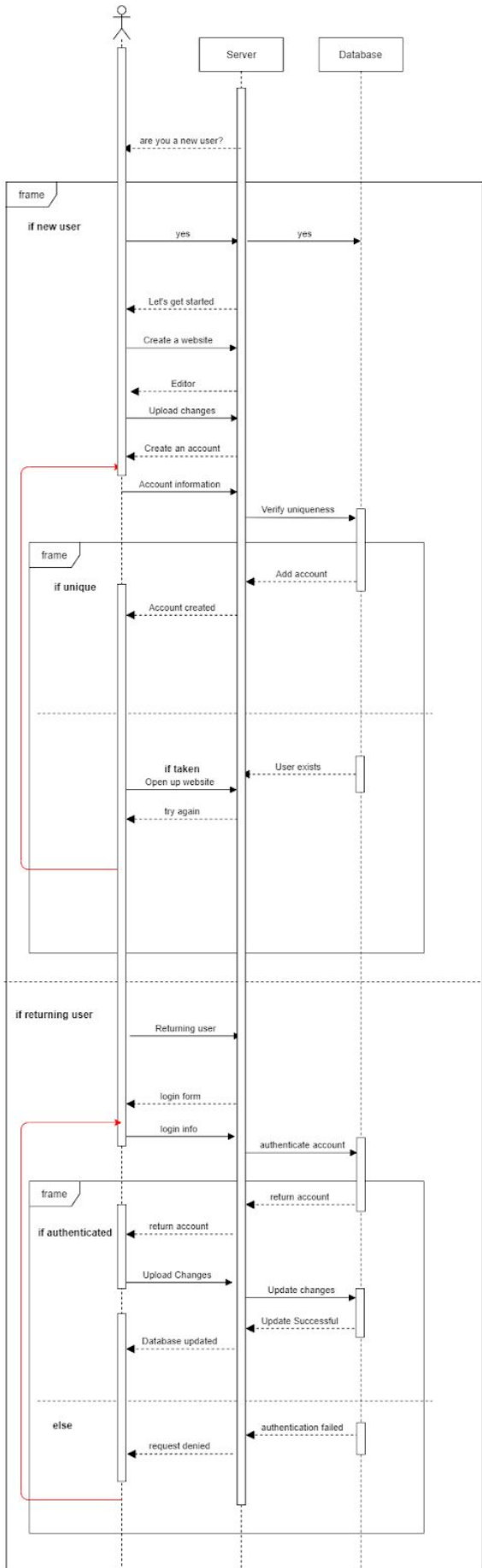
Overall System  
UML Architecture Diagram



