Requirements Specification

for

Vex

Version 1.7

4F00 Project Stage Three

Prepared by Group 4

Lingjue Zhou 5606611

Yue Guan 5727698

Yufan Shen 5533252

Jingxiang Ma 5704929

Maka Hu 5706791

Penghan Yan 5949292

Xiaoyu Wang 6360135

April. 21, 2020

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Penghan Yan	2.6	Reference sorted from a to z, Some grammar correction	1.1
Lingjue Zhou	4.17	Rewrite hardware interface	1.2
Lingjue Zhou	4.18	Add Appendix Analysis Models	1.3
Lingjue Zhou	4.19	Add Appendix Code Conversion	1.4
Penghan Yan & Xiaoyu Wang	4.20	Update user interface images, move components, attributes, toolbar definition to Appendix	1.5
Penghan Yan	4.20	Added SQL schema, corrected SQL naming issue	1.6
Lingjue Zhou	4.21	Add Appendix Hazard Tree	1.7

1. Introduction

1.1 Purpose

In general, the website builder is the tool, which can allow users to construct websites based on the user's ideas without coding manually. The purpose of this project to create an online web-based website and run on services provided by the web builder. To create a website, all the user need is an internet connection and a web browser. Our website builder will be easy to get hands-on even for first-timers who don't have any coding background, though, users should spend time to familiarize the web interface, features and plug-in tools.

1.2 Document Conventions

We refer the host machine of a company intend to provide web design service(security)

- *Brock server* is interchangeable with *Sandcastle server* in this document.
- When referring to a machine, we use italic text such as *Brock Server*, and **Bold** for users.

FREQ	Functional Requirement
PREQ	Performance Requirements
SREQ	Security Requirements
NREQ	Nonfunctional Requirements

1.3 Intended Audience and Reading Suggestions

This document is intended to assist and inform individuals who will make use of this software for testing purposes and implementations, for both testers and users. This document should also be applied by developers and project managers during the development.

1.4 Project Scope

The scope of the project is creating a website builder that allows users or people who don't even have any coding background to create their websites. The website builder includes a drag and drop feature applied for creating pages with layouts based on the user's personal preferences. Users are required to login so that all the functions will be available. On the other hand, if users don't want to log in, the features, such as saving the webpages for future editing and sharing user's page with other clients, will be limited.

1.5 References

Templates, resources and software use for specification:

1. Amanda, C. (2017, January 23). *Metadata Delivery - Crossref*. Retrieved from

https://www.crossref.org/services/metadata-delivery/

2. Antonielli, A. (2013, March 1). *Yeats's Digital Identity: Q&A with Web Editor Neil Mann* Retrieved from https://doaj.org/article/95eef53f9e834fe58efa1e74ade52d55
3. Content management system. (2020). Retrieved 6 February 2020, from https://en.wikipedia.org/wiki/Content_management_system
4. Icons – Graphic resources: https://www.zondicons.com/icons.html
5. Sketch – For use to design interface
6. StarUML - For use cases and Sequence diagrams
7. Turner, W., & Leonard, S. (2017). *JavaScript for sound artists: learn to code with the Web Audio Api*. Boca Raton: CRC Press.
8. WIX (2020). Retrieved 6 February 2020, from http://www.wix.com

2. Overall Description

2.1 Product Perspective

The *Vex System* is a CMS based system that is created for use to design webpages, it is a web-based system implementing the client-server model (See Figure 1), The *Vex System* provides a simple mechanism for users to design and share webpages.

The following are the main features that are included in *Vex System*:

- Cross-platform support: Offers operating support for most of the known and commercial operating systems.
- Drag and drop component support: The system allows the user to drag an HTML component and drop into the working project to design their webpage.
- Storage support: The system allows the user to create their accounts in the system and save their designed webpages in our *cloud services*.
- Number of users being supported by the system: Though the number is explicitly not mentioned but the system can support a large number of online users at a time.

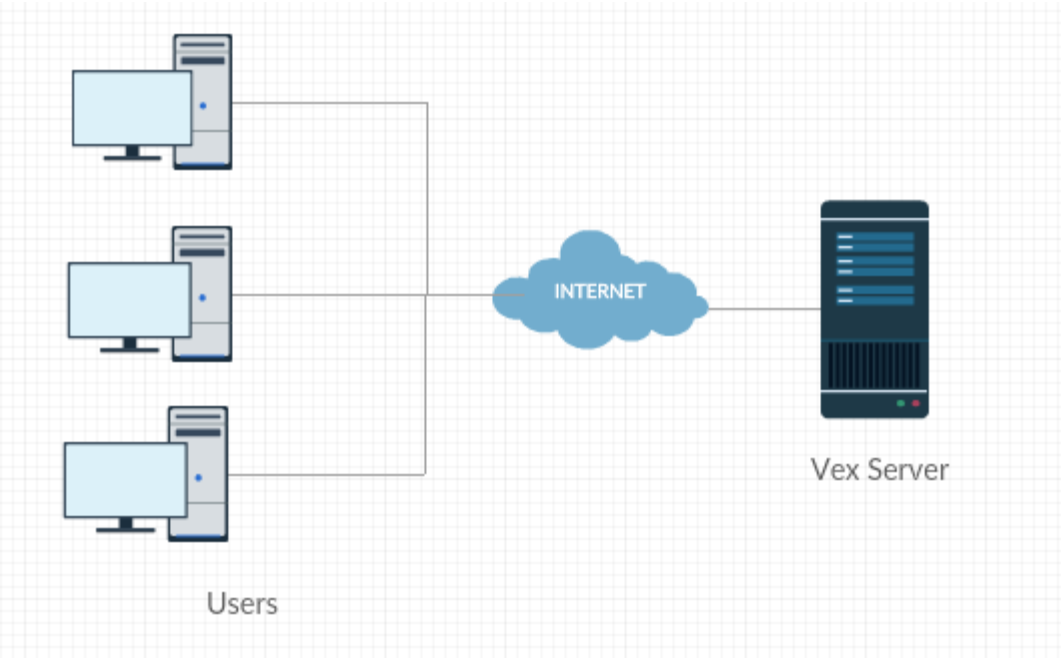


Figure 1: Overall view of the system components

2.2 Product Features

Product Features include:

1. The ability for users to register/login to the system through the hosted webpage.
2. The ability for users to create/modify/delete a webpage project.
3. The ability to save a webpage project into the database.
4. The ability for users to drag & drop web components into their current webpage project.
5. The ability for users to public their own webpages.
6. The ability for users to request an assistant from technical support.
7. The ability for system user to create/modify/delete/restore the tag component.
8. The ability for system user to access and modify an existing webpage project.
9. The ability for system user to control user access.
10. The ability to download the project source code.

2.3 User Classes and Characteristics

There are two types of users that interact with the system, **users** need to get service to create a webpage, technical users maintain the system feature, create/modify the components feature.

- **Users**:

  The main users for the system are the ones who will be accessing the system most frequently. These users will use the server as a tool to design a webpage as their own, by selecting different HTML components. Their privilege level is medium since the other users cannot access the webpage that doesn't belong to them, as long as the webpage is not publicly visible.

- **Technical Users**:

  This user class which is expected to have high educational level, or experience, and have high the highest privileges within the system, since technical user have the ability to solve any issues that user experienced when using *Vex System*, they have the ability to maintain current HTML components and create new HTML component template.

2.4 Operating Environment

Support operating system:

- Windows Server 2012, 2012 R2, 2016, 2019
- Ubuntu Server x86 or x65 18.0 or later
- Debian Server x86 or x65 10.2.0 or later
- CentOS Server x86 or x64 8.0 or later

This system can be setup on both in Linux and Windows based servers, but it must install those environments:

- Apache HTTP Server 2.x (httpd) with PHP 5.6.x enable
- PostgreSQL 9.6 or later

To access the local server should made with one of the following browsers with these versions or later:

- Google Chrome Version 71.0.3578.98
- Microsoft Edge 40.15063.674.0
- Firefox Standard Release 65.0
- Safari 12.0.2

2.5 Design and Implementation Constraints

Constraints for development of the software include:

- **Language requirements**

The software must be designed such that it can run on the sandcastle at Brock University. The language requires including but not limited to Javascript, PHP.

- **Communications protocols**

Require HTTP/HTTPS protocols, port: 80/433

- **Security considerations**

Any user's information must store safe and secure, the password must be encrypted in the database, at least using the SHA256 cryptographic hash function.

2.6 User Documentation

An online tutorial will be provided once a user first login into the system, the tutorial will demonstrate the work sequence on how to create a new webpage project, the exists user can review the tutorial as well, by clicking the tutorial button.

Manuals will be provided to the technical users, which shows detailed instruction that how they manage the toolbar components, system config, troubleshot issue once an error code provided.

2.7 Assumptions and Dependencies

Assumptions:

- The tag component code e.g. , <p> should formatted correctly. At least have a starting and closing tag for a component.
- The users installed one of the supported browsers. (Sec. 2.4)
- The users have a decent internet connection.
- Server administrator is responsible for information security & provide proper ports to the public.
- Users will not intentionally destroy the system, e.g. submit an unsupported file (virus, etc.)
- The server will not be violating any Internet Ethnic or Cultural Rules and won't be blocked by the ISP.

Dependencies:

- We are dependent on our hosting service to provide 24/7 online services. (Internet connection, power supply)

- Website interface must be friendly and easy to use.

3. System Features

3.1 User Can Sign Up/Login into The System

A register page will provide to user, since all the webpage projects will relate to the account when the users wish to start to design a webpage.

Priority: Very High

3.1.1 Stimulus/Response Sequences

It will have to promote a registration page, in order to register, users will be required to provide information such that username, password, and email, etc. information, and pass the validation (e.g. reCAPTCHA from Google), see the sequence diagram for Sign up and Login Fig 3.1.1.1 and 3.1.1.2

