



# Maggotty High Alumni Association – US northern Region Website

**Documentation**

Version 2.0 • December 20, 2021

The Maggotty High Alumni Association - US North Region website will allow The Maggotty High Alumni to join in the association and make contributions to the high school.

*The Mondays*

*1 Normal Ave*

*Montclair*

*07043*

*United States*

Last edited: 20 December 2021

Copyright © The Mondays. All rights reserved.

No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without prior written permission from The Mondays.

All copyright, confidential information, patents, design rights, and all other intellectual property rights of whatsoever nature contained herein are and shall remain the sole and exclusive property of The Mondays. The information furnished herein is believed to be accurate and reliable.

However, no responsibility is assumed by The Mondays for its use or any infringements of patents or other rights of third parties resulting from its use.

The “The Mondays” name and “The Mondays” logo are trademarks or registered trademarks of “The Mondays.”

All other trademarks are the property of their respective owners.

# System Design Document

## The Mondays

# Alumni Dynamic Website

Version: 2.0    Revision Date December 20, 2021

<b>Professor</b>	
[Name]: Dr. H. Johnson	[Email]: johnsonh@montclair.edu

<b>Project Manager/Software Architect</b>	
[Name]: Davaughn Hoots	[Email]: hootsd1@montclair.edu

<b>QA Lead/Software Architect/Tester</b>	
[Name]: Nesreen Salah	[Email]: salahn1@mail.montclair.edu

<b>Lead Programmer/UX &amp; UI Designer</b>	
[Name]: Rafael Hidalgo	[Email]: hidalgor2@montclair.edu

<b>Lead UX &amp; UI Designer/Full Stack Developer</b>	
[Name]: Laura Paulino	[Email]: adamespaul1@mail.montclair.edu

# Table of Contents

<b>Executive Summary</b>	<b>10</b>
Purpose of this document	11
Identification	12
Scope	13
Relationship to Other Plans	14
Methodology, Tools, and Techniques	14
Methodologies	14
The Unified Process (UP)	14
Tools	14
Documentation Tools	14
Planning Management Tools	14
Design Tools	15
Front End Development Tools	16
Back End Development Tools	17
Database Tools	17
IDEs	18
Techniques	18
End to End Testing:	18
BlackBox Testing:	18
WhiteBox Testing:	18
Object-oriented analysis and design (OOAD):	18
Representativeness heuristic:	18
Policies, Directives, and Procedures	19
Policies	19
Directives	19
Procedures	19
<b>Design Overview</b>	<b>20</b>
Background Information	20
Constraints	21

---

Design Goals	22
Environment Usage	22
Interface Compatibility	22
Operating System Compatibility:	22
Device Compatibility:	22
Availability	22
Efficiency	23
Security	23
Maintenance	23
Size performance and goals	24
User Characteristics	25
User Objectives	25
Users without Accounts	25
Alumni	25
Assistant Treasurer	26
Treasurer	26
Vice-President	27
President	28
Admin	29
<b>System Architecture</b>	<b>30</b>
Hardware Architecture	30
Webserver	30
Processor	30
RAM	31
SSD	31
Power Supply	31
Networking Devices	32
Modem	32
Router	32
Firewall	32
Software Architecture	33
Content Authoring	33

Mobile Support	33
Media and Files	33
User interactivity	34
Payment and Shopping Cart Functionality	34
Account management	35
Registration	35
Communications Architecture	36
User-System Communication	36
Hardware Communication	42
<b>Data Design</b>	<b>44</b>
Database Management System Files	44
<b>Detailed Design</b>	<b>46</b>
Hardware Detailed Design	46
Web Server Specifics	46
Networking Specifics	47
Software Detailed Design	48
Accounts Page	48
Calendar Page	57
Files and Media Page	67
Virtual Meetings Page	77
News Page	86
Volunteer Page	99
Donations Page	109
Shop Page	117
<b>Human-Machine Interface</b>	<b>142</b>
Interface Design Rules	142
Registration	142
Login	143
Alumni Profile Page	144
Donations	145
Volunteering	146
Products and Shopping	146

---

Contact Us	149
News	149
<b>Testing</b>	<b>150</b>
Overview	150
Objective	150
Scope	150
Test Approach	151
Testing Items	151
Successful Criteria	152
Testing Methodologies	153
Module Test Cases	156
Login Registration	156
Viewing Featured Pages	159
Purchasing	161
Cart Page	163
Create / Edit / Delete / Events	164
Event Registration	166
Profile Customization	168
Delete / Edit/ Create Volunteer Opportunity	170
Delete / Edit User Profile	173
Payment Page	175
Testing Conclusion	178
<b>Appendix A</b>	<b>179</b>
User Manual	179
Installation of necessary software	179
Getting the Website to Run	181
Website Navigation Instructions	184
Homepage:	184
Registration:	185
User Account:	186
Profile:	187
Donate:	188

Shop:	189
Cart:	190
Navigation Menu:	191
Alumni News:	192
History:	193
Events:	194
Admin:	195
AdminPage:	198
Data Dictionary	199
Overview	199
Field Definitions	199
Alumni	200
Cart	202
Events	203
Orders	204
News	204
Product	206
Volunteer	207
Glossary of Terms	209

# Document History

Paper copies are valid only on the day they are printed. Contact the author if you are in any doubt about the accuracy of this document.

---

## Revision History

**Table 0.1** A record of the Revisions made to the project with each contributor.

Revision Number	Revision Date	Summary of Changes	Author
1	October 11th, 2021	Added Requirements	Laura Paulino
2	October 16th, 2021	Added Specifications	Davaughn Hoots
3	October 18th, 2021	Edited Specifications to meet Professors suggestions	Rafael Hidalgo
4	October 20th, 2021	Added User Details Table	Rafael Hidalgo
5	November 1st, 2021	Revised Specifications to add clarity	Rafael Hidalgo
6	November 6th, 2021	Added first cases of use case diagrams	Davaughn Hoots, Laura Paulino, Rafael Hidalgo, Nesreen Salah
7	November 13th, 2021	Revised Use Case Diagrams	Davaughn Hoots, Laura Paulino, Rafael Hidalgo, Nesreen Salah
8	November 30th, 2021	Revised Use Case Diagrams	Davaughn Hoots
9	December 5th, 2021	Added Activity Diagrams	Davaughn Hoots
10	December 15th, 2021	Added Technical Details and Summaries to Use Cases.	Nesreen Salah
11	December 18th, 2021	Added User Manual	Rafael Hidalgo/Laura Paulino
12	December 20th, 2021	Ran Testing	Davaughn Hoots/Nesreen Salah

---

## Reference Documents

**Table 0.2** The following documents are used as a base of information for the contents of this project document.

Document Name	Version	Author
<i>Software Engineering: Theory and Practice</i>	4th	<i>Shari Lawrence Pfleeger</i>
"Top-Down Programming in Large Systems," in <i>Debugging Techniques in Large Systems</i>		R. Ruskin, ed., Englewood Cliffs
<i>IEEE/ISO/IEC 12207-2017 - ISO/IEC/IEEE International Standard - Systems and software engineering -- Software life cycle processes</i>		IEEE
<i>UML 2 and the Unified Process: Practical Object-Oriented Analysis and Design</i>	2nd	Addison-Wesley
<i>developer.IBM Article: An introduction to the Unified Modeling Language</i>		Donald Bell
<i>developer.IBM Article: The UML 2 class diagram</i>		Donald Bell
<i>CSIT 515 Lectures</i>		Dr. H. Johnson

## Distribution List

**Table 0.3** This document has been distributed to:

Name	Position	Company
Dr. H. Johnson	Professor	Montclair State University

# 1 Executive Summary

## 1.1 Purpose of this document

This Document provides developers with guidance on the system architecture by defining the system's architecture and tracking the required information. The development life cycle includes incremental and iterative production of design documents based on the project's specifics and the development methodology used.

## 1.2 Identification

This Document will follow the following standards:

- ANSI Y32.9, American National Standard Graphic Symbols for Electrical Wiring and Layout Diagrams Used in Architecture and Building Construction.<sup>5</sup>
- ANSI/IEEE Std 260.3™, American National Standard for Mathematical Signs and Symbols for Use in Physical Sciences and Technology.
- IEC 60050, IEC International Electrotechnical Vocabulary.<sup>6</sup>
- IEC Multilingual Dictionary—Electricity, Electronics, and Telecommunications, 2005, Edition 6.7
- IEEE SA Standards Board Bylaws, New York: Institute of Electrical and Electronics Engineers, Incorporated, 2019.
- IEEE SA Standards Board Operations Manual, New York: Institute of Electrical and Electronics Engineers, Incorporated, 2019.
- IEEE Standards Dictionary Online.<sup>8</sup>
- IEEE Std 91™, IEEE Standard Graphic Symbols for Logic Functions. 9, 10
- IEEE Std 260.1™, IEEE Standard Letter Symbols for Units of Measurement (SI Units, Customary InchPound Units, and Certain Other Units).
- IEEE Std 260.4™, IEEE Standard Letter Symbols and Abbreviations for Quantities Used in Acoustics.
- IEEE Std 270™, IEEE Standard Definitions for Selected Quantities, Units, and Related Terms, with Special Attention to the International System (SI).
- IEEE Std 315™, IEEE Standard Graphic Symbols for Electrical and Electronics Diagrams (Including Reference Designation Letters).
- IEEE Std 945™, IEEE Recommended Practice for Preferred Metric Units for Use in Electrical and Electronics Science and Technology.
- IEEE Std 991™, IEEE Standard for Logic Circuit Diagrams.
- IEEE/ASTM SI 10, American National Standard for Metric Practice.
- ISO/IEC Directives, Sixth Edition, 2011, Part 2, Rules for the structure and drafting of International Standards.
- APA Style
- MLA Standard

## Scope

**Table 1.1** The parts of the Unified Process that were covered in this project.

Includes	
Specifications	This phase contains the capture of all possible requirements for the system being developed, documented in a document called a requirements specification
Design	This phase involves analyzing the requirements specifications from the initial phase and developing the system design. An overview of this system design helps in defining the overall system architecture, specifying hardware and system requirements.
Implementation	As a result of the system's design, small unit programs are developed and then integrated into the next phase. Unit testing involves developing and evaluating each unit to determine its functionality.
Testing	In the implementation phase, all the units developed are integrated into a system after each has been tested. Once this process has been completed, the entire system is tested for faults or errors.

**Table 1.2** The parts of the Unified Process that will not be covered in this project.

Excludes	
Deployment of system	The product is released into the market once it has completed a functional and non-functional test.
Maintenance	Occasionally, issues arise in the client environment. Patches are released to resolve these issues. New versions are also released to enhance the product. These changes are then delivered to the customer through maintenance.

## 1.3 Relationship to Other Plans

This document contains the Specifications and Designs related to:

- CSIT 515 Dynamic Website Requirements part 1
- CSIT 515 Dynamic Website Requirements part 2
- Data Dictionary

## 1.4 Methodology, Tools, and Techniques

### 1.4.1 Methodologies

#### 1.4.1.1 The Unified Process (UP)

Unified process (UP) is an architecture-centric, use-case-driven, iterative, and incremental development process that leverages unified modeling language and complies with the system process engineering metamodel.

#### 1.4.1.2 Tools

##### 1.4.1.2.1 Documentation Tools

###### **Google Workspace**

<https://workspace.google.com/>

Google Workspace is a collection of cloud computing, productivity, and collaboration tools, software, and products developed and marketed.

###### **Microsoft 365**

<https://www.office.com/>

Microsoft 365 is a suite of apps that help you stay connected and get things done.

##### 1.4.1.2.2 Planning Management Tools

###### **Trello**

<https://trello.com/>

Trello is a web-based, Kanban-style list-making application developed by Trello Enterprise, a subsidiary of Atlassian.

#### 1.4.1.2.3 Design Tools

##### **Adobe Creative Cloud**

<https://www.adobe.com/creativecloud.html>

Adobe Photoshop is a raster graphics editor developed and published by Adobe Inc. for Windows and macOS.

##### **UMLet**

<https://www.umlet.com/>

UMLet is a free, open-source UML tool with a simple user interface: draw UML diagrams fast, build sequence and activity diagrams from plain text, export diagrams to eps, pdf, jpg, SVG, and clipboard, share diagrams using Eclipse, and create new, custom UML elements. UMLet runs a stand-alone or Eclipse plug-in on Windows, OS X, and Linux.

#### 1.4.1.2.4 Front End Development Tools

##### **Bootstrap-vue**

<https://bootstrap-vue.org/>

Bootstrap-Vue is a component library based entirely on the popular Bootstrap frontend CSS framework. It provides several Vue.js components that ultimately render Bootstrap HTML markup decorated with the correct classes to be styled by the Bootstrap CSS files.

##### **Vue.js**

<https://vuejs.org/>

The Progressive JavaScript Framework.

##### **Axios**

<https://www.axios.com/>

Axios is a promise-based HTTP client for node.js and the browser. It is isomorphic, it can run in the browser and Node.js with the same codebase).

##### **VueCookies**

<https://www.npmjs.com/package/vue-cookies>

A simple Vue.js plugin for handling browser cookies

##### **VueCropper**

<https://www.npmjs.com/package/vue-cropper>

A simple Vue picture clipping plugin.

##### **VueRouter**

<https://router.vuejs.org/>

Vue Router is the official router for Vue.js (opens new window). It deeply integrates with Vue.js core to make building Single Page Applications with Vue.js a breeze.

#### 1.4.1.2.5 Back End Development Tools

##### **Node.js**

<https://nodejs.org/>

Node.js is an open-source, cross-platform, back-end JavaScript runtime environment that runs on the V8 engine and executes JavaScript code outside a web browser.

##### **Express.js**

<https://expressjs.com/>

Express is a minimal and flexible Node.js web application framework that provides robust features for web and mobile applications.

##### **CORS**

<https://expressjs.com/en/resources/middleware/cors.html>

CORS is a node.js package for providing a Connect/Express middleware that a Javascript Program can use to enable CORS with various options.

##### **Mongoose**

<https://mongoosejs.com>

Mongoose is an Object Data Modeling (ODM) library for MongoDB and Node.js. It manages relationships between data, provides schema validation, and translates between objects in code and the representation of those objects in MongoDB.

#### 1.4.1.2.6 Database Tools

##### **MongoDB**

<https://www.mongodb.com>

MongoDB is an open-source NoSQL database management program. NoSQL is used as an alternative to traditional relational databases. NoSQL databases are useful for working with

large sets of distributed data. MongoDB is a tool that can manage document-oriented information, store or retrieve information.

#### 1.4.1.2.7 IDEs

##### **Visual Studio Code**

<https://code.visualstudio.com/>

Visual Studio Code is a source-code editor made by Microsoft for Windows, Linux, and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git.

## 1.4.2 Techniques

### 1.4.2.1 End to End Testing:

End to End Testing involves a software testing method that validates entire software from starting to the end along with its integration with external interfaces.

### 1.4.2.2 BlackBox Testing:

BlackBox testing involves testing a system with no prior knowledge of its internal workings.

### 1.4.2.3 WhiteBox Testing:

WhiteBox Testing involves software testing in which the internal structure or the program or the code is hidden and nothing is known about it.

### 1.4.2.4 Object-oriented analysis and design (OOAD):

The website will be developed using object-oriented programming, as well as using visual modeling throughout the software development process to guide stakeholder communication and product quality.

### 1.4.2.5 Representativeness heuristic:

The design of the website will lean on the already developed color scheme of the Maggotty High official website.

## 1.5 Policies, Directives, and Procedures

### 1.5.1 Policies

Due to the length of the class, the detail of the project will be limited in its scope to only having a local working model. The implementation of a server will be considered in this document but will not be fully realized. All project details may not be completed due to time constraints but will be listed in this document.

Upon the recommendation of the class, this project has been developed according to the Unified Process.

In order to achieve the course objectives, the IEEE, and IBM standards are used; we strive to adhere to the course teachings as often as possible.

All external Texts are taken from citations used in the class textbooks or Lecture slides.

### 1.5.2 Directives

The main goal of this project is to show a working example of the course and to reinforce the lessons taught in the Lecture.

We will only be using 4 of the 9 Diagrams used in a typical software engineering project.

### 1.5.3 Procedures

The project will be restricted to using Javascript for all its main functionalities while using HTTP and CSS in order to support it.

Any additional libraries will also be Javascript-oriented.

## 2 Design Overview

### 2.1 Background Information

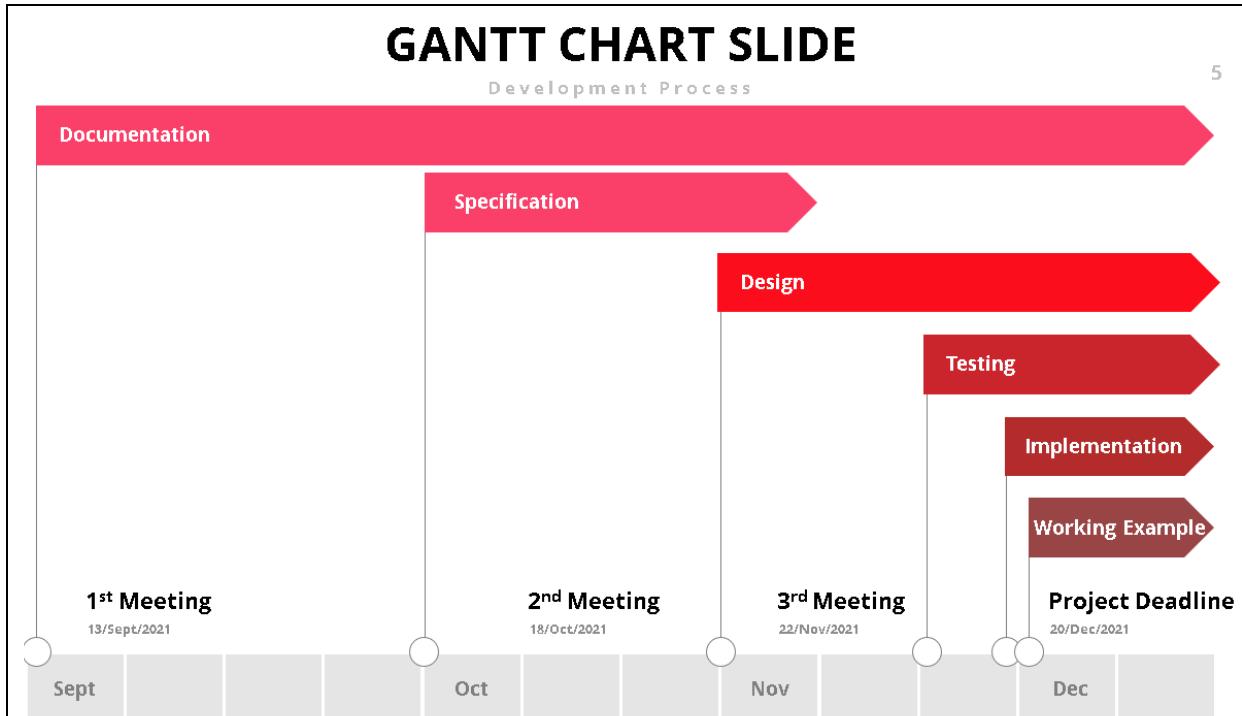
The Maggotty High Alumni Association – US northern Region Website will allow alumni from the US northern region to join the Maggotty High Alumni Association. The Association's primary focus is supporting the educational development of the students of Maggotty High through monetary contributions and donations given by the school's alumni.

Alumni will register and join the association through the website, make monetary contributions, register for volunteering and social events of the Alumni Association's choosing, and access Maggotty High educational news. The website will also allow the Maggotty High Alumni Association – US northern Region to accept monetary contributions and hold volunteering and social events of their choosing.

The website shall allow for easy collaboration and networking for alumni. Through online payment options and volunteer sign-up boards, it will also allow alumni to contribute and give Maggotty High. These online payment options and volunteer sign-up boards will be posted and maintained by the Maggotty High Alumni Association – US northern Region.

## 2.2 Constraints

In order to develop the most effective product that "The Mondays" could possibly produce during such a short time period. A Gantt chart is used to track each step of the development process.



**Figure 2.1** A Gantt chart showing the workflow of the team and project development.

Each flag represents a different part of the Unified Process.

Documentation: Changing and updating the system documentation to reflect changes throughout the entire development process.

**Table 1.1** provides a brief explanation of what each step of the process is intended to accomplish.

Within the constraints of time, each step of the process was efficiently completed.

## 2.3 Design Goals

### 2.3.1 Environment Usage

The website will be hosted on an online server with Distributed Denial of Service(DDOS) protection such as Cloudflare to ensure constant accessibility. The website will be accessible from any device with internet browsing capabilities, including personal computers (PCs), laptops, tablets, and cell phones.

### 2.3.2 Interface Compatibility

#### 2.3.2.1 Operating System Compatibility:

Website hosting will be through any service or web server that delivers the needs of a dynamic web site's front and backend development and ongoing maintainability.

The website will be accessible through any operating system that can run a web browser capable of running a dynamic website. Google Chrome, Mozilla Firefox, Microsoft Edge, and Safari are the recommended browsers due to their cross-platform compatibility.

#### 2.3.2.2 Device Compatibility:

The website will dynamically change how it loads to meet the needs of a wide array of devices from Desktops to Mobile Phones of all kinds. It will choose when to load certain page elements based on the device and connection type (wired vs wireless)

### 2.3.3 Availability

Due to online server hosting, the website will be available 24 hours a day, seven days a week. Even in an emergency where the primary server hosting the website goes down, a backup server will load it.

### 2.3.4 Efficiency

The website will allow multiple users to register, process payments, update schedules, and host events without crashing due to its dynamic nature.

The website will use algorithms to make logical decisions. Some of these decisions will be:

- Loading and manipulating primitive data types (e.g., integers and floats) and non-primitive classes) to optimize loading times based on connection type(wired vs. wireless) and speed
- Choosing image type (e.g., JPEG and BMP) and size to display information to the user
- Adjusting website display size to current user screen size, and even turning off some features depending on device constraints. (For example, turning off touch screen compatibility when no touch screen interfaces are detected.)

The overall goal of these implementations will be to ensure faster loading times and an overall smoother experience while users are using the website.

### 2.3.5 Security

The website will use payment services such as Stripe and Paypal to handle payment processing and donations securely.

Website performance and security will be done using a website security service such as Cloudflare. They have variable pricing and scalable options, which the website will utilize at every stage of its life span.

### 2.3.6 Maintenance

If the website needs maintenance, only a section of the website will be inaccessible during the maintenance. This dynamic design will allow users to access parts of the website that are not under maintenance.

### 2.3.7 Size performance and goals

The website will be on a server with high-speed Internet capability. The Maggotty High Alumni Association – US northern Region will determine the physical machine Region-owned server and contain a database. The web page will be limited to 1 MB whenever possible (Size limitation for most successful websites).

Web pages within the website that have media content will have a maximum loading time of three seconds. Text-only web pages will have a maximum loading time of one second.

## 2.4 User Characteristics

The website will have seven user types: Alumni, Administrators (Admins), Assistant Treasurer, Treasurer, President, Vice-President, and Users without Accounts.

### 2.4.1 User Objectives

#### 2.4.1.1 Users without Accounts

Front-end users of the website. They will have access to the following:

- Place Own Shop Orders
- View Calendar Events and Polls
- View News
- Make Donations

#### 2.4.1.2 Alumni

- Front-end users of the website. They will have access to the following:
- Manage Own Account
- Place Own Shop Orders
- View Calendar Events and Polls
- Manage Own Files and Media
- Manage Own Virtual Meetings
- View News
- Sign up for Volunteer Positions
- Make Donations

#### 2.4.1.3 Assistant Treasurer

Front-end user with special access to financial reports. The assistant treasurer will have the ability to:

- Manage Own Account
- Manage All Orders
- Calendar Events and Polls
- Manage Own Virtual Meetings
- View News
- Sign up for Volunteer Positions
- Make Donations
- Manage Donations

#### 2.4.1.4 Treasurer

Front-end user with special access to specific reports. The treasurer will have the ability to:

- Manage Own Account
- Manage All Orders
- Calendar Events and Polls
- Manage Own Virtual Meetings
- View News
- Sign up for Volunteer Positions
- Make Donations
- Manage Donations

#### 2.4.1.5 Vice-President

Back-end user the Vice-president will have the ability to:

- Manage Own Account
- Manage All Accounts
- Place Own Shop Orders
- Manage All Orders
- Manage Products listed on the shop
- View Calendar Events and Polls
- Manage All Calendar Events and Polls
- Manage Own Files and Media
- Manage All Files and Media
- Manage Own Virtual Meetings
- Manage All Virtual Meetings
- View News
- Manage News
- Sign up for Volunteer Positions
- Manage Volunteer Listings and Volunteer sign up lists
- Make Donations
- Manage Donations

#### 2.4.1.6 President

Back-end user the president will have the ability to:

- Manage Own Account
- Manage All Accounts
- Place Own Shop Orders
- Manage All Orders
- Manage Products listed on the shop
- View Calendar Events and Polls
- Manage All Calendar Events and Polls
- Manage Own Files and Media
- Manage All Files and Media
- Manage Own Virtual Meetings
- Manage All Virtual Meetings
- View News
- Manage News
- Sign up for Volunteer Positions
- Manage Volunteer Listings and Volunteer sign up lists
- Make Donations
- Manage Donations

#### 2.4.1.7 Admin

Back-end user of the website. The website launch will start with one Admin and will have scalability options if more Admins are needed:

- Manage Own Account
- Manage All Accounts
- Place Own Shop Orders
- Manage All Orders
- Manage Products listed on the shop
- View Calendar Events and Polls
- Manage All Calendar Events and Polls
- Manage Own Files and Media
- Manage All Files and Media
- Manage Own Virtual Meetings
- Manage All Virtual Meetings
- View News
- Manage News
- Sign up for Volunteer Positions
- Manage Volunteer Listings and Volunteer sign up lists
- Make Donations
- Manage Donations

## 3 System Architecture

### 3.1 Hardware Architecture

#### 3.1.1 Webserver



**Figure 3.1**

A Web server requires a computer connected directly to the Internet, usually via an ethernet network (LAN/WAN). Low-end computers (80386-based PCs or 68040 Macintoshes) can run Web servers, but if you want your server to be responsive, you will need a more powerful machine (such as a Pentium or PowerPC Mac). In order to operate effectively, a Web server must be equipped with a large and fast hard drive as well as plenty of RAM (at least 16 MB).

Item	Web server (minimal)	Web server (recommended)	Combined Web & Database Server (minimal)	Combined Web & Database Server (recommended)
Processor	1,6 GHz CPU	2 x 1,6 GHz CPU	2 x 1,6 GHz CPU	4 x 1,6 GHz CPU
RAM	1,75 GB RAM	3,5 GB RAM	3,5 GB RAM	7 GB RAM
HDD	1x 250+ GB of free space or more is recommended for the webshop data (non-system drive is preferred)			
SSD	1x 250+ GB of free space or more is recommended for the software that is listed in the software requirements (system drive)			
Power Supply	500 GD, 80+ GOLD 500W			

**Table 3**

#### Processor

**Pentium:** Refers to a series of microprocessors produced by Intel that are compatible with the x86 architecture.

## RAM

**Crucial RAM:** The Crucial RAM brand of memory is a line of RAM manufactured by Micron, an electronics company that produces chips for gaming PCs. It holds up well against RAM manufactured by other companies.

## SSD

**M.2 NVMe Interface Internal Solid State Drive:** An M.2 SSD is a small form factor solid-state drive (SSD) that is used in internally mounted storage expansion cards.

## Power Supply

**EVGA Power Supply:** EVGA Power Supply family comes in 5 different models, each has several sub-models and different wattage ratings.

### 3.1.2 Networking Devices



**Figure 3.2**

#### Modem

A modem is a device typically provided to you by your Internet Service Provider enabling one or more computers to connect to the Internet.



**Figure 3.3**

#### Router

A router is a hardware device that directs data traffic between two or more networks.



**Figure 3.4**

#### Firewall

A firewall is either hardware or software that acts as a filter or gatekeeper to stop unauthorized users from accessing a firm's internal or sensitive data or applications.

## 3.2 Software Architecture

### 3.2.1 Content Authoring

The website will provide a section for users to define and publish user-specific content. content authoring includes:

- The uploading of files and media - text, images, and videos
- Creation of calendar events, polls
- Image galleries
- Managing account information such as profile picture and bio

Users will be able to create, delete, and update content swiftly to make sure that other users are intuned with what is going on with them.

### 3.2.2 Mobile Support

The website will have mobile support and adaptability using the following features:

- Content automation adaption for display on mobile devices
- Preview in different screen resolutions
- Optimization for mobile data speeds
- Easy to navigate

Users will be able to access the website from their mobile devices. The dynamic structure of the website will optimize the website for mobile and desktop browsing via combined features such as big finger-friendly buttons and navigation, swipe and zoom options, and the ability to change between landscape and portrait mode.

### 3.2.3 Media and Files

The website will have File and Media hosting so that the users will be able to upload and manage their media and view the media of others.

- Media and File hosting will consist of:
- Offsite Media Hosting
- Media Streaming
- Automatic thumbnail creation of images

### 3.2.4 User interactivity

The website will have WebForms, allowing users to interact with Admins and each other.