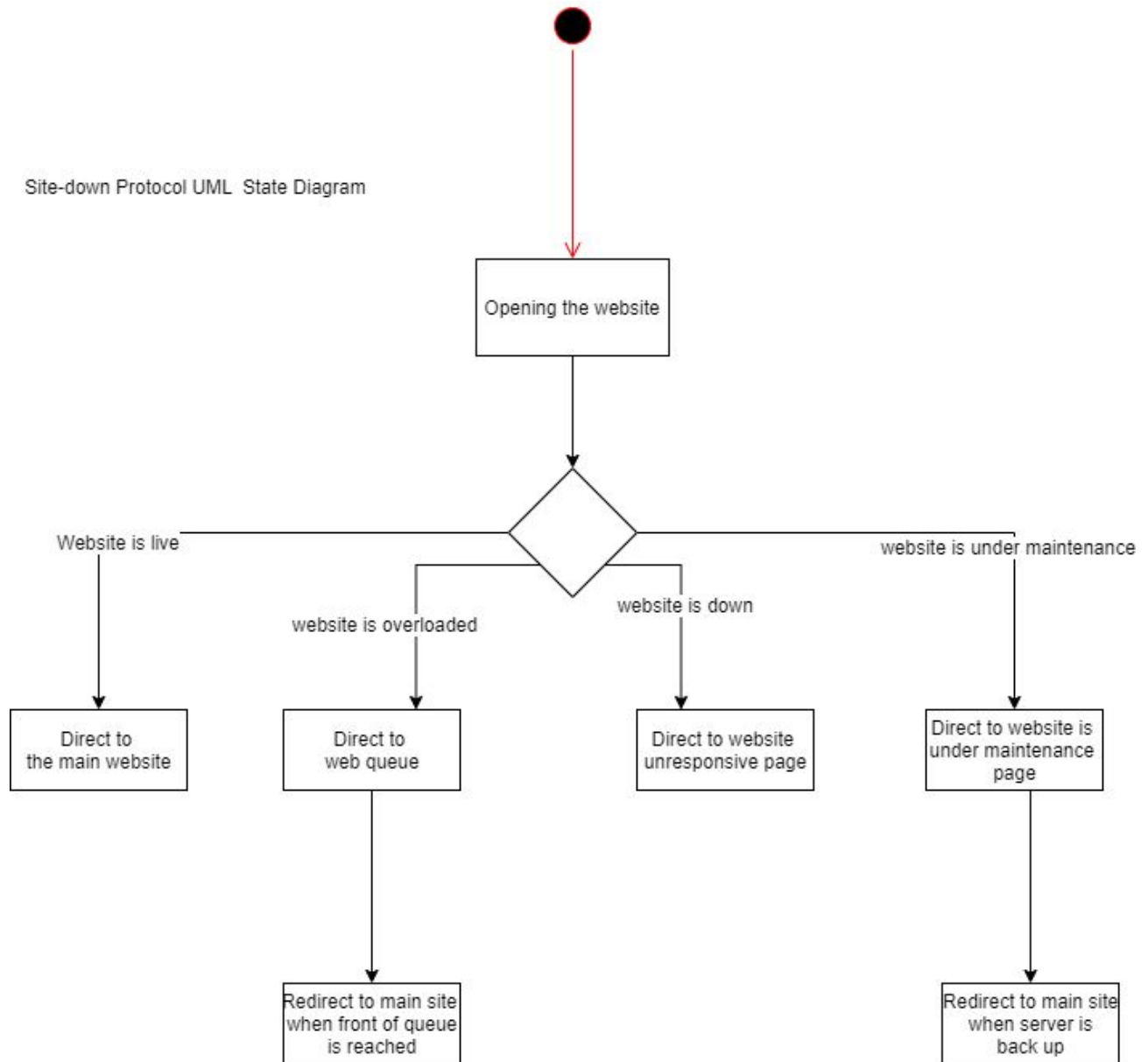
Overall System UML State Diagram

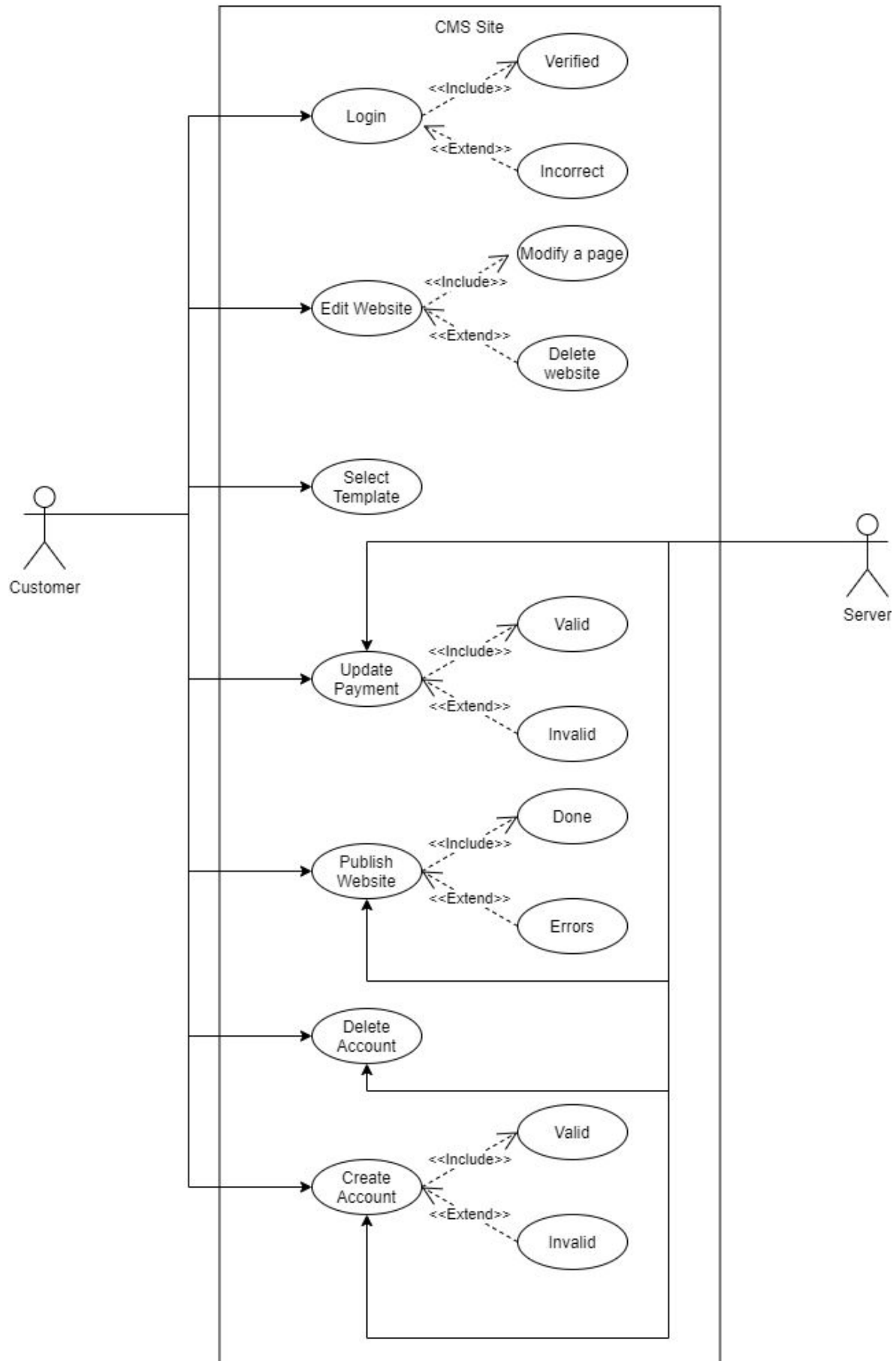


Overall System UML Sequence Diagram



Site-down Protocol UML State Diagram







Example of different areas the web page will be divided into

## **Appendix C: Issues List**

- Adding the undo button
- Encryption of the data
- Injection checking
- Avoiding deadlocks
- Getting drag and drop to work
- Avoiding crashing of the server and site
- Adding multiple users to edit one web site
- How much source code access
- Type of payments supported
- Mobile compatibility
- Error reporting, how in depth

## **Appendix D: Design Requirements**

The following design requirements will be used in implementing this project:

- Easy to use
- Easily change webpages
- Editor loads web pages and makes changes/saves quickly
- Signifiers for each affordance
- Signifiers for processes that take time to show progress
- Build for primary persona
- Easy to learn
- Drag & drop
- What you see is what you get (WYSIWYG)
- Web pages display on mobile
- avoid technical jargon on the UI
- avoid abbreviations (use natural language)
- retrieve feedback
- shortcuts macros, and special keys for common tasks
- indication of errors



## **Appendix E: Revision History**

## CHANGE REQUEST LOG

<b>Project Name</b>	Brix.ca	<b>Date</b>	April 21, 2020
<b>Project Number</b>		<b>Document Number</b>	
<b>Project Manager</b>	Bikram	<b>Project Owner/Client</b>	Group 5

[illegible]


For additional free project management templates, visit [www.mypmllc.com/project-management-resources/free-project-management-templates](http://www.mypmllc.com/project-management-resources/free-project-management-templates)

# Change Request Form

Project Name	Brix	Date	April 21 <sup>st</sup> 2020
Project Number	5	Requestor	Bikram
Project Manager		Project Owner	

Describe the Requested Change
1.1 Updated functionality in purpose statement
1.2 Updated document conventions based on the response from version 1
1.5 Added references
6 added 2 definitions
Throughout document: added conventions to version 1 and 2