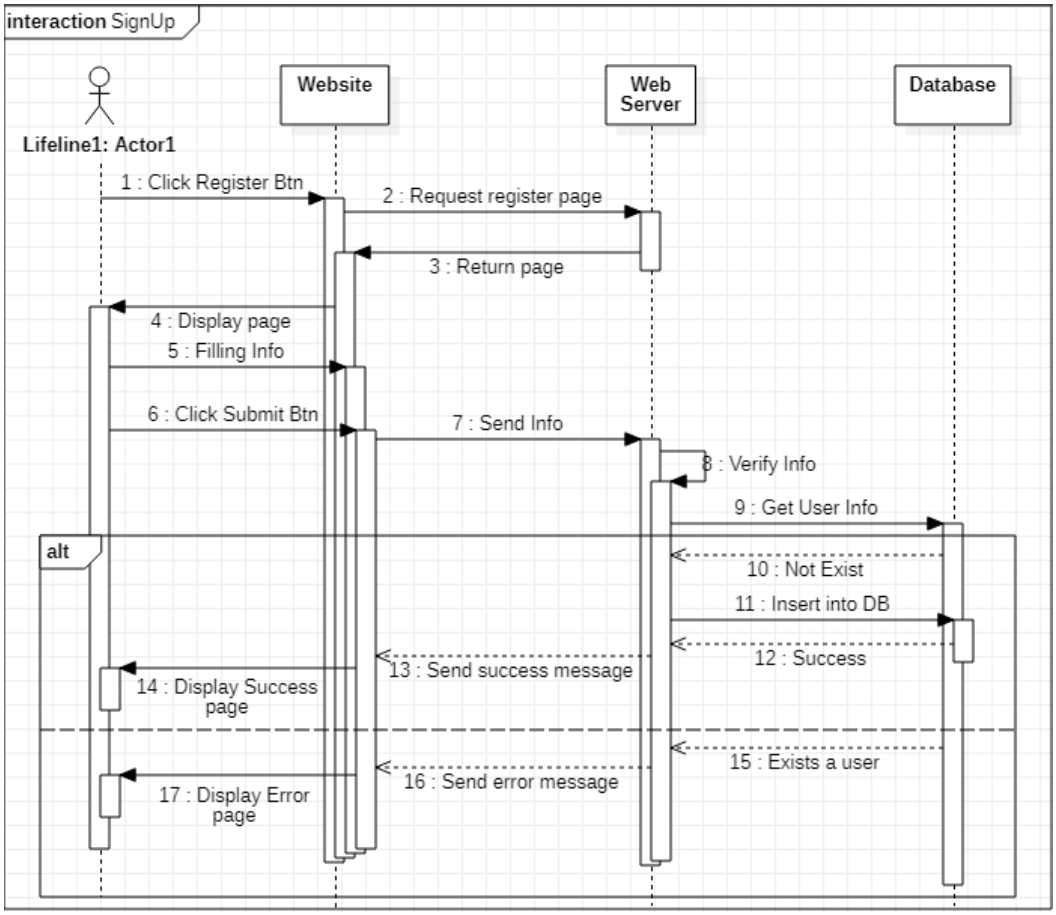


Figure 3.1.1.1 Sequence diagram for Sign up

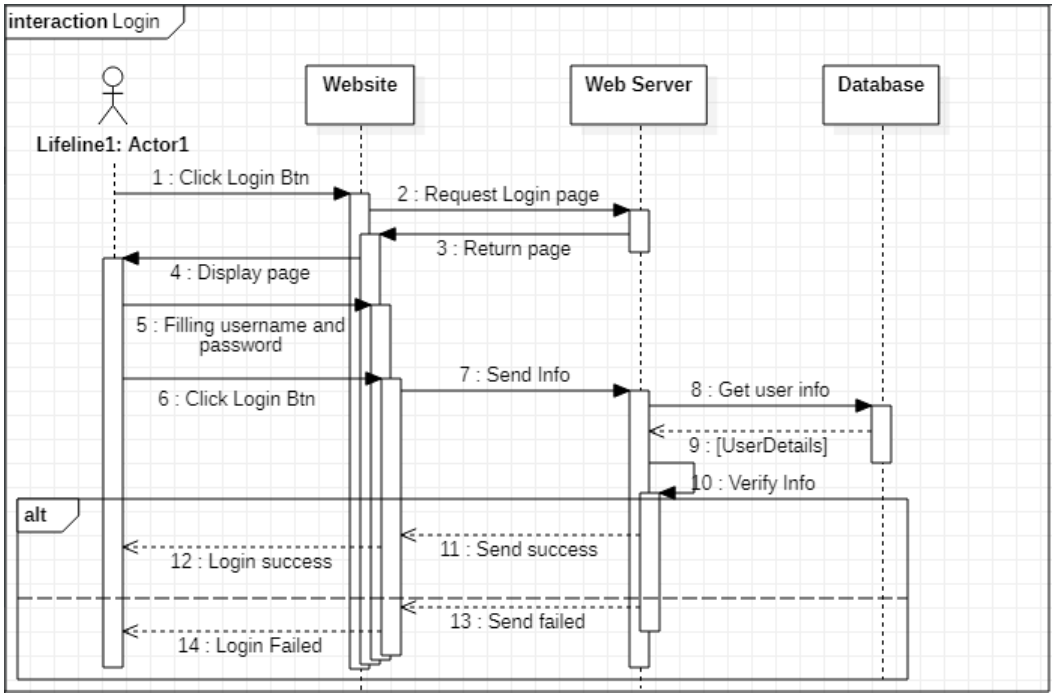


Figure 3.1.1.2 Sequence diagram for Login

3.1.2 Functional Requirements

- FREQ-1: Users should be able to register an account.
- FREQ-2: Users should be able to login once they registered an account.
- FREQ-3: System should be able to prevent duplicated username.
- FREQ-4: System should require user a strong password.
- FREQ-5: System should be able to establish that a computer user is human.

3.2 User Friendly Editing Environment

We provide a web interface to users. Although the users are most likely to lack of writing HTML code, they can easily use the drag and drop feature, such as drag a “textfile” component and put it below of a picture.

Priority: Very High

3.2.1 Stimulus/Response Sequences

To use the editor page, users might enter though the home page, users might authenticate themselves in order to use the *cloud system*. Once the authentication is complete, user will be presented the project list, after select/create a webpage project, users can start working on their webpage. See Editor Interface in Sec. 4.1.2

3.2.2 Functional Requirements

- FREQ-6: All the HTML components provided by Vex System should perform their function. That is, when a user drags a defined component into their page, the corresponding HTML code should be inserted into the working webpage.

- FREQ-7: The components in working webpage have ability to repositioning.

3.3 User Can Save/Discard a Webpage

We offer user to store their unfinished or finished webpage in our cloud services, which user could continue to design their webpage next time. The saved webpage project will appear on the project list, which relates to the user, and selecting one of the projects will enter the editor mode and restore the previous work.

Priority: Medium

3.3.1 Stimulus/Response Sequences

After user authenticated themselves and modified their webpage, user may click the save button in the top right corner, a saving procedure will be running at the backend. After saved successfully, user will be notified. If user wish to discard a specific webpage project, user could select a webpage project they wish to discard, a warning message will popup, after double confirming, the webpage will be marked as deleted, and user won't see that webpage project in the project list. See Save/Discard Sequence in Fig.3.3.1.1

3.3.2 Functional Requirements

- FREQ-8: User should be able to save a current working webpage into database.
- FREQ-9: User should be able to delete a webpage project from the database.
- FREQ-10: System should be able to detect failure when inserting/updating the database, it will undo the saving/discarding operation and notify technical user.
- FREQ-11: User should be able to download the webpage project source code into user's local computer.

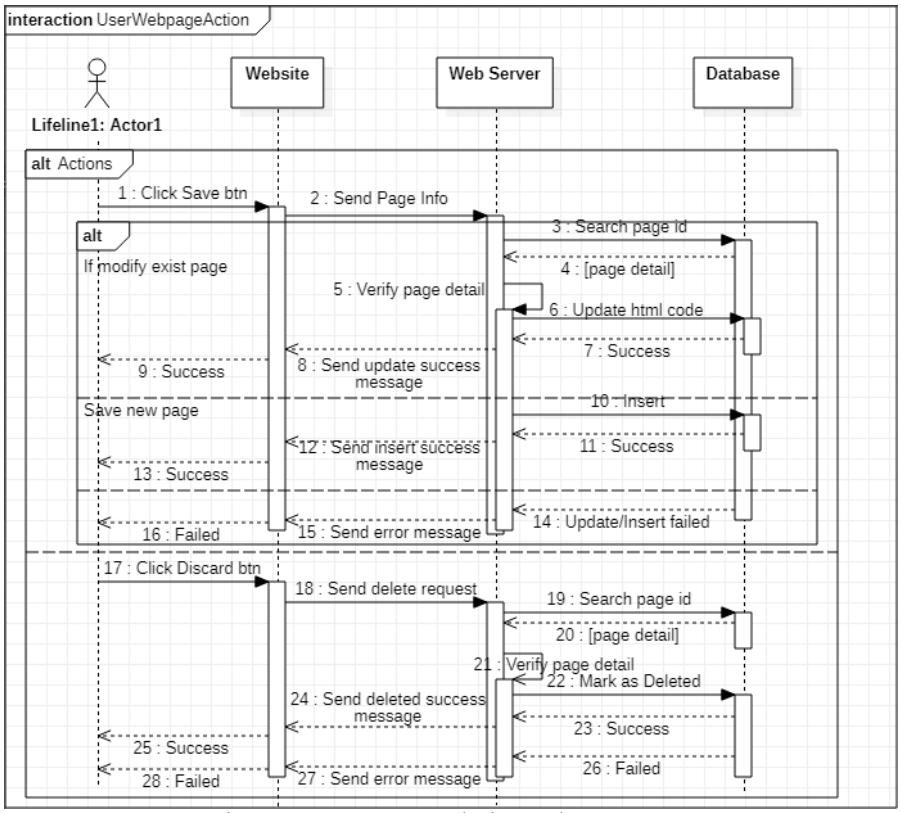


Figure 3.3.1.1 Save/Discard Sequence

3.4 User Can Publish a Webpage

We provide a live system for those who don't have a web server or elastic compute from Google or Amazon. The user could publish their designed webpage and the system will generate a URL for that page. After the webpage has been publicly visible, anyone could access their pages by using that URL.

Priority: Low

3.4.1 Stimulus/Response Sequences

After user authenticated themselves and was in the project list, the user could select a webpage project, and click the set to the public button, it will promote a message dialog that shows the URL, and user will see the status of this project is public or not. This also could be done in the editor, a public switch/button will be in the top right corner.
Revising the webpage visibility to private are the same, click the public button will set the webpage to non-accessible from the public.

3.4.2 Functional Requirements

- FREQ-12: User should be able to set the webpage project to be accessible or non-accessible.
- FREQ-13: System should be able to verify the user have permission to access the webpage that is not set to public.

3.5 Technical User Can Add/Modify/Delete HTML Components

The HTML components toolbar is managed and maintained by the technical users, they could add a new HTML component template into the system and let the normal user to use.

Priority: High

3.5.1 Stimulus/Response Sequences

After technical user authenticated themselves, they will be directed to the manager page. For the HTML components set.
The technical user may select to create a new component, and it requires the user to set the component name, input the template code, and icon pic.
They may select a exists component and modify the information of it. In case of debugging, it could set this component to disabled, thus, it won't be shown in the toolbar.
They also can delete existing components, it will pop up a dialog to double confirm the action by clicking the delete button.

3.5.2 Functional Requirements

- FREQ-14: Technical users have ability to add a new HTML component.
- FREQ-15: Technical users have ability to modify a exists component and could set this component enables or not.
- FREQ-16: Technical user have ability to delete a exists component.

3.6 Technical User Can Be Able to Modify Their Users' Webpages

Technical users could help user to solve some issues that happened during user designing, user may send an issue request (ticket), and the technical user could receive the email message and access the user's webpage. They may also to set accessibility to the published page, if those webpages violate the regulation or law.

Priority: Medium

3.6.1 Stimulus/Response Sequences

After technical user authenticated themselves, they will be directed to the manager page. They can access the user's webpage project through URL or through searching the project name or username feature and enter to the editor mode.

3.6.2 Functional Requirements

- FREQ-17: Technical user should be able to search webpage projects by username or project name.
- FREQ-18: Technical user have ability to access all the webpage projects.
- FREQ-19: Technical user have ability to modify/delete the webpage projects.
- FREQ-20: Technical user have ability to set accessibility of webpage project.

3.7 Technical User Can set User accessibility

Technical users could be able to privilege user access, which can grand access as the technical user and prevent user publish the webpage which violates copyright, law, etc.

Priority: Low

3.7.1 Stimulus/Response Sequences

After technical user authenticated themselves, they will enter the user account management page. They can ban the users or set a user as admin by clicking the options.

3.7.2 Functional Requirements

- FREQ-21: Technical user have ability to ban a user.
- FREQ-22: Technical user have ability to set a user to admin permission.
- FREQ-23: Once a user was banned by technical user, user will not be able to login into the system, and published page will not be able to access as well.

4. External Interface Requirements

The website will be provided for this system for those people who are lack of knowledge to make a website.

For these target users, the website offers two features depends on their login status. First, for non-login users, they do not need to create their own account and passwords and able to quickly use some basic functions of the site. For example, using tools to create own static web pages and export HTML and JavaScript code. However, this kind of users cannot use functions such as Publish, Live.

Second, login status requires users to create their own account passwords. If users have their own account passwords, some additional features will be open. For example, they can provide to administer with bugs encountered in the design and publish/live their own websites.

For all users, the site offers all features for free. Therefore, any payment information request will not be considered in UI designing.

4.1 User Interfaces

(When a user uses the webpage for the first time, a pre-guide is provided by the webpage internally)

4.1.1 Home Page and Account Creation

The Home page of the website includes user's guidance, login and register function.
The login and sign up button (Figure. 4.1 and Figure. 4.2) are at the top-right corner of the page.
Once users create their own accounts, they can log in to the system through the login page. Also, publish/live function and save function will be available to these users.