User interactivity consists of:

- Web forms for user interactions
- Allow visitors to provide feedback on the website.
- Virtual meeting spaces
- The news page will be where users can see event details and updates
- An outside messenger group will provide direct interaction between users.
- Support chat where users can reach out to Admins for support

Users will interact easily and quickly respond to one another on the appropriate boards without confusion or incident.

### 3.2.5 Payment and Shopping Cart Functionality

The website will have Payment gateways and a shopping cart feature for easy shopping and donating.

- Payment functionality consists of:
  - A payment gateway (Paypal, Stripe, Epay)
  - Secure payment landing page
  - Payment encryption protected
- Shopping Cart Functionality consists of:
  - Customized checkout feature
  - Built-in payment feature/gateway (for donation, purchasing of tickets)
  - Save credit card information, and order history
  - Support Multi-currency transaction
  - Provide quotes (for ticket) that can be manually modified/updated before publishing

Users will be able to purchase products, tickets, or make donations, be transported to a payment gateway where all payment security will be hosted, then brought back to the website seamlessly for continued browsing.

### 3.2.6 Account management

Users will be able to manage their accounts.

Account management consists of:

- Changing email and passwords
- Account deactivation
- changing or updating payment methods
- Changing or updating billing and shipping address

### 3.2.7 Registration

Users will be able to register a new alumni account.

The registration consists of:

- Name (First, Last)
- School name and Student ID from graduated school
- Email address and password
- Shipping and billing address

## 3.3 Communications Architecture

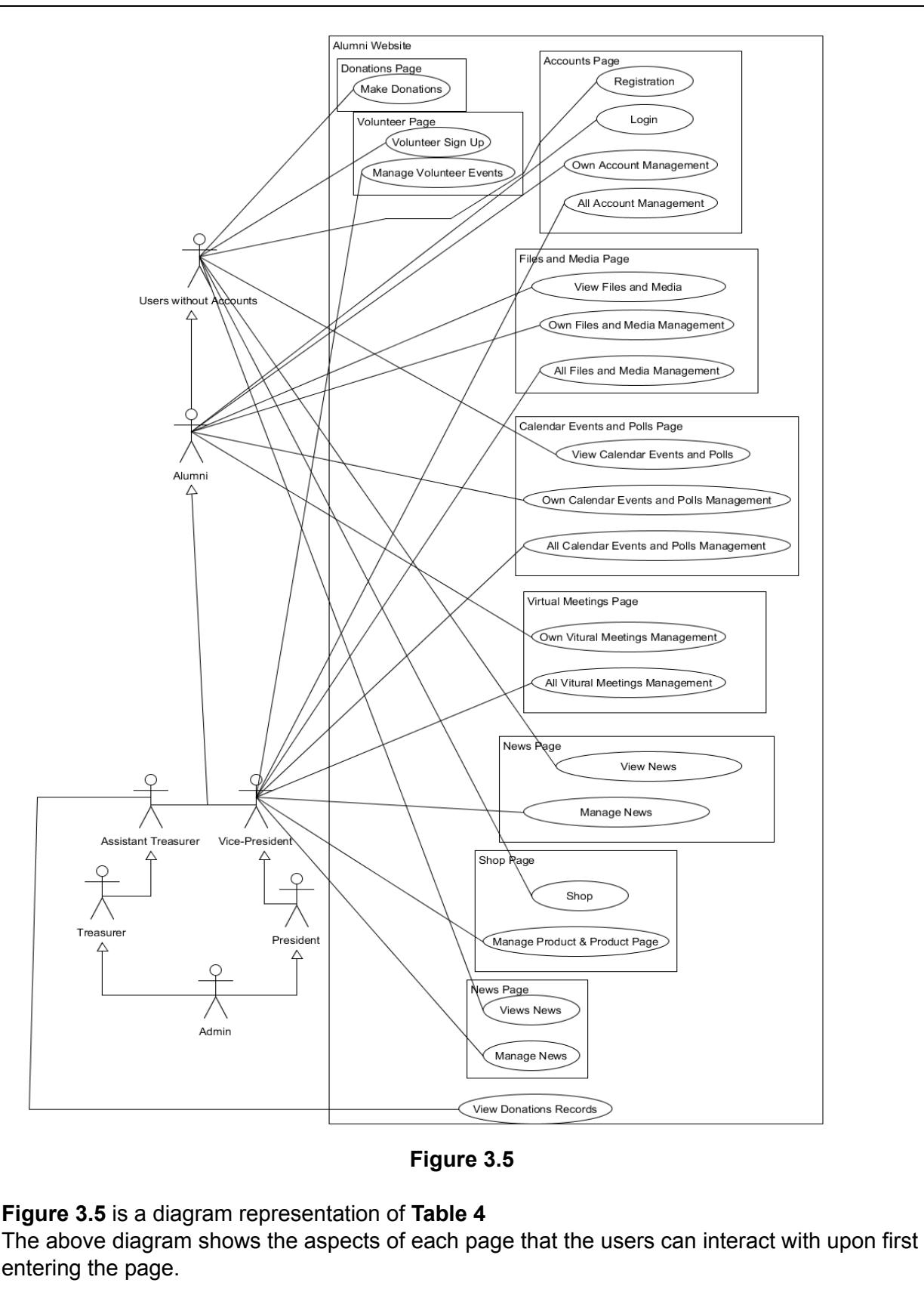
### 3.3.1 User-System Communication

The system is designed so that each page can be navigated from the top menu bar, but each page is independent of one another. Information is retrieved from listings and records that are maintained in a database that is held on the webserver through MongoDB.

	Alumni	Admin	Assistant Treasurer	Treasurer	Vice President	President	Users without Accounts
Permissions							
Manage Own Account	*	*	*	*	*	*	
Manage Other Accounts (Including access to additional functionalitie s)		*			*	*	
Place Own Shop Orders	*	*	*	*	*	*	*
Manage All Orders		*		*		*	
Manage Products listed on the shop		*			*	*	
View Calendar Events and Polls	*	*	*	*	*	*	* Only view

Manage All Calendar Events and Polls		*			*	*	
Manage Own Files and Media	*	*	*	*	*	*	
Manage All Files and Media		*			*	*	
Manage Own Virtual Meetings	*	*	*	*	*	*	
Manage All Virtual Meetings		*			*	*	
View News	*	*	*	*	*	*	*
Manage News		*			*	*	
Sign up for Volunteer Positions	*	*	*	*	*	*	
Manage Volunteer Listings and Volunteer sign-up lists.		*			*	*	
Make Donations	*	*	*	*	*	*	*
Manage Donations		*	*	*			

Table 4

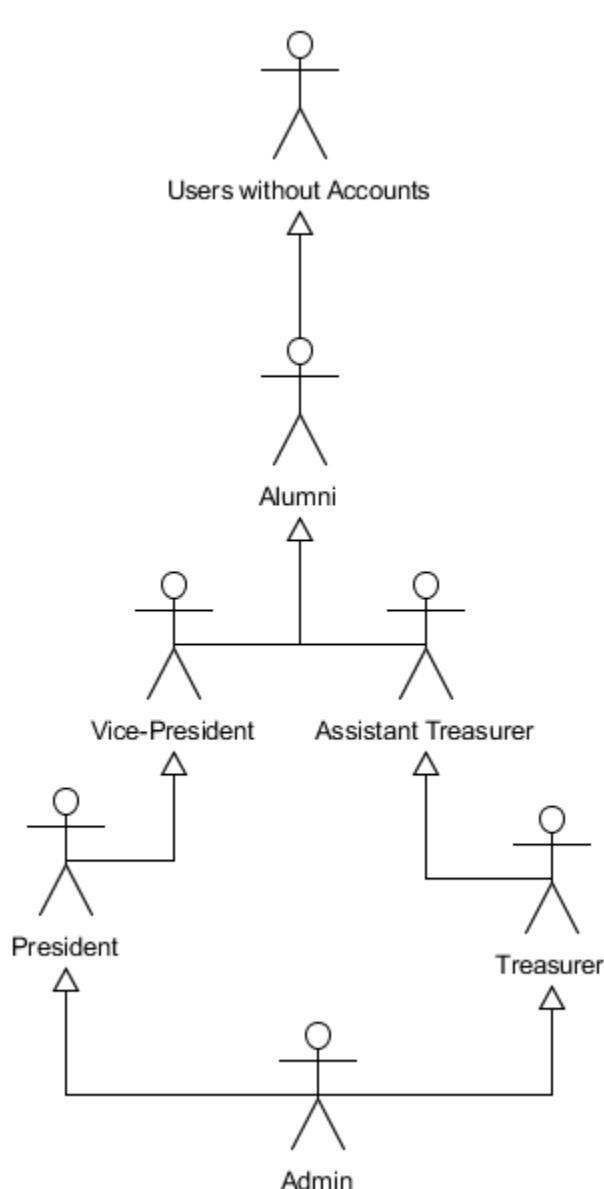
**Figure 3.5**

**Figure 3.5** is a diagram representation of **Table 4**

The above diagram shows the aspects of each page that the users can interact with upon first entering the page.

**Figure 3.5** shows the user hierarchy and the interactivity of each page across the entire system.

Once logged in the system checks to see which user has logged in and travels down the hierarchy in order to determine what access and privileges they have. The accounts hierarchy system works as follows:



**Figure 3.6**

Each stick figure represents an Actor in the system. The hierarchy is from top-down where the

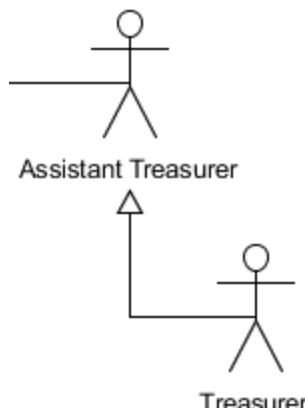
Users with the least permissions start at the top, and more permissions are added through inheritance as you traverse downward.

**Figure 3.6** shows the relationship each User object has with one another. Each user has the ability to do all the things that the user listed directly above them (Denoted by a line with an arrow pointing to them) can do.



**Figure 3.7**

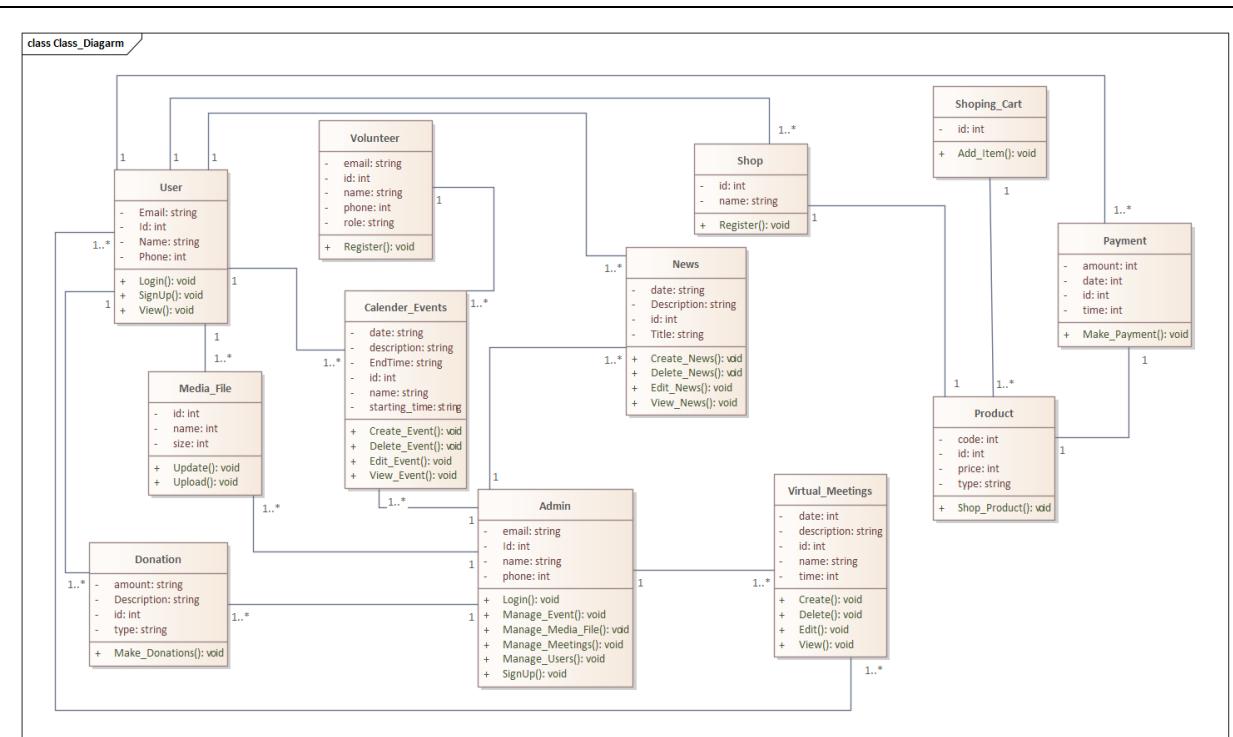
The Vice-President and Assistant Treasurer relationship displays what happens when two Users sit on the same level of permissions. The users at the same level do not have access to the permissions that the other users at that level possess



**Figure 3.8**

The Assistant Treasurer and Treasurer relationship display how inheritance works. A relationship inheritance is determined by the arrow, as the Actor that it points to gives all of its permissions to the Actor at the other end.

As shown in **Figure 3.8**, the Treasurer is able to perform all of the functions of the Assistant Treasurer. This relationship is one-directional, meaning that the Treasurer may have permissions that the Assistant Treasurer does not have access to.

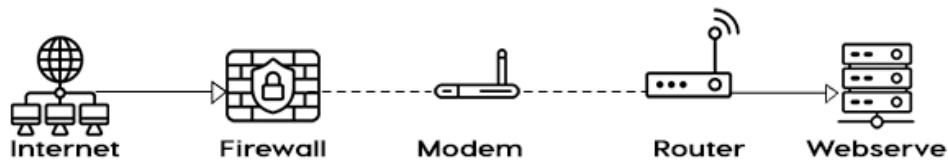


**Figure 4.3**

Classes are drawn in class diagrams to give the user a general understanding of how classes are implemented. Various types of associations are present to illustrate the relationship between classes. Each class has different multiplicities to illustrate how they will interrelate.

- Below you will find examples of these multiplicities.
- A User can shop for many products
  - But each Shop session can be applied to 1 user.

### 3.3.2 Hardware Communication



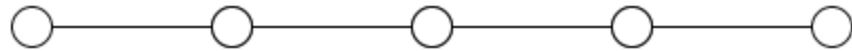
**Figure 3.9**

This is a graphical representation of the Network set up at the initial launch.

As shown in **Figure 3.9**, the network design also illustrates the direction of data flow between the point at which users attempt to connect to the network and the point at which they connect to the webserver.

Users will have to be allowed access to the network through the firewall (**Figure 3.4**).

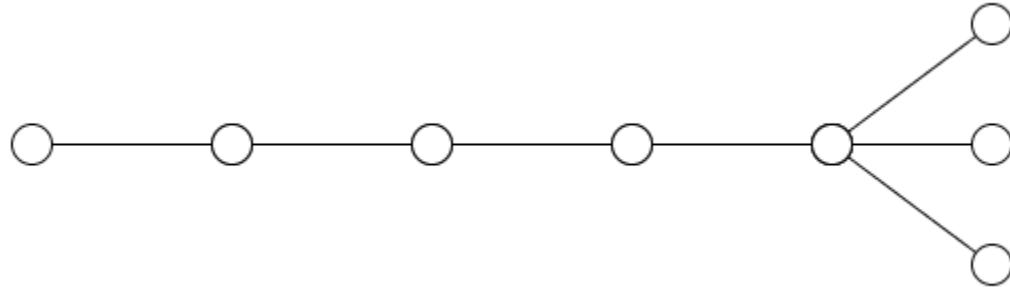
Once the firewall allows them access, their connection will go through the modem (**Figure 3.2**) to the router (**Figure 3.3**) and then finally to the webserver (**Figure 3.1**), where they will be able to access the website.



**Figure 3.10**

The Network topology is linear. Represented by **Figure 3.10**

To avoid adversely affecting the operation of the website, it may become necessary to move the database to its own independent server as it grows. Although this will affect the performance of lookups, a hybrid topology can mitigate this.



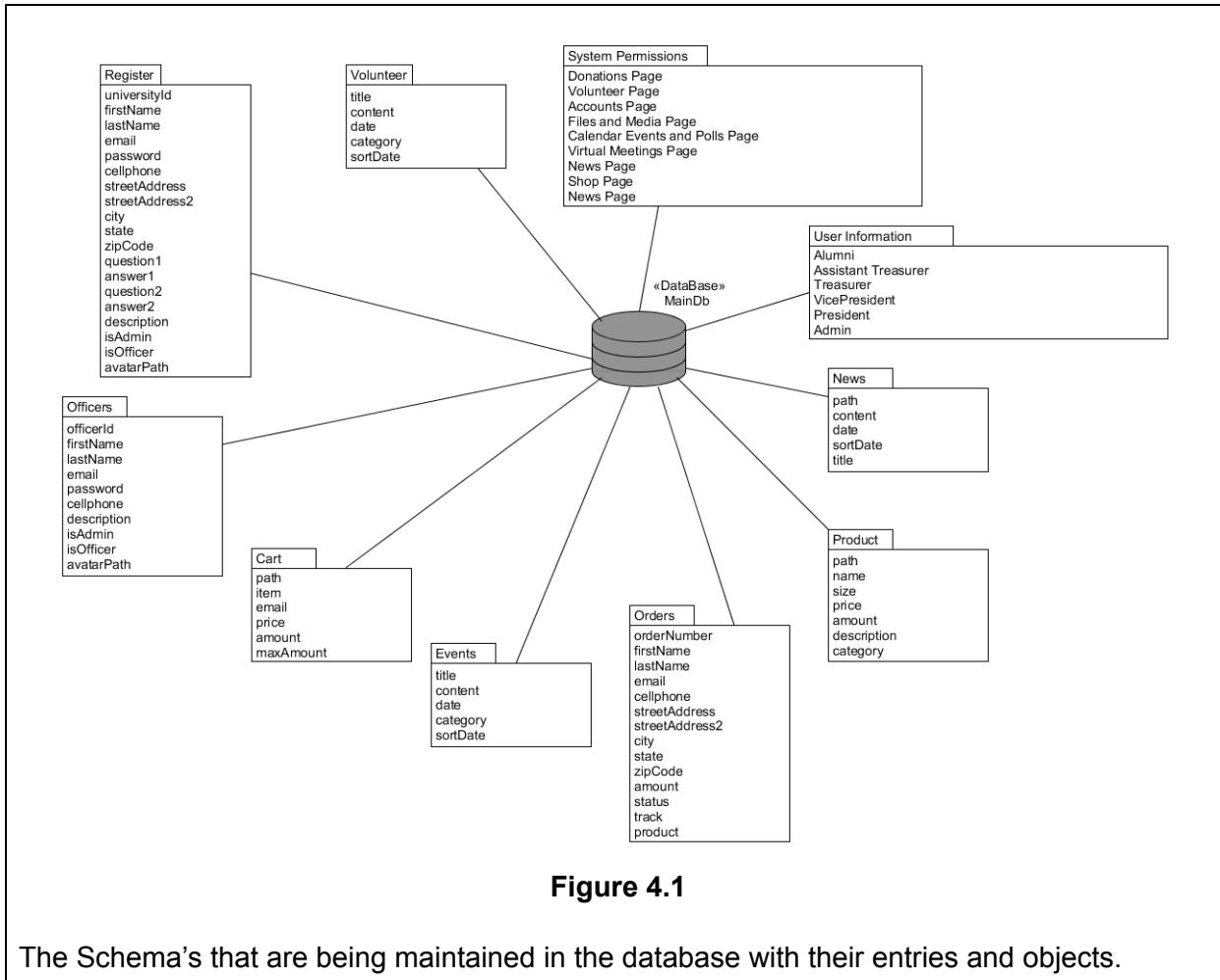
**Figure 3.11**

A Hybrid network topology, Linear-Star.

The network diagram in **Figure 3.11** can be used to expand the webserver using external servers if necessary.

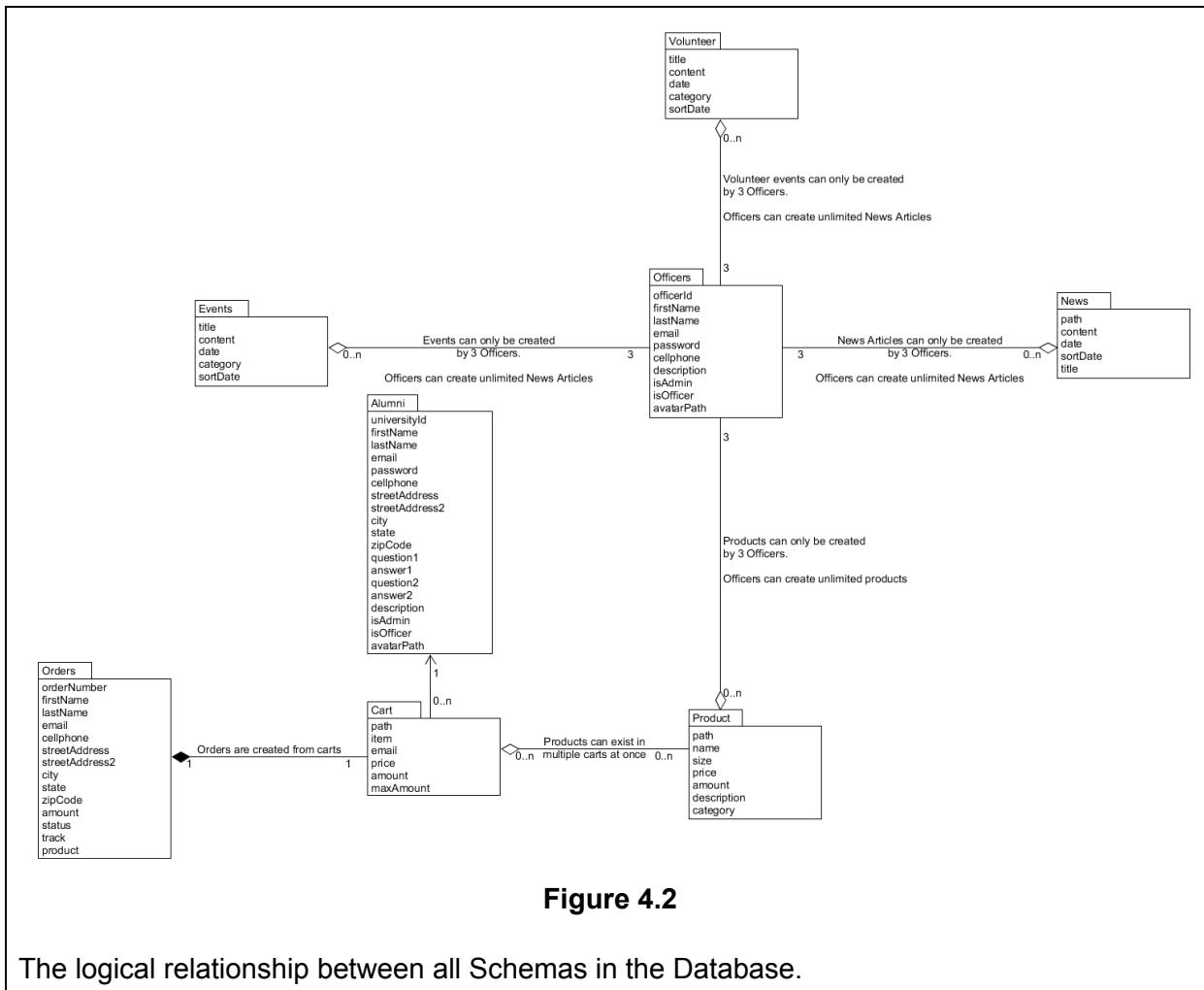
# 4 Data Design

## 4.1 Database Management System Files



**Figure 4.1**

The Schema's that are being maintained in the database with their entries and objects.



The **Data Dictionary** section contains more detailed information about the entities listed in both **Figure 4.1**, and **Figure 4.2**.

## 5 Detailed Design

### 5.1 Hardware Detailed Design

#### 5.1.1 Web Server Specifics

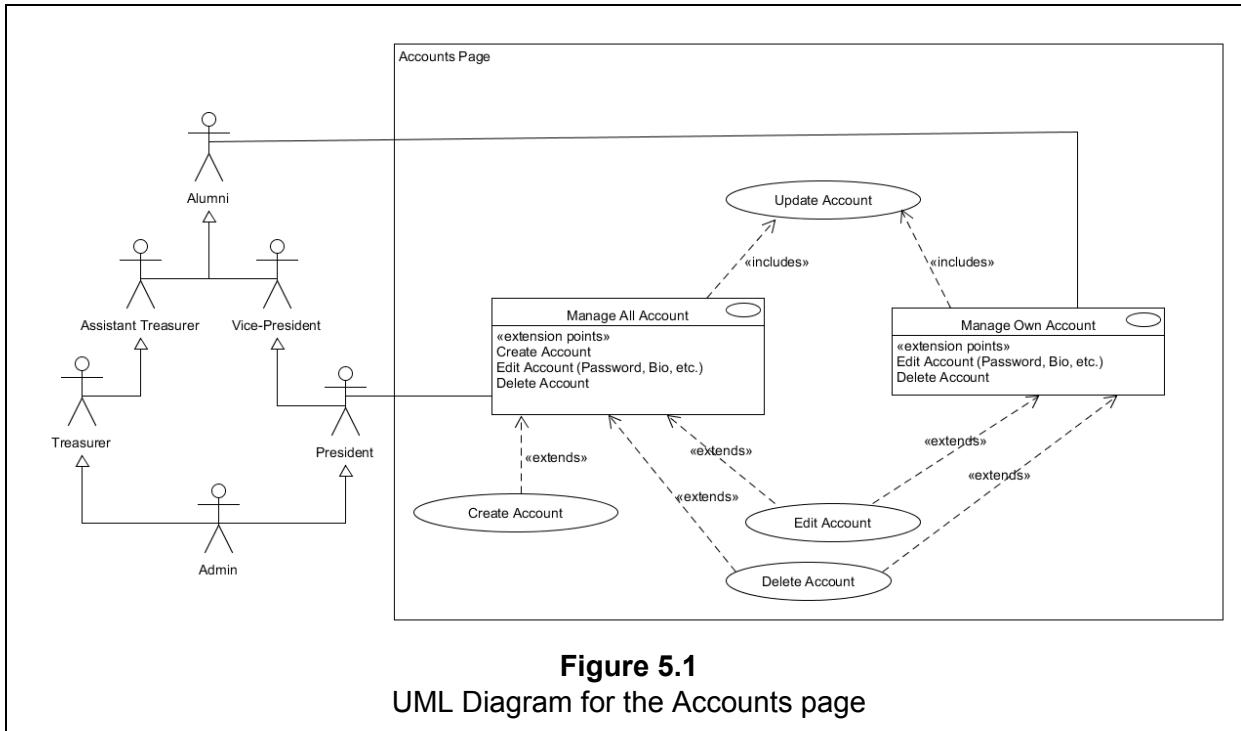
Webserver	Name	Price ~ USD	Purchase Link
Processor	Intel® Xeon® Bronze 3206R Processor (11M Cache, 1.90 GHz)	\$355	<a href="https://www.intel.com">Intel.com</a>
RAM	Crucial RAM 16GB DDR4 3200MHz CL22	\$61.99	<a href="https://www.amazon.com">Amazon.com</a>
SSD	SAMSUNG 980 SSD 1TB	\$109.99	<a href="https://www.amazon.com">Amazon.com</a>
Power Supply	500 GD, 80+ GOLD 500W	\$39.99	<a href="https://www.amazon.com">Amazon.com</a>
Total		\$566.97	