Describe the Reason for the Request
1.1 No longer support HTML and CSS direct access
1.2 version 1 conventions belonged in glossary
1.5 Version 1 did not have references, added those used since then
6 these were in version 1 section 1.2 but not in 6
Throughout document: previous convention not consistent with requirements, added conventions to version 1 and 2 material

Risk Identification/Analysis
n/a

Impact Analysis	
Work Products to be Modified	Version Number
1. minimal, one line changed	
2. Minimal, deleting previous section 1.2 (it is already contained in section 6)	
3. minimal added a table	
4. minimal, 2 section added	
5. minimal, some buttons may require formatting	
Describe the impact of the suggested change to work that is already complete.	

Quality Impact			
Additional Quality Assurance or Quality Control Activities			
1. may require grammar correction			
2. Will require review for update section 1.2			
3.none			
4.none			
5. Conventions added throughout document will need peer review			
<i>Describe the impact of the change to quality assurance activities and quality control activities.</i>			
Schedule Impact			
New Deliverables Description	Effort Hours	Date Required	Impact to Other Delivery Dates
1. update 1.1	1	April 22 <sup>nd</sup> 2020	Minimal
2. update 1.2	3	April 22 <sup>nd</sup> 2020	Minimal
3. add references	1	April 22 <sup>nd</sup> 2020	Minimal
4.added 2 definitions	0.1	April 22 <sup>nd</sup> 2020	Minimal
5. implement conventions throughout document	2	April 22 <sup>nd</sup> 2020	Minimal
<i>Based on the impact, state the estimated date for implementing the requested change. State the new estimated project completion date.</i>			

Budget Impact			
New Deliverables Description	Lessen or Eliminate Other Expenses? Please describe.	Cost of New Deliverable	Total
1. update 1.1	n/a	n/a	n/a
2. update 1.2	N/a	n/a	n/a
3. add references	N/a	N/a	N/a
4.added 2 definitions	N/a	N/a	N/a
5. implement conventions throughout document	N/a	N/a	N/a
<i>Describe the overall impact to budget/cost.</i>			

Decision	
<input type="checkbox"/> Approved	<input type="checkbox"/> Rejected
<input type="checkbox"/> Approved with modifications	<input type="checkbox"/> Deferred

Decision	
<input type="checkbox"/> Approved	<input type="checkbox"/> Rejected
Justifications	
Additional Comments	

\_\_\_\_\_  
Approver's Printed Name

\_\_\_\_\_  
Date

\_\_\_\_\_  
Title

\_\_\_\_\_  
Signature

For additional project management templates visit [www.mypmlc.com/project-management-resources/free-project-management-templates](http://www.mypmlc.com/project-management-resources/free-project-management-templates).

# Change Request Form

Project Name	Brix	Date	Saturday April 18 <sup>th</sup> , 2020
Project Number		Requestor	Brock
Project Manager	Bikram	Project Owner	Wes

Describe the Requested Change
Multiple changes to sections 3.1, 3.2, 3.4, 3.16, 3.19, 5
3.1 Replaced REQ-4 with Recover password
3.2 Replaced REQ-10 with validate password and removed REQ-13,REQ-14,REQ-15
3.4, 3.5, 3.7 removed entirely
3.16 removed REQ-84 REQ-85. Added REQ-90, REQ-91, REQ-92
3.19 Changed some sentences in 3.19.2 and REQ-106. Removed REQ-110

Describe the Reason for the Request
Mostly clarifying things and removing parts that were too low priority to implement

Risk Identification/Analysis
No Risk involved. Just modifying language in the CMS

Impact Analysis	
Work Products to be Modified	Version Number
1. CMS will include new features mentioned in the section changes	
2.	
3.	
Describe the impact of the suggested change to work that is already complete.	

<b>Quality Impact</b>			
Additional Quality Assurance or Quality Control Activities			
1. Improvement to SRS.			
2.			
3.			
Describe the impact of the change to quality assurance activities and quality control activities.			
<b>Schedule Impact</b>			
New Deliverables Description	Effort Hours	Date Required	Impact to Other Delivery Dates
1. Just reviewing and approving the changes	<30 mins	April 21st	None
2.			
3.			
Based on the impact, state the estimated date for implementing the requested change. State the new estimated project completion date.			