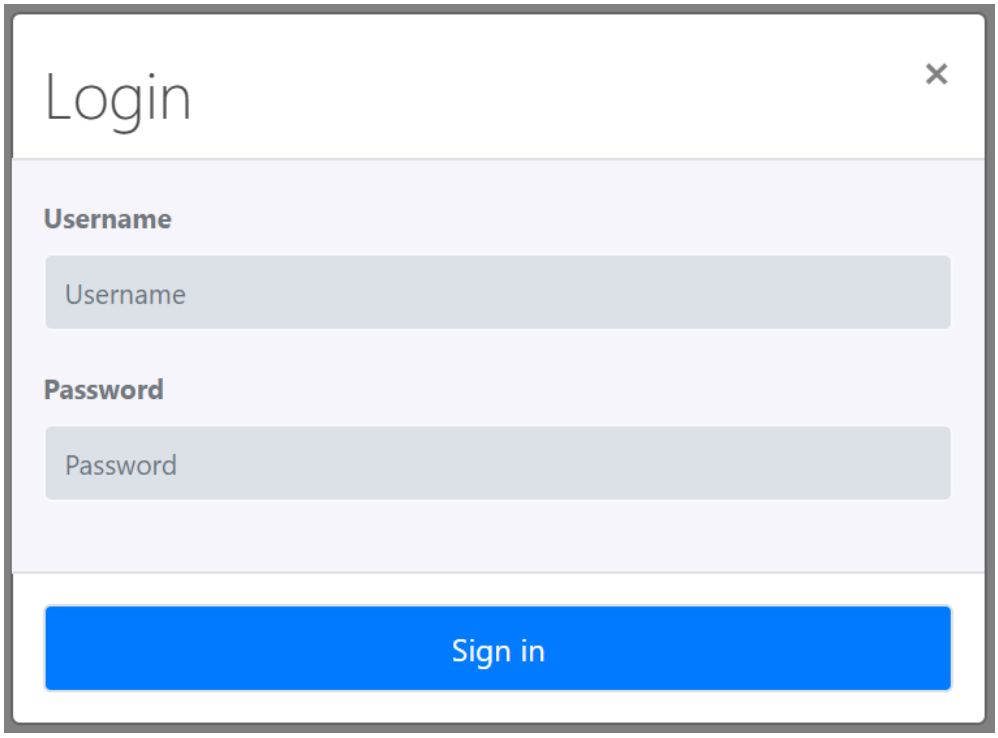


Figure 4.1 Log in Page

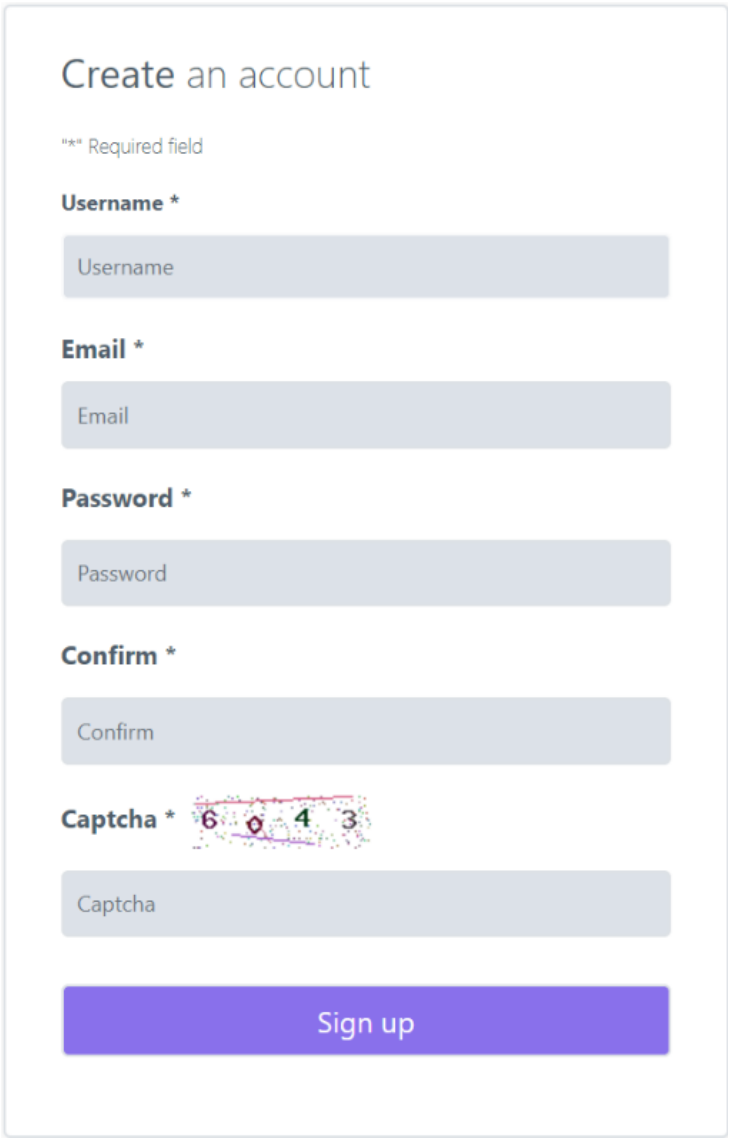


Figure 4.2 Sign Up Page

4.1.2 Editor Interface

Figure 4.3 is a webpage of the main interface of the editor and it allows user to design their website. This page includes a project tree that can overview the projects which the user designed, a component tool panel that allows user to drag and drop the web components for designing, a main panel of users' website, a style modify panel that allows user to modify the components they selected and a tool bar at the top of page. If users did not log in to the system, they can still use the editor on this page to design the webpage but exporting HTML code is not available for them.

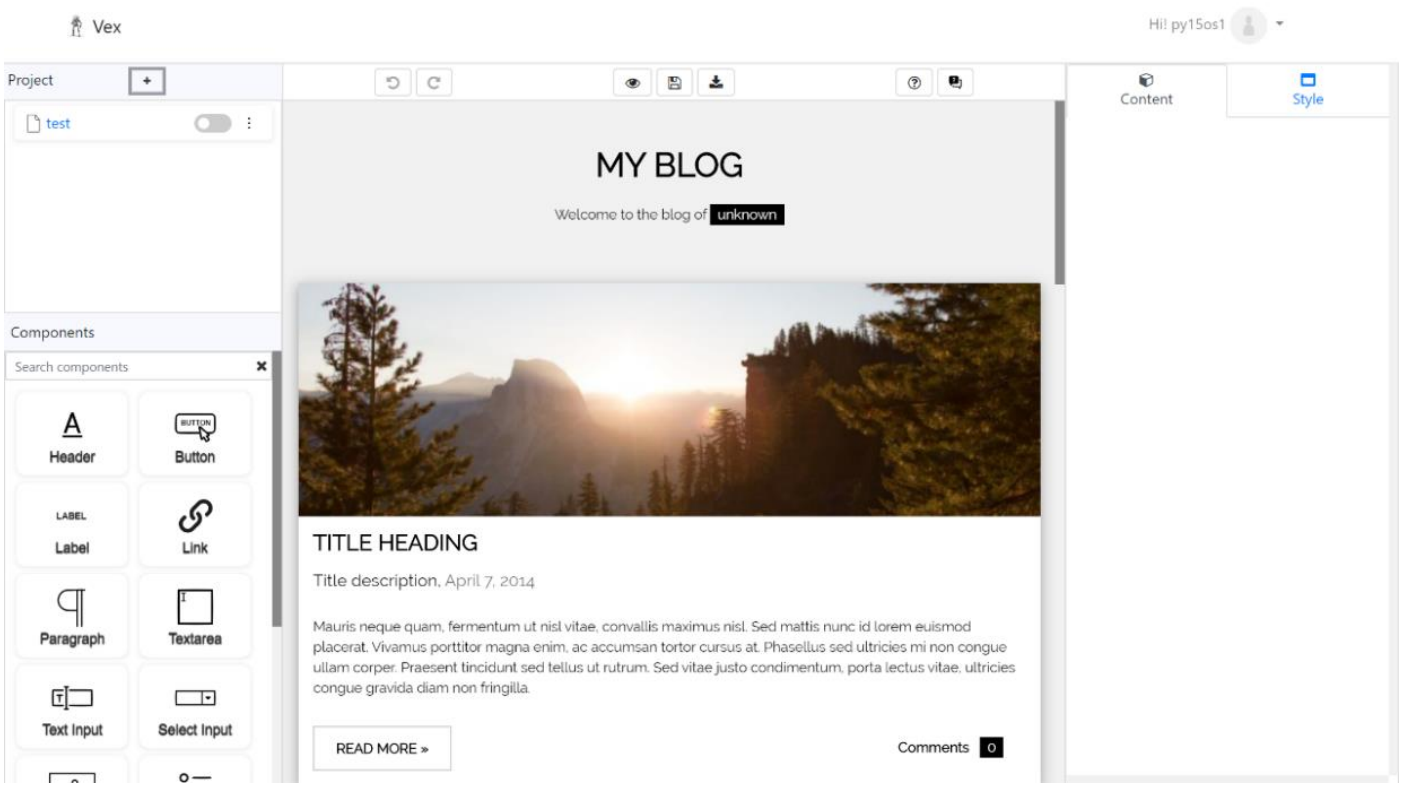


Figure 4.3 Editor Page

4.1.3 Components Panel

Components panel (Figure 4.4) contains plenty of HTML components that helps users to create their websites simply. Users can easily drag the components into the corresponding position instead of typing code to generate a web page. If the user wants to place a component to the page, he can do this by clicking the mouse on the selected component and holding. Move the mouse to find a correct place to drop the component. Once the place be confirmed, release the mouse and the component will be automatically placed to the designed area. For more details of functions of each component, please see *Appendix D*.

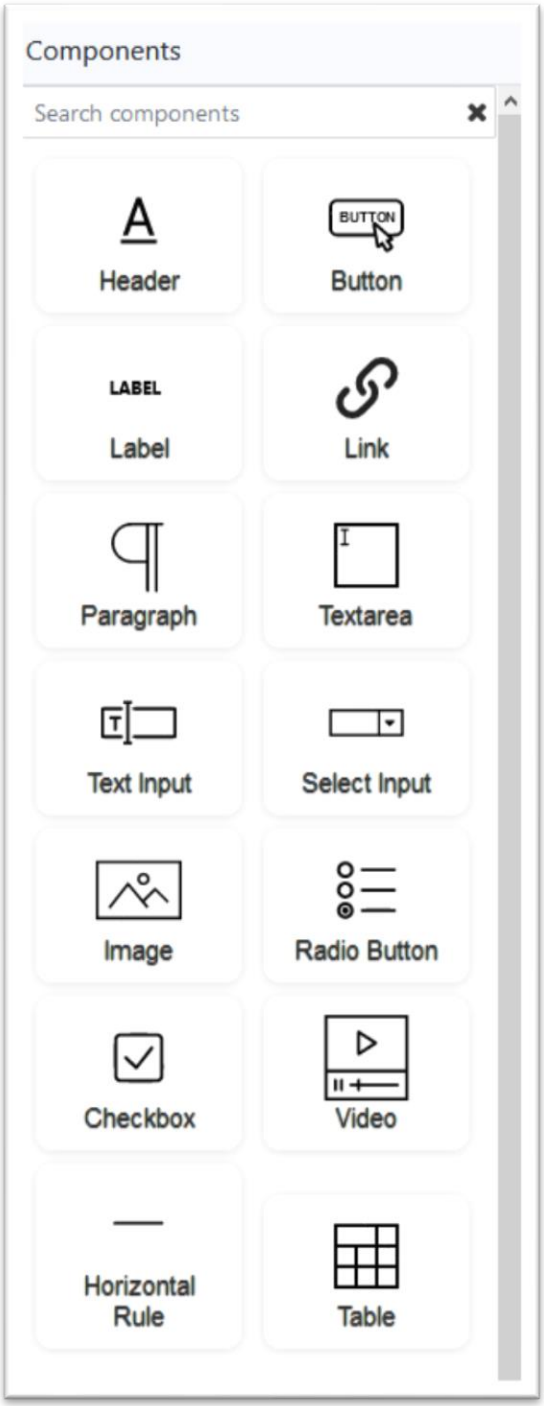


Figure 4.4 Component Panel

4.1.4 Style Panel

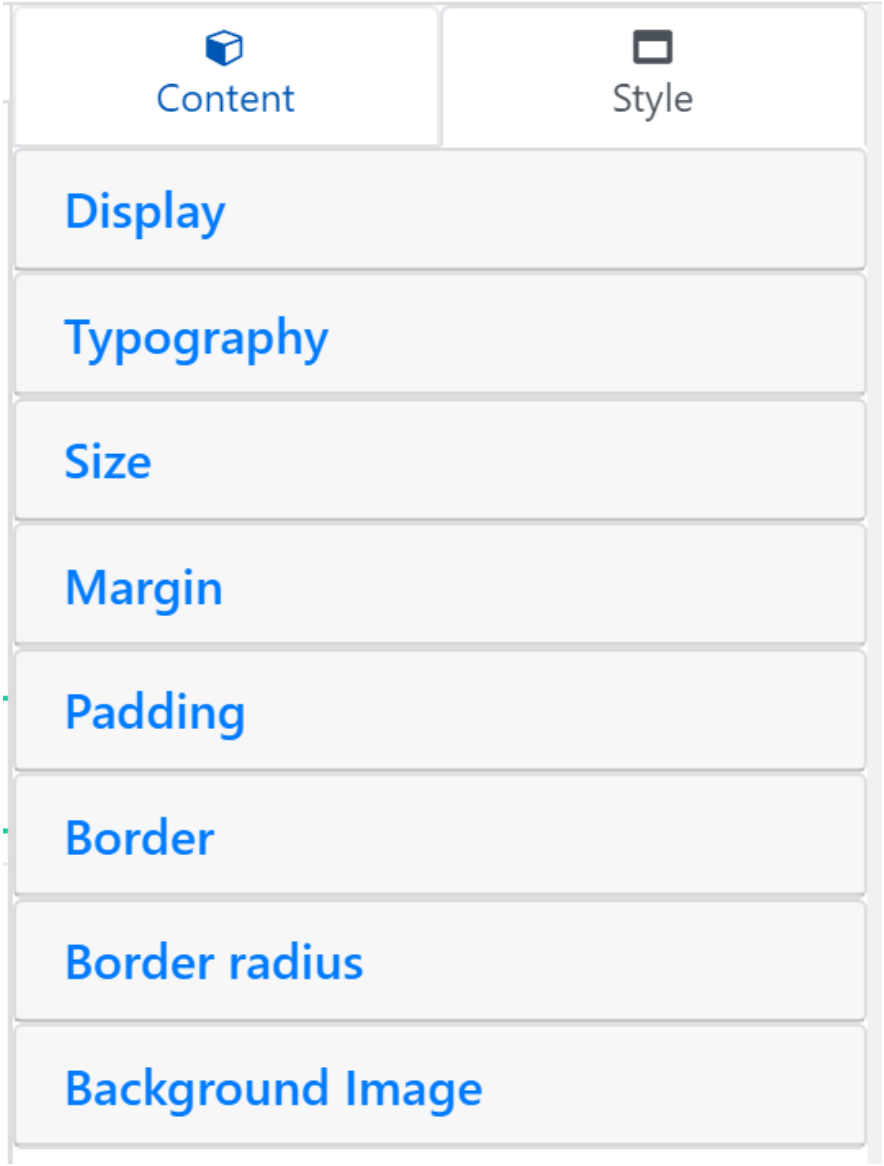


Figure 4.5 Style Tool Panel

A style tool provides a function that allows designers to modify the components of the web page such as Display, Typography, Size, Margin, Padding, Border, Border radius, and Background Image. This is located on the right-hand side of the editor page. For more details about style panel, see *Appendix E*.