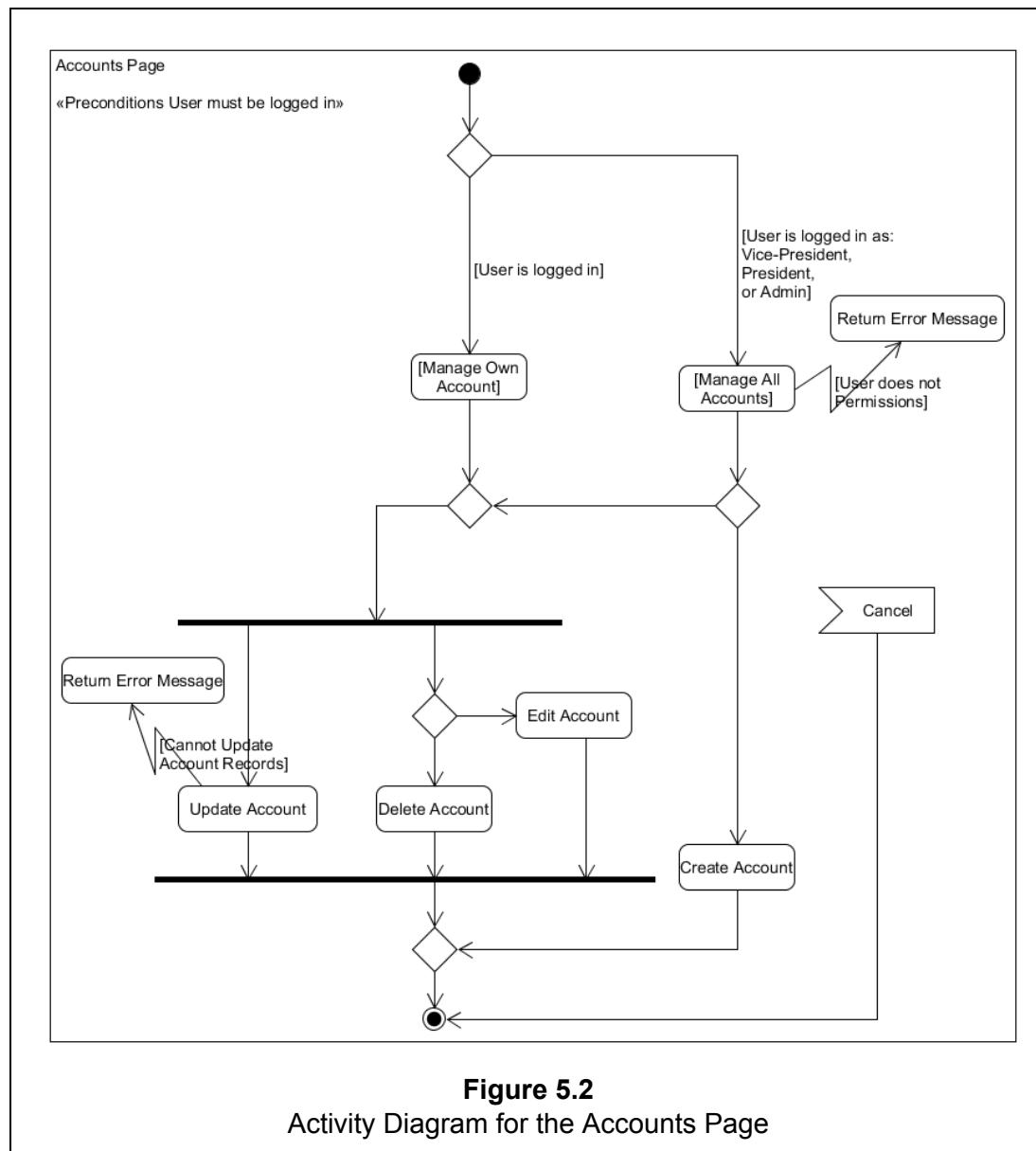
### 5.1.2 Networking Specifics

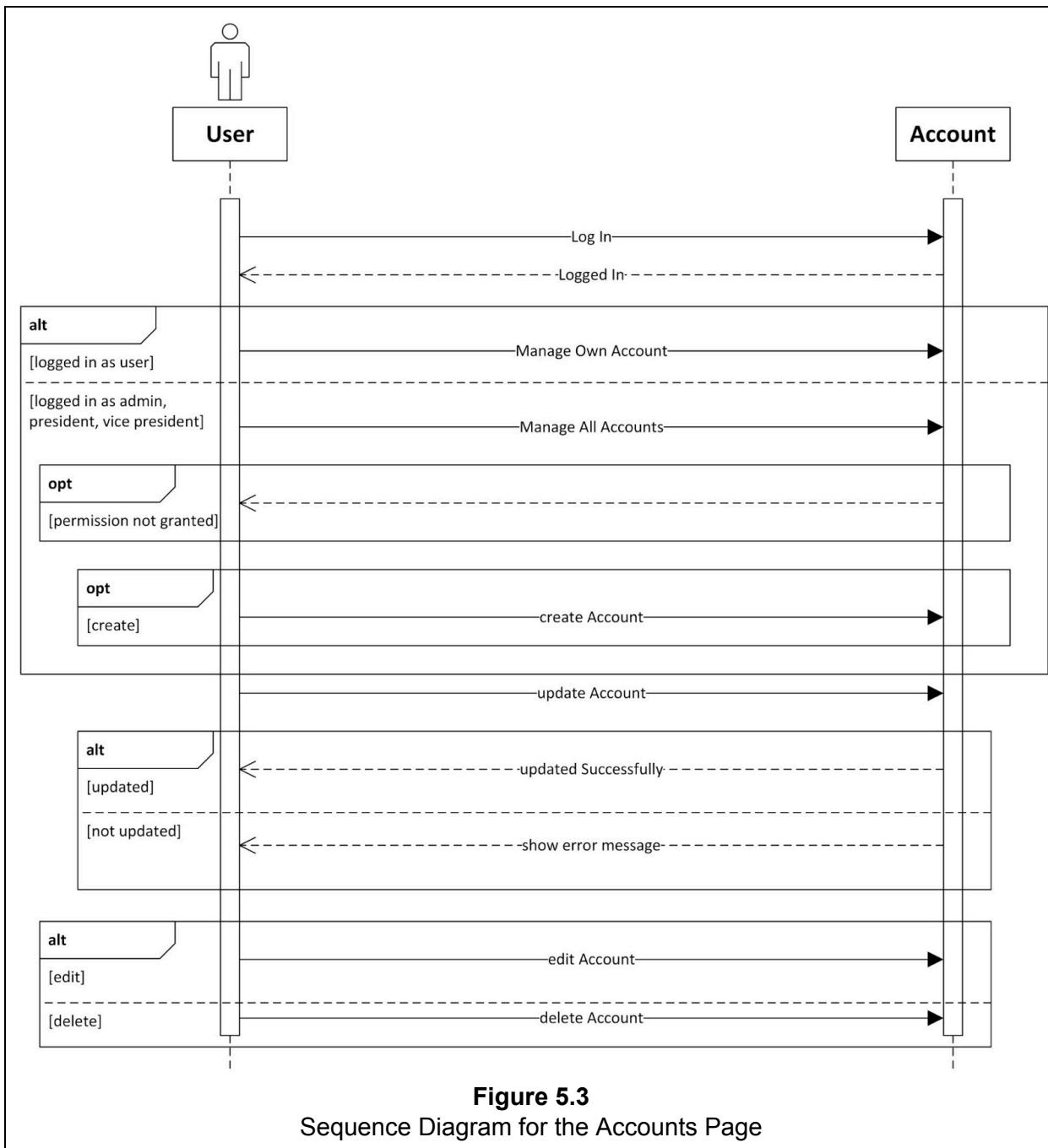
Network	Name	Price ~ USD	Purchase Link
Modem	NETGEAR - 16 x 4 DOCSIS 3.0 Cable Modem	\$44.99	<a href="#">Bestbuy.com</a>
Router	NETGEAR - Dual-Band AC1200 Router with 8 x 4 DOCSIS 3.0 Cable Modem	\$82.99	<a href="#">Bestbuy.com</a>
Firewall	Firewalla Blue Plus	\$199.00	<a href="#">Firewalla.com</a>
Ethernet Cables x 3	Insignia™ - 8' Cat-6 Ethernet Cable - Gray	\$44.97(14.99 x 3)	<a href="#">Bestbuy.com</a>
Total		\$370.96	

## 5.2 Software Detailed Design

### 5.2.1 Accounts Page







**Figure 5.3**  
Sequence Diagram for the Accounts Page

The Accounts page use case diagram shows 3 primary actors: the Alumni, the Admin, and the President. Alumni will be able to manage their own accounts which include the ability to edit accounts, update accounts, and delete accounts. The President and Admin will both be able to manage all accounts which include the ability to create an account, edit an account, update an account, and delete an account.

Use Case:	Manage Own Account
ID:	Figure 5.1
Summary Description:	The system will take the User to the Manage Account page to manage their Account.
Preconditions:	The User must be logged in.
Postconditions:	On success: this system will update Account  On Failure: No updates will occur
Primary Actors:	Alumni, Assistant Treasurer, Treasurer
Secondary Actors:	
Trigger:	The User clicked on the “manage Account” button
Main Flow:	<ol style="list-style-type: none"> <li>1. The user clicks on the “manage Account” button.</li> <li>2. The system takes the user to a page where they have two options:                     <ul style="list-style-type: none"> <li>o 2.1. Extends::Edit an Account</li> <li>o 2.3. Extends::Delete an Account</li> </ul> </li> <li><b>3. INCLUDE::Update Account Records</b></li> </ol>
Alternate Flows:	<ol style="list-style-type: none"> <li>1a. The user attempts to Manage someone else’s Account.                     <ul style="list-style-type: none"> <li>• 1a.1. The system returns an error message that says, “You do not have permissions to change this Account.”</li> </ul> </li> <li>3a. The user decides not to manage any events.                     <ul style="list-style-type: none"> <li>• 3a.1. The system returns the user to the Account Page.</li> </ul> </li> </ol>

Use Case:	Manage All Account
ID:	Figure 5.1
Summary Description:	The system will take the User to the Manage Account page.
Preconditions:	The User must be logged in as: <ul style="list-style-type: none"> <li>● Vice-President</li> <li>● President</li> <li>● Admin</li> </ul>
Postconditions:	On success: this system will update the Account.  On Failure: No updates will occur
Primary Actors:	Vice-President, President, Admin
Secondary Actors:	
Trigger:	The User clicked on the “manage Account” button
Main Flow:	<ol style="list-style-type: none"> <li>1. The user clicks on the “manage Account” button.</li> <li>2. The system takes the user to a page where they have two options:           <ul style="list-style-type: none"> <li>○ 2.1. Extends::Create/Publish a new Account</li> <li>○ 2.2. Extends::Edit an Account</li> <li>○ 2.3. Extends::Delete an Account</li> </ul> </li> <li><b>3. INCLUDE::Update Account Records</b></li> </ol>
Alternate Flows:	<ol style="list-style-type: none"> <li>3a. The user decides not to manage any events.           <ul style="list-style-type: none"> <li>● 3a.1. The system returns the user to the Account Page.</li> </ul> </li> </ol>

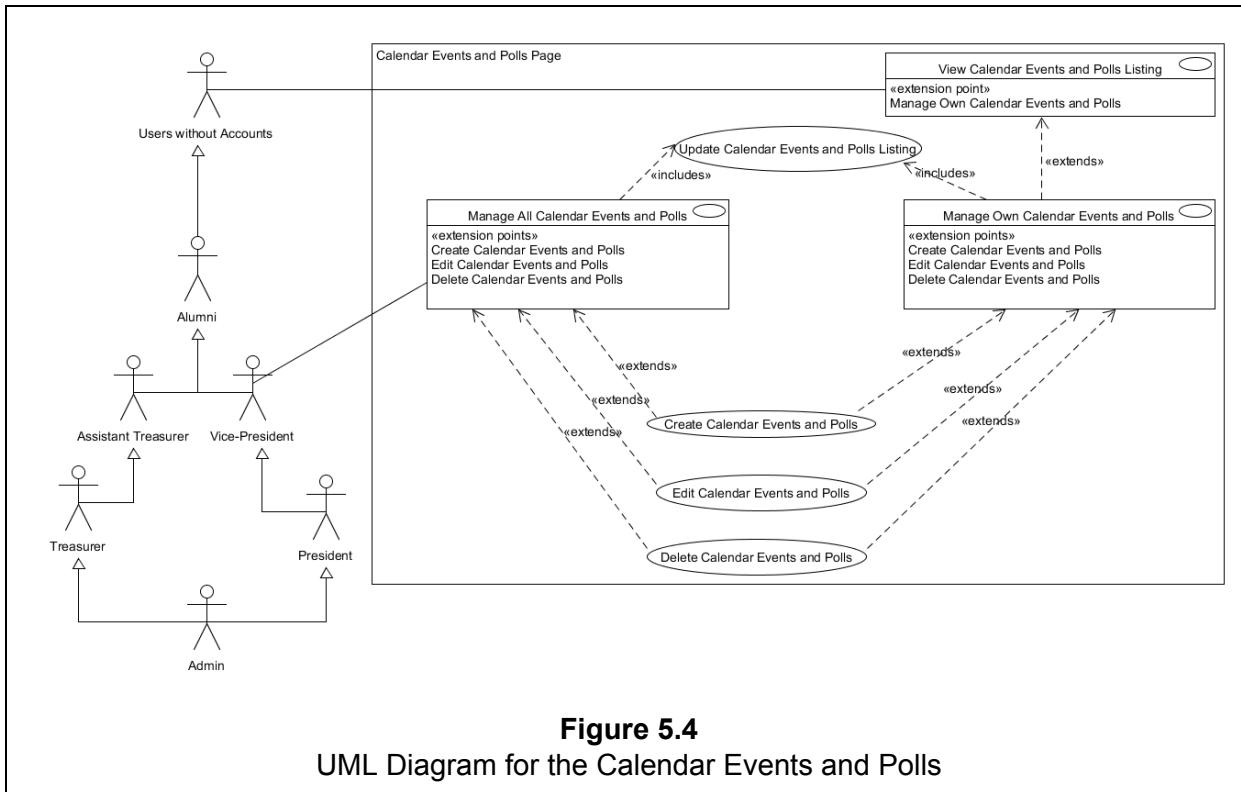
Use Case:	Update Account Records
ID:	Figure 5.1
Summary Description:	The system updates the Account Records.
Preconditions:	
Postconditions:	On success: this system will Update Account Records.  On Failure: No updates will occur
Primary Actors:	
Secondary Actors:	
Trigger:	The user is finished managing the Account.
Main Flow:	<ol style="list-style-type: none"><li>1. The user has either created, edited, or deleted an Account.</li><li>2. The system now updates the Account.</li></ol>
Alternative Flows:	<p>2a. The system cannot update the Account Records.</p> <ul style="list-style-type: none"><li>• 2a.1. The system presents the user with an error message that states, “the Account Records cannot be updated at this time.”</li><li>• 2a.2. The system returns the user to the Accounts page.</li></ul>

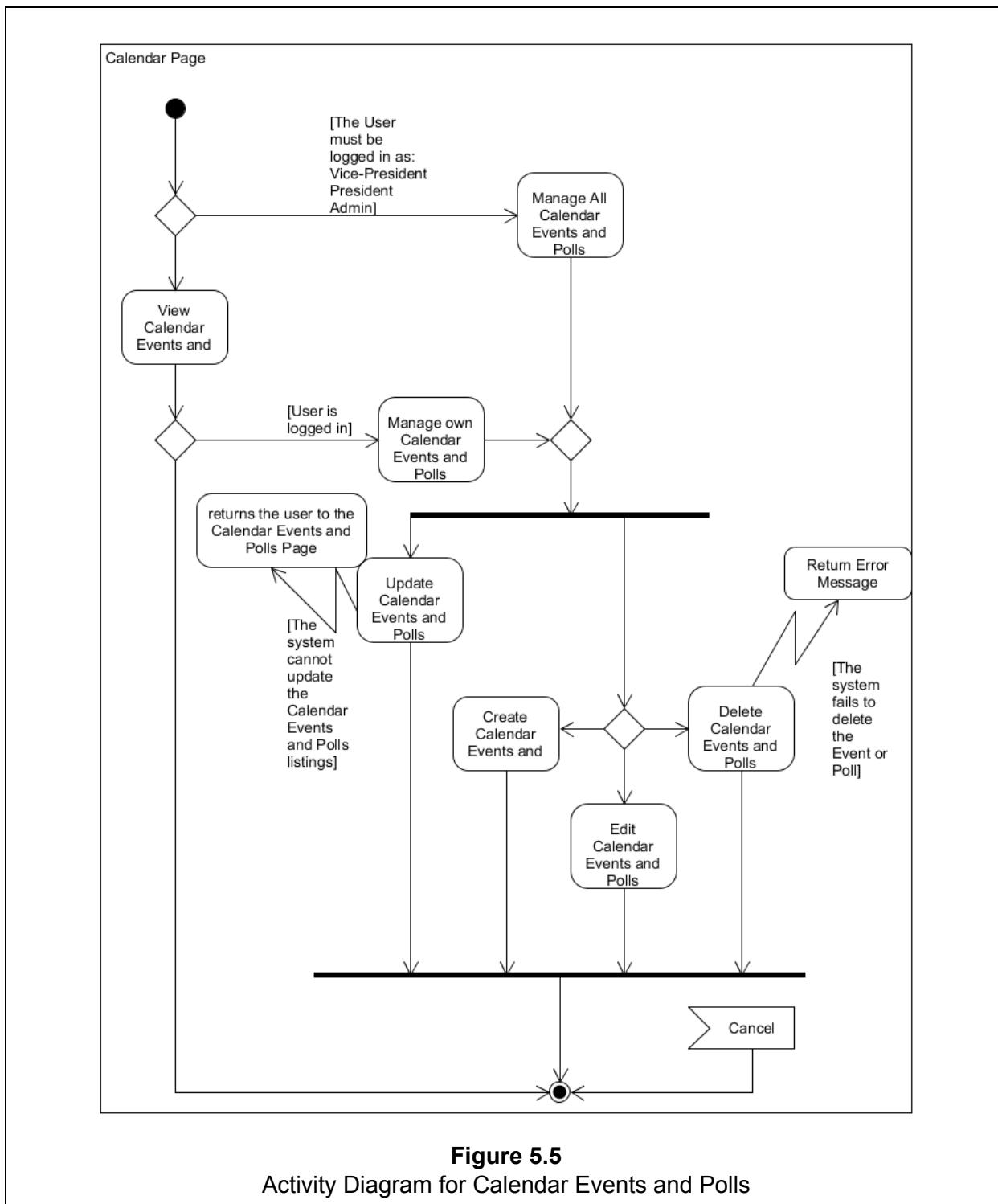
Use Case:	Create Account
ID:	Figure 5.1
Summary Description:	The User will create an Account.
Preconditions:	<p>The User must be logged in as:</p> <ul style="list-style-type: none"> <li>● Vice-President</li> <li>● President</li> <li>● Admin</li> </ul>
Postconditions:	<p>On success: The User will have created an Account.</p> <p>On Failure: No updates will occur</p>
Primary Actors:	Vice-President, President, Admin
Secondary Actors:	
Trigger:	The user clicked the “Create an Account” button
Main Flow:	<ol style="list-style-type: none"> <li>1. The user clicks the “Create an Account” button</li> <li>2. The system will load a page where the user can fill out an info form about the Account.</li> <li>3. The system will also present the user with two buttons on the page: <ul style="list-style-type: none"> <li>○ 3.1. "Create Account" button</li> <li>○ 3.2. "cancel" button</li> </ul> </li> </ol>
Alternate Flows:	<p>3a: user clicks the “Create Account” button.</p> <ul style="list-style-type: none"> <li>● 3a.1. Update Account Records use case kicks in.</li> <li>● 3a.2. The system updates the Account Records with the created Account being posted.</li> </ul> <p>3b: user clicks the “cancel button.”</p> <ul style="list-style-type: none"> <li>● 3b.1. The system returns the user to the Account Page</li> </ul>

Use Case:	Edit Account
ID:	Figure 5.1
Summary Description:	The User will Edit an Existing Account
Preconditions:	<p>The User must be logged in.</p> <p>Users who can only Edit their Account:</p> <ul style="list-style-type: none"> <li>• Alumni,</li> <li>• Assistant Treasurer,</li> <li>• Treasurer</li> </ul> <p>Users who can Edit all Accounts:</p> <ul style="list-style-type: none"> <li>• Vice-President</li> <li>• President</li> <li>• Admin</li> </ul> <p>The User must be Managing Account</p>
Postconditions:	<p>On success: The User will have edited an Account.</p> <p>On Failure: No updates will occur</p>
Primary Actors:	Users excluding those without accounts
Secondary Actors:	
Trigger:	The User clicked on the “Edit Account” button
Main Flow:	<ol style="list-style-type: none"> <li>1. The system shows an “Edit Account” button</li> <li>2. The user clicks the “Edit Account” button</li> <li>3. The system will load a page to fill in an “Account” information form.</li> <li>4. The system will also present the user with two buttons on the page:                     <ol style="list-style-type: none"> <li>a. 4.1. “Update Account” button</li> <li>b. 4.2. “cancel” button</li> </ol> </li> </ol>
Alternate Flows:	<p>4a: user clicks the “Update Account” button.</p> <ul style="list-style-type: none"> <li>• 4a.1. Update Account Records listing use case kicks in.</li> <li>• 4a.2. The system updates the Account Records with the published Account being posted.</li> </ul> <p>4b: user clicks the “cancel button.”</p> <ul style="list-style-type: none"> <li>• 4b.1. The system returns the user to the Account Page</li> </ul>

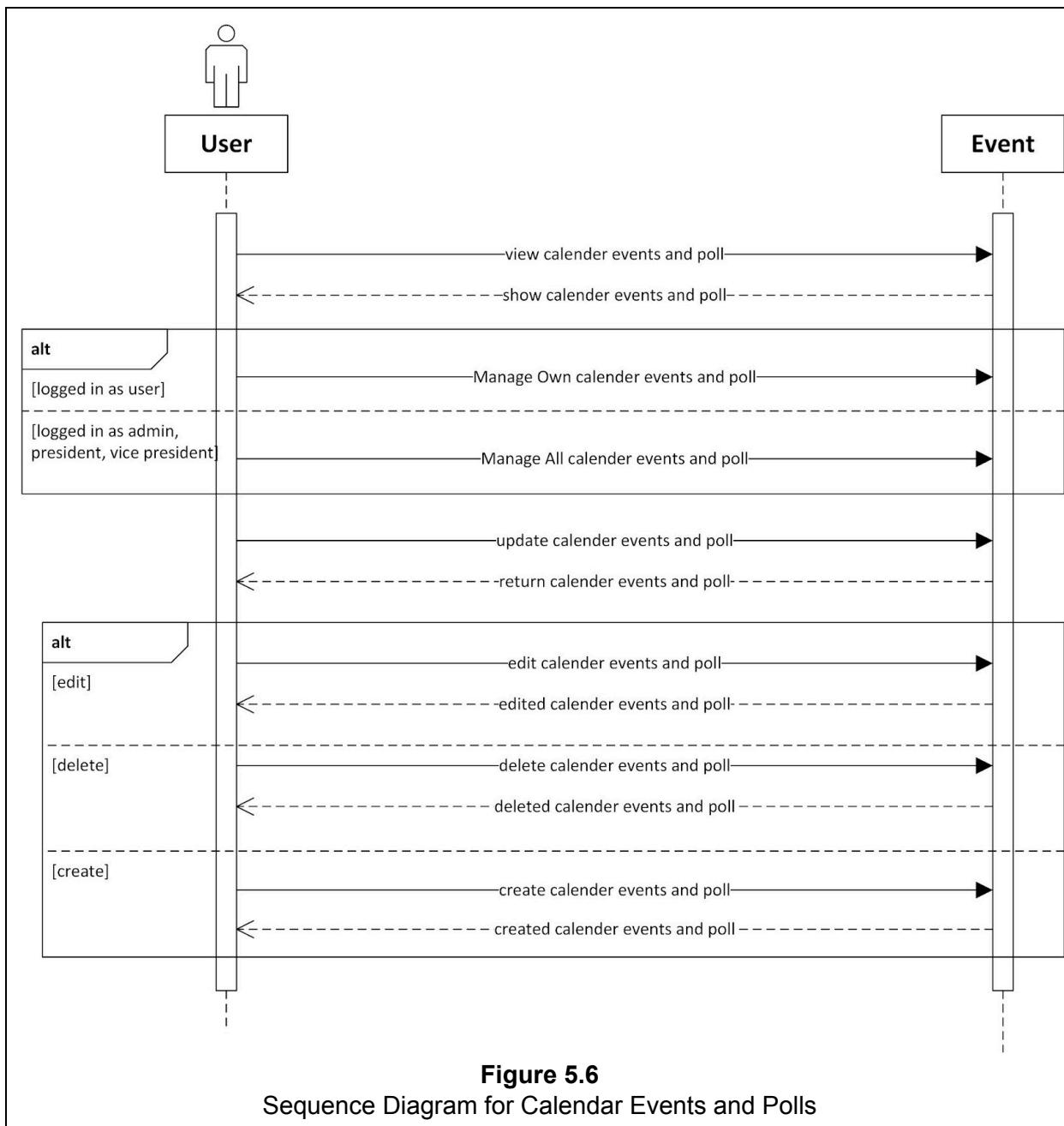
Use Case:	Delete Account
ID:	Figure 5.1
Summary Description:	The system deletes an existing Account.
Preconditions:	<p>The User must be logged in.</p> <p>Users who can only Delete their Accounts:</p> <ul style="list-style-type: none"> <li>• Alumni,</li> <li>• Assistant Treasurer,</li> <li>• Treasurer</li> </ul> <p>Users who can Delete All Accounts:</p> <ul style="list-style-type: none"> <li>• Vice-President</li> <li>• President</li> <li>• Admin</li> </ul> <p>The User must be Managing Accounts</p>
Postconditions:	<p>On success: The User will have Deleted an Account.</p> <p>On Failure: No updates will occur</p>
Primary Actors:	Users excluding those without accounts
Secondary Actors:	
Trigger:	The User clicked on the “Delete Account” button
Main Flow:	<ol style="list-style-type: none"> <li>1. The system shows a “Delete Account” button</li> <li>2. The user clicks the “Delete Account” button</li> <li>3. The system updates the Account page with the deleted Account being removed.           <ul style="list-style-type: none"> <li>○ Return the deleted Account status to the Manage Account Use case to update Account Records.</li> </ul> </li> </ol>
Alternate Flows:	<p>3a. The system fails to delete the Account for some unknown reason.</p> <p>3a.1. The system alerts the user with an error message.</p> <p>3a.2. Go back to step 1.</p>

## 5.2.2 Calendar Page





**Figure 5.5**  
Activity Diagram for Calendar Events and Polls



**Figure 5.6**  
 Sequence Diagram for Calendar Events and Polls

All the actors in the use case diagram are primary actors. Any user, with or without an account, will be able to view calendar events and polls listings as well as manage their own calendar events and polls. The Vice-President, President, and Admin will be able to manage all calendar events and polls, create calendar events and polls, edit calendar events and polls, update calendar events and polls listing, and delete calendar events and polls. The Treasurer, Assistant Treasurer, and Alumni will be able to manage their own calendar events and polls. create calendar events and polls, edit calendar events and polls. Update calendar events and polls listing, and delete calendar events and polls.