<b>Budget Impact</b>			
New Deliverables Description	Lessen or Eliminate Other Expenses? Please describe.	Cost of New Deliverable	Total
1. It's just changing the SRS wording	None	\$0	\$0
2.			
3.			
Describe the overall impact to budget/cost.			

<b>Decision</b>	
<input type="checkbox"/> Approved	<input type="checkbox"/> Rejected
<input type="checkbox"/> Approved with modifications	<input type="checkbox"/> Deferred
<b>Justifications</b>	
<b>Additional Comments</b>	

\_\_\_\_\_  
Approver's Printed Name

\_\_\_\_\_  
Date



---

Title

---

Signature

For additional project management templates visit [www.mypmllc.com/project-management-resources/free-project-management-templates](http://www.mypmllc.com/project-management-resources/free-project-management-templates).

# Change Request Form

Project Name	Brix	Date	Saturday April 20 <sup>th</sup> , 2020
Project Number	5	Requestor	Eduardo
Project Manager	Bikram	Project Owner	Wes

Describe the Requested Change
Multiple changes to sections 3.6, 3.15, 3.16, 3.17, 3.18, 3.19 3.21, 3.22, 3.23
3.6 Removed and added requirements, also implemented changes of operation
3.15 Removed and added requirements, also implemented changes of operation
3.16 Removed Site-Down protocol as it was not implemented in final release
3.17 Added some requirements
3.18 Removed and added requirements, also implemented changes of operation
3.19 Removed Billing system as it was not implemented
3.21 Removed Analytics as it was not implemented
3.22 Removed Commerce Support system as it was not implemented
3.23 Added and removed requirements that are not yet implemented

Describe the Reason for the Request
After careful consideration, we felt the need to correct any ambiguity in the document.

Risk Identification/Analysis
No Risk involved. Just modifying language in the CMS

Impact Analysis	
Work Products to be Modified	Version Number
1. CMS will include new features mentioned in the section changes	
2. CMS will remove unused features mentioned in the section changes	
3.	
Describe the impact of the suggested change to work that is already complete.	

<b>Quality Impact</b>			
Additional Quality Assurance or Quality Control Activities			
1. Adding features that was not mentioned			
2. Following implemented features actually added			
3. Changed ways of some of the requirements			
Describe the impact of the change to quality assurance activities and quality control activities.			
<b>Schedule Impact</b>			
New Deliverables Description	Effort Hours	Date Required	Impact to Other Delivery Dates
1. Just reviewing and approving the changes	<90 mins	April 21st	None
2.			
3.			
Based on the impact, state the estimated date for implementing the requested change. State the new estimated project completion date.			

<b>Budget Impact</b>			
New Deliverables Description	Lessen or Eliminate Other Expenses? Please describe.	Cost of New Deliverable	Total
1. Changes to the document	None	\$0	\$0
2.			
3.			
Describe the overall impact to budget/cost.			

<b>Decision</b>	
<input type="checkbox"/> Approved	<input type="checkbox"/> Rejected
<input type="checkbox"/> Approved with modifications	<input type="checkbox"/> Deferred
<b>Justifications</b>	
<b>Additional Comments</b>	

\_\_\_\_\_  
Approver's Printed Name

\_\_\_\_\_  
Date

---

Title

---

Signature

For additional project management templates visit [www.mypmllc.com/project-management-resources/free-project-management-templates](http://www.mypmllc.com/project-management-resources/free-project-management-templates).

# Change Request Form

Project Name	Brix	Date	<b>Saturday April 18<sup>th</sup>, 2020</b>
Project Number		Requestor	Emmanuel
Project Manager	Bikram	Project Owner	Wes

Describe the Requested Change
Multiple changes to sections 3.8, 3.9, 3.10, 3.11, 3.12, as well as Section 6 Appendix B
3.8 Multiple word changes made to improve clarity objective clarity
3.9 Word changes made to improve clarity
3.10 Word changes made to improve clarity
Multiple Requirements changes to add and clarify the project goals
3.11 and 3.12 Have now been combined into a single section as the objective has changed.
Section 6 Appendix B Architecture and State models updated to better reflect current system

Describe the Reason for the Request
After careful consideration, we felt the need to correct any ambiguity in the document.