Content Tool

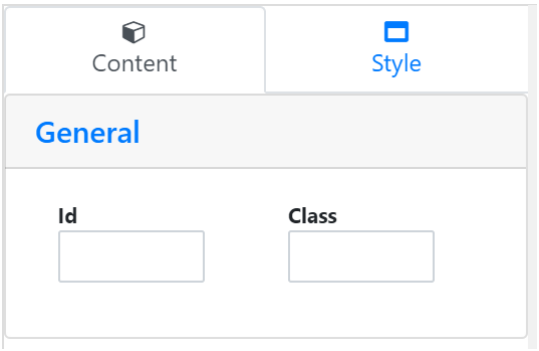


Figure 4.6 Content Tool bar

A content panel provides a function that allows the designers to modify the attributes of the components they selected. For all the components, they can modify the ID and Class of the CSS style. For specific components, they have different attributes. For example, designer can modify the header component size from 1 to 6.

4.1.5 Tool bar

Figure 4.7 Toolbar

The toolbar provides basic functions for user. It provides undo/redo, preview, save, and download buttons, it also provides user to get help. For more detailed on each function, see *Appendix F*.

4.1.6 Template

Template is a pre-designed web page that can be directly used by designers. Designers can use the template by clicking the specified template on the left-hand side panel into the editor. Then all the components on that template will be placed to corresponding position on the editor. After implementing the template, the designer can also modify the editor such that he can delete the components which he does not want, or he can change the style of each component, even more he can add new components based on the template.

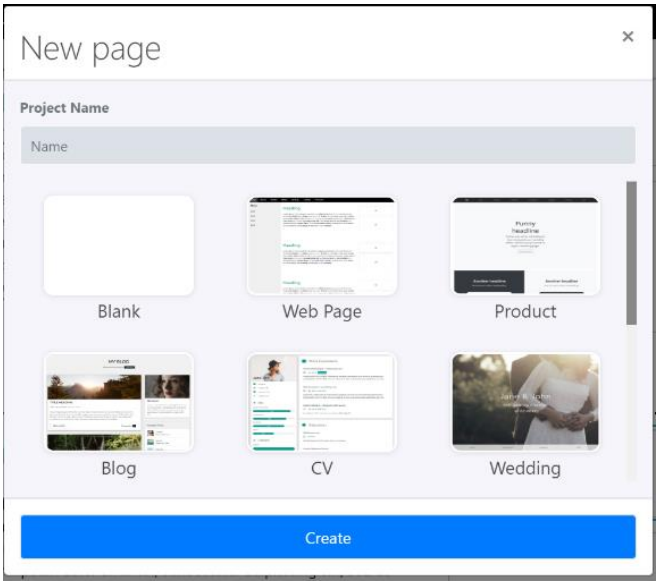


Figure 4.8 Template

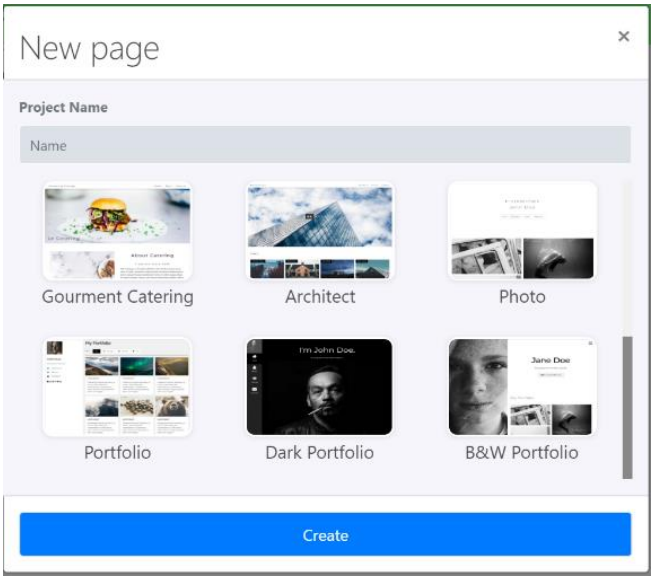


Figure 4.9 Template cont.

4.1.3 My page

My page is a collection of personal web pages of the users. All personally designed web pages will be stored in this place. If the users click their Project at the right top, they can intuitively see their designed web pages. Also, users have the most permission to operate the settings of their web pages. For example, rename, publishing, deleting, adding, and modifying for the selected web page.

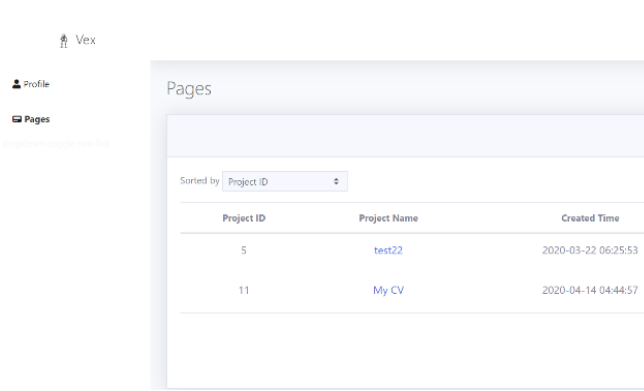


Figure 4.10 Project List Page

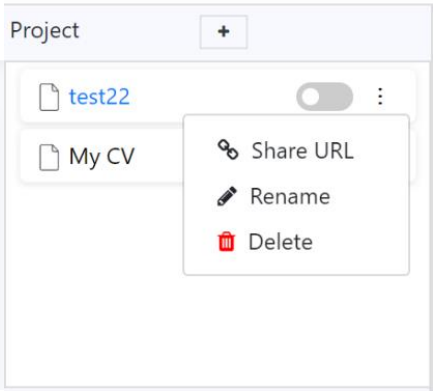


Figure 4.11 Project Modification

4.1.7 Administrator page

When a user logs in with an administrator account, the administrator can go to the administrator's page (Right Top click Web Manage). The administrator has the highest authority of the system, and they can view all the projects stored in the database and the basic information (encrypted passwords) of all users in the administrator page. At the same time, the page provides a search function. Administrators can provide project_id, user_id, email, or type for precise or range search. The above function is used on:

- User issue tracking and resolution
- Illegal web page blocking function
- Illegal user blocking function
- Upgrade a user to administrator user

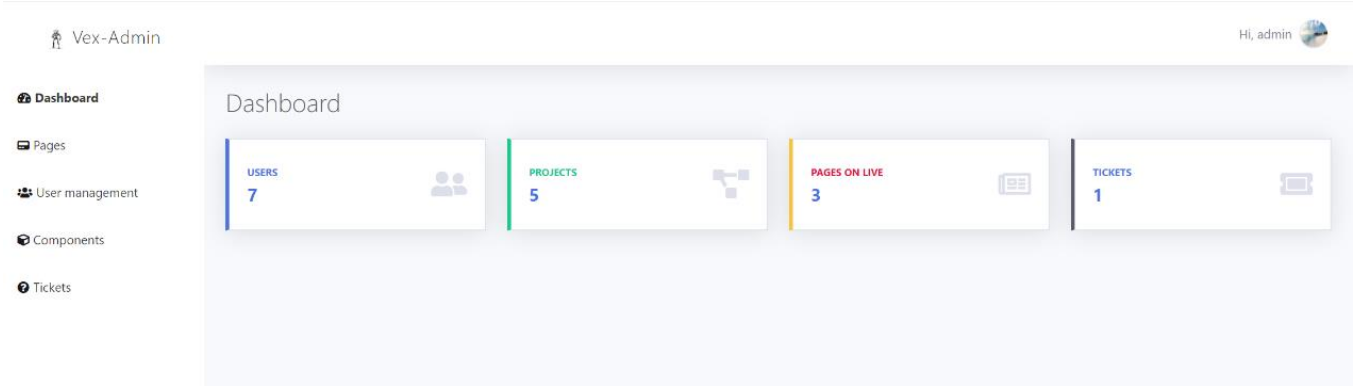


Figure 4.12 Administrator's Dashboard

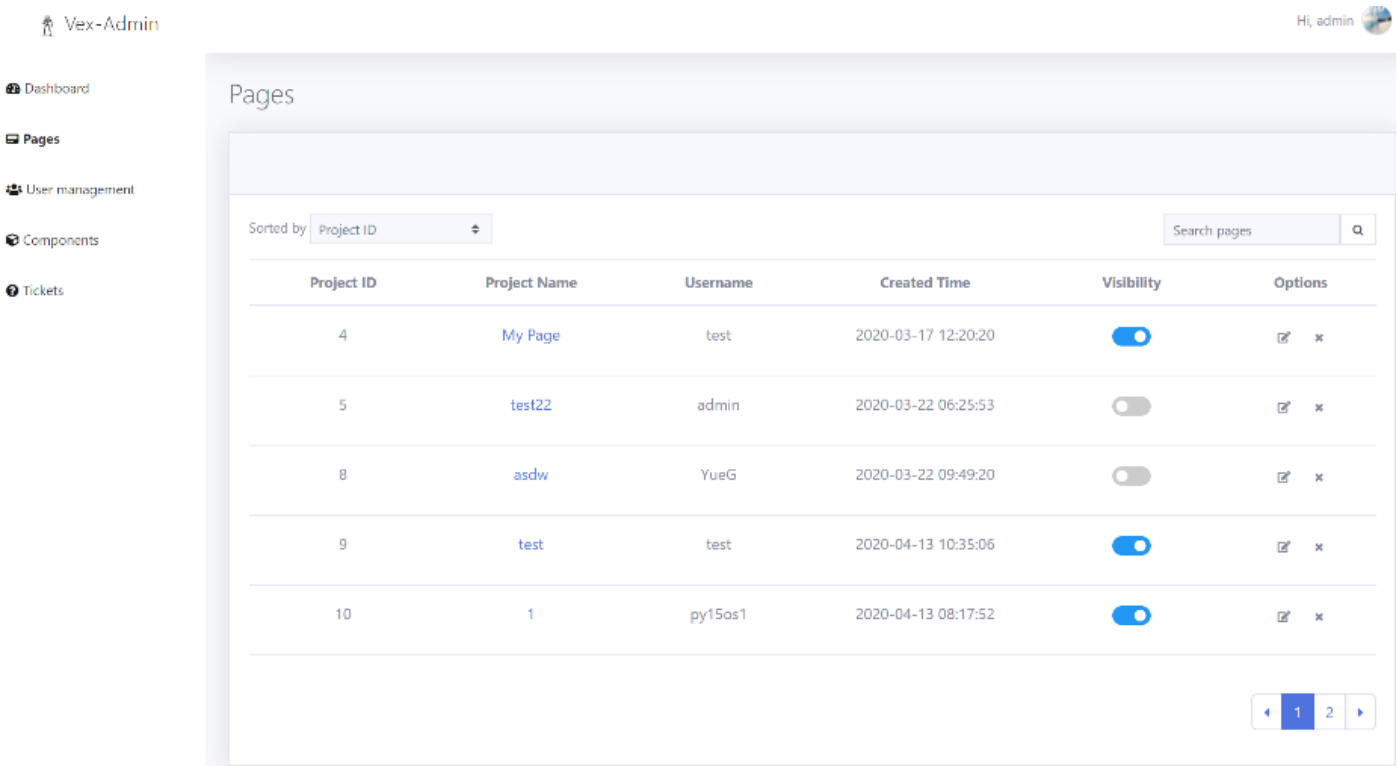


Figure 4.13 Administrator's Project Management Page

4.1.8 Components editor page

(All operations will modify the Data of the Components table in the database.)
This is a page that allows administrators or programmers to quickly add, delete, and modify components. The administrator can directly modify or add functions in the editor page without logging in to the background.