Use Case:	View Calendar Events and Polls
ID:	Figure 5.2
Summary Description:	The system will take the User to the Calendar page to view Calendar Events and Polls.
Preconditions:	
Postconditions:	On success: this system will show the Calendar Events and Polls. On Failure: Page fails to load
Primary Actors:	All Users
Secondary Actors:	
Trigger:	The User clicked on the “Calendar” Tab
Main Flow:	<ol style="list-style-type: none"> <li>1. The Use Case starts when the User clicks on the “Calendar” link from the navigation menu.</li> <li>2. The system loads the Calendar Events and Polls Page, which shows the Calendar Events and Polls from newest to oldest.           <ol style="list-style-type: none"> <li>a. 2.1 Extends::Manage Own Calendar Events and Polls</li> </ol> </li> </ol>
Alternate Flows:	

Use Case:	Manage Own Calendar Events and Polls
ID:	Figure 5.2
Summary Description:	The system will take the User to the Manage Calendar Events and Polls page, where they will manage their Events and Polls.
Preconditions:	The User must be logged in.
Postconditions:	On success: this system will update Calendar Events and Polls.  On Failure: No updates will occur
Primary Actors:	Alumni, Assistant Treasurer, Treasurer
Secondary Actors:	
Trigger:	The User clicked on the “manage Calendar Events and Polls” button
Main Flow:	<ol style="list-style-type: none"> <li>1. The user clicks on the “manage Calendar Events and Polls” button.</li> <li>2. The system takes the user to a page where they have two options:                     <ol style="list-style-type: none"> <li>a. 2.1. Extends::Create a new Event or Poll</li> <li>b. 2.2. Extends::Edit Calendar Event or Poll</li> <li>c. 2.3. Extends::Delete an Event or Poll</li> </ol> </li> </ol> <p><b>3. INCLUDE::Update Calendar Events and Polls Listing</b></p>
Alternate Flows:	<ol style="list-style-type: none"> <li>1a. The user attempts to Manage someone else’s Event or Poll.                     <ul style="list-style-type: none"> <li>• 1a.1. The system returns an error message that says, “You do not have permissions to change this Event/Poll.”</li> </ul> </li> <li>3a. The user decides not to manage any events.                     <ul style="list-style-type: none"> <li>• 3a.1. The system returns the user to the Calendar Events and Polls Page.</li> </ul> </li> </ol>

Use Case:	Manage All Calendar Events and Polls
ID:	Figure 5.2
Summary Description:	The system will take the User to the Manage Calendar Events and Polls page.
Preconditions:	<p>The User must be logged in as:</p> <ul style="list-style-type: none"> <li>● Vice-President</li> <li>● President</li> <li>● Admin</li> </ul>
Postconditions:	<p>On success: this system will update Calendar Events and Polls.</p> <p>On Failure: No updates will occur</p>
Primary Actors:	Vice-President, President, Admin
Secondary Actors:	
Trigger:	<p>The User clicked on the “manage Calendar Events and Polls” button</p>
Main Flow:	<ol style="list-style-type: none"> <li>1. The user clicks on the “manage Calendar Events and Polls” button.</li> <li>2. The system takes the user to a page where they have two options:           <ol style="list-style-type: none"> <li>a. 2.1. Extends::Create a new Event or Poll</li> <li>b. 2.2. Extends::Edit Calendar Event or Poll</li> <li>c. 2.3. Extends::Delete an Event or Poll</li> </ol> </li> <li><b>3. INCLUDE::Update Calendar Events and Polls Listing</b></li> </ol>
Alternate Flows:	<p>3a. The user decides not to manage any events.</p> <ul style="list-style-type: none"> <li>● 3a.1. The system returns the user to the Calendar Events and Polls Page.</li> </ul>

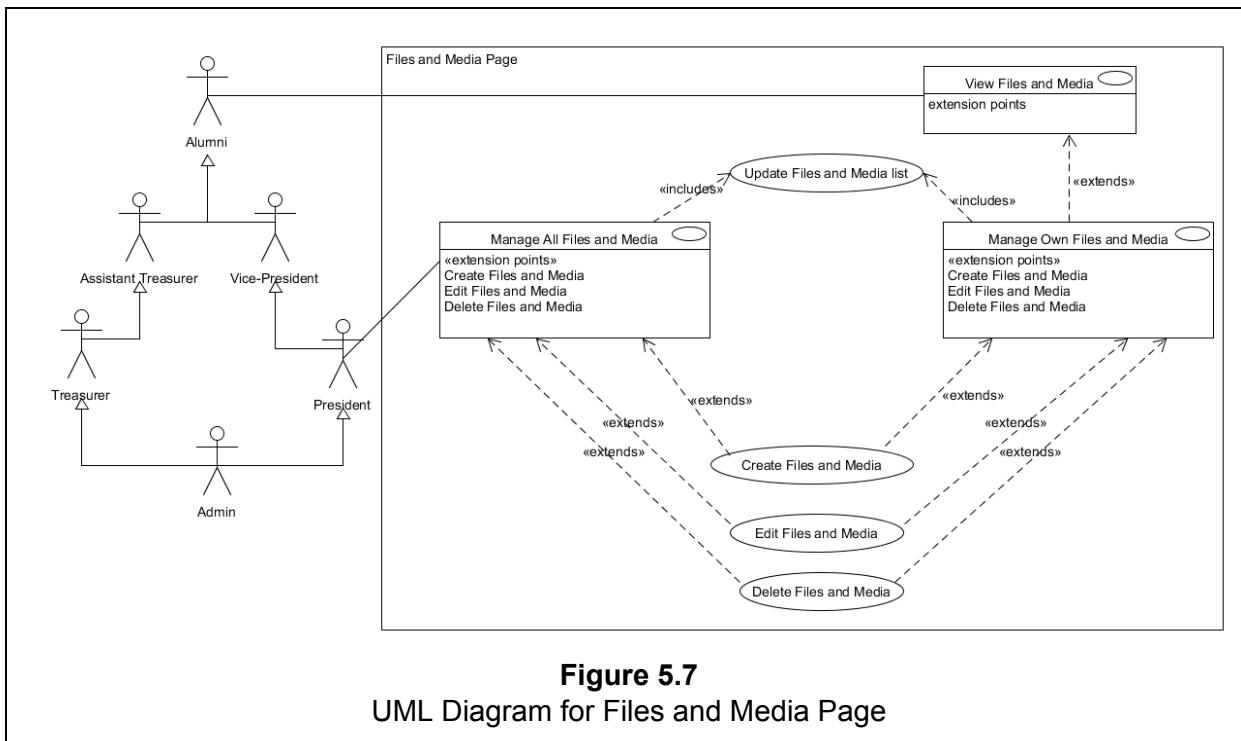
Use Case:	Update Calendar Events and Polls
ID:	Figure 5.2
Summary Description:	The system updates the Calendar Events and Polls listing, shown on the Calendar Events and Polls page.
Preconditions:	
Postconditions:	On success: this system will update Calendar Events and Polls listings.  On Failure: No updates will occur
Primary Actors:	
Secondary Actors:	
Trigger:	The user is finished managing Calendar Events and Polls.
Main Flow:	1. The user has created, edited, or deleted an event or poll. 2. The system updates the Calendar Events and Polls listings on the Events and Polls page.
Alternative Flows:	2a. The system cannot update the Calendar Events and Polls listings. <ul style="list-style-type: none"><li>● 2a.1. The system presents the user with an error message that states, “the Calendar Events and Polls listings cannot be updated at this time.”</li><li>● 2a.2. The system returns the user to the Calendar Events and Polls Page.</li></ul>

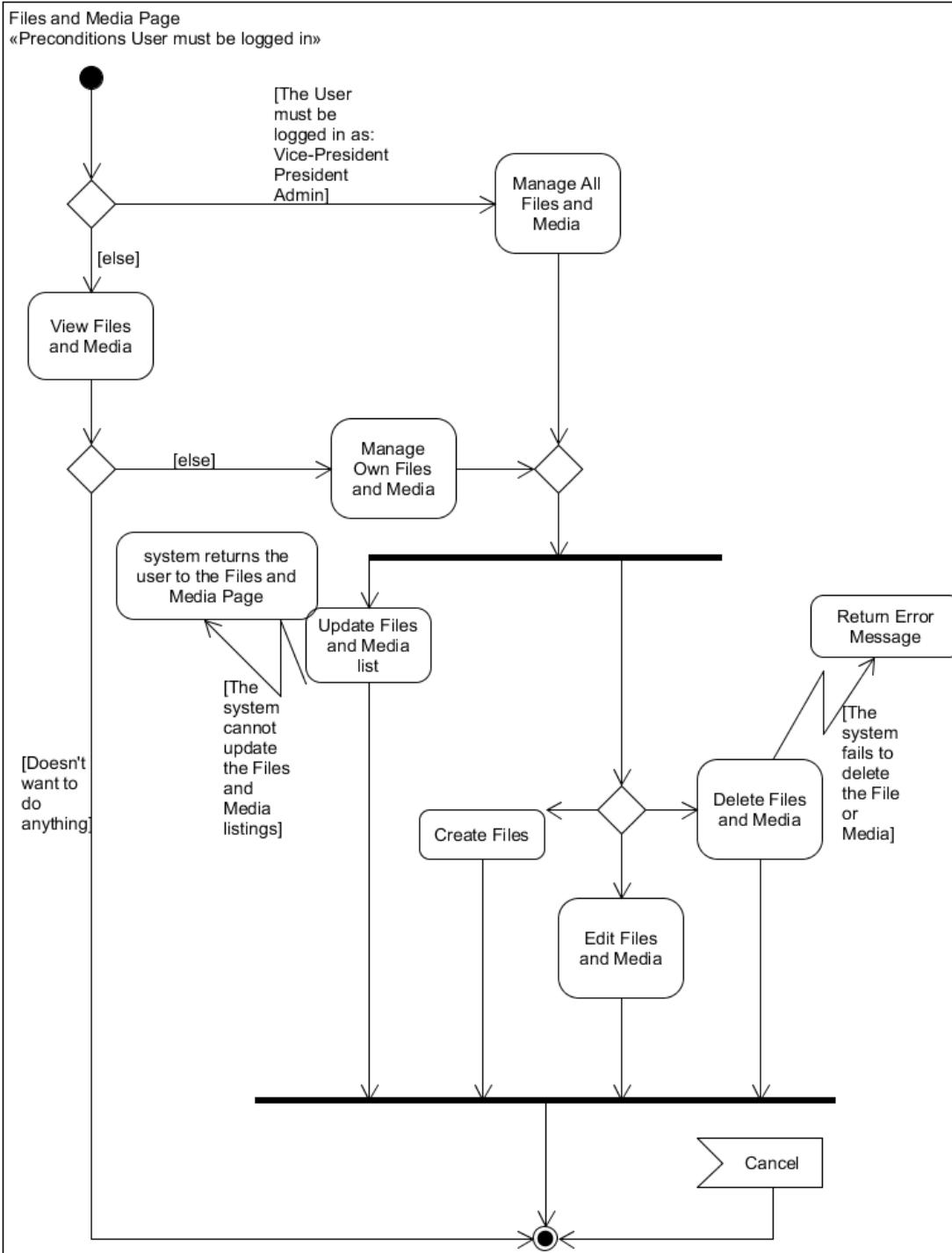
Use Case:	Create Calendar Events and Polls
ID:	Figure 5.2
Summary Description:	The User will create a Calendar Event or Poll. Events and Polls are owned by the person who created them.
Preconditions:	<p>The User must be logged in.</p> <p>The User must be Managing Calendar Events and Polls</p>
Postconditions:	<p>On success: The User will have created an Event or Poll.</p> <p>On Failure: No updates will occur</p>
Primary Actors:	Users excluding those without accounts
Secondary Actors:	
Trigger:	The user clicked the “Create an Event or Poll” button
Main Flow:	<ol style="list-style-type: none"> <li>1. The user clicks the “Create an Event or Poll” button</li> <li>2. The system will load a page to fill in an “Event or Poll” information form.</li> <li>3. The system will also present the user with two buttons on the page: <ul style="list-style-type: none"> <li>o 3.1. "publish Event or Poll" button</li> <li>o 3.2. "cancel" button</li> </ul> </li> </ol>
Alternate Flows:	<p>3a: user clicks the “publish Event or Poll” button.</p> <ul style="list-style-type: none"> <li>• 3a.1. Update Calendar Events and Polls Listing use-case Activates.</li> <li>• 3a.2. The system updates the Updates Event Listings with the published event being posted.</li> </ul> <p>3b: user clicks the “cancel button.”</p> <ul style="list-style-type: none"> <li>• 3b.1. The system returns the user to the Calendar Events and Polls Page</li> </ul>

Use Case:	Edit Calendar Events and Polls
ID:	Figure 5.2
Summary Description:	The User will Edit an Existing Event or Poll
Preconditions:	<p>The User must be logged in.</p> <p>Users who can only Edit their Events or Polls:</p> <ul style="list-style-type: none"> <li>• Alumni,</li> <li>• Assistant Treasurer,</li> <li>• Treasurer</li> </ul> <p>Users who can Edit all Events and Polls:</p> <ul style="list-style-type: none"> <li>• Vice-President</li> <li>• President</li> <li>• Admin</li> </ul> <p>The User must be Managing Calendar Events and Polls</p>
Postconditions:	<p>On success: The User will have edited an Event or Poll.</p> <p>On Failure: No updates will occur</p>
Primary Actors:	Users excluding those without accounts
Secondary Actors:	
Trigger:	The User clicked on the “Edit an Event or Poll” button
Main Flow:	<ol style="list-style-type: none"> <li>1. The system shows an “Edit an Event or Poll” button</li> <li>2. The user clicks the “Edit an Event or Poll” button</li> <li>3. The system will load a page to fill in an “Event or Poll” information form.</li> <li>4. The system will also present the user with two buttons on the page:                     <ol style="list-style-type: none"> <li>a. 4.1. "publish Event or Poll" button</li> <li>b. 4.2. "cancel" button</li> </ol> </li> </ol>
Alternate Flows:	<p>4a: user clicks the “publish Event or Poll” button.</p> <ul style="list-style-type: none"> <li>• 4a.1. Update Event Listings use case kicks in.</li> <li>• 4a.2. The system updates the Updates Event Listings with the published event being posted.</li> </ul> <p>4b: user clicks the “cancel button.”</p> <ul style="list-style-type: none"> <li>• 4b.1. The system returns the user to the Calendar Events and Polls Page</li> </ul>

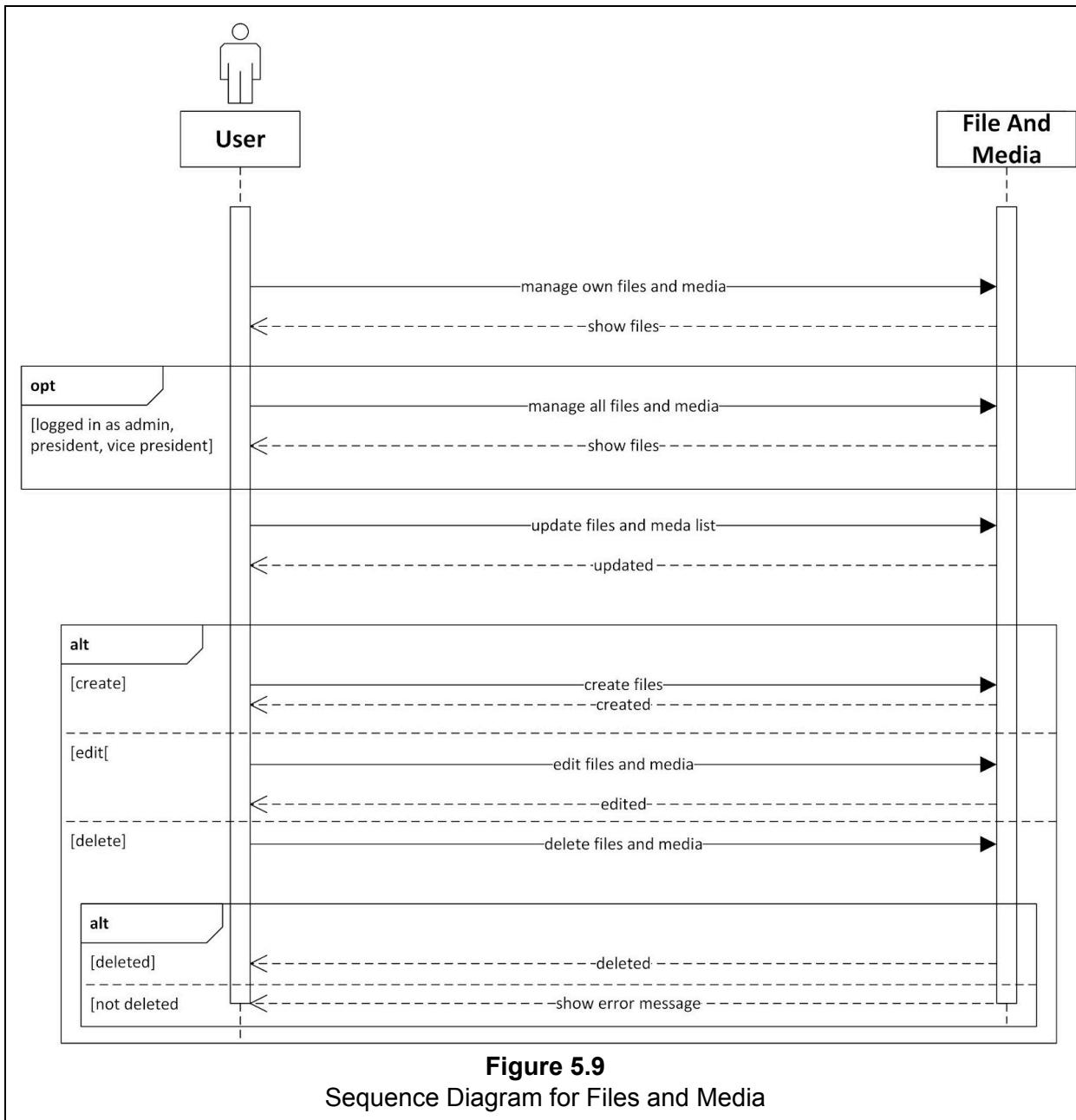
Use Case:	Delete Calendar Events and Polls
ID:	Figure 5.2
Summary Description:	The system deletes an existing Calendar Event or Poll.
Preconditions:	<p>The User must be logged in.</p> <p>Users who can only Delete their Events or Polls:</p> <ul style="list-style-type: none"> <li>• Alumni,</li> <li>• Assistant Treasurer,</li> <li>• Treasurer</li> </ul> <p>Users who can Delete all Events and Polls:</p> <ul style="list-style-type: none"> <li>• Vice-President</li> <li>• President</li> <li>• Admin</li> </ul> <p>The User must be Managing Calendar Events and Polls</p>
Postconditions:	<p>On Success: The chosen existing Event or Poll is deleted.</p> <p>On Failure: The chosen existing Event or Poll is not deleted.</p>
Primary Actors:	Users excluding those without accounts
Secondary Actors:	
Trigger:	User clicks the manageableEvent or Poll" button
Main Flow:	<ol style="list-style-type: none"> <li>1. The system shows a “Delete Event or Poll” button</li> <li>2. The user clicks the “Delete Event or Poll” button</li> <li>3. The system updates the Calendar Events and Polls page, removing the deleted Event or Poll.             <ul style="list-style-type: none"> <li>○ Returns the deleted Event or Poll status to the Manage Calendar Events and Polls Use case, updating the Calendar Events and Polls.</li> </ul> </li> </ol>
Alternate Flows:	<p>3a. The system fails to delete the Event or Poll for unknown reasons.</p> <ul style="list-style-type: none"> <li>• 3a.1. The system alerts the user with an error message.</li> <li>• 3a.2. Go back to step 1.</li> </ul>

### 5.2.3 Files and Media Page





**Figure 5.8**  
Activity Diagram for Files and Media



**Figure 5.9**  
 Sequence Diagram for Files and Media

All users with accounts are primary actors in this use case diagram. Any user with an account will be able to view files and media. The Treasurer, Assistant Treasurer, and Alumni will be able to manage their own files and media which includes the ability to create, edit, update, and delete their files and media. The President and Admin will be able to manage all files and media which includes the ability to create files and media, edit files and media, update files and media list, and delete files and media.

Use Case:	View Files and Media
ID:	Figure 5.3
Summary Description:	The system will take the User to the Files and Media page, where they will View Files and Media.
Preconditions:	
Postconditions:	
Primary Actors:	Users excluding those without accounts
Secondary Actors:	
Trigger:	The User clicked on the “Files and Media” Tab
Main Flow:	<ol style="list-style-type: none"><li>1. The Use Case starts when the User clicks on the “Files and Media” link from their profile menu.</li><li>2. The system loads the Files and Media Gallery, which shows the Files and Media from newest to oldest.</li></ol>
Alternate Flows:	

Use Case:	Manage Own Files and Media
ID:	Figure 5.3
Summary Description:	The system will take the User to the Manage Files and Media page to manage their Files and Media.
Preconditions:	The User must be logged in.
Postconditions:	On success: this system will update Files and Media.  On Failure: No updates will occur
Primary Actors:	Alumni, Assistant Treasurer, Treasurer
Secondary Actors:	
Trigger:	The User clicked on the “manage Files and Media” button
Main Flow:	<ol style="list-style-type: none"> <li>1. The user clicks on the “manage Files and Media” button.</li> <li>2. The system takes the user to a page where they have two options:                     <ul style="list-style-type: none"> <li>o 2.1. Extends::Create/Publish a new File/Media</li> <li>o 2.2. Extends::Edit a File or Media</li> <li>o 2.3. Extends::Delete a File or Media</li> </ul> </li> <li><b>3. INCLUDE::Update Files and Media list</b></li> </ol>
Alternate Flows:	<ol style="list-style-type: none"> <li>1a. The user attempts to Manage someone else’s Files and Media.                     <ul style="list-style-type: none"> <li>• 1a.1. The system returns an error message that says, “You do not have permissions to change these Files and Media.”</li> </ul> </li> <li>3a. The user decides not to manage any events.                     <ul style="list-style-type: none"> <li>• 3a.1. The system returns the user to the Files and Media Page.</li> </ul> </li> </ol>

Use Case:	Manage All Files and Media
ID:	Figure 5.3
Summary Description:	The system will take the User to the Manage Files and Media page.
Preconditions:	The User must be logged in as: <ul style="list-style-type: none"> <li>● Vice-President</li> <li>● President</li> <li>● Admin</li> </ul>
Postconditions:	On success: this system will update Files and Media.  On Failure: No updates will occur
Primary Actors:	Vice-President, President, Admin
Secondary Actors:	
Trigger:	The User clicked on the “manage Files and Media” button
Main Flow:	<ol style="list-style-type: none"> <li>1. The user clicks on the “manage Files and Media” button.</li> <li>2. The system takes the user to a page where they have two options: <ul style="list-style-type: none"> <li>○ 2.1. Extends::Create/Publish a new File/Media</li> <li>○ 2.2. Extends::Edit a File or Media</li> <li>○ 2.3. Extends::Delete a File or Media</li> </ul> </li> <li><b>3. INCLUDE::Update Files and Media list</b></li> </ol>
Alternate Flows:	3a. The user decides not to manage any events. <ul style="list-style-type: none"> <li>● 3a.1. The system returns the user to the Files and Media Page.</li> </ul>

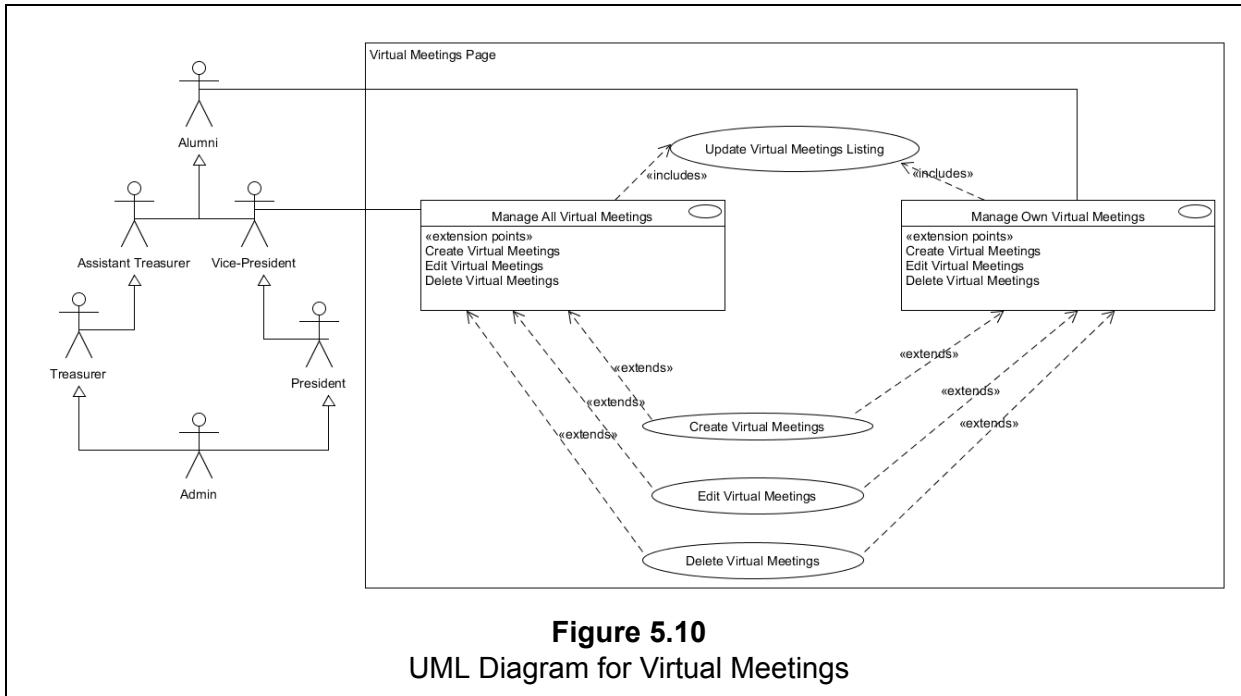
Use Case:	Update Files and Media list
ID:	Figure 5.3
Summary Description:	The system updates the Files and Media listing, which is shown on the Calendar Events and Polls page.
Preconditions:	
Postconditions:	On success: this system will update Files and Media listings. On Failure: No updates will occur
Primary Actors:	
Secondary Actors:	
Trigger:	The user is finished managing Files and Media.
Main Flow:	<ol style="list-style-type: none"><li>1. The user has either created, edited, or deleted a File or Media.</li><li>2. The system now updates the Files and Media listings on the Files and Media page.</li></ol>
Alternative Flows:	<p>2a. The system cannot update the Files and Media listings.</p> <ul style="list-style-type: none"><li>• 2a.1. The system presents the user with an error message that states, “the Files and Media listings cannot be updated at this time.”</li><li>• 2a.2. The system returns the user to the Files and Media Page.</li></ul>

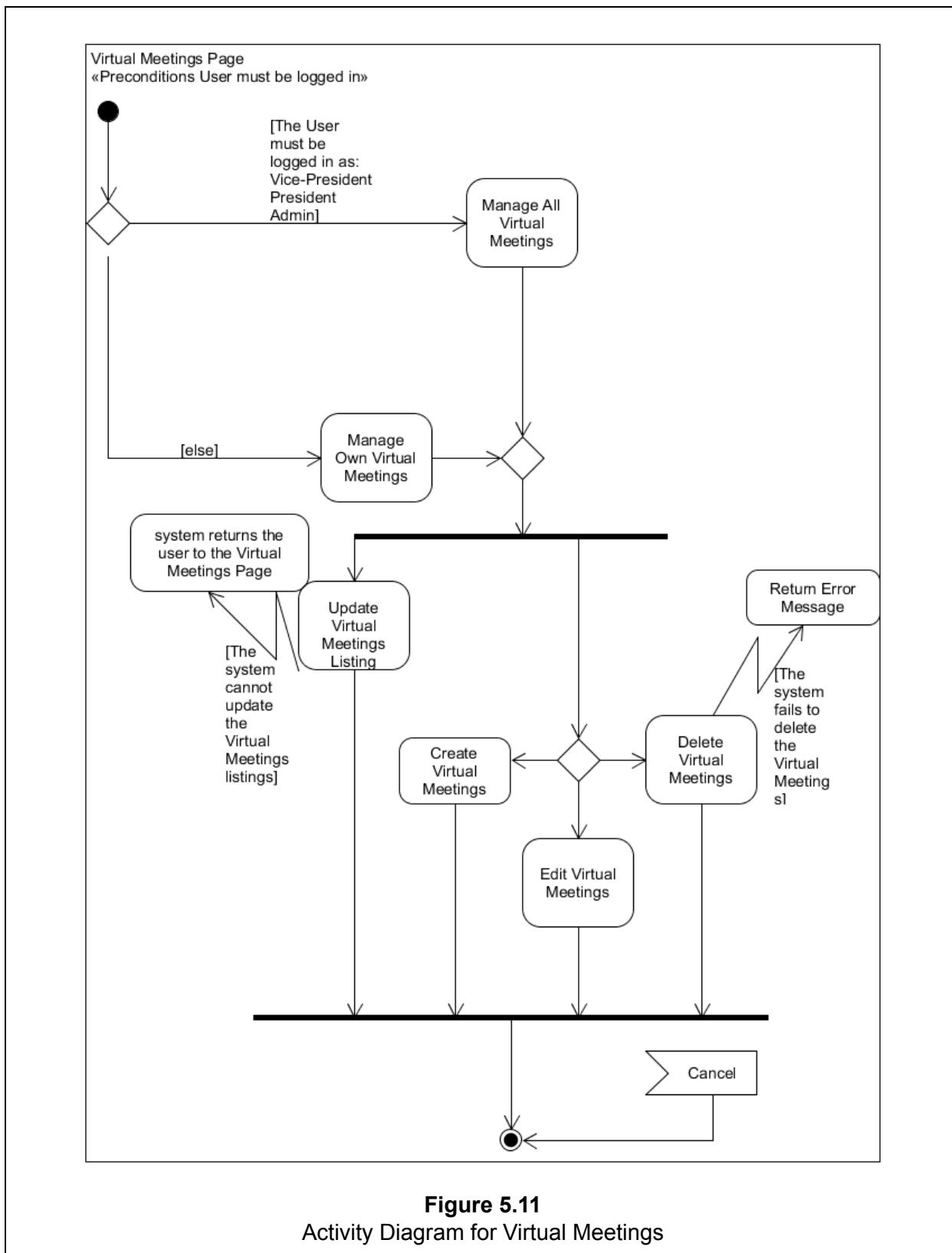
Use Case:	Create Files and Media
ID:	Figure 5.3
Summary Description:	The User will create a File or Media. Files and Media are owned by the person who created them.
Preconditions:	The User must be logged in. The User must be Managing Files and Media
Postconditions:	On success: The User will have created a File or Media. On Failure: No updates will occur
Primary Actors:	Users excluding those without accounts
Secondary Actors:	
Trigger:	The user clicked the “Create a File or Media” button
Main Flow:	<ol style="list-style-type: none"> <li>1. The user clicks the “Create File or Media” button</li> <li>2. The system will load a page where the user can upload a File or Media.</li> <li>3. The system will also present the user with two buttons on the page: <ul style="list-style-type: none"> <li>○ 3.1. "publish File or Media" button</li> <li>○ 3.2. "cancel" button</li> </ul> </li> </ol>
Alternate Flows:	<p>3a: user clicks the “publish File or Media” button.</p> <ul style="list-style-type: none"> <li>● 3a.1. Update Files and Media list use case kicks in.</li> <li>● 3a.2. The system updates the Update Files and Media list with the published File or Media being posted.</li> </ul> <p>3b: user clicks the “cancel button.”</p> <ul style="list-style-type: none"> <li>● 3b.1. The system returns the user to the Files and Media Page</li> </ul>