Risk Identification/Analysis
No Risk involved. Just modifying language in the SRS.

Impact Analysis	
Work Products to be Modified	Version Number
1. SRS will be updated	
2. CMS will include new features mentioned in the section changes	
3. CMS will be better reflected by the new models	
Describe the impact of the suggested change to work that is already complete.	

Quality Impact			
Additional Quality Assurance or Quality Control Activities			
1. Improvement to SRS.			
2.			
3.			
Describe the impact of the change to quality assurance activities and quality control activities.			
Schedule Impact			
New Deliverables Description	Effort Hours	Date Required	Impact to Other Delivery Dates
1. Just reviewing and approving the changes	<30 mins	April 21st	None
2.			
3.			
Based on the impact, state the estimated date for implementing the requested change. State the new estimated project completion date.			

Budget Impact			
New Deliverables Description	Lessen or Eliminate Other Expenses? Please describe.	Cost of New Deliverable	Total
1. It's just changing the SRS wording and models	None	\$0	\$0
2.			
3.			
Describe the overall impact to budget/cost.			

Decision	
<input type="checkbox"/> Approved	<input type="checkbox"/> Rejected
<input type="checkbox"/> Approved with modifications	<input type="checkbox"/> Deferred
Justifications	
Additional Comments	

Approver's Printed Name

Date

---

Title

---

Signature

For additional project management templates visit [www.mypmllc.com/project-management-resources/free-project-management-templates](http://www.mypmllc.com/project-management-resources/free-project-management-templates).

# Change Request Form

Project Name	Group 5	Date	<b>Sunday April 20<sup>th</sup>, 2020</b>
Project Number	5	Requestor	Kaya Ege
Project Manager	Bikram	Project Owner	Wes

Describe the Requested Change
Multiple changes to sections 3.11, 3.19, 4.1
3.11 – Small change added to indicate that HTML components can be added to the editor.
3.19 – Removed part of a section that no longer applies to the CMS. For example, the outage redirect section is removed.
4.1 – Some changes to clarify sections that no longer apply to the CMS. For example, keyboard shortcuts are removed.

Describe the Reason for the Request
After careful consideration, we felt the need to correct any ambiguity in the document.

Risk Identification/Analysis
No Risk involved. Just modifying language in the CMS.

Impact Analysis	
Work Products to be Modified	Version Number
1. The SRS will be modified according to the requested changes.	
2.	
3.	
Describe the impact of the suggested change to work that is already complete.	



<b>Quality Impact</b>			
Additional Quality Assurance or Quality Control Activities			
1. Improvement to the SRS.			
2.			
3.			
Describe the impact of the change to quality assurance activities and quality control activities.			
<b>Schedule Impact</b>			
New Deliverables Description	Effort Hours	Date Required	Impact to Other Delivery Dates
1. Reviewing and approving the changes.	< 30 min	April 21 <sup>st</sup>	None
2.			
3.			
Based on the impact, state the estimated date for implementing the requested change. State the new estimated project completion date.			

<b>Budget Impact</b>			
New Deliverables Description	Lessen or Eliminate Other Expenses? Please describe.	Cost of New Deliverable	Total
1. Only changing the SRS wording	None	0	0
2.			
3.			
Describe the overall impact to budget/cost.			

<b>Decision</b>	
<input type="checkbox"/> Approved	<input type="checkbox"/> Rejected
<input type="checkbox"/> Approved with modifications	<input type="checkbox"/> Deferred
<b>Justifications</b>	
<b>Additional Comments</b>	

\_\_\_\_\_  
 Approver's Printed Name  
 Change Request Form

\_\_\_\_\_  
 Date

---

Title

---

Signature

For additional project management templates visit [www.mypmllc.com/project-management-resources/free-project-management-templates](http://www.mypmllc.com/project-management-resources/free-project-management-templates).