When the administrator clicks the components edit button from the administrator page, the webpage will jump to the Components editor page with the administrator identification. In this page, we have four functions: added, deleted, modified, and search.

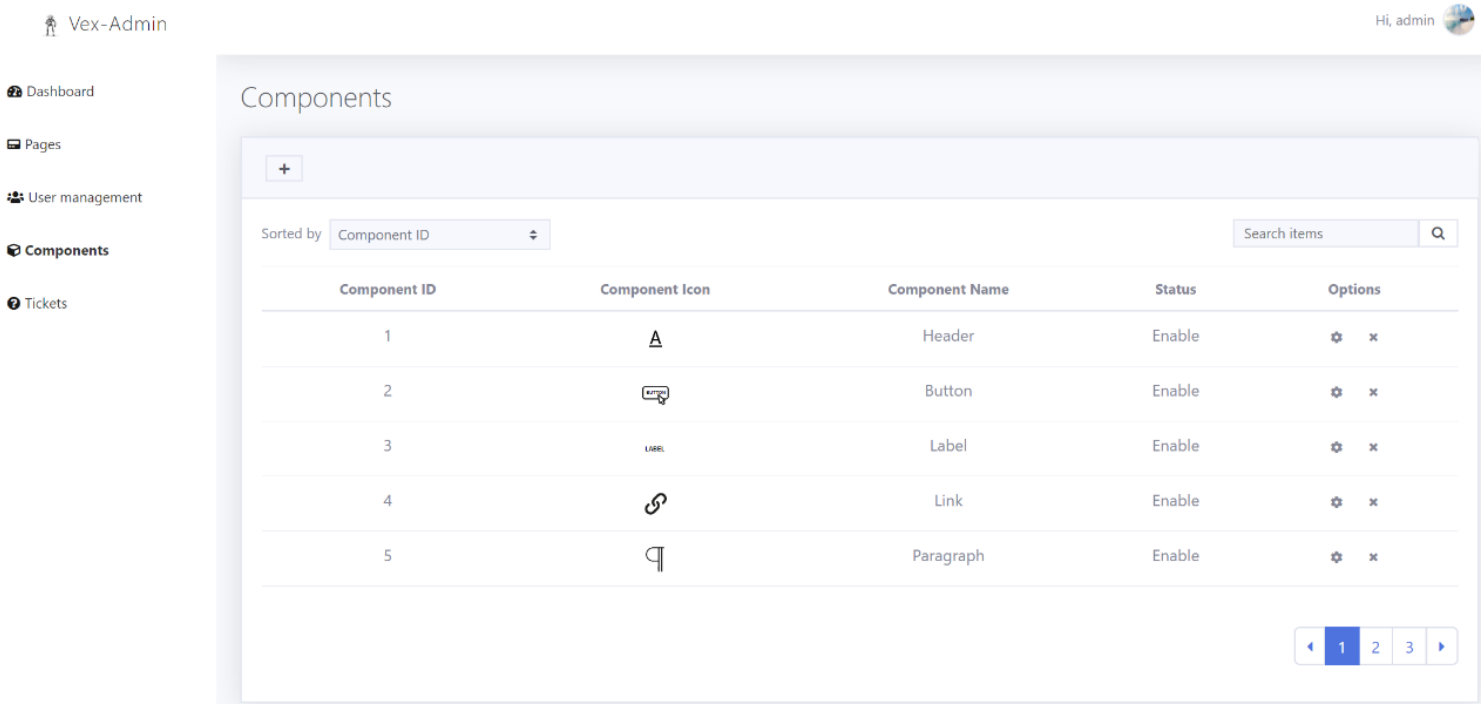


Figure 4.14 Administrator's Components Management Page

- Add: After the administrator clicks the Add button, he needs to provide the corresponding code and photo, and the system will automatically add this function to the editor page.

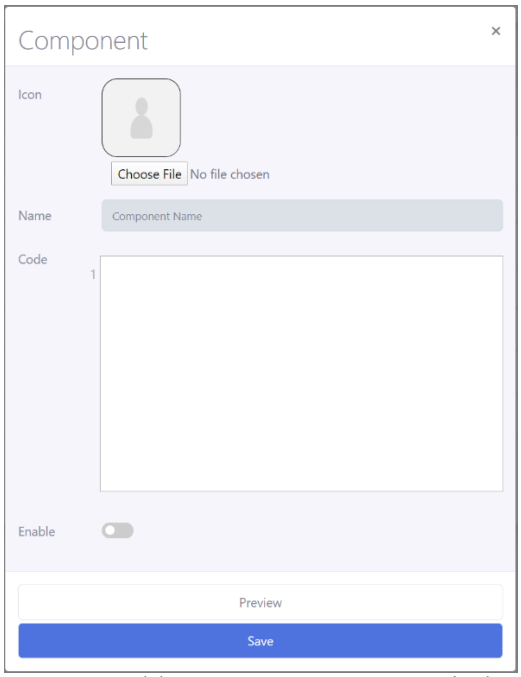


Figure 4.15 Add Component Popup Window

- Delete: After the administrator clicks the Delete button (Options then click “Cross”), he needs to choose which components needed to delete, and the system will automatically delete this function from the editor page.

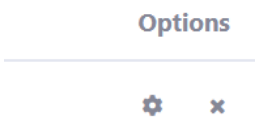


Figure 4.16 Component Option Buttons (Modify & Delete)

- Modify: After the administrator clicks the Modify button, he needs to choose which components needed to Modify (Options, then click “manage”). When the administrator modified and confirmed the page, the system will automatically modify this function from the editor page.

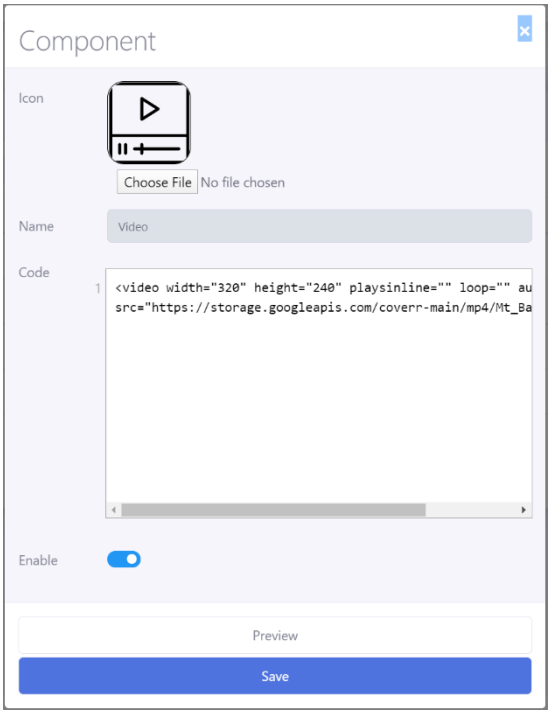


Figure 4.17 Modify Component Popup Window

- Search: This feature can help administrators more easily find specific components by keywords. After selecting the component, the administrator can delete or modify it.

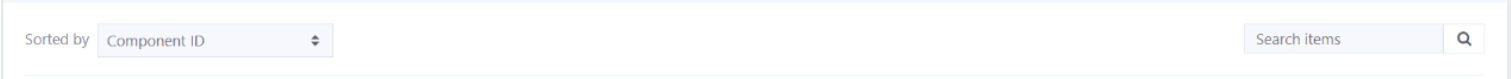


Figure 4.18 Search Component Panel

4.1.9 Tickets page

If the users use the bug report function, the administrator can make communication from the Tickets option of the administrator page. The administrator can manage and search the user's question submission. Then, they can answer the specified questions, and change the status of the questions (solved, un-solve).

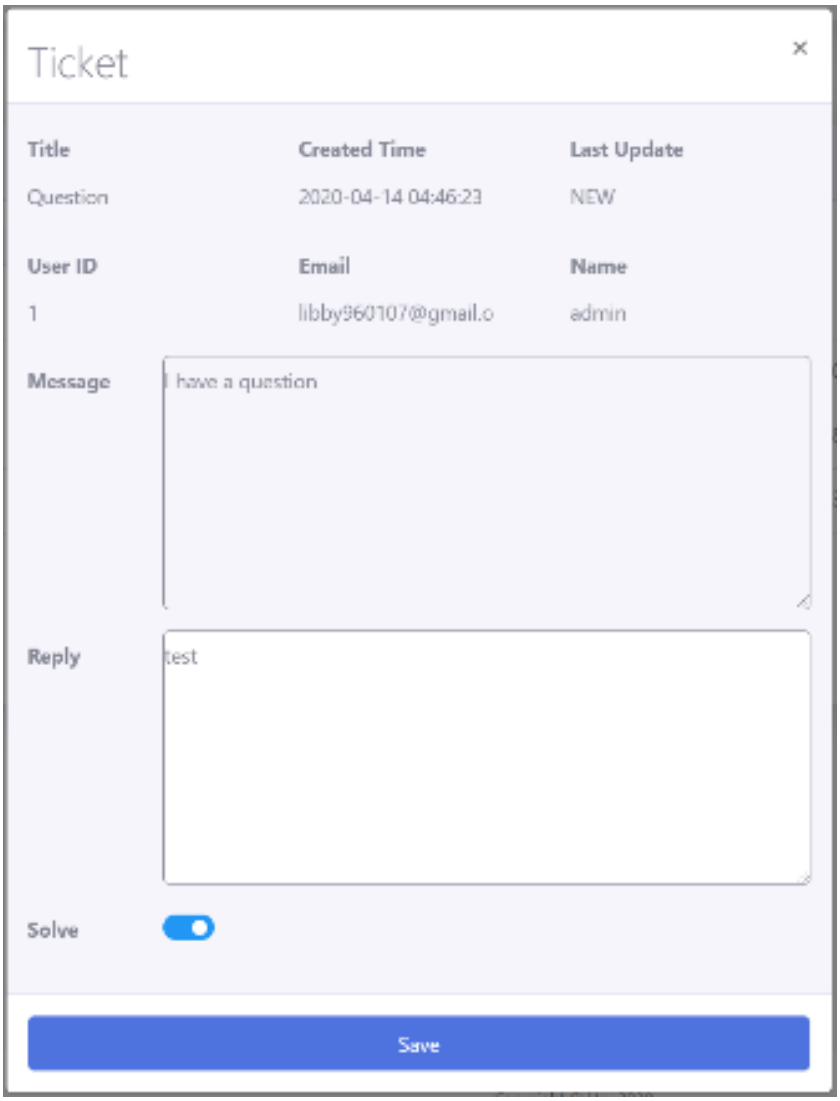


Figure 4.19 Reply Popup Window

4.2 Hardware Interfaces

No extra hardware interfaces are needed. The system will use the standard hardware and data communications resources. This includes, but is not limited to, the general Ethernet network/T1 connection at the server/hosting site, network servers, and network management tools.

4.3 Communication Interfaces

4.3.1 Website and Database
When logging in, storing designed web pages, and publish / live web pages, there is a communication interface between the website and its DB, the algorithm in the website and its DB storage.

4.3.2 User Log in
The web browser will have communication with the database. When the users log in to the system from the web browser, then the web browser will send a request from the web browser to the system.

4.3.3 Website Storage
When storing designed web pages, the user's web code will be encoded and stored in the user's personal database.

4.3.4 Publish/Live
Finally, when user publish / live their website, the webpage code is organized and stored in the system database and it's open to download / view permissions for all users.

4.4 Software Interfaces

We have created our own website to provide services. Users can edit the web page on the website and download the code. Also, the administrator can block the users or designated webpage. And they can search the bug's webpages by the user report and solve system bugs and optimize the system.

For users who are not logged in, they can still experience the simple features of the site in a non-login state. However, download and publish functions are only available after login

When the administrator is logged in, the user can also edit the editor of the web page. For example, administrators can upload new components without background login

4.5 Database

The database we use which is PostgreSQL and being provided by Brock server. This system which contains four tables which are **vex_user, vex_product, vex_component** and **vex_ticket**.

4.5.1 Table: vex_user
There are two types of users: users who registered by themselves and administrators, each has different permissions. **user_id** is used to define different users. It is a primary key and is unique. **username** is the account of the user and it is unique.

user_id, **username**, **password**, **create_time**, **email**, **type** and **is_enable** are required when creating a user, so they cannot be empty.