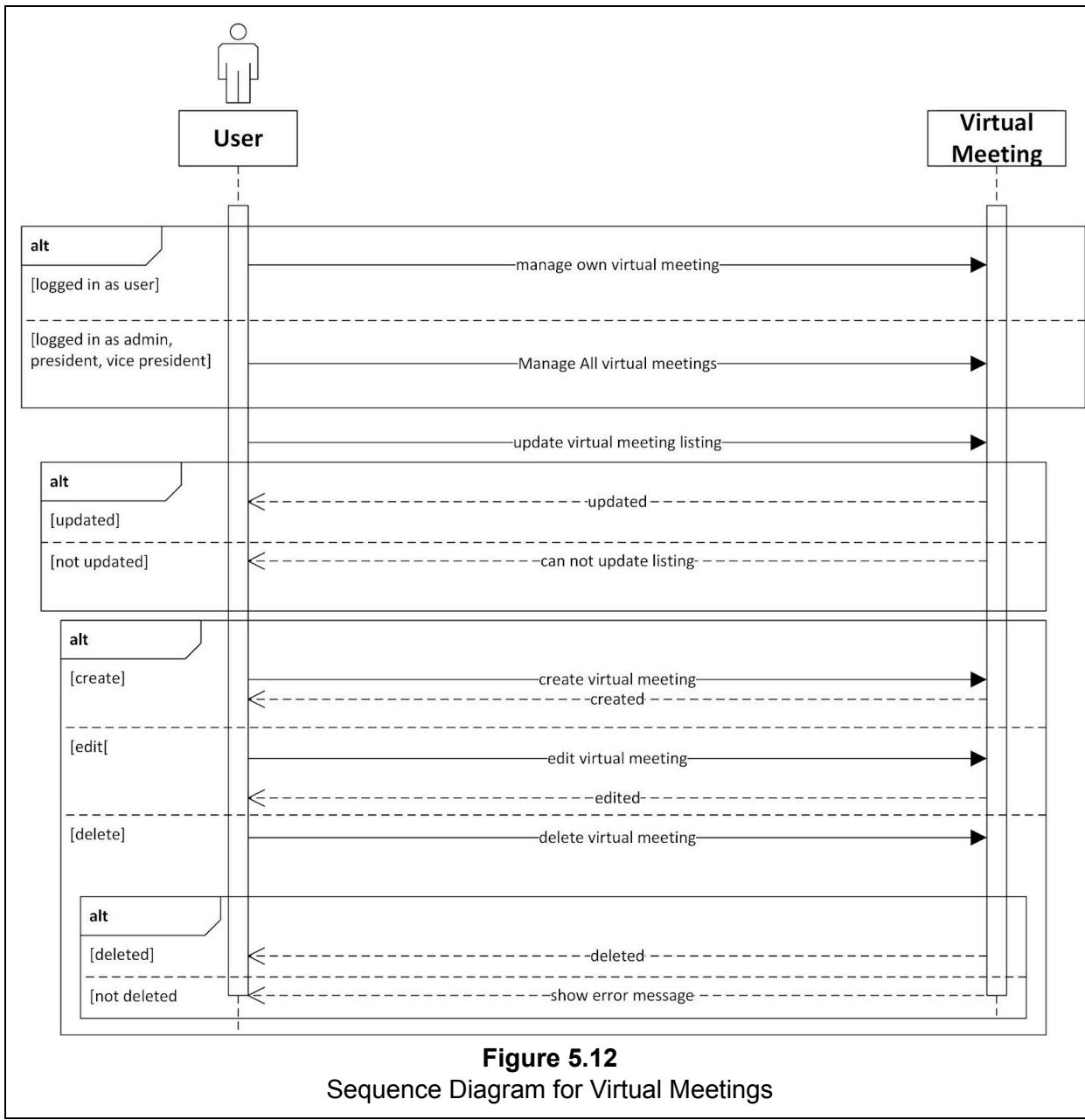
Use Case:	Edit Files and Media
ID:	Figure 5.3
Summary Description:	The User will Edit an Existing File or Media
Preconditions:	<p>The User must be logged in.</p> <p>Users who can only Edit their Files and Media:</p> <ul style="list-style-type: none"> <li>• Alumni,</li> <li>• Assistant Treasurer,</li> <li>• Treasurer</li> </ul> <p>Users who can Edit all Files and Media:</p> <ul style="list-style-type: none"> <li>• Vice-President</li> <li>• President</li> <li>• Admin</li> </ul> <p>The User must be Managing Files and Media</p>
Postconditions:	<p>On success: The User will have edited a File or Media.</p> <p>On Failure: No updates will occur</p>
Primary Actors:	Users excluding those without accounts
Secondary Actors:	
Trigger:	The User clicked on the “Edit a File or Media” button
Main Flow:	<ol style="list-style-type: none"> <li>1. The system shows an “Edit a File or Media” button</li> <li>2. The user clicks the “Edit a File or Media” button</li> <li>3. The system will load a page to fill in a “File or Media” information form.</li> <li>4. The system will also present the user with two buttons on the page:             <ol style="list-style-type: none"> <li>a. 4.1. “publish File or Media” button</li> <li>b. 4.2. “cancel” button</li> </ol> </li> </ol>
Alternate Flows:	<p>4a: user clicks the “publish Event or Poll” button.</p> <ul style="list-style-type: none"> <li>• 4a.1. Update Files and Media list use case kicks in.</li> <li>• 4a.2. The system updates the Update Files and Media list with the published File or Media being posted.</li> </ul> <p>4b: user clicks the “cancel button.”</p> <ul style="list-style-type: none"> <li>• 4b.1. The system returns the user to the Files and Media Page</li> </ul>

Use Case:	Delete Files and Media
ID:	Figure 5.3
Summary Description:	The system deletes an existing File or Media.
Preconditions:	<p>The User must be logged in.</p> <p>Users who can only Delete their Files or Media:</p> <ul style="list-style-type: none"> <li>• Alumni,</li> <li>• Assistant Treasurer,</li> <li>• Treasurer</li> </ul> <p>Users who can Delete all Files or Media:</p> <ul style="list-style-type: none"> <li>• Vice-President</li> <li>• President</li> <li>• Admin</li> </ul> <p>The User must be Managing Files and Media</p>
Postconditions:	<p>On Success: The chosen existing File or Media is deleted.</p> <p>On Failure: The chosen existing File or Media is not deleted.</p>
Primary Actors:	Users excluding those without accounts
Secondary Actors:	
Trigger:	User clicks the “Delete Files and Media” button
Main Flow:	<ol style="list-style-type: none"> <li>1. The system shows a “Delete Files and Media” button</li> <li>2. The user clicks the “Delete Files and Media” button</li> </ol> <p>The system updates the Files and Media page with the deleted File or Media being removed.</p> <ul style="list-style-type: none"> <li>○ Returns the deleted File or Media status to the Manage Files and Media Use case, updating the Files and Media list.</li> </ul>
Alternate Flows:	<p>3a. The system fails to delete the File or Media for some unknown reason.</p> <p>3a.1. The system alerts the user with an error message.</p> <p>3a.2. Go back to step 1.</p>

## 5.2.4 Virtual Meetings Page







**Figure 5.12**  
 Sequence Diagram for Virtual Meetings

All users with an account are primary actors in this use case diagram. This includes the Alumni, Assistant Treasurer, Treasurer, Vice President, President, and the Admin. Alumni will be able to manage their own virtual meetings which include the ability to create virtual meetings, edit virtual meetings, update virtual meetings listing, and delete virtual meetings. The Vice-President, President, and Admin will be able to manage all virtual meetings which include the ability to create virtual meetings, edit virtual meetings, update virtual meeting listings, and delete virtual meetings.

Use Case:	Manage Own Virtual Meetings
ID:	Figure 5.4
Summary Description:	The system will take the User to the Manage Virtual Meetings page to manage their Virtual Meeting.
Preconditions:	The User must be logged in.
Postconditions:	On success: this system will update Virtual Meetings.  On Failure: No updates will occur
Primary Actors:	Alumni, Assistant Treasurer, Treasurer
Secondary Actors:	
Trigger:	The User clicked on the “manage Virtual Meetings” button
Main Flow:	<ol style="list-style-type: none"> <li>1. The user clicks on the “manage Virtual Meetings” button.</li> <li>2. The system takes the user to a page where they have two options: <ul style="list-style-type: none"> <li>○ 2.1. Extends::Create/Publish a new Virtual Meeting</li> <li>○ 2.2. Extends::Edit a Virtual Meeting</li> <li>○ 2.3. Extends::Delete a Virtual Meeting</li> </ul> </li> <li><b>3. INCLUDE::Update Virtual Meeting listing</b></li> </ol>
Alternate Flows:	<p>1a. The user attempts to Manage someone else’s Virtual Meeting.</p> <ul style="list-style-type: none"> <li>● 1a.1. The system returns an error message that says, “You do not have permissions to change these virtual Meetings.”</li> </ul> <p>3a. The user decides not to manage any events.</p> <ul style="list-style-type: none"> <li>● 3a.1. The system returns the user to the Virtual Meetings Page.</li> </ul>

Use Case:	Manage All Virtual Meetings
ID:	Figure 5.4
Summary Description:	The system will take the User to the Manage Virtual Meetings page.
Preconditions:	<p>The User must be logged in as:</p> <ul style="list-style-type: none"> <li>• Vice-President</li> <li>• President</li> <li>• Admin</li> </ul>
Postconditions:	<p>On success: this system will update Virtual Meetings.</p> <p>On Failure: No updates will occur</p>
Primary Actors:	Vice-President, President, Admin
Secondary Actors:	
Trigger:	The User clicked on the “manage Virtual Meetings” button
Main Flow:	<ol style="list-style-type: none"> <li>1. The user clicks on the “manage Virtual Meetings” button.</li> <li>2. The system takes the user to a page where they have two options:             <ul style="list-style-type: none"> <li>○ 2.1. Extends::Create/Publish a new Virtual Meeting</li> <li>○ 2.2. Extends::Edit a Virtual Meeting</li> <li>○ 2.3. Extends::Delete a Virtual Meeting</li> </ul> </li> <li><b>3. INCLUDE::Update Virtual Meeting listing</b></li> </ol>
Alternate Flows:	<ol style="list-style-type: none"> <li>3a. The user decides not to manage any events.             <ul style="list-style-type: none"> <li>● 3a.1. The system returns the user to the Virtual Meetings Page.</li> </ul> </li> </ol>

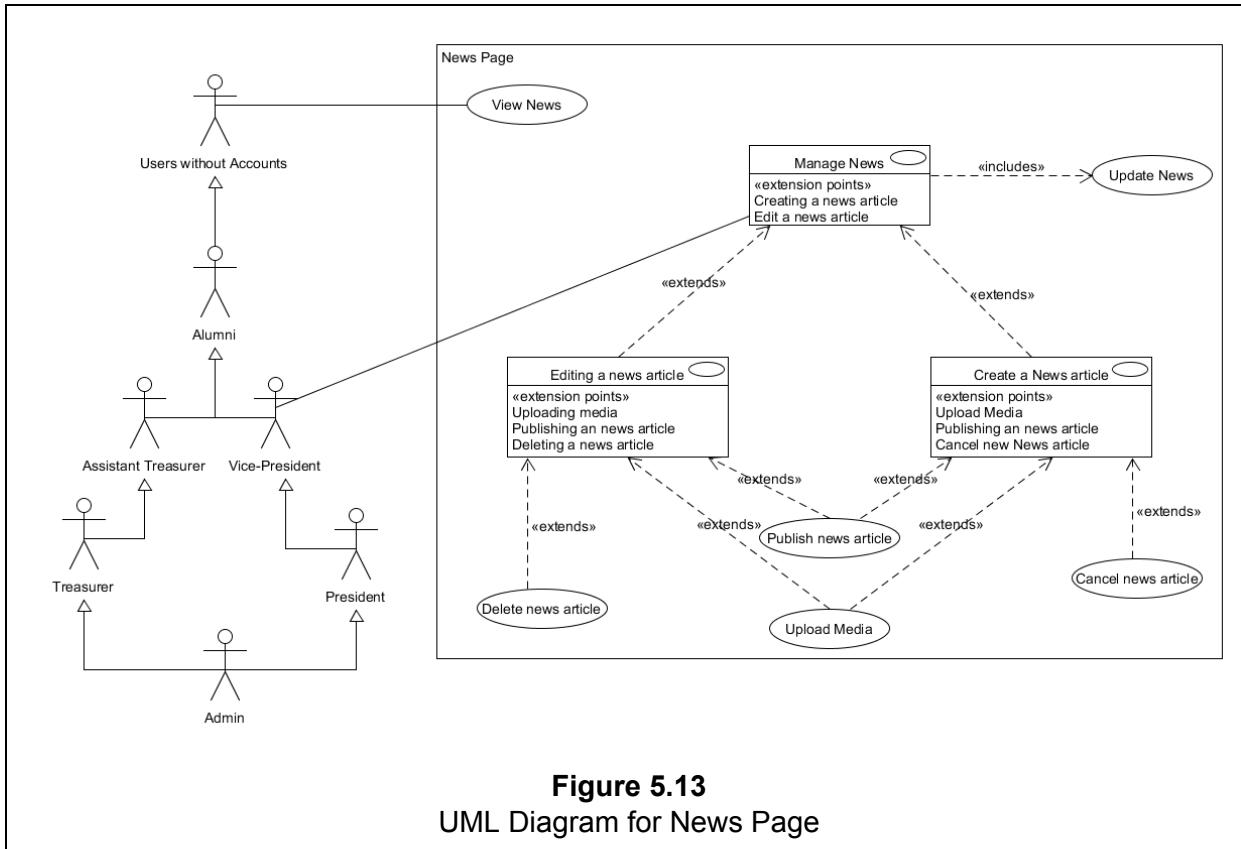
Use Case:	Update Virtual Meetings Listing
ID:	Figure 5.4
Summary Description:	The system updates the Virtual Meetings listing, shown on the Virtual Meetings page.
Preconditions:	
Postconditions:	On success: this system will update Virtual Meetings listings.  On Failure: No updates will occur
Primary Actors:	
Secondary Actors:	
Trigger:	The user is finished managing Virtual Meetings.
Main Flow:	<ol style="list-style-type: none"> <li>1. The user has either created, edited, or deleted a Virtual Meeting.</li> <li>2. The system now updates the Virtual Meetings listings on the Virtual Meetings page.</li> </ol>
Alternative Flows:	<p>2a. The system cannot update the Virtual Meetings listings.</p> <ul style="list-style-type: none"> <li>• 2a.1. The system presents the user with an error message that states, “the Virtual Meetings listings cannot be updated at this time.”</li> <li>• 2a.2. The system returns the user to the Virtual Meetings Page.</li> </ul>

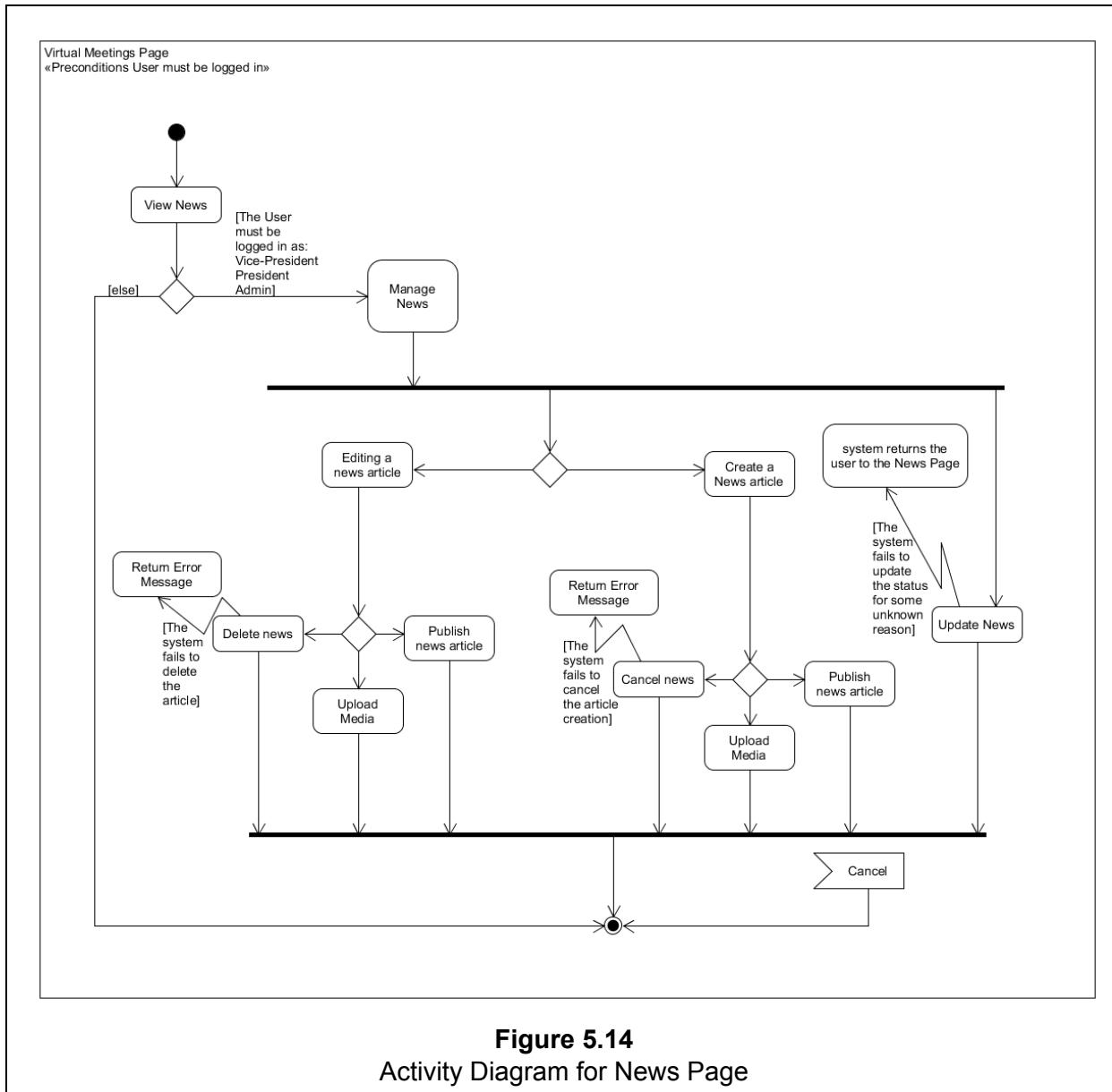
Use Case:	Create Virtual Meetings
ID:	Figure 5.4
Summary Description:	The User will create a Virtual Meeting. Virtual Meetings are owned by the person who created them.
Preconditions:	<p>The User must be logged in.</p> <p>The User must be Managing Virtual Meetings</p>
Postconditions:	<p>On success: The User will have created a Virtual Meeting.</p> <p>On Failure: No updates will occur</p>
Primary Actors:	Users excluding those without accounts
Secondary Actors:	
Trigger:	The user clicked the “Create a Virtual Meeting” button
Main Flow:	<ol style="list-style-type: none"> <li>1. The user clicks the “Create a Virtual Meeting” button</li> <li>2. The system will load a page where the user can fill out an info form about the Virtual Meeting.</li> <li>3. The system will also present the user with two buttons on the page:                     <ul style="list-style-type: none"> <li>o 3.1. “publish Virtual Meeting” button</li> <li>o 3.2. “cancel” button</li> </ul> </li> </ol>
Alternate Flows:	<p>3a: user clicks the “publish Virtual Meeting” button.</p> <ul style="list-style-type: none"> <li>• 3a.1. Update Virtual Meeting listing use case kicks in.</li> <li>• 3a.2. The system updates the Update Virtual Meeting listing with posted Virtual Meeting.</li> </ul> <p>3b: user clicks the “cancel button.”</p> <ul style="list-style-type: none"> <li>• 3b.1. The system returns the user to the Virtual Meeting Page</li> </ul>

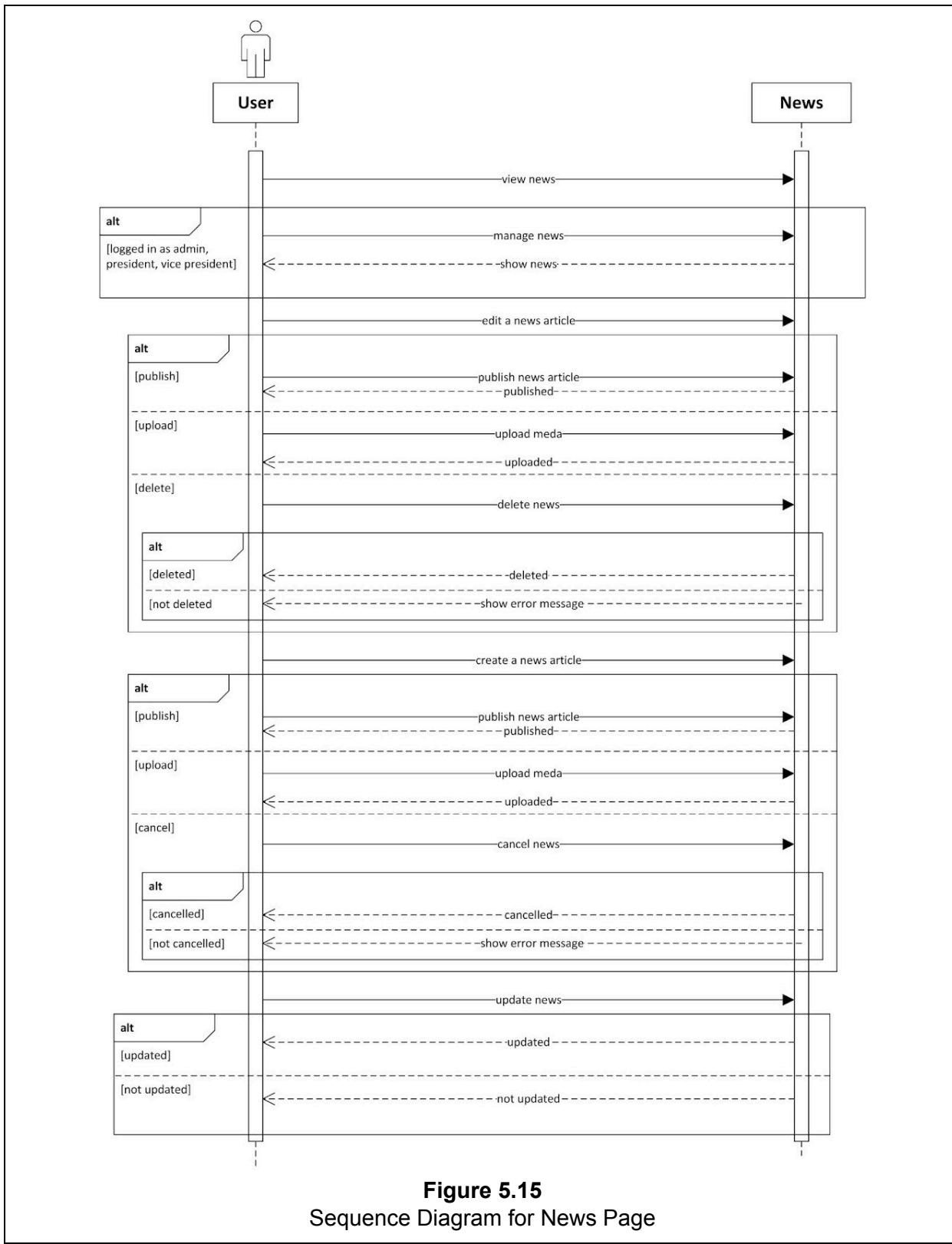
Use Case:	Edit Virtual Meetings
ID:	Figure 5.4
Summary Description:	The User will Edit an Existing Virtual Meeting
Preconditions:	<p>The User must be logged in.</p> <p>Users who can only Edit their Virtual Meetings:</p> <ul style="list-style-type: none"> <li>• Alumni,</li> <li>• Assistant Treasurer,</li> <li>• Treasurer</li> </ul> <p>Users who can Edit all Virtual Meetings:</p> <ul style="list-style-type: none"> <li>• Vice-President</li> <li>• President</li> <li>• Admin</li> </ul> <p>The User must be Managing Virtual Meetings</p>
Postconditions:	<p>On success: The User will have edited a Virtual Meeting.</p> <p>On Failure: No updates will occur</p>
Primary Actors:	Users excluding those without accounts
Secondary Actors:	
Trigger:	The User clicked on the “Edit Virtual Meetings” button
Main Flow:	<ol style="list-style-type: none"> <li>1. The system shows an “Edit Virtual Meetings” button</li> <li>2. The user clicks the “Edit Virtual Meeting” button</li> <li>3. The system will load a page to fill in a “Virtual Meeting” information form.</li> <li>4. The system will also present the user with two buttons on the page:             <ol style="list-style-type: none"> <li>a. 4.1. "publish Virtual Meeting" button</li> <li>b. 4.2. "cancel" button</li> </ol> </li> </ol>
Alternate Flows:	<p>4a: user clicks the “publish Virtual Meeting” button.</p> <ul style="list-style-type: none"> <li>• 4a.1. Update Virtual Meetings listing use case kicks in.</li> <li>• 4a.2. The system updates the Update Virtual Meetings recording with the published Virtual Meeting.</li> </ul> <p>4b: user clicks the “cancel button.”</p> <ul style="list-style-type: none"> <li>• 4b.1. The system returns the user to the Virtual Meeting Page</li> </ul>

Use Case:	Delete Virtual Meetings
ID:	Figure 5.4
Summary Description:	The system deletes an existing Virtual Meeting.
Preconditions:	<p>The User must be logged in.</p> <p>Users who can only Edit their Virtual Meetings:</p> <ul style="list-style-type: none"> <li>• Alumni,</li> <li>• Assistant Treasurer,</li> <li>• Treasurer</li> </ul> <p>Users who can Edit all Virtual Meetings:</p> <ul style="list-style-type: none"> <li>• Vice-President</li> <li>• President</li> <li>• Admin</li> </ul> <p>The User must be Managing Virtual Meetings</p>
Postconditions:	<p>On success: The User will have edited a Virtual Meeting.</p> <p>On Failure: No updates will occur</p>
Primary Actors:	Users excluding those without accounts
Secondary Actors:	
Trigger:	The User clicked on the “Delete Virtual Meetings” button
Main Flow:	<ol style="list-style-type: none"> <li>1. The system shows a “Delete Virtual Meetings” button</li> <li>2. The user clicks the “Delete Virtual Meetings” button</li> </ol> <p>The system updates the Virtual Meetings page, removing the deleted Virtual Meeting.</p> <ul style="list-style-type: none"> <li>○ Returns the deleted Virtual Meeting status to the Manage Virtual Meetings Use case, updating the Virtual Meetings listing.</li> </ul>
Alternate Flows:	<p>3a. The system fails to delete the Virtual Meetings for unknown reasons.</p> <p>3a.1. The system alerts the user with an error message.</p> <p>3a.2. Go back to step 1.</p>

## 5.2.5 News Page







All users are primary actors in this use case diagram. Any user, with or without an account, will be able to view news. The Vice-President. President and Admin will be able to manage and update the news as well as create, publish, edit, delete and cancel a news article.