# Change Request Form

Project Name	Brix	Date	Saturday April 18th, 2020
Project Number		Requestor	Tyler
Project Manager	Bikram	Project Owner	Wes

Describe the Requested Change
Multiple changes to sections 2.2, 2.4, 2.6, 2.7
2.2, Removed User Admin Functions that weren't implemented
2.4, Removed components that were not integrated in the system
2.6, Removed information that will not be provided in User documentation
2.7, Removed Third-party component that wasn't used and added open source libraries that were used

Describe the Reason for the Request
Mostly clarifying things and removing parts that were too low priority to implement

Risk Identification/Analysis
No Risk involved. Just modifying language in the CMS

Impact Analysis	
Work Products to be Modified	Version Number
1. CMS will include new features mentioned in the section changes	
2.	
3.	
Describe the impact of the suggested change to work that is already complete.	

<b>Quality Impact</b>			
Additional Quality Assurance or Quality Control Activities			
1. Improvement to SRS.			
2.			
3.			
Describe the impact of the change to quality assurance activities and quality control activities.			
<b>Schedule Impact</b>			
New Deliverables Description	Effort Hours	Date Required	Impact to Other Delivery Dates
1. Just reviewing and approving the changes	<30	April 21st	None
2.			
3.			
Based on the impact, state the estimated date for implementing the requested change. State the new estimated project completion date.			

<b>Budget Impact</b>			
New Deliverables Description	Lessen or Eliminate Other Expenses? Please describe.	Cost of New Deliverable	Total
1. It's just changing the SRS wording	None	\$0	\$0
2.			
3.			
Describe the overall impact to budget/cost.			

<b>Decision</b>	
<input type="checkbox"/> Approved	<input type="checkbox"/> Rejected
<input type="checkbox"/> Approved with modifications	<input type="checkbox"/> Deferred
<b>Justifications</b>	
<b>Additional Comments</b>	

\_\_\_\_\_  
Approver's Printed Name

\_\_\_\_\_  
Date

\_\_\_\_\_  
Title

\_\_\_\_\_  
Signature

For additional project management templates visit [www.mypmllc.com/project-management-resources/free-project-management-templates](http://www.mypmllc.com/project-management-resources/free-project-management-templates).

# Change Request Form

Project Name	Group 5	Date	<b>Sunday April 20<sup>th</sup>, 2020</b>
Project Number	5	Requestor	Wes Nantais
Project Manager	Bikram	Project Owner	Group 5

Describe the Requested Change
Multiple changes to all sections. General editing and formatting as well.

Describe the Reason for the Request
SRS needs updated to match current implementation

Risk Identification/Analysis

Impact Analysis	
Work Products to be Modified	Version Number
1. The SRS	
2.	
3.	
Describe the impact of the suggested change to work that is already complete.	

<b>Quality Impact</b>			
Additional Quality Assurance or Quality Control Activities			
1.			
2.			
3.			
Describe the impact of the change to quality assurance activities and quality control activities.			
<b>Schedule Impact</b>			
New Deliverables Description	Effort Hours	Date Required	Impact to Other Delivery Dates
1.			
2.			
3.			
Based on the impact, state the estimated date for implementing the requested change. State the new estimated project completion date.			

<b>Budget Impact</b>			
New Deliverables Description	Lessen or Eliminate Other Expenses? Please describe.	Cost of New Deliverable	Total
1.			
2.			
3.			
Describe the overall impact to budget/cost.			

<b>Decision</b>	
<input type="checkbox"/> Approved	<input type="checkbox"/> Rejected
<input type="checkbox"/> Approved with modifications	<input type="checkbox"/> Deferred
<b>Justifications</b>	
<b>Additional Comments</b>	

\_\_\_\_\_  
 Approver's Printed Name  
 Change Request Form

\_\_\_\_\_  
 Date

---

Title

---

Signature

For additional project management templates visit [www.mypmllc.com/project-management-resources/free-project-management-templates](http://www.mypmllc.com/project-management-resources/free-project-management-templates).