4.5.2 Table vex_product
The web pages created by the user will be stored in this table. If the product is not published, access to this table is restricted to the user and the administrator. Also, the administrator can find and modify the target page based on **product_id** or **user_id**.

product_id is the primary key of the table, and **user_id** is a foreign key, which connects the **user_id** of **vex_user**. And, **product_id**, **user_id**, **product_name**, **create_time**, and **is_delete** cannot be empty.

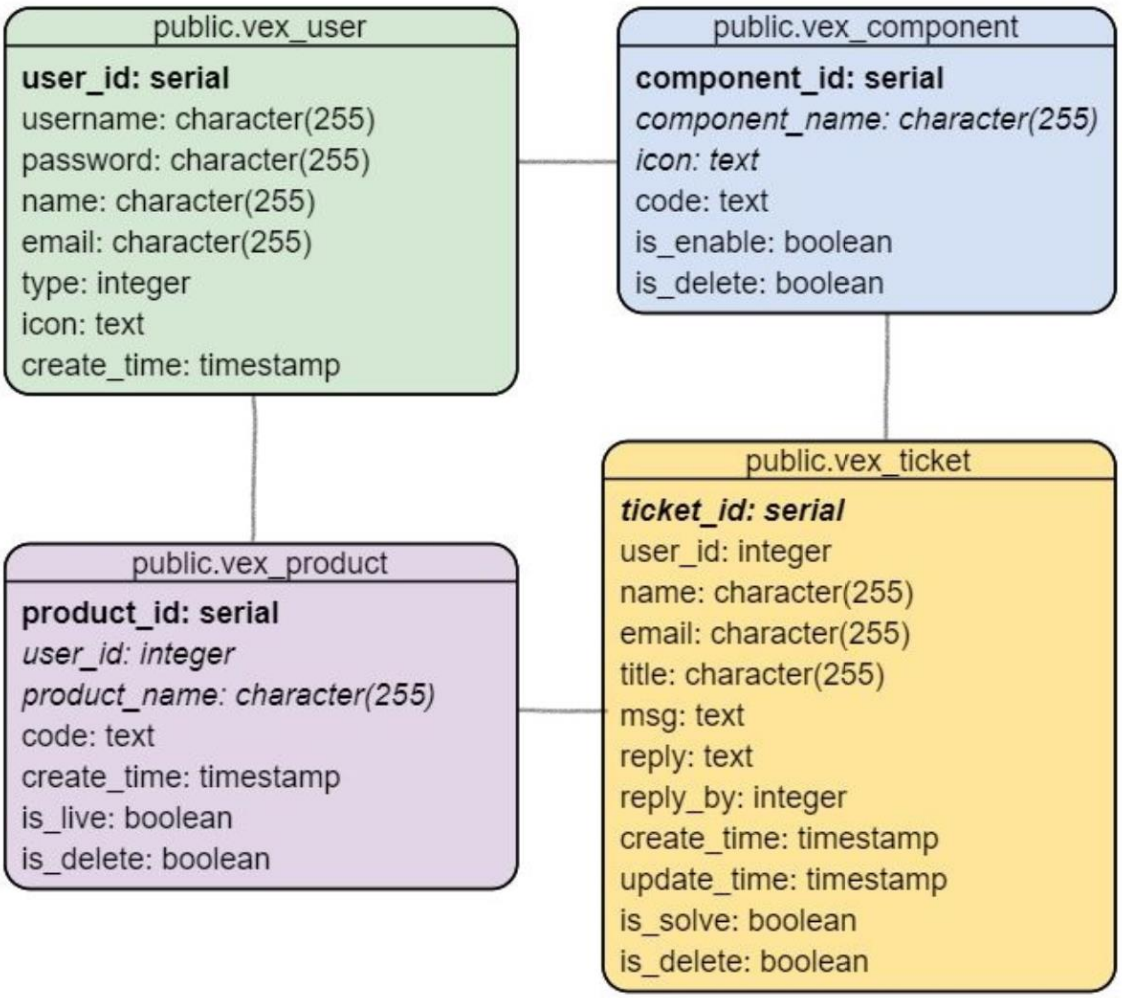
4.5.3 Table Component

The Component table stores all the functions of the web editor, such as the drag and drop function, the button's implementation function, and the browsing's view function.

compoent_id is the primary key for this table. And all entries in the table are not allowed empty.

4.5.4 Table vex_ticket

The table of **vex_ticket** which has 12 fields, and **ticket_id**, **title**, **name**, **email, msg** and **create_time** are required fields. Users will enter the contact information, which is title, name, email, and msg. after system manager, it will fill the **reply**, **reply_by**, **update_time** and **is_solve**



SQL Schema:

```
CREATE TABLE vex user
(
    user id serial NOT NULL PRIMARY KEY,
    username character(255) NOT NULL UNIQUE,
    password character(255) NOT NULL,
    name character(255),
    email character(255),
    type integer DEFAULT 1,
    icon text,
    create time timestamp,
    is enable boolean DEFAULT true
) ;

CREATE TABLE vex_component
(
    component_id serial NOT NULL PRIMARY KEY,
    component_name character(255),
    icon text,
    code text,
    is_enable boolean DEFAULT true,
    is_delete boolean DEFAULT false
);

CREATE TABLE vex_product
(
    product_id serial NOT NULL PRIMARY KEY,
    user_id integer NOT NULL,
    product_name character(255),
    code text,
    create_time timestamp,
    is_live boolean DEFAULT false,
    is_delete boolean DEFAULT false
);

CREATE TABLE vex_ticket
(
    ticket_id serial NOT NULL PRIMARY KEY,
    user_id integer,
    name character(255),
    email character(255),
    title character(255),
    msg text,
    reply text,
    reply_by integer,
    create_time timestamp,
    update_time timestamp,
    is_solve boolean DEFAULT false,
    is_delete boolean DEFAULT false
);
```

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- **PREQ-1:** Time needed to open the web editor should be less than 10 seconds.
  It should not take too much time to open this editor. Less waiting time leads to happier users.
- **PREQ-2:** The time for users to download their work using the website should not exceed 1 minute.
  The designed web page downloaded by the user should be a .zip file or .rar file. So it is reasonable to assume that if the download process takes longer than 1 minute, there must be some undesirable thing happening, such as a poor connection.
- **PREQ-3:** The publishing time for users of the system server should not exceed 5 minutes. Because we have restrictions on the maximum size of a user could send to the system server.
- **PREQ-4:** The processing time for the publishing web page on the system server should not exceed one business day.

5.2 Security Requirements

Although we make assumptions that the overall security level is taken care of by the system administrator on the back end, some security requirements on the system contribute positively to privacy and integrity.

- **SREQ-1:** Whenever a user fails to authenticate themselves five times consecutively within 10 minutes, their account is locked for fifteen minutes. And they will be notified about the status of their account.
- **SREQ-2:** The user password in the database should be encrypted. When the user is publishing a website and other users are not allow direct access to the information by URL, they need the link.
- **SREQ-3:** The process of encryption and decryption should not corrupt any information.
- **SREQ-4:** Some functions of the system should be available and only available for people who have the correct credentials. People having the correct credentials then who can add/delete/modify the designed web page.

5.3 Software Quality Attributes

This system is a graphical interactive interface, which can give users a good operating experience and simple and easy to understand operating methods.

- **NREQ-1** The time for the editor service to recover from a system failure should not exceed 1.5 hours.
- **NREQ-2** The algorithm should be designed to be reusable. In the future when developing similar software, it should be easily augmented.
- **NREQ-3** The system should have the ability to add additional functions for the edit web page. It should have the ability to design more templet in the future. Changing software interfaces and result layout should be flexible.
- **NREQ-4** Maintainability of this software is expected to be high. The editor interface will able to be updated when new features were added in the future.

Appendix A: Glossary

1. CMS - Content management system, is a software application that can be used to manage the creation and modification of digital content.

2. Captcha - Completely Automated Public Turing test to tell Computers and Humans Apart is a type of challenge–response test used in computing to determine whether or not the user is human.

3. SHA - Secure Hash Algorithm is a set of cryptographic hash functions.

4. Cloud Service – A cloud server is a virtual server running in a cloud computing environment. It is built, hosted and delivered via a cloud computing platform via the internet, and can be accessed remotely.

5. DIV - The div tag is known as Division tag. The div tag is used in HTML to make divisions of content in the web page.

6. Container - Container is a DIV tag in HTML which divide a web page into several boxes.

7. HTML - Hypertext Markup Language (HTML) is the standard markup language for documents designed to be displayed in a web browser.

8. Publish - Publishing your website, then others can see your website project by input URL (uniform resource locator).

9. Database - A database is a collection of information that is organized so that it can be easily accessed, managed and updated.

10. Export - It allows users to export webpage code to a specified text file, or desktop

Appendix B: Analysis Models

The following use case diagrams represents the entire system. Fig. B-1 which represent user interact with the system, and Fig.B-2 and B-3 represents the system manager interact with the web management system.