Use Case:	View News
ID:	Figure 5.5
Summary Description:	The system will take the User to the News page to View News.
Preconditions:	
Postconditions:	
Primary Actors:	All Users
Secondary Actors:	
Trigger:	The User clicked on the “News” Tab
Main Flow:	<ol style="list-style-type: none"><li>1. The Use Case starts when the Alumni clicks on the “News” links from the navigation menu.</li><li>2. The system loads the News Page, Which shows the news from the newest to the oldest.</li></ol>
Alternate Flows:	

Use Case:	Manage News
ID:	Figure 5.5
Summary Description:	The system will take the User to the Manage News page, where they will create a News article or Manage a News article.
Preconditions:	<p>The User must be logged in as:</p> <ul style="list-style-type: none"> <li>• Vice-President</li> <li>• President</li> <li>• Admin</li> </ul> <p>The User must currently be on the News Page.</p>
Postconditions:	The system must Update the News Page reflecting the changes made by the User.
Primary Actors:	Vice-President, President, Admin
Secondary Actors:	
Trigger:	The User clicked on the “Manage News Button” on the News Page.
Main Flow:	<ol style="list-style-type: none"> <li>1. The Use Case starts when the User clicks on the “Manage News Button” on the News Page.</li> <li>2. The system will load a page that will present two options:             <ul style="list-style-type: none"> <li>◦ 2.1 Extends:: Creating a news article</li> <li>◦ 2.2 Extends:: Edit a news article</li> </ul> </li> <li>3. <b>INCLUDE::Update News</b></li> </ol>
Alternative Flows	3a. The user decides not to manage any News articles. <ul style="list-style-type: none"> <li>• 3a.1. The system returns the user to the View News Page.</li> </ul>

Use Case:	Update News
ID:	Figure 5.5
Summary Description:	The system will update the news page with the user's changes.
Preconditions:	
Postconditions:	The system must Update the News Page reflecting the changes made by the User.
Primary Actors:	Vice-President, President, Admin
Secondary Actors:	
Trigger:	The User must be a Managing News.
Main Flow:	<ol style="list-style-type: none"> <li>1. The user has just completed managing news.</li> <li>2. The system then pushes the updates to the news page.</li> </ol>
Alternative Flows	<p>2a. The system fails to update the status for some unknown reason.</p> <ul style="list-style-type: none"> <li>• 2a.1. The system alerts the user with an error message.</li> <li>• 2a.2. The User is taken back to the Manage News page.</li> </ul>

Use Case:	Editing a news article
ID:	Figure 5.5
Summary Description:	The User will Edit an Existing news article
Preconditions:	<p>The User must be logged in as:</p> <ul style="list-style-type: none"> <li>● Vice-President</li> <li>● President</li> <li>● Admin</li> </ul> <p>The User must be Managing News</p>
Postconditions:	<p>On success: The User will have edited a News Article</p> <p>On Failure: No updates will occur</p>
Primary Actors:	Vice-President, President, Admin
Secondary Actors:	
Trigger:	The User clicked on the “Edit News Article” button
Main Flow:	<ol style="list-style-type: none"> <li>1. The system shows an “Edit News Article” button</li> <li>2. The user clicks the “Edit News Article” button</li> <li>3. The system will load a page to fill in a “News Article” information form.</li> <li>4. The system will also present the user with two buttons on the page:                     <ol style="list-style-type: none"> <li>a. 4.1. “publish News Article” button</li> <li>b. 4.2. “cancel” button</li> </ol> </li> </ol>
Alternative Flows	<p>4a: user clicks the “publish News Article” button.</p> <ul style="list-style-type: none"> <li>● 4a.1. Update News use case kicks in.</li> <li>● 4a.2. The system updates the News Page with the published News Page.</li> </ul> <p>4b: user clicks the “cancel button.”</p> <ul style="list-style-type: none"> <li>● 4b.1. The system returns the user to the News Page</li> </ul>

Use Case:	Create a news article
ID:	Figure 5.5
Summary Description:	The User will create a news article
Preconditions:	<p>The User must be logged in as:</p> <ul style="list-style-type: none"> <li>• Vice-President</li> <li>• President</li> <li>• Admin</li> </ul> <p>The User is Managing news articles</p>
Postconditions:	<p>On success: The User will have created a news article.</p> <p>On Failure: No updates will occur</p>
Primary Actors:	Vice-President, President, Admin
Secondary Actors:	
Trigger:	The user clicked the “Create a news article” button
Main Flow:	<ol style="list-style-type: none"> <li>1. The user clicks the “Create a news article” button</li> <li>2. The system will load a page to fill in a “news article” form.</li> <li>3. The system will also present the user with two buttons on the page: <ul style="list-style-type: none"> <li>○ 3.1. “publish news article” button</li> <li>○ 3.2. “cancel” button</li> </ul> </li> </ol>
Alternative Flows	<p>3a: user clicks the “publish news article” button</p> <ul style="list-style-type: none"> <li>• 3a.1. Update News use case kicks in.</li> <li>• 3a.2. The system updates the Update News with the published article being posted.</li> </ul> <p>3b: user clicks the “cancel button.”</p> <ul style="list-style-type: none"> <li>• 3b.1. The system returns the user to the News Page.</li> </ul>

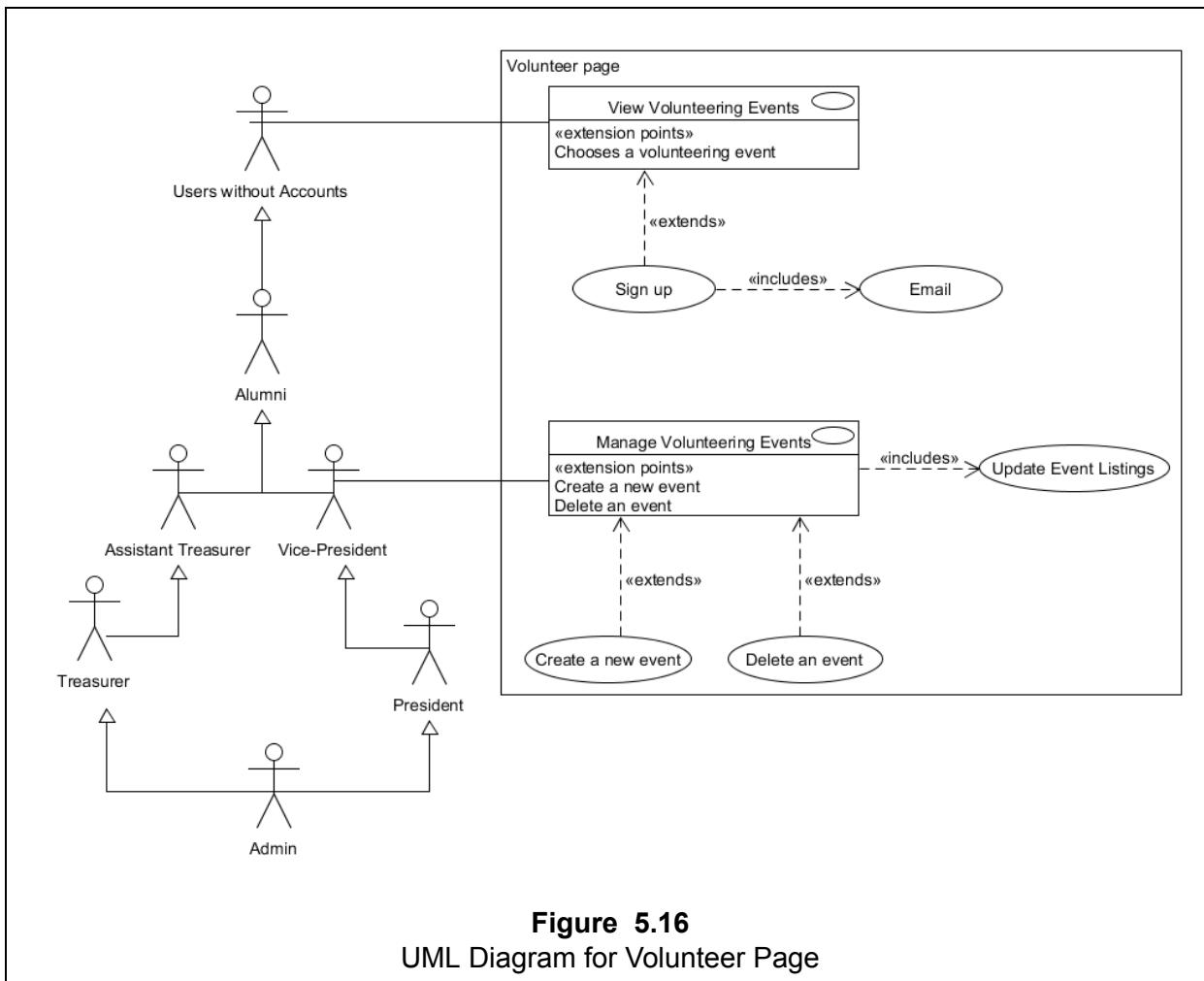
Use Case:	Publish news article
ID:	Figure 5.5
Summary Description:	The system publishes the new/edited news article to the news page.
Preconditions:	<p>The User must be logged in as:</p> <ul style="list-style-type: none"> <li>• Vice-President</li> <li>• President</li> <li>• Admin</li> </ul> <p>Users must be either editing or creating a news article.</p>
Postconditions:	<p>On Success: News article is published.</p> <p>On Failure: News article is not published.</p>
Primary Actors:	Vice-President, President, Admin
Secondary Actors:	
Trigger:	User clicks publish article button
Main Flow:	<ol style="list-style-type: none"> <li>1. The system will show a “Publish News Article” button along the bottom of the news page.</li> <li>2. The User will click “Publish News Article.”</li> <li>3. The system updates the news page with the published news article.             <ul style="list-style-type: none"> <li>○ Returns the publish article status to the Manage News Use case, where it will update the news.</li> </ul> </li> </ol>
Alternative Flows	<p>3a. The system fails to update the status for some unknown reason.</p> <ul style="list-style-type: none"> <li>• 3a.1. The system alerts the user with an error message.</li> <li>• 3a.2. Go back to step 1.</li> </ul>

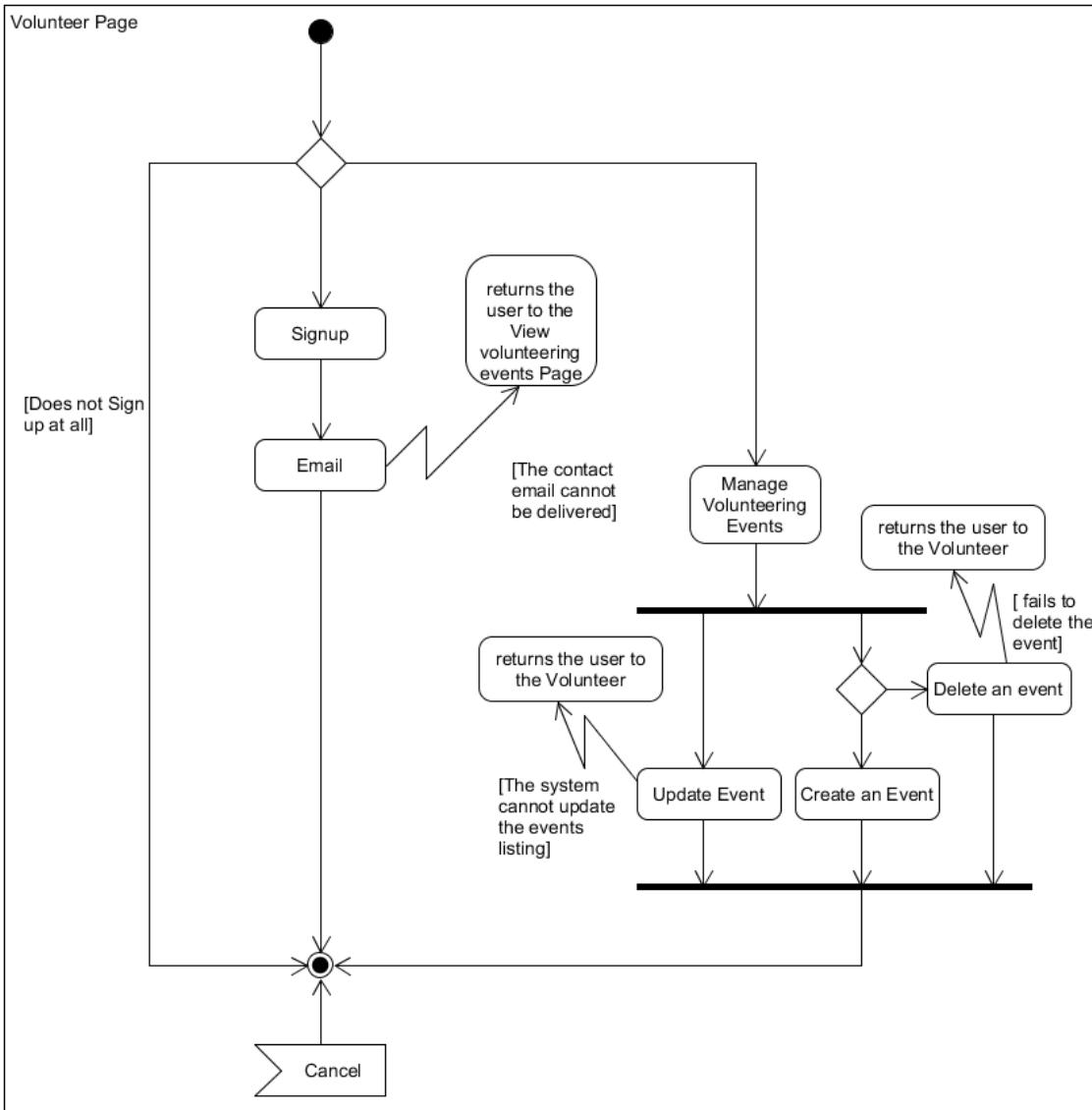
Use Case:	Upload Media
ID:	Figure 5.5
Summary Description:	The system loads a box that allows the user to select media included in the news article.
Preconditions:	<p>The User must be logged in as:</p> <ul style="list-style-type: none"> <li>• Vice-President</li> <li>• President</li> <li>• Admin</li> </ul> <p>Users must be either editing or creating a news article.</p>
Postconditions:	<p>On Success: Media is uploaded and inserted into articles.</p> <p>On Failure: Media is uploaded and inserted into an article.</p>
Primary Actors:	Vice-President, President, Admin
Secondary Actors:	
Trigger:	User clicks the “Insert Media” button
Main Flow:	<ol style="list-style-type: none"> <li>1. The system shows an “Insert Media” button</li> <li>2. The user clicks the “Insert Media” button</li> <li>3. The system loads a media selector in which the user is prompted to select media to be inserted.</li> <li>4. Afterward, the User will choose the media to be inserted and click "Apply."</li> <li>5. Upon selecting the appropriate media, the system will place it in the appropriate section of the article as defined by the User.</li> </ol>
Alternative Flows	<p>5a. The system fails to update the article for some unknown reason.</p> <ul style="list-style-type: none"> <li>• 5a.1. The system alerts the user with an error message.</li> <li>• 5a.2. Go back to step 3.</li> </ul>

Use Case:	Delete new article
ID:	Figure 5.5
Summary Description:	The system deletes an existing news article.
Preconditions:	<p>The User must be logged in as:</p> <ul style="list-style-type: none"> <li>● Vice-President</li> <li>● President</li> <li>● Admin</li> </ul> <p>Users must be editing a news article.</p>
Postconditions:	<p>On Success: Existing article is deleted.</p> <p>On Failure: Existing article is not deleted.</p>
Primary Actors:	Vice-President, President, Admin
Secondary Actors:	
Trigger:	User clicks the “Delete News Article” button
Main Flow:	<ol style="list-style-type: none"> <li>1. The system shows a “Delete News Article” button</li> <li>2. The user clicks the “Delete News Article” button</li> <li>3. The system updates the News page with the deleted article being removed.                     <ul style="list-style-type: none"> <li>○ Returns the deleted article status to the Manage News Use case, where it will update the news.</li> </ul> </li> </ol>
Alternative Flows	<p>3a. The system fails to delete the article for some unknown reason.</p> <ul style="list-style-type: none"> <li>● 3a.1. The system alerts the user with an error message.</li> <li>● 3a.2. Go back to step 1.</li> </ul>

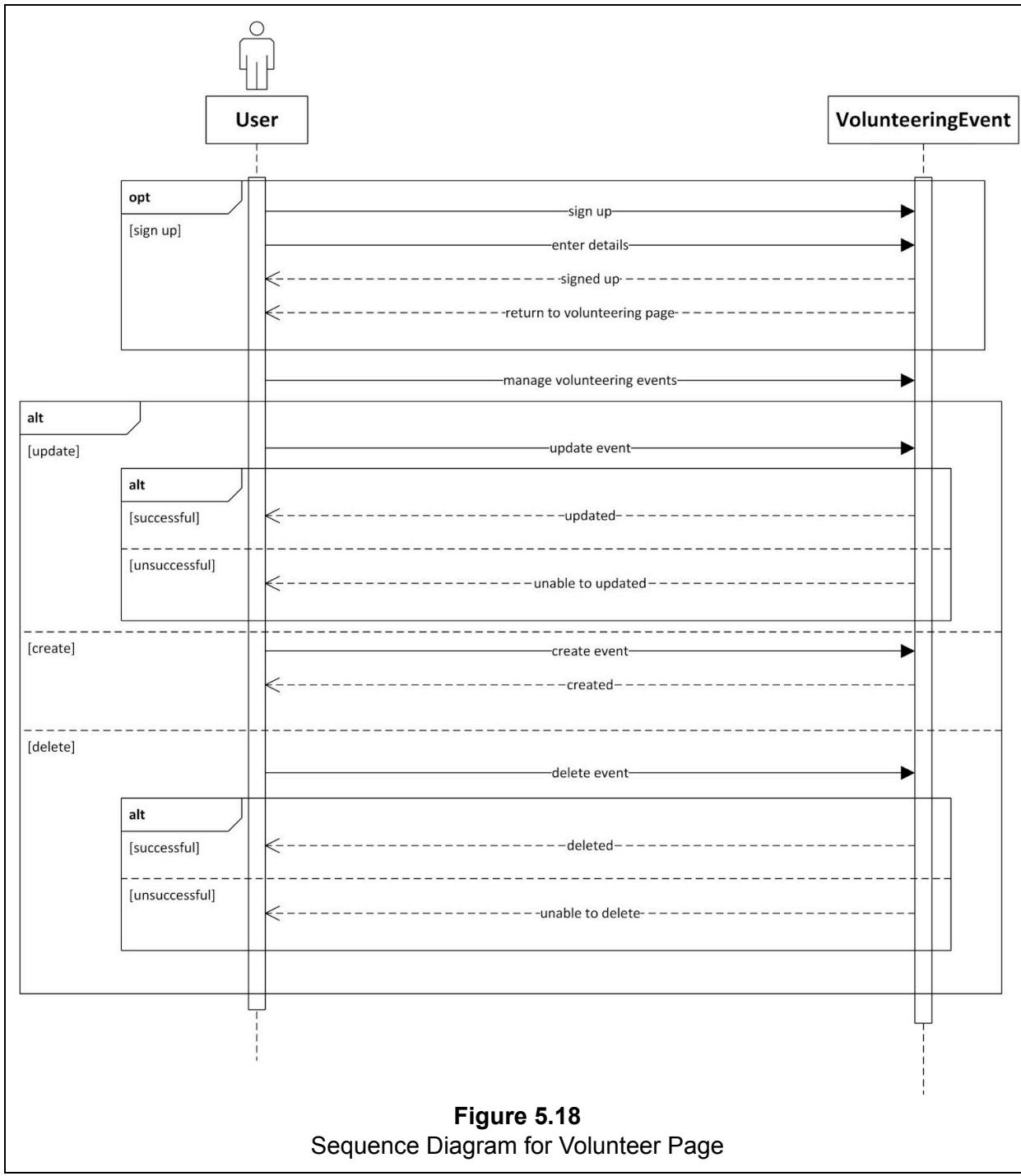
Use Case:	Cancel news article
ID:	Figure 5.5
Summary Description:	The system cancels the creation of a new news article.
Preconditions:	<p>The User must be logged in as:</p> <ul style="list-style-type: none"> <li>• Vice-President</li> <li>• President</li> <li>• Admin</li> </ul> <p>The user must be creating a new news article.</p>
Postconditions:	<p>On Success: The creation of a news article is canceled.</p> <p>On Failure: The creation of a news article is not canceled.</p>
Primary Actors:	Vice-President, President, Admin
Secondary Actors:	
Trigger:	User clicks the “Cancel News Article” button
Main Flow:	<ol style="list-style-type: none"> <li>1. The system shows a “Cancel News Article” button</li> <li>2. The user clicks the Cancel News Article” button</li> <li>3. The system updates the News page, indicating that the canceled article is not uploaded.           <ul style="list-style-type: none"> <li>○ Returns the canceled article status to the Manage News Use case, where it will update the news.</li> </ul> </li> <li>4. The User is taken back to the Manage News page</li> </ol>
Alternative Flows	<p>3a. The system fails to cancel the article creation for unknown reasons.</p> <ul style="list-style-type: none"> <li>• 3a.1. The system alerts the user with an error message.</li> <li>• 3a.2. The User is taken back to the Manage News page</li> </ul>

## 5.2.6 Volunteer Page





**Figure 5.17**  
Activity Diagram for Volunteer Page



**Figure 5.18**  
 Sequence Diagram for Volunteer Page

Any user, with or without accounts, are primary actors in this use case diagram and will be able to view volunteering events and choose events to sign up for and receive emails for. The vice president, President, and Admin will be able to manage volunteering events which include the ability to create a new event, update event listings, and delete an event.

Use Case:	View volunteering events
ID:	Figure 5.6
Summary Description:	The system will take the User to the Volunteer page to view volunteering events.
Preconditions:	
Postconditions:	
Primary Actors:	All Users
Secondary Actors:	
Trigger:	The User clicked on the “Volunteer Tab.”
Main Flow:	<ol style="list-style-type: none"> <li>1. The Use Case starts when the Alumni clicks on the “Volunteer” link from the navigation menu.</li> <li>2. The system loads the Volunteer Page, which shows the Volunteer events from newest to oldest.</li> <li>3. The system shows a “sign-up” button next to each volunteering event. <ul style="list-style-type: none"> <li>o 3.1 Extends::Sign Up</li> </ul> </li> </ol>
Alternative Flows:	3a. Users do not sign up for a volunteering event.

Use Case:	Sign Up
ID:	Figure 5.6
Summary Description:	The system will take the User to a page that will list the positions that the user can choose from.
Preconditions:	The user has picked the volunteering event they want to participate in.
Postconditions:	The user's contact information should be submitted into a form on success.  On Failure: the user gives an error message telling them they cannot sign up for that particular position or event.
Primary Actors:	All Users
Secondary Actors:	
Trigger:	The User clicked on the "sign up" button next to the event listing.
Main Flow:	<ol style="list-style-type: none"> <li>1. The user clicks the sign-up button</li> <li>2. The user is prompted to enter their contact information into a form.</li> <li>3. Once completed, they will click on a button that says "submit."</li> </ol>
Alternative Flows:	3a. The user cancels their registration for the event. <ul style="list-style-type: none"> <li>• 3a.1. The User presses the "Cancel."</li> <li>• 3a.2. The system returns the user to the View volunteering events Page.</li> </ul>

Use Case:	Email
ID:	Figure 5.6
Summary Description:	The system emails the user's contact information to the contact email related to the event.
Preconditions:	The user has signed up for a volunteer position.
Postconditions:	On success: the user's information should be forwarded to the appropriate email related to the event.  On Failure: the system should return a message stating that the email cannot be delivered.
Primary Actors:	
Secondary Actors:	
Trigger:	The user has clicked the “submit” button
Main Flow:	<ol style="list-style-type: none"> <li>1. The system will take the link to the sign-up Publish form and put it in an email.</li> <li>2. This system will then email the link to the form to the contact email related to the event.</li> </ol>
Alternative Flows:	<p>2a. The contact email cannot be delivered.</p> <ul style="list-style-type: none"> <li>• 2a.1. The system gives the user an error message stating that “the contact information cannot be forwarded.”</li> <li>• 2a.2. The system returns the user to the View volunteering events Page.</li> </ul>

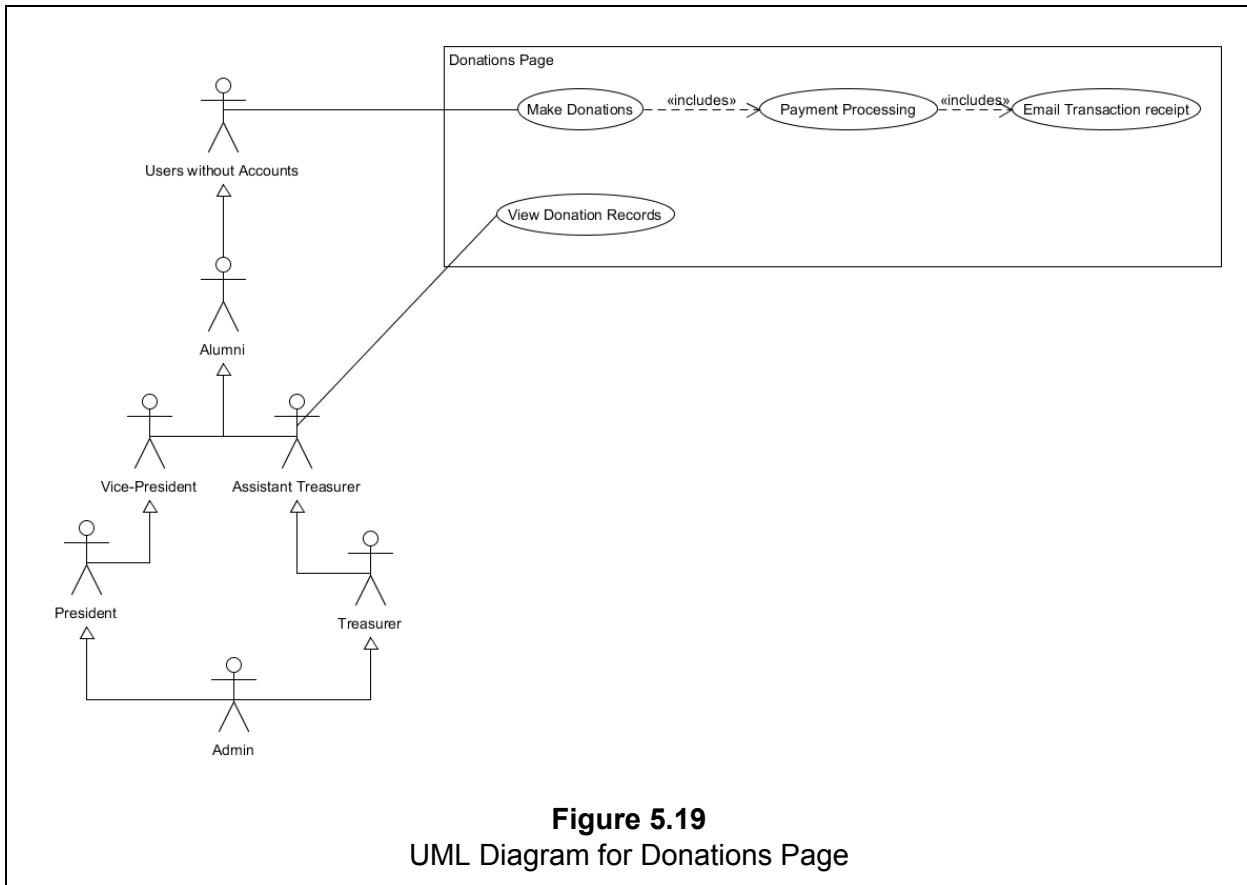
Use Case:	Manage Volunteering Events
ID:	Figure 5.6
Summary Description:	The system will take the User to the Manage Volunteering Events page.
Preconditions:	<p>The User must be logged in as:</p> <ul style="list-style-type: none"> <li>● Vice-President</li> <li>● President</li> <li>● Admin</li> </ul>
Postconditions:	<p>On success: this system will update event listings.</p> <p>On Failure: No updates will occur</p>
Primary Actors:	Vice-President, President, Admin
Secondary Actors:	
Trigger:	The User clicked on the “manage events” button
Main Flow:	<ol style="list-style-type: none"> <li>1. The user clicks on the “manage events” button.</li> <li>2. The system takes the user to a page where they have two options:             <ul style="list-style-type: none"> <li>○ 2.1. Extends::Create a new Event</li> <li>○ 2.2. Extends::Delete an event</li> </ul> </li> <li><b>3. INCLUDE::Update Event Listings</b></li> </ol>
Alternative Flows:	<p>3a. The user decides not to manage any events.</p> <ul style="list-style-type: none"> <li>● 3a.1. The system returns the user to the View volunteering events Page.</li> </ul>

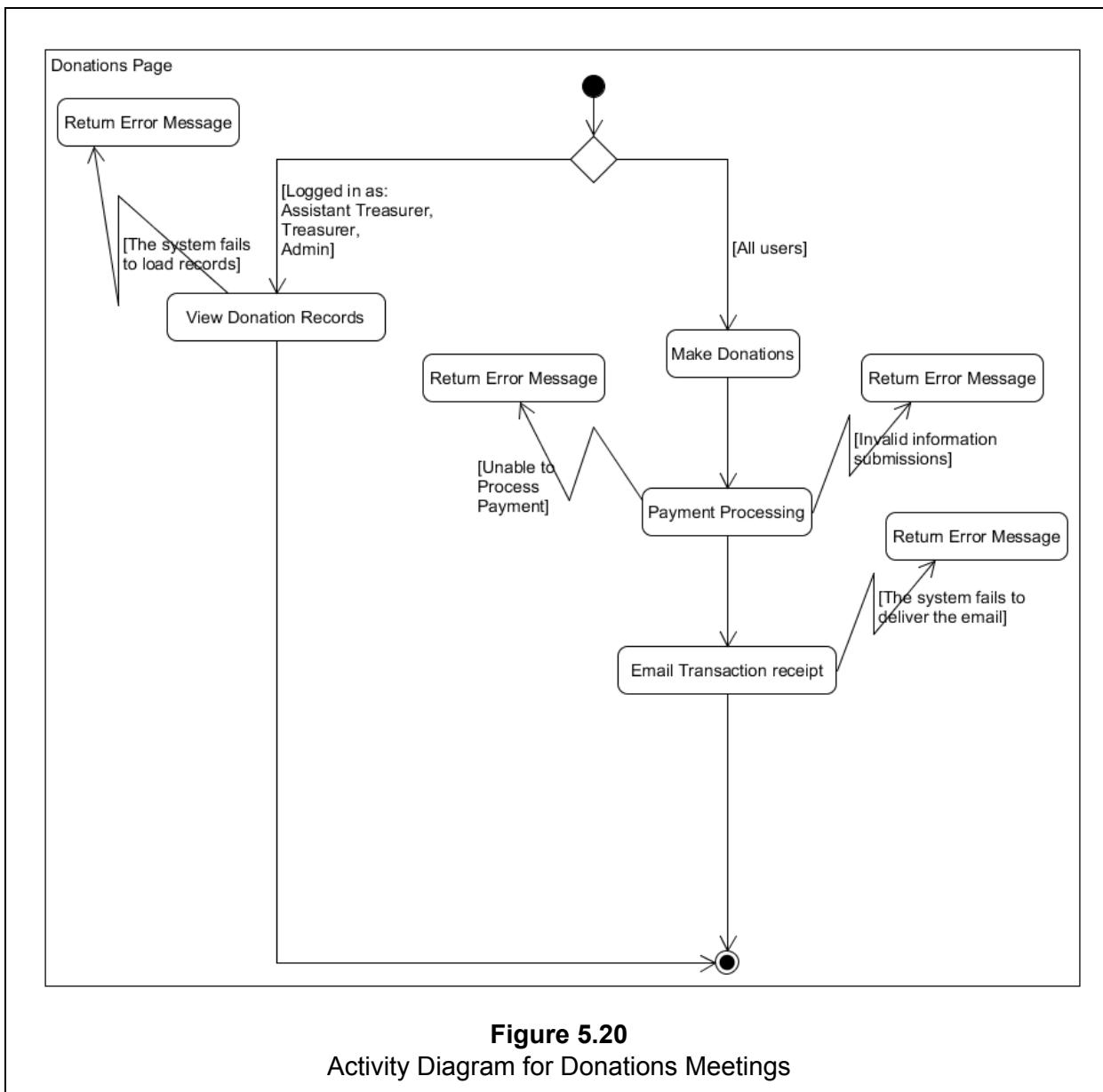
Use Case:	Update Event Listings
ID:	Figure 5.6
Summary Description:	The system updates the event listing, shown on the Volunteer page.
Preconditions:	
Postconditions:	On success: this system will update event listings.  On Failure: No updates will occur
Primary Actors:	Vice-President, President, Admin
Secondary Actors:	
Trigger:	The user is finished managing volunteer events.
Main Flow:	<ol style="list-style-type: none"> <li>1. The user has either created or deleted an event.</li> <li>2. The system now updates the event listings on the volunteer page.</li> </ol>
Alternative Flows:	<p>2a. The system cannot update the events listing.</p> <ul style="list-style-type: none"> <li>• 2a.1. The system presents the user with an error message that states, “the event listings cannot be updated at this time.”</li> <li>• 2a.2. The system returns the user to the Volunteer Page.</li> </ul>

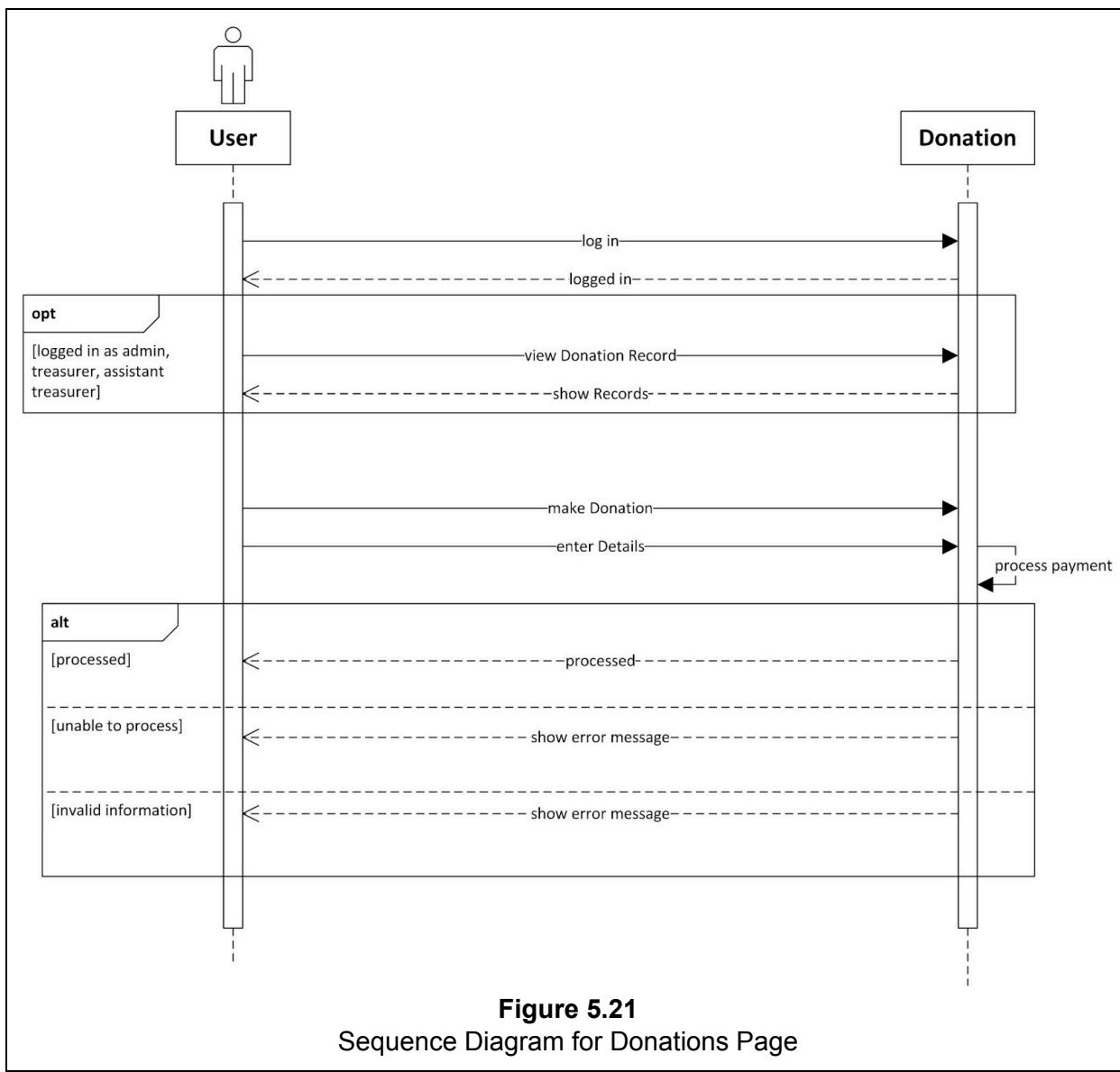
Use Case:	Create a new event
ID:	Figure 5.6
Summary Description:	The User will create a new event.
Preconditions:	<p>The User must be logged in as:</p> <ul style="list-style-type: none"> <li>• Vice-President</li> <li>• President</li> <li>• Admin</li> </ul> <p>The User is Managing Volunteering Events</p>
Postconditions:	<p>On success: The User will have created a new event.</p> <p>On Failure: No updates will occur</p>
Primary Actors:	Vice-President, President, Admin
Secondary Actors:	
Trigger:	The user clicked the “Create a new event” button
Main Flow:	<ol style="list-style-type: none"> <li>1. The user clicks the “Create a new event” button</li> <li>2. The system will load a page to fill in an “event information” form.</li> <li>3. The system will also present the user with two buttons on the page:                     <ul style="list-style-type: none"> <li>○ 3.1. “publish event” button</li> <li>○ 3.2. “cancel” button</li> </ul> </li> </ol>
Alternative Flows:	<p>3a: user clicks the “publish event” button.</p> <ul style="list-style-type: none"> <li>● 3a.1. Update Event Listings use case kicks in.</li> <li>● 3a.2. The system updates the Updates Event Listings with the published event being posted.</li> </ul> <p>3b: user clicks the “cancel button.”</p> <ul style="list-style-type: none"> <li>● 3b.1. The system returns the user to the Volunteer Page.</li> </ul>

Use Case:	Delete an event
ID:	Figure 5.6
Summary Description:	The system deletes an existing event.
Preconditions:	<p>The User must be logged in as:</p> <ul style="list-style-type: none"> <li>• Vice-President</li> <li>• President</li> <li>• Admin</li> </ul> <p>The User is Managing Volunteering Events</p>
Postconditions:	<p>On Success: The current event is deleted.</p> <p>On Failure: The current event is not deleted.</p>
Primary Actors:	Vice-President, President, Admin
Secondary Actors:	
Trigger:	The user clicked the “Delete event” button
Main Flow:	<ol style="list-style-type: none"> <li>1. The user clicks the “Delete event” button</li> <li>2. The system updates the Updates Event Listings with the deleted event being removed.           <ul style="list-style-type: none"> <li>○ Returns the deleted event status to the Manage Volunteering Events Use case, where it will update the news.</li> </ul> </li> </ol>
Alternative Flows:	<p>2a. The system fails to delete the event for some unknown reason.</p> <p>3a.1. The system alerts the user with an error message.</p> <p>3a.2. Go back to the Manage Volunteering Events Use case.</p>

### 5.2.7 Donations Page







**Figure 5.21**  
 Sequence Diagram for Donations Page

Any user, with or without accounts, are the primary actors in this use case diagram and will be able to make donations. The donations system will process the payment and email a transaction receipt. The Assistant Treasurer, Treasurer, and Admin will be able to view donation records.

Use Case:	Make Donations
ID:	Figure 5.7
Summary Description:	The system will take the User to the Donations page, where they will Make Donations.
Preconditions:	
Postconditions:	Users' payments should be processed, and a receipt should be emailed to them.
Primary Actors:	All Users
Secondary Actors:	
Trigger:	The User clicked on the “Donations” Tab
Main Flow:	<ol style="list-style-type: none"> <li>1. The Use Case starts when the Alumni clicks on the “Donations” link from the navigation menu.</li> <li>2. The system loads the Donations Page, which shows the make donation form with a “donate” button.</li> <li>3. The form will list the following donation amounts:           <ul style="list-style-type: none"> <li>o 10\$</li> <li>o 20\$</li> <li>o 50\$</li> <li>o 100\$</li> <li>o Custom amount</li> </ul> </li> <li>4. Once the user selects their amount, they will click the “donate” button to be redirected to the payment info page.</li> <li>5. <b>INCLUDE::Payment Processing</b></li> </ol>
Alternative Flows:	6a. User chooses not to donate and just leaves the page.

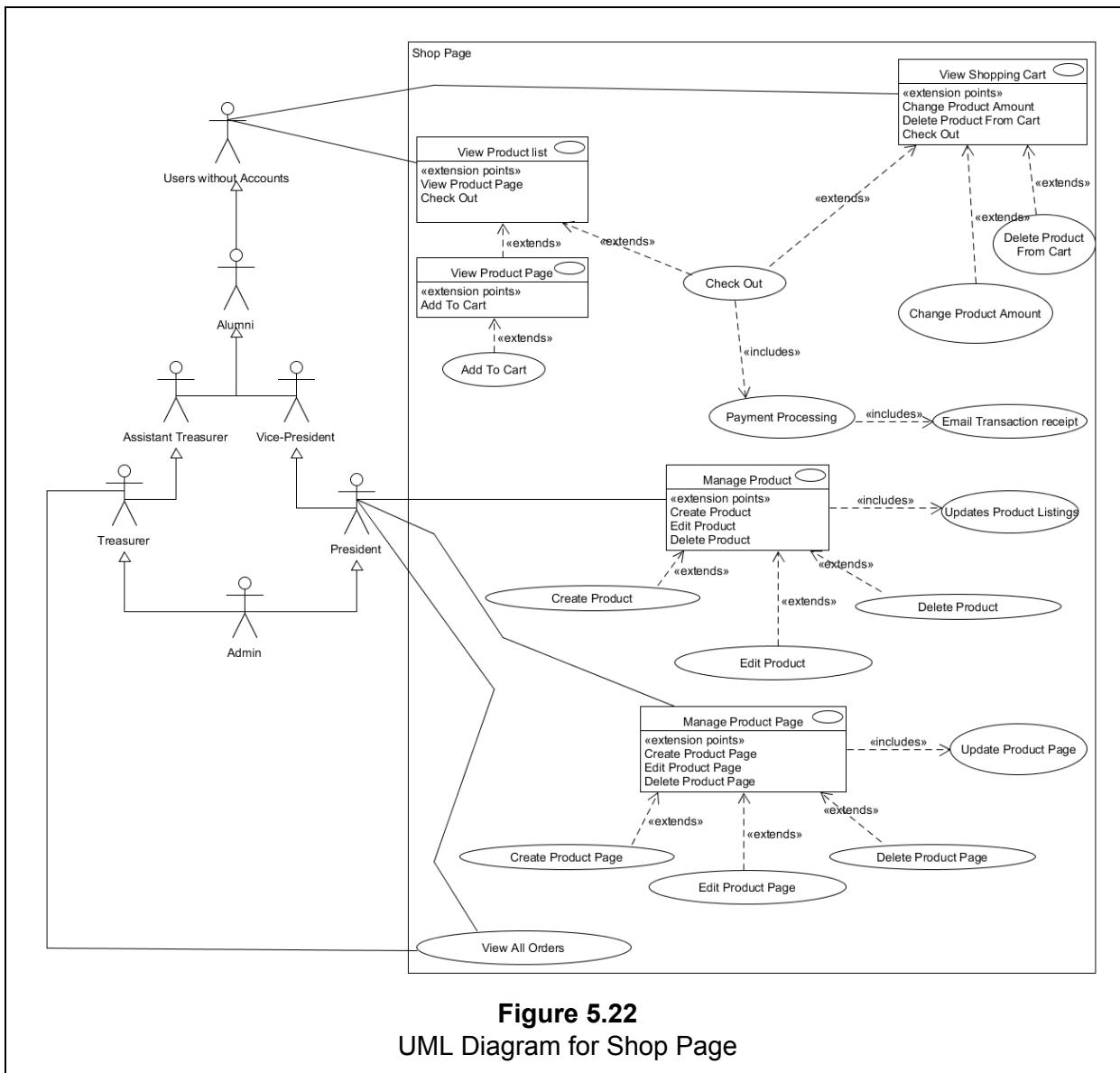
Use Case:	Payment Processing
ID:	Figure 5.7
Summary Description:	For Visa and Mastercard, there will be a form asking for important information regarding payment purchase.
Preconditions:	User clicked the “donate” button
Postconditions:	On success: The users’ payment information should be read, and the user should be charged.  On failure: The user is not charged and is sent back to the make donations page.
Primary Actors:	All Users
Secondary Actors:	
Trigger:	The User clicked on the “donate” button
Main Flow:	<ol style="list-style-type: none"> <li>1. User clicks the “donate” button</li> <li>2. The system loads the “Payment Info Page.”</li> <li>3. User fills out the form on the page</li> <li>4. Upon valid entries for all requirements, the system will navigate to the "Payment receipt" page.</li> <li>5. The system will display, "Thank you for your donation!"</li> <li>6. <b>INCLUDE::Email Transaction receipt</b></li> </ol>
Alternative Flows:	4a. The user does not submit valid entries into the payment form <ul style="list-style-type: none"> <li>• 4a.1. The user does not submit valid name info                 <ul style="list-style-type: none"> <li>◦ 4a.1.1 An invalid entry will return the statement “Please enter Full name as shown on the card.” by the system bolded in red font.</li> </ul> </li> <li>• 4a.2. The user does not submit valid Card info                 <ul style="list-style-type: none"> <li>◦ 4a.2.1 An invalid entry will return the statement “Please enter the Card number.” by the system bolded in red font.</li> </ul> </li> <li>• 4a.3. The user does not submit valid CVV Number info                 <ul style="list-style-type: none"> <li>◦ 4a.3.1 An invalid entry will return the statement “Please enter CVV number.” by the system bolded in red font.</li> </ul> </li> <li>• 4a.4. The user does not submit valid billing address info</li> </ul>

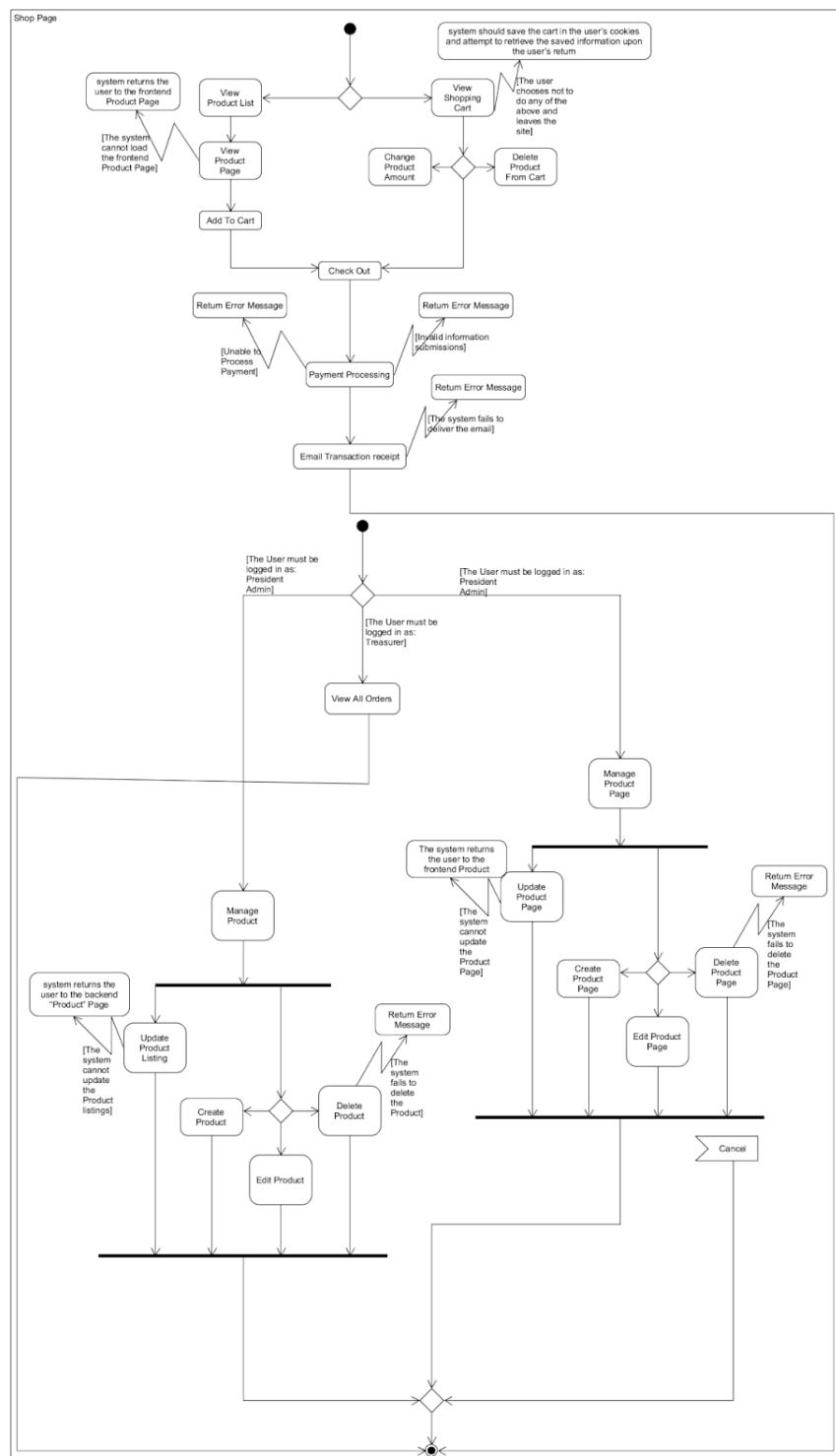
- |  |   |
|--|---|
|  | <ul style="list-style-type: none"><li>○ 4a.4.1 An invalid entry will return the statement “Please enter a billing address.” by the system bolded in red font.</li><li>● 4a.5. The user does not submit valid email info<ul style="list-style-type: none"><li>○ 4a.5.1 An invalid entry will return the statement “Please enter email for transaction receipt” by the system bolded in red font.</li></ul></li></ul> <p>4b. The system cannot process the payment.</p> <ul style="list-style-type: none"><li>● 4b.1. The system alerts the user with an error message.</li><li>● 4b.2. Go back to step 3.</li></ul> <p>5a. The system fails to load the “ThankYou” message</p> <ul style="list-style-type: none"><li>● 5a.1. The system gives an error message and tells the user their receipt is on the way through email.</li></ul> |
|--|---|

Use Case:	Email Transaction receipt
ID:	Figure 5.7
Summary Description:	The system will email a copy of the receipt to the user.
Preconditions:	Payment has been successfully processed.
Postconditions:	On success: the user receives an email copy of the receipt.  On failure: The system sends an error message to the admin, letting them know that the email copy of the receipt could not be delivered.
Primary Actors:	Payment Processing system
Secondary Actors:	
Trigger:	Payment has been successfully processed.
Main Flow:	<ol style="list-style-type: none"><li>1. Payment Processing system sends payment complete status.</li><li>2. The system will then send a copy of the receipt to the email address indicated on the payment form</li></ol>
Alternate Flows:	<ol style="list-style-type: none"><li>2a. The system fails to deliver the email.<ul style="list-style-type: none"><li>• 2a.1. The system sends an error message to the admin, letting them know the receipt could not be delivered.</li><li>• 2a.2. The system logs the failed receipt delivery with a copy of the receipt.</li></ul></li></ol>

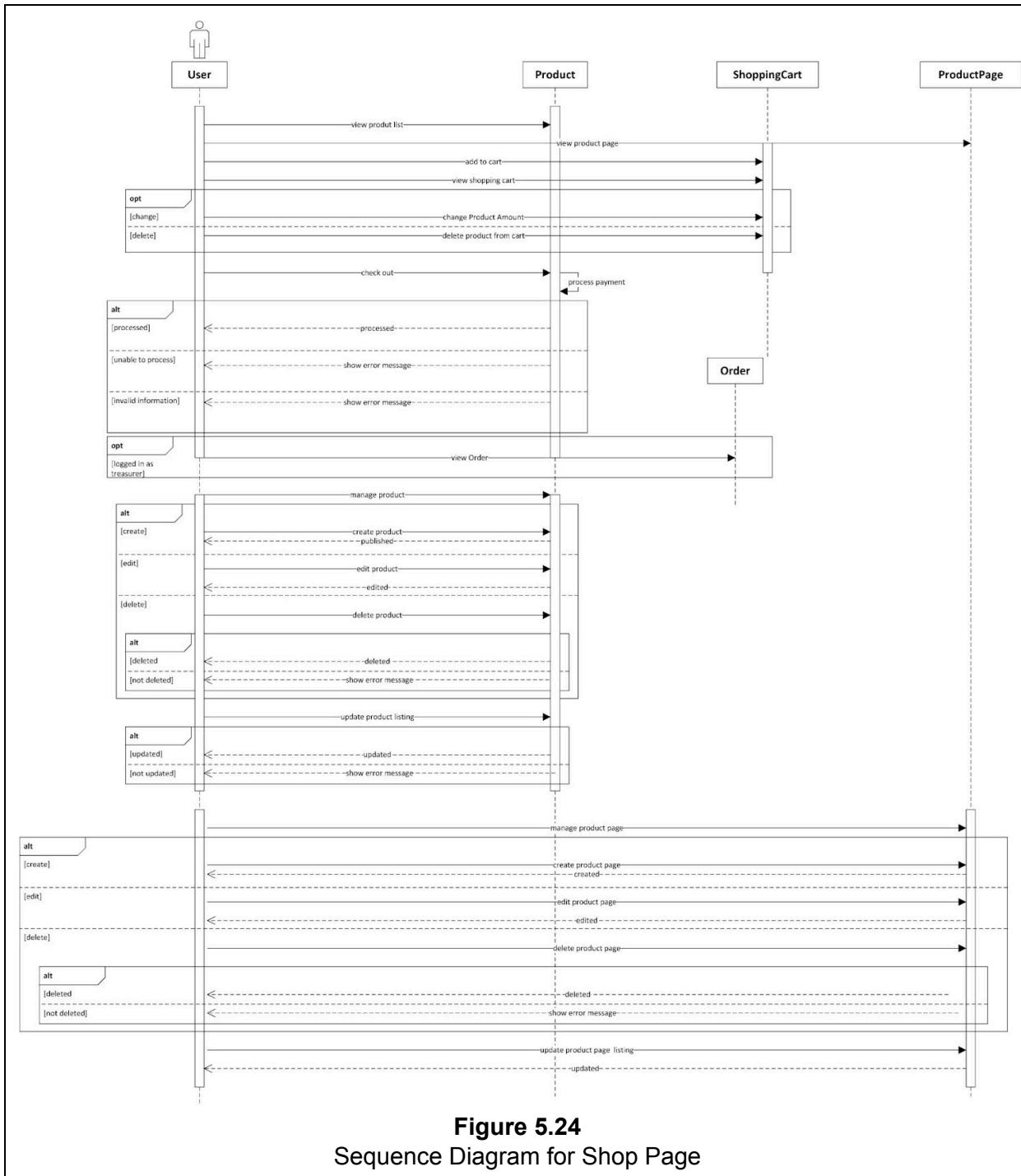
Use Case:	View Donation Records
ID:	Figure 5.7
Summary Description:	The system will take the User to the Donations Record page, where they will View the Donation Records.
Preconditions:	User must be signed in as: <ul style="list-style-type: none"><li>● Assistant Treasurer,</li><li>● Treasurer,</li><li>● Admin</li></ul>
Postconditions:	
Primary Actors:	Assistant Treasurer, Treasurer, Admin
Secondary Actors:	
Trigger:	The User clicks on the “View Records” button.
Main Flow:	<ol style="list-style-type: none"><li>1. The User clicks on the “View Records” button.</li><li>2. The system loads the list of records from newest to oldest.</li></ol>
Alternate Flows:	2a. The system fails to load records. 2a.1. The system sends an error message to the User.

## 5.3 Shop Page





**Figure 5.23**  
Activity Diagram for Shop Page



All the actors in this use case diagram are primary actors. Any user, with or without an account, will be able to view the product list, view the product page, add to cart, and check out as well as view their shopping cart, change the product amount, and delete the product from the cart. The Treasurer, President, and Admin will be able to view all orders. However, only the President and Admin will be able to manage products, create products, edit products, update product listing, and delete products. They will also be able to manage product pages, create product pages, edit product pages, update product pages, and delete product pages.