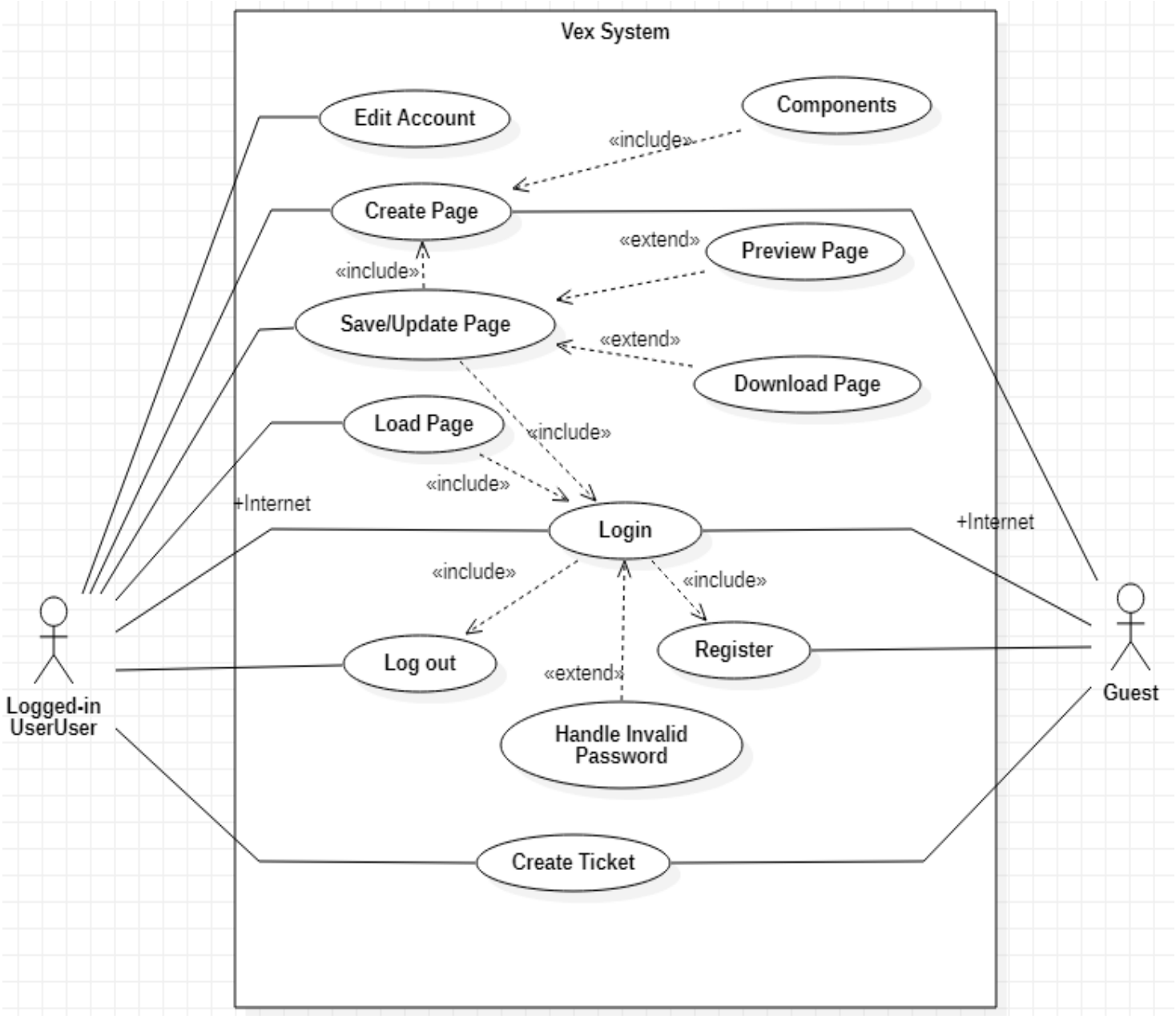


Figure B-1 User Interface Use Case Diagram

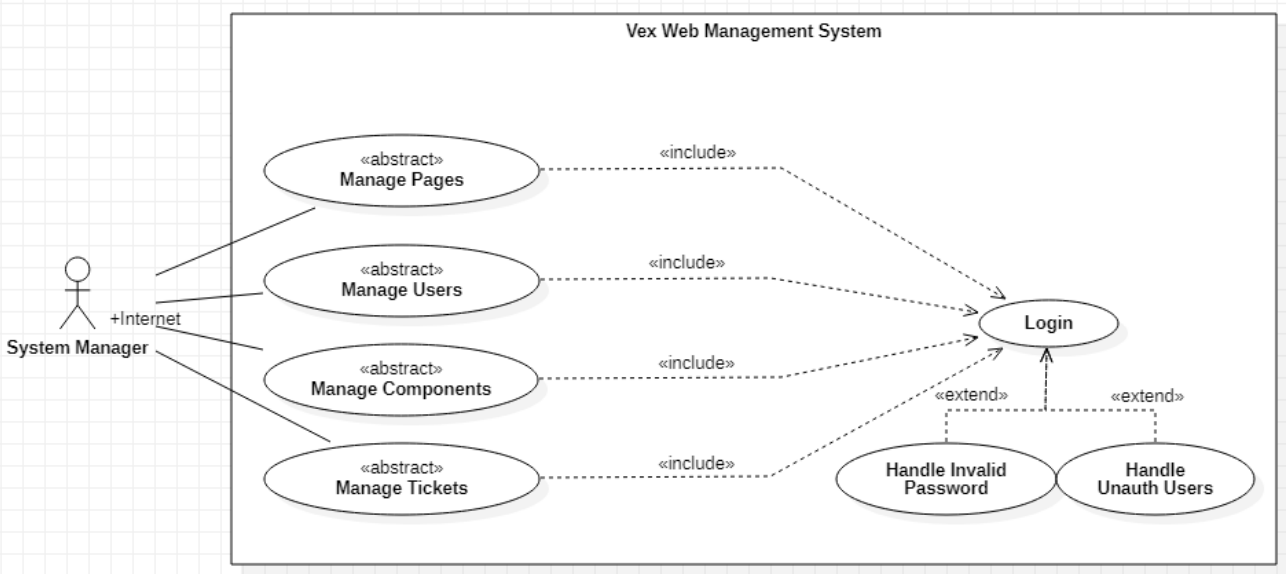


Figure B-2 Web Management Use Case Diagram

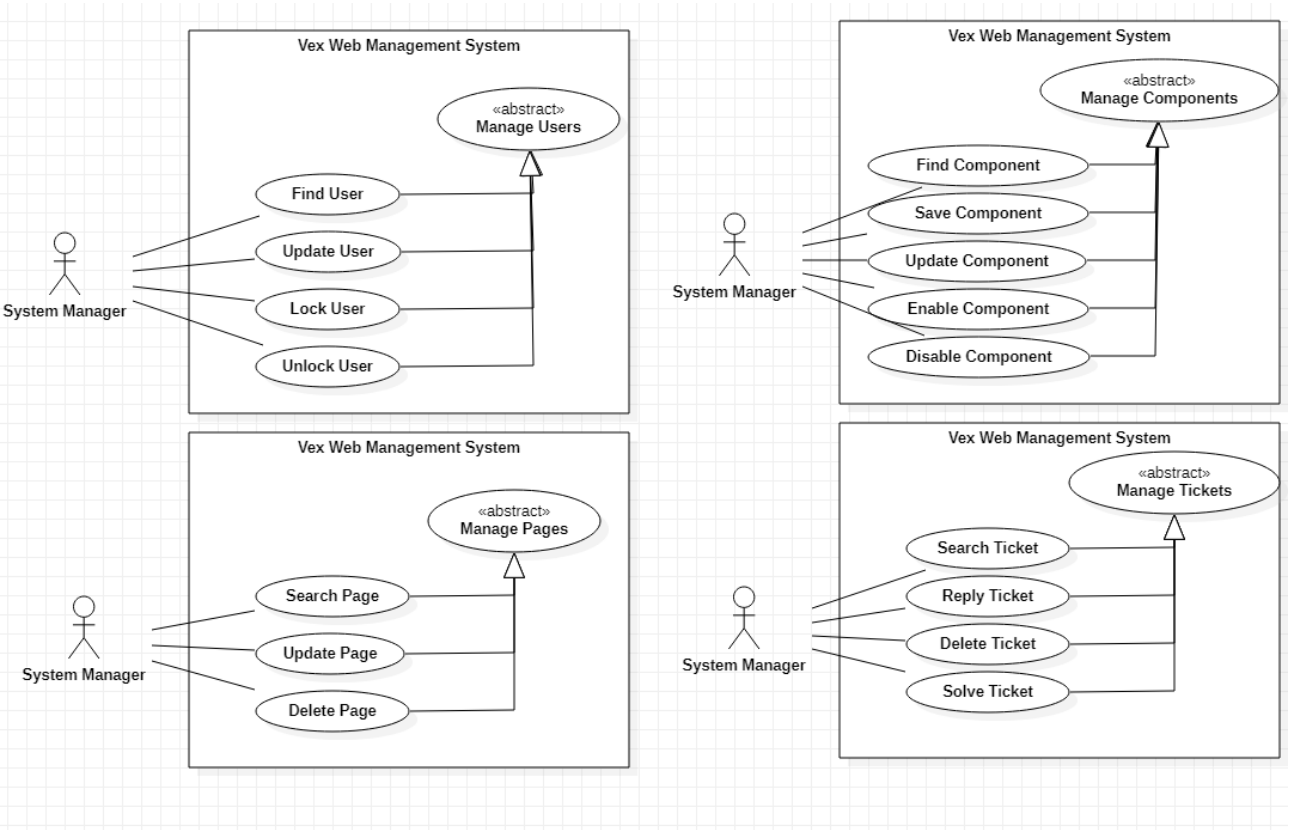


Figure B-3 Users, Components, Pages, and Tickets for Web Management

Appendix C: Code Conversion Table

This table defines the system code, to reference the system process status.

CODE	Code Conversion
200	Normal/Success
300	Login Required
301	User does not exist
302	Incorrect Password
303	Blocked User
304	Access Denied
305	Invalid sorting field
306	Component does not exist
307	Ticket does not exist
401	List user failed
402	Update user profile failed
403	Update user icon failed
404	Update user password failed
405	Disable user failed
406	Delete component failed
407	List component failed
408	Update component failed
409	Save component failed
410	Update live failed
411	Update page name failed
412	Save page failed
413	Update page failed
414	Delete page failed
415	List page failed
416	Save ticket failed
417	Delete ticket failed
418	Update ticket failed
419	List ticket failed
420	List live page failed
500	Database Error

Appendix D: Default Components

- Header

Figure D-1 Header Element

Header is a HTML tag works as a container for introductory content. It typically has six different sizes of header represents as 1 to 6 from large to small respectively.

- **Button**

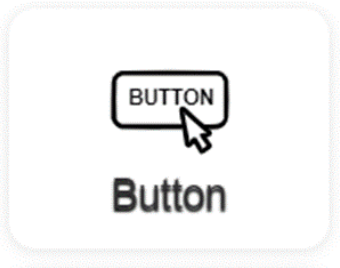


Figure D-2 Button Element

Button is a Button tag in HTML which allows users to click and design a trigger event. For example, users can design a login button by dragging a button tool into the editor and design a click event that can jump to another page for users to log in.

- **Label**

Figure D-3 Label Element

Label is a HTML tag that can defines a label for several elements such as color, URL, etc.

- **Link**

Figure 4 Link Tool

Link allows users to add a Hyperlink in a web image. Also, users can use Link to make a Hyperlink as a text in the website body.

- **Paragraph**

Figure D-5 Paragraph Element

This is a paragraph tag in HTML. It allows the designer to type a paragraph on their web page with no character limitation.

- **Image**

Figure D-6 Image Element

Image allows users to upload their local image to the system and place it to the corresponding position. User can resize their image through style panel on the right-hand side.

- **Text Input**

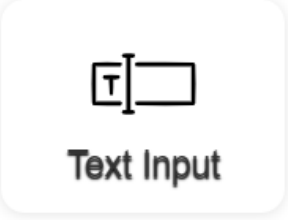


Figure D-7 Text Element

Text input bar allows users to add an input field in the designed web page so that website

users can type briefly text in the field and the designer can collect text inside the specified text field for other functions.

- **Select input bar**

Figure D-9 Select Element

Select input bar is a tool that allows user to create a drop-down list which contains several selections can be chosen. The drop-down list usually only shows the selection which is chosen. However, once website's users click on the list, it will expand and show every selection that it contains.

- **Text Area**

Figure D-10 Text Area Element

This is a sort of Text input field which is very similar to Text Input Bar. However, the text input bar only allows users to type short-length text. Text Area provides more limitation on the length of characters so that when the designer wants web users to type more text, he can drag this tool instead of the text input bar.

- **Checkbox**

Figure D-11 Checkbox Element

Checkbox is a tool that allows user to create a checkbox which contains a button and users can click to check. Eventually, the checkbox will submit the event whether the user

checked to the webserver.

- **Radio Button**

Figure D-12 Radio Button

Radio button works as a select box that allows users to choose their selection. The difference between radio button and checkbox is that users can choose more than one selection on checkbox, but they can only choose one selection on radio button.

- **Video option**

Figure D-13 Video Element

A video option is a tool that allows the user to create a video on the designed website. Also, users are expected to input the video URL and choose the size of the video, whether to mute, whether to loop, whether to auto-play, and whether to be controlled.

- **Horizontal Rule**

Figure D-14 Horizontal Element

Horizontal is a tool that allows users to create a horizontal line in the body of the designed website.

Appendix E: Attributes Panel

- **Display**

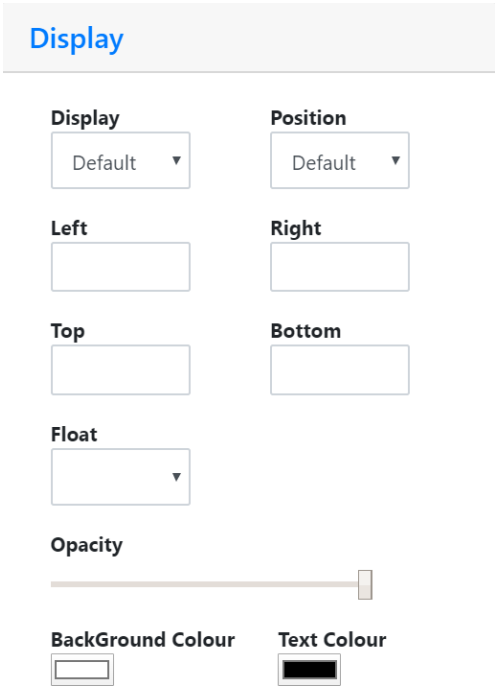


Figure E-1 Display Attribute

Display is a part of style. It allows the users to change the position of components. Such as Left, Right, Top, and Bottom of the component. Transparency is also available to users here.

- **Typography**

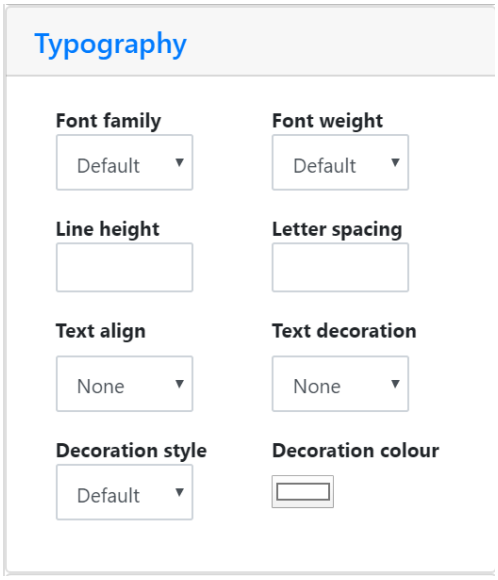


Figure E-2 Typography Attribute

Typography is a part of style. It allows the users to change the font format. For example, font family, font-weight, line height, letter spacing, text-align, text-decoration, decoration style, and decoration color.

- **Size**

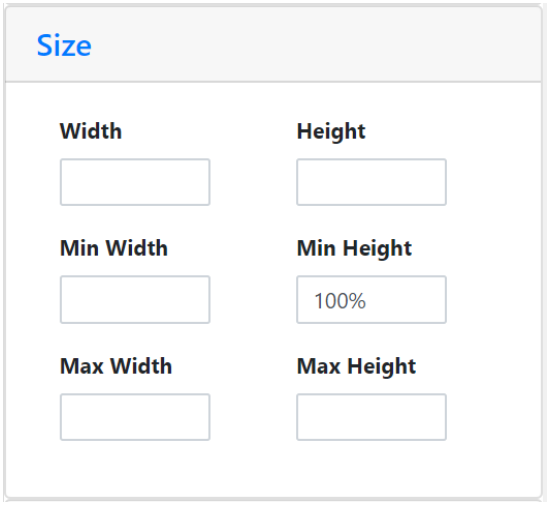


Figure E-3 Size Attribute

Size is a part of style. It allows the users to change the photo format. For example, photo width, height, min width, min height, max width, and max height.

- **Margin**

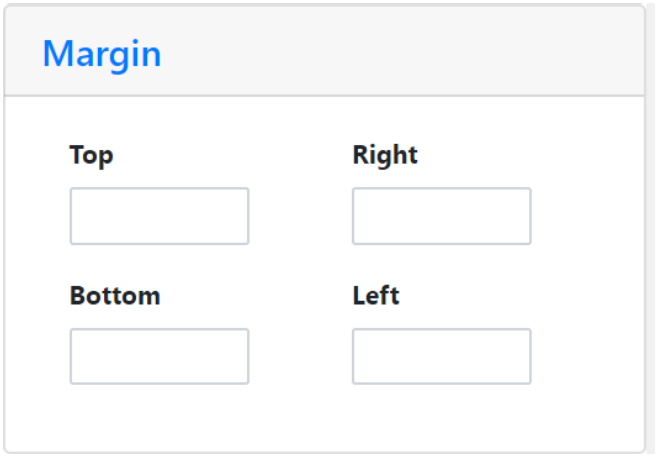


Figure E-4 Margin Attribute

Size is a part of style. It allows the users to modify the component's margin. Four parameters are expected to be input by the user, Margin Top, Margin Right, Margin Bottom, Margin Left.

- **Padding**

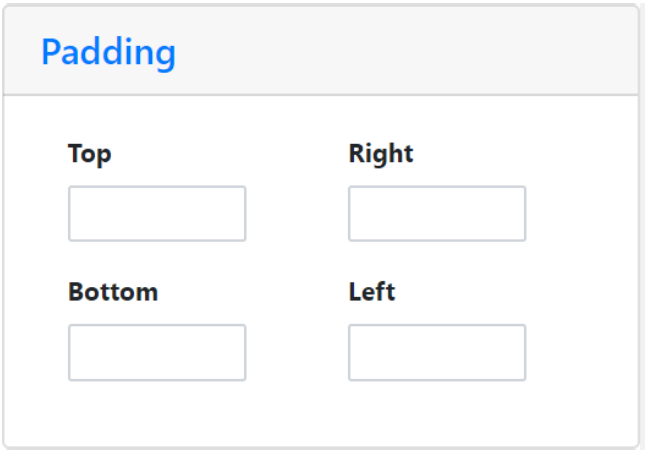


Figure E-5 Padding Attribute

Size is a part of style. It allows the users to adjust the distance between the internal element and itself. Four parameters are expected to be input by the user, Padding Top, Padding Right, Padding Bottom, Padding Left.

- **Border**

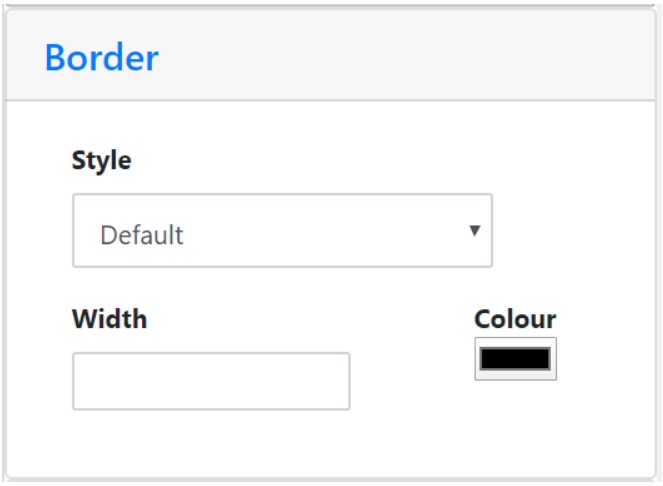


Figure E-6 Border Attribute

Border is a part of style. It allows the users to choose the border type of components. There are several types can be chosen such as solid, dotted, and dashed. Also, color and width can change here.

- **Border radius**

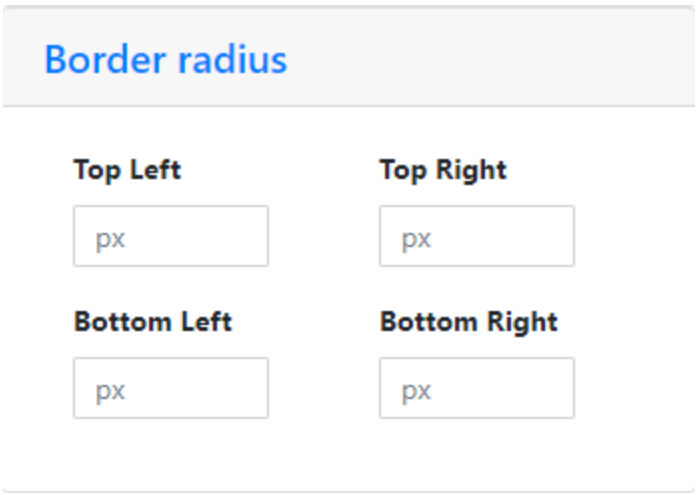


Figure E-7 Border Radius Attribute

Border radius is a part of style. It allows the user to choose the border-radius of components. Several radii can be chosen such as Top Left, Top Right, Bottom Left, and Bottom Right.

- **Background Image**

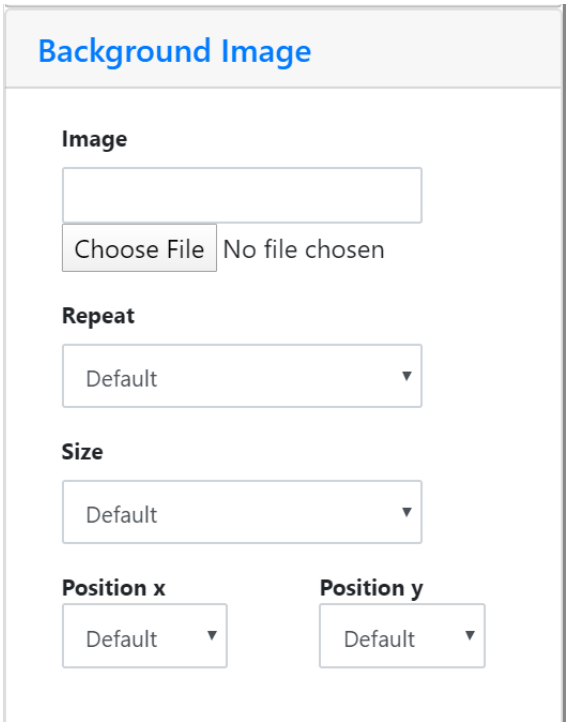


Figure E-8 Background Image Attribute

Background Image is a part of style. It allows the user to choose the background image of the website. Users are expected to choose an image file, and set the size, position, and the format of the image.

Appendix F: Toolbar Panel

- **Undo/redo**

Figure F-1 Undo and Redo Button

To optimize user experience, Undo / Redo function is available to the users. When the user fails to add / remove / modify their design page, the Undo button can help the designer quickly withdraw the previous step to save time. When the user regrets the withdrawal, the Redo button can quickly return to the previous state. The Redo function also allows users to quickly compare the different effects of this step.

- **Download**

Figure F-2 Download Code Button

This function is only available to those users who are already registered and logged in. Download button allows the users to export webpage HTML code to a specified .html file, or desktop (automatically creates a .html file). This code is a static webpage HTML code, which means website designers can paste directly to use elsewhere. Also, when first download the web page, the system will automatically check the project name. If the project no name, same name as before saved page, or the page name has illegal characters, the project will not be allowed to download.

- **Preview**

Figure F-3 Preview Button

Preview button allows designers to preview the webpage that they are working on. When click the button, the current webpage that in the main content panel will be opened on a new window and the designer can interact with it as a real webpage.

- **Save**

Figure F-4 Save Button

This function is only available to those users who are registered and logged in. It allows the designers to save their project into the server so that once they log in with the same account, the saved projects can be loaded to the user's side.

- **Tutorial**

Figure F-5 Tutorial Button

The user who logged in first time will be automatically loaded a tutorial that helps the user to learn how to use our website. This button provides a review of the tutorial progression. If the user forgets how to use the website, then he can go through our tutorial again by clicking this button. If the user wants to skip the tutorial, there is also an option to close it when tutorial is ongoing.

- **Support**

Figure F-6 Support Button

This button allows the users to submit a question or report a bug that find on this website. If users find frustration of using the website or some technique issues takes place, they can use this button to submit a ticket to the administrator and administrator can try to help the users to solve their problems.

Appendix G: Hazard Tree

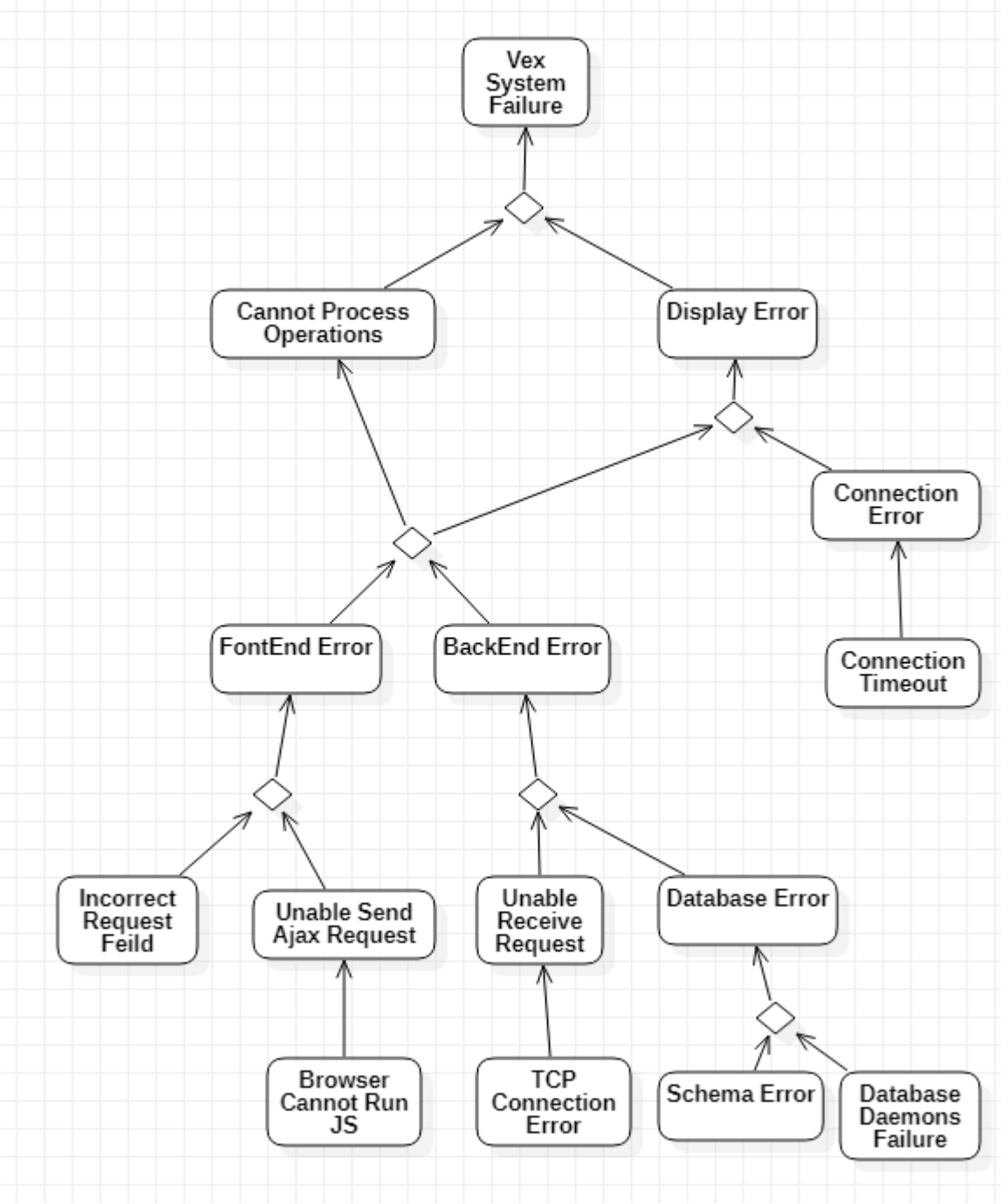


Figure G Hazard Tree