Use Case:	View Shopping Cart
ID:	Figure 5.8
Summary Description:	The system will open a Shopping Cart page Where the Users will be able to change Product amount, delete Products from their cart, and checkout:
Preconditions:	The User clicked on the shopping cart icon
Postconditions:	The system loads the shopping page where users will browse the Products they are interested in.
Primary Actors:	All Users
Secondary Actors:	
Trigger:	The User clicked on the shopping cart icon
Main Flow:	<ol style="list-style-type: none"><li>1. The Use Case starts when the User clicks on the shopping cart icon from the navigation menu.</li><li>2. The system loads the Shopping Cart page, which allows them to see the Products in their cart as well as change Product amount, delete Products from their cart, and checkout:<ul style="list-style-type: none"><li>o Extends:: Change Product Amount</li><li>o Extends:: Delete Product From Cart</li><li>o Extends:: Check Out</li></ul></li></ol>
Alternate Flows:	<ol style="list-style-type: none"><li>1a. The user chooses not to do any of the above and leaves the site.<ol style="list-style-type: none"><li>1. 1a.1. The system should save the cart in the user's cookies and attempt to retrieve the saved information upon the user's return.</li></ol></li></ol>

Use Case:	Change Product Amount
ID:	Figure 5.8
Summary Description:	The user will change the Product amount on the shopping cart page through a number input box.
Preconditions:	The User clicked on the shopping cart icon
Postconditions:	On success: The Product amount will be changed.  On failure: No updates will occur
Primary Actors:	All Users
Secondary Actors:	
Trigger:	The user clicked with the number input box
Main Flow:	<ol style="list-style-type: none"> <li>1. The Use Case starts when the User clicks on the number input box from the shopping cart page.</li> <li>2. The user will be able to enter any number except for “0.”</li> </ol>
Alternate Flows:	<ol style="list-style-type: none"> <li>2a. The user chooses to enter the number “0.”             <ol style="list-style-type: none"> <li>2. 2a.1. The system should prompt the user with a message that states, “Would you like to remove Product from cart?”</li> <li>3. 2a.2 If the user selects “yes,” the Delete Product From Cart Use case activates.</li> </ol> </li> </ol>

Use Case:	Delete Product From Cart
ID:	Figure 5.8
Summary Description:	On the shopping cart page, the User will be able to Delete a Product From the Cart by clicking on a button with an x.
Preconditions:	The User clicked on the shopping cart icon
Postconditions:	On success: The Product will be removed from the Cart.  On failure: No updates will occur
Primary Actors:	All Users
Secondary Actors:	
Trigger:	The user clicked on the delete button
Main Flow:	<ol style="list-style-type: none"> <li>1. The Use Case starts when the User clicks on the delete from the shopping cart page.</li> <li>2. The system should prompt the user with a message that states, "Would you like to remove Product from cart?"</li> <li>3. The system will present the User with two options:                     <ul style="list-style-type: none"> <li>o 3.1. "Remove Product" button</li> <li>o 3.2. "cancel" button</li> </ul> </li> </ol>
Alternate Flows:	3a: user clicks the "Remove Product" button. <ul style="list-style-type: none"> <li>• 3a.1. The system will remove the Product from the User's cart.</li> <li>• 3a.2. The system updates the Shopping Cart Page with the removed Product being gone.</li> </ul> 3b: user clicks the "cancel button." <ul style="list-style-type: none"> <li>• 3b.1. The system returns the user to the Shopping Cart Page</li> </ul>

Use Case:	Check Out
ID:	Figure 5.8
Summary Description:	<p>There will be a checkout button which, when pressed, will:</p> <ul style="list-style-type: none"> <li>a. Display the items and the quantity of the items in a list</li> <li>b. Have the price of each item displayed next to it (price = Unit price * Amount)</li> <li>c. A total sum of the added item prices will be displayed at the bottom of the list, next to a confirm checkout button.</li> </ul>
Preconditions:	The User's cart has more than 0 items in it.
Postconditions:	<p>On success: The user will be taken to a payment processing page to complete their payment.</p> <p>On failure: The system will return an error and send the user back to the Shopping Cart Page</p>
Primary Actors:	All Users
Secondary Actors:	
Trigger:	The User clicked on the Check-Out Button
Main Flow:	<ol style="list-style-type: none"> <li>1. The Use case starts when the User clicks on the “Check Out Button.”</li> <li>2. The User will be prompted to pay using Paypal or Credit/Debit. <ul style="list-style-type: none"> <li>o 2.1 INCLUDE:: Payment Processing</li> </ul> </li> <li>3. Whichever they choose, they will be taken to the appropriate payment gateway.</li> </ol>
Alternate Flows:	<p>2a. The User picks Paypal</p> <ul style="list-style-type: none"> <li>• 2a.1 The User is taken to the Paypal portal to process their payment.</li> </ul> <p>2b The User chooses Credit/Debit</p> <ul style="list-style-type: none"> <li>• 2b.1 The Payment Processing Use Case kicks in.</li> </ul>

Use Case:	Payment Processing
ID:	Figure 5.8
Summary Description:	For Visa and Mastercard, there will be a form asking for important information regarding payment purchase.
Preconditions:	User clicked the “donate” button
Postconditions:	<p>On success: The users’ payment information should be read, and the user should be charged.</p> <p>On failure: The user is not charged and is sent back to the make donations page.</p>
Primary Actors:	All Users
Secondary Actors:	
Trigger:	The User clicked on the “donate” button
Main Flow:	<ol style="list-style-type: none"> <li>1. User clicks the “donate” button</li> <li>2. The system loads the “Payment Info Page.”</li> <li>3. User fills out the form on the page</li> <li>4. Upon valid entries for all requirements, the system will navigate to the "Payment receipt" page.</li> <li>5. The system will display, "Thank you for your donation!"</li> <li>6. INCLUDE::Email Transaction receipt</li> </ol>
Alternative Flows:	<p>4a. The user does not submit valid entries into the payment form</p> <ul style="list-style-type: none"> <li>• 4a.1. The user does not submit valid name info             <ul style="list-style-type: none"> <li>◦ 4a.1.1 An invalid entry will return the statement “Please enter Full name as shown on the card.” by the system bolded in red font.</li> </ul> </li> <li>• 4a.2. The user does not submit valid Card info             <ul style="list-style-type: none"> <li>◦ 4a.2.1 An invalid entry will return the statement “Please enter the Card number.” by the system bolded in red font.</li> </ul> </li> <li>• 4a.3. The user does not submit valid CVV Number info             <ul style="list-style-type: none"> <li>◦ 4a.3.1 An invalid entry will return the statement “Please enter CVV number.” by the system bolded in red font.</li> </ul> </li> <li>• 4a.4. The user does not submit valid billing address info</li> </ul>

- |  |   |
|--|---|
|  | <ul style="list-style-type: none"><li>○ 4a.4.1 An invalid entry will return the statement “Please enter a billing address.” by the system bolded in red font.</li><li>● 4a.5. The user does not submit valid email info<ul style="list-style-type: none"><li>○ 4a.5.1 An invalid entry will return the statement “Please enter email for transaction receipt” by the system bolded in red font.</li></ul></li></ul> <p>4b. The system cannot process the payment.</p> <ul style="list-style-type: none"><li>● 4b.1. The system alerts the user with an error message.</li><li>● 4b.2. Go back to step 3.</li></ul> <p>5a. The system fails to load the “ThankYou” message</p> <ul style="list-style-type: none"><li>● 5a.1. The system gives an error message and tells the user their receipt is on the way through email.</li></ul> |
|--|---|

Use Case:	Email Transaction receipt
ID:	Figure 5.8
Summary Description:	The system will email a copy of the receipt to the user.
Preconditions:	Payment has been successfully processed.
Postconditions:	On success: the user receives an email copy of the receipt.  On failure: The system sends an error message to the admin, letting them know that the email copy of the receipt could not be delivered.
Primary Actors:	Payment Processing system
Secondary Actors:	
Trigger:	Payment has been successfully processed.
Main Flow:	<ol style="list-style-type: none"><li>1. Payment Processing system sends payment complete status.</li><li>2. The system will then send a copy of the receipt to the email address indicated on the payment form</li></ol>
Alternate Flows:	<ol style="list-style-type: none"><li>2a. The system fails to deliver the email.<ul style="list-style-type: none"><li>• 2a.1. The system sends an error message to the admin, letting them know the receipt could not be delivered.</li><li>• 2a.2. The system logs the failed receipt delivery with a copy of the receipt.</li></ul></li></ol>

Use Case:	View Product list
ID:	Figure 5.8
Summary Description:	The system will take the User to the shop page, where the Product List will be loaded.
Preconditions:	
Postconditions:	The system loads the Product List for the User to view
Primary Actors:	All Users
Secondary Actors:	
Trigger:	The User clicked on the Shop tab
Main Flow:	<ol style="list-style-type: none"><li>1. The Use Case starts when the User clicks on the “Shop ” link from the navigation menu.</li><li>2. The system loads the Product list, which displays the Products on the shop page from newest to oldest.<ul style="list-style-type: none"><li>o 2.1 Extends::View frontend Product Page</li></ul></li></ol>
Alternate Flows:	

Use Case:	View Product Page
ID:	Figure 5.8
Summary Description:	The system will take the user to the frontend Product Page page, where they will have a more detailed view of a particular Product.
Preconditions:	The User has selected a product they wish to view more information about.
Postconditions:	On success: The system loads the frontend Product Page, which gives the User a more detailed view of a particular Product.  On failure: The system displays an error message.
Primary Actors:	All Users
Secondary Actors:	
Trigger:	The User clicked on a Product
Main Flow:	<ol style="list-style-type: none"><li>1. The Use case starts when the user clicks on a Product.</li><li>2. The system will load the front-end Product Page where more specific information about the Product will be available and add the Product to the shopping cart.</li></ol>
Alternate Flows:	<ol style="list-style-type: none"><li>2a. The system cannot load the frontend Product Page<ul style="list-style-type: none"><li>• 2a.1. The system presents the user with an error message that states, “the Product listings cannot be updated at this time.”</li><li>• 2a.2. The system returns the user to the frontend Product Page.</li></ul></li></ol>

Use Case:	Add To Cart
ID:	Figure 5.8
Summary Description:	The system will add the Product to the Users Cart
Preconditions:	The User is viewing the frontend Product Page
Postconditions:	On success: The product is added to the Users Cart On failure: The Product is not added to the Users Cart
Primary Actors:	All Users
Secondary Actors:	
Trigger:	The User clicked on the “Add to Cart” button
Main Flow:	<ol style="list-style-type: none"><li>1. The Use case starts when the user clicks on the “Add to Cart” button.</li><li>2. The system will update the Users Cart with the newly added Product</li></ol>
Alternate Flows:	

Use Case:	Manage Product
ID:	Figure 5.8
Summary Description:	The system will take the User to the backend “Product” Page.
Preconditions:	The User must be logged in as: <ul style="list-style-type: none"> <li>● President</li> <li>● Admin</li> </ul>
Postconditions:	This system will update the backend “Manage Product” Page on success.  On Failure: No updates will occur
Primary Actors:	President, Admin
Secondary Actors:	
Trigger:	The User clicked on the “Manage Product” Page button
Main Flow:	<ol style="list-style-type: none"> <li>1. The user clicks on the backend “Manage Product” Page button.</li> <li>2. The system takes the user to a page where they have two options:                     <ul style="list-style-type: none"> <li>○ 2.1. Extends::Create a new Product</li> <li>○ 2.2. Extends::Edit Product</li> <li>○ 2.3. Extends::Delete a Product</li> </ul> </li> <li>3. INCLUDE::Update Product Listing</li> </ol>
Alternate Flows:	3a. The user decides not to manage any Products. <ul style="list-style-type: none"> <li>● 3a.1. The system returns the user to the backend “Product” Page.</li> </ul>

Use Case:	Updates Product Listings
ID:	Figure 5.8
Summary Description:	The system updates the Product listing, shown on the “Product” page’s backend.
Preconditions:	<p>The User must be logged in as:</p> <ul style="list-style-type: none"> <li>• President</li> <li>• Admin</li> </ul> <p>The User must be Managing Product</p>
Postconditions:	<p>On success: this system will update Product listings.</p> <p>On Failure: No updates will occur</p>
Primary Actors:	President, Admin
Secondary Actors:	
Trigger:	The user is finished managing the Product.
Main Flow:	<ol style="list-style-type: none"> <li>1. The user has either created, edited, or deleted a Product.</li> <li>2. The system now updates the Product Listings on the Shop page.</li> </ol>
Alternative Flows:	<p>2a. The system cannot update the Product listings.</p> <ul style="list-style-type: none"> <li>• 2a.1. The system presents the user with an error message that states, “the Product listings cannot be updated at this time.”</li> <li>• 2a.2. The system returns the user to the backend “Product” Page.</li> </ul>

Use Case:	Create Product
ID:	Figure 5.8
Summary Description:	The User will create a Product.
Preconditions:	<p>The User must be logged in as:</p> <ul style="list-style-type: none"> <li>● President</li> <li>● Admin</li> </ul> <p>The User must be Managing Product</p>
Postconditions:	<p>On success: The User will have created a Product.</p> <p>On Failure: No updates will occur</p>
Primary Actors:	President, Admin
Secondary Actors:	
Trigger:	The user clicked the “Create a Product” button
Main Flow:	<ol style="list-style-type: none"> <li>1. The user clicks the “Create a Product” button</li> <li>2. The system will load a page to fill in a “Product” information form.</li> <li>3. The system will also present the user with two buttons on the page:                     <ul style="list-style-type: none"> <li>○ 3.1. "Publish Product" button</li> <li>○ 3.2. "cancel" button</li> </ul> </li> </ol>
Alternate Flows:	<p>3a: user clicks the "Publish Product" button.</p> <ul style="list-style-type: none"> <li>● 3a.1. Update Product Listing use case kicks in.</li> <li>● 3a.2. The system updates the Product Listings with the Published Product being posted.</li> </ul> <p>3b: user clicks the "cancel button."</p> <ul style="list-style-type: none"> <li>● 3b.1. The system returns the user to the backend “Product” Page</li> </ul>

Use Case:	Edit Product
ID:	Figure 5.8
Summary Description:	The User will Edit an Existing Product
Preconditions:	<p>The User must be logged in as:</p> <ul style="list-style-type: none"> <li>• President</li> <li>• Admin</li> </ul> <p>The User must be Managing a Product</p>
Postconditions:	<p>On success: The User will have edited a Product.</p> <p>On Failure: No updates will occur</p>
Primary Actors:	President, Admin
Secondary Actors:	
Trigger:	The User clicked on the “Edit a Product” button
Main Flow:	<ol style="list-style-type: none"> <li>1. The system shows an “Edit a Product” button</li> <li>2. The user clicks the “Edit a Product” button</li> <li>3. The system will load a page to fill in a “Product” information form.</li> <li>4. The system will also present the user with two buttons on the page: <ul style="list-style-type: none"> <li>○ 4.1. "Publish Product" button</li> <li>○ 4.2. "cancel" button</li> </ul> </li> </ol>
Alternate Flows:	<p>4a: user clicks the “Publish Product” button.</p> <ul style="list-style-type: none"> <li>● 4a.1. Update Product Listings use case kicks in.</li> <li>● 4a.2. The system updates the Product Listings with the Published Product being posted.</li> </ul> <p>4b: user clicks the “cancel button.”</p> <ul style="list-style-type: none"> <li>● 4b.1. The system returns the user to the backend “Product” Page</li> </ul>

Use Case:	Delete Product
ID:	Figure 5.8
Summary Description:	The system deletes an existing Product.
Preconditions:	<p>The User must be logged in as:</p> <ul style="list-style-type: none"> <li>● President</li> <li>● Admin</li> </ul> <p>The User must be Managing a Product</p>
Postconditions:	<p>On Success: The chosen existing Product is deleted.</p> <p>On Failure: The chosen existing Product is not deleted.</p>
Primary Actors:	President, Admin
Secondary Actors:	
Trigger:	User clicks the “Delete Product” button
Main Flow:	<ol style="list-style-type: none"> <li>1. The system shows a “Delete Product” button</li> <li>2. The user clicks the “Delete Product” button</li> <li>3. The system updates the Product Listings with the deleted Product being removed.                     <ul style="list-style-type: none"> <li>○ Returns the deleted Product status to the Manage Product Use case, where it will update the Shop Page.</li> </ul> </li> </ol>
Alternate Flows:	<p>3a. The system fails to delete the Product for some unknown reason.</p> <ul style="list-style-type: none"> <li>● 3a.1. The system alerts the user with an error message.</li> <li>● 3a.2. Go back to step 1.</li> </ul>

Use Case:	Manage Product Page
ID:	Figure 5.8
Summary Description:	The system will take the User to the frontend “Manage Product Page.”
Preconditions:	The User must be logged in as: <ul style="list-style-type: none"> <li>● President</li> <li>● Admin</li> </ul>
Postconditions:	On success: this system will update the frontend Product Page. On Failure: No updates will occur
Primary Actors:	President, Admin
Secondary Actors:	
Trigger:	The User clicked on the “Manage Product Page” button
Main Flow:	<ol style="list-style-type: none"> <li>1. The user clicks on the “Manage Product Page” button.</li> <li>2. The system takes the user to a page where they have two options:           <ul style="list-style-type: none"> <li>○ 2.1. Extends::Create a new Product Page</li> <li>○ 2.2. Extends::Edit Product Page</li> <li>○ 2.3. Extends::Delete a Product Page</li> </ul> </li> <li>4. INCLUDE::Update Product Page</li> </ol>
Alternate Flows:	<ol style="list-style-type: none"> <li>3a. The user decides not to manage the Product Page.           <ul style="list-style-type: none"> <li>● 3a.1. The system returns the user to the frontend Product Page.</li> </ul> </li> </ol>

Use Case:	Update Product Page
ID:	Figure 5.8
Summary Description:	The system updates the Product Page.
Preconditions:	
Postconditions:	On success: this system will update the Product Page.  On Failure: No updates will occur
Primary Actors:	
Secondary Actors:	
Trigger:	The user is finished managing the Product Page.
Main Flow:	<ol style="list-style-type: none"><li>1. The user has either created, edited, or deleted a Product Page.</li><li>2. The system now updates the Product Page</li></ol>
Alternative Flows:	<ol style="list-style-type: none"><li>2a. The system cannot update the Product Page.<ul style="list-style-type: none"><li>• 2a.1. The system presents the user with an error message stating, “the Product Page cannot be updated at this time.”</li><li>• 2a.2. The system returns the user to the frontend Product Page.</li></ul></li></ol>

Use Case:	Create Product Page
ID:	Figure 5.8
Summary Description:	
Preconditions:	<p>The User must be logged in as:</p> <ul style="list-style-type: none"> <li>• President</li> <li>• Admin</li> </ul> <p>The User must be Managing a Product Page</p>
Postconditions:	<p>On success: The User will have created a Product Page.</p> <p>On Failure: No updates will occur</p>
Primary Actors:	Users excluding those without accounts
Secondary Actors:	
Trigger:	The user clicked the “Create a Product Page” button
Main Flow:	<ol style="list-style-type: none"> <li>1. The user clicks the “Create a Product Page” button</li> <li>2. The system will load the Product Page with new options to customize the UI.</li> <li>3. The system will also present the user with two buttons on the page: <ul style="list-style-type: none"> <li>○ 3.1. "Publish Product Page" button</li> <li>○ 3.2. "cancel" button</li> </ul> </li> </ol>
Alternate Flows:	<p>3a: user clicks the “Publish Product Page” button.</p> <ul style="list-style-type: none"> <li>• 3a.1. Update Product Page use case kicks in.</li> <li>• 3a.2. The system updates the Product Page.</li> </ul> <p>3b: user clicks the “cancel button.”</p> <ul style="list-style-type: none"> <li>• 3b.1. The system returns the user to the frontend Product Page</li> </ul>

Use Case:	Edit Product Page
ID:	Figure 5.8
Summary Description:	The User will Edit an Existing Product Page
Preconditions:	<p>The User must be logged in as:</p> <ul style="list-style-type: none"> <li>● President</li> <li>● Admin</li> </ul> <p>The User must be Managing a Product Page</p>
Postconditions:	<p>On success: The User will have edited a Product Page.</p> <p>On Failure: No updates will occur</p>
Primary Actors:	Users excluding those without accounts
Secondary Actors:	
Trigger:	The User clicked on the “Edit a Product Page” button
Main Flow:	<ol style="list-style-type: none"> <li>1. The system shows an “Edit a Product Page” button</li> <li>2. The user clicks the “Edit a Product Page” button</li> <li>3. The system will load the Product Page with new options to customize the UI.</li> <li>4. The system will also present the user with two buttons on the page:                     <ul style="list-style-type: none"> <li>○ 4.1. "Publish Product Page" button</li> <li>○ 4.2. "cancel" button</li> </ul> </li> </ol>
Alternate Flows:	<p>4a: user clicks the “Publish Product” button.</p> <ul style="list-style-type: none"> <li>● 4a.1. Update Product Page use case kicks in.</li> <li>● 4a.2. The system updates the Updates Product Page</li> </ul> <p>4b: user clicks the “cancel button.”</p> <ul style="list-style-type: none"> <li>● 4b.1. The system returns the user to the frontend Product Page</li> </ul>

Use Case:	Delete Product Page
ID:	Figure 5.8
Summary Description:	The system deletes an existing Product Page.
Preconditions:	<p>The User must be logged in as:</p> <ul style="list-style-type: none"> <li>• President</li> <li>• Admin</li> </ul> <p>The User must be Managing a Product Page</p>
Postconditions:	<p>On Success: The chosen existing Product Page Is deleted.</p> <p>On Failure: The chosen existing Product Page Is not deleted.</p>
Primary Actors:	Users excluding those without accounts
Secondary Actors:	
Trigger:	User clicks the “Delete Product Page” button
Main Flow:	<ol style="list-style-type: none"> <li>1. The system shows a “Delete Product Page” button</li> <li>2. The user clicks the “Delete Product Page” button</li> <li>3. The system updates the Product Page Listings with the deleted Product being removed.           <ul style="list-style-type: none"> <li>○ Returns the deleted Product Page status to the Manage Product Page Use case, where it will update the Shop page.</li> </ul> </li> </ol>
Alternate Flows:	<p>3a. The system fails to delete the Product Page for unknown reasons.</p> <ul style="list-style-type: none"> <li>• 3a.1. The system alerts the user with an error message.</li> <li>• 3a.2. Go back to step 1.</li> </ul>

Use Case:	View All Orders
ID:	Figure 5.8
Summary Description:	The system will take the User to the shop Orders page to view All Orders.
Preconditions:	User must be signed in as: <ul style="list-style-type: none"><li>• Treasurer,</li><li>• Admin</li></ul>
Postconditions:	
Primary Actors:	Treasurer, Admin
Secondary Actors:	
Trigger:	The User clicks on the “View All Orders” button.
Main Flow:	<ol style="list-style-type: none"><li>1. The User clicks on the “View All Orders” button.</li><li>2. The system loads the list of Orders from newest to oldest.</li></ol>
Alternate Flows:	<ol style="list-style-type: none"><li>2a. The system fails to load Orders.<ul style="list-style-type: none"><li>• 2a.1. The system sends an error message to the User.</li></ul></li></ol>

## 6 Human-Machine Interface

### 6.1 Interface Design Rules

#### 6.1.1 Registration

In order to proceed with creating an account, the user will be taken to the registration page. In order to complete the process, they must enter the following information:

1. Personal info:

- a. First Name\*
- b. Last Name\*
- c. Student Id\*

2. Account info:

- a. Email Address\*
- b. Password\*
- c. Re-enter password\*

3. Mailing address:

- a. Street Address\*
- b. Street Address 2 (Apt/Unit)
- c. City\*
- d. State\*
- e. Zip Code\*

Additionally, the user must choose two security questions for account authorization whenever requesting account changes (email or password), signing in from an unfamiliar device, or forgetting their password\*

To complete the registration process, all fields marked with an asterisk (\*) must be filled out.

Passwords created by users must satisfy the following criteria.

- One uppercase letter is required
- One number is required

- One special character is required
- Minimum of eight characters

### 6.1.2 Login

The system will take the user to the login page, where they will need to enter their login credentials, which are:

- User Name\*
- Password\*
- Forgot Password (Security Question)

The user will be directed to the main page of the website when he or she enters the correct username and correct password.

In order to log in successfully, the user must complete all fields marked with an asterisk (\*).

A message stating, "incorrect login information, please try again," will appear if the user enters the wrong username or wrong password.

In the event that the user clicks the "Forgot Password" link, the system will direct the user to the "Forgot Password?" page, where they must enter their email address. The user will be directed to the security question page as soon as he or she enters the correct email address. The security question page will ask the user to answer security questions that were defined at the time the user registered. An error message will appear if the user enters the incorrect answer. Upon entering the correct answer, the system will direct the user to the "Reset password" page.

Once the user reaches the password reset page, he or she will be required to enter and re-enter a new password of their choice.

In order for the password to be valid, it must satisfy the following conditions.

- One uppercase letter is required
- One number is required
- One special character is required

- Minimum of eight characters

As long as both the "New password" and the "Re-enter password" match, a message that indicates "The password has been successfully reset" will appear, directing the user to the login page.

In the event that "New password" and "Re-enter password" do not match, a message that says "Please re-enter your new password" will appear.

### 6.1.3 Alumni Profile Page

Alumni can edit and update their profile information on their alumni profile page.

- Images can be uploaded by users for their profile pictures
- A 1:1 aspect ratio will be automatically applied to profile images as well as automatically scaled down to 180 x 180 pixels.
- Users can write "First Name," "Last Name," and "About Me." descriptions.

The status section will allow users to post images and videos in addition to their status updates.

1. There is a text area that states, "Post your status here!".
2. There is an icon for a camera that opens an importer for files
  - a. The file importer will include the following features:
    - i. "Back" button, "Forward" button, and "Up" button
    - ii. A clickable box displaying the file directory will be provided with an adjacent refresh button.
    - iii. There is a search box within the file viewer which searches the current and lower-level folders and displays files that match the search expression
    - iv. There will be a section that displays the devices and opens the devices in a tree-like structure to reveal hidden folders.
    - v. The file viewer provides access to the contents of the folder, and, if the search is active, the contents of nested folders and the current folder.
    - vi. Details about the file
    - vii. File name box

- viii. Type of files being shown for import in the file viewer section. Changing the file type also limits what the file viewer section will show.
  - ix. An import and cancel button
3. By clicking on the post button, the status update will be posted to the Alumni profile page and any accompanying images or videos into the Alumni's gallery.

#### 6.1.4 Donations

The donate page offers users the opportunity to make a contribution using the provided payment gateways.

There will be a message at the top of the donate page that states, "Donations are important for the operation and maintenance of the Maggoty High Alumni Association website.".

Donations may be made in the form of 10\$, 20\$, 50\$, or 100\$, or they may be made in the form of a custom amount (5\$ minimum). The user will be redirected to the payment information page after selecting an option and clicking on the yellow "donate" button.

#### Payment Information Page

- For Visa and Mastercard, there will be a form asking for important information regarding payment purchase.

There will be multiple text boxes that will have the following:

1. A text box for "Please enter Full name as shown on the card."
  - a. An invalid entry will return the same statement by the system bolded in red font.
2. A text box for 16-digit "Please enter the Card number."
3. An invalid entry will return the same message by the system bolded in red font.
4. A text box for 3-digit, "Please enter CVV number."
  - a. An invalid entry will return the same statement by the system bolded in red font.
5. A text box for the billing address, "Please enter a billing address."
  - a. An invalid entry will return the same statement by the system bolded in red font.

6. Upon valid entries for all requirements, the system will navigate to the "Payment receipt" page.

#### Payment receipt

- The system will display, "Thank you for your donation!"

#### 6.1.5 Volunteering

Different categories of volunteering will be available on the Volunteer page, such as:

- Taking part in charitable events (for example, feeding the homeless)
- The volunteer can work at a local educational institution (for example, tutoring kids with a similar interest as the user's specialization).
- Working in hospitals and healthcare facilities as a volunteer

As soon as the user selects a position, they will receive an email that includes all available job openings in the category they selected.

#### 6.1.6 Products and Shopping

A shopping area will be made available for users on the website, where they will be able to view product display pages and manage their shopping baskets.

When users first enter the shopping area through the menu system, they will see the products listed for sale by the Alumni Association.

The display page will show a:

- Products name
- Unit price
- Product image
- Available amount

The product display page will show only the available items available for purchase currently. When the user clicks on the product, it will send them to a detail page that will list the products:

- Product's name
- Unit price
- Product image
- Available amount
- Product Description

There will also be a box where users can select the amount of the product they want. As well as an "add to cart button."

#### Shopping cart management page

The shopping cart management page allows users to manage the products they have selected and added for purchase to their carts.

1. The shopping cart management page will display an array of items that the user clicked "add to cart" on.
2. If the user's cart is empty upon arrival, the page will display the message, "uh oh, looks like your cart is empty, time to go shopping!"
3. If there are items in the user's cart, then it will display the page will display the following information:
  - a. A link to the product display page embedded into the product display name
  - b. Quantity of purchase
  - c. There will be a delete button under each item.
4. If the user clicks the delete button, the page will refresh and update the current list of items available in the cart.
  - a. If the cart is empty, then the page will display the message, "uh oh, looks like your cart is empty, time to go shopping!"
5. There will be a checkout button which, when pressed, will:
  - a. Display the items and the quantity of the items in a list

- b. Have the price of each item displayed next to it (price = Unit price \* Amount)
- c. A total sum of the added item prices will be displayed at the bottom of the list, next to a confirm checkout button.

Once the user presses the confirm checkout button, the system will load the payment info page.

### Payment Info Page

- For Visa, Amex, and Mastercard, there will be a form asking for important information regarding payment purchase.

There will be multiple text boxes that will have the following:

1. A text box for the email address, "Please enter email for transaction receipt "
  - a. An invalid entry will return the same statement by the system bolded in red font.
2. A text box for "Please enter Full name as shown on the card."
  - a. An invalid entry will return the same statement by the system bolded in red font.
3. A text box for 16-digit "Please enter the Card number."
4. An invalid entry will return the same message by the system bolded in red font.
5. A text box for 3-digit, "Please enter CVV number."
  - a. An invalid entry will return the same statement by the system bolded in red font.
6. A text box for the billing address, "Please enter a billing address."
  - a. An invalid entry will return the same statement by the system bolded in red font.
7. Upon valid entries for all requirements, the system will navigate to the "Payment receipt" page.

### Payment receipt

- The system will display, "Thank you for your purchase! We hope you enjoy your products!"

### 6.1.7 Contact Us

The users will be able to contact the Association by clicking the "Contact Us" button, which will lead them to a contact page that asks for their:

- Email Address text box
- Subject text box
- Question or query text box

Once the user fills in this information and presses send, the page will display a message that says, "You will receive a response in 2-3 business days," with the help desk operational hours.

### 6.1.8 News

The News link will take the users to a page where users will be able to either:

- View event details and updates
- Edit event details and updates

The only users who will edit news will be Admins and the President.

# 7 Testing

## 7.1 Overview

The test plan is intended to support the operation of the system as a whole so that all of its features operate according to expectations. There will be several tests performed prior to delivery, as well as recommendations for how to optimize the system's performance. The purposes, scope, and testing strategy of the plan are discussed in the following sections.

## 7.2 Objective

The following objectives are supported by this document:

- Analyze existing project information
- Determine testing strategies and approaches
- Determine which features should be tested
- Provide a list of the recommended test requirements
- Assess the risks associated with the testing strategy
- List of deliverables for testing activities
- Improve the quality of the product delivered by reducing bugs within the system

## 7.3 Scope

In essence, the test plan is designed to demonstrate high-level approaches and strategies for testing Maggotty High Alumni's official website. As a result, test engineers will have guidance on how to conduct tests and have a comprehensive understanding of why it is so important to conduct testing. A key function of this audit is to validate that the website functions in accordance with the customer's requirements. To ensure the product is delivered on time, the entire software specification will be followed. Due to the limited time available for delivering the software application, The Monday's has prioritized the features and testing that will be done according to their importance. It is permissible for all testing of very low priority to be excluded from the document, but it is acceptable for quality assurance to be performed.

## 7.4 Test Approach

The overall objective of the test plan is to identify and resolve any problems within the system. Moreover, you will ensure that all functional requirements are met in accordance with the client's requirements. During testing, the Mondays will test all of the required functions. These tests include:

- Interface Testing
- Functionality Testing
- Design process
- Compatibility Testing
- Usability Testing
- Performing Testing
- Security Testing

## 7.5 Testing Items

Here, the Mondays will provide a list of all of the components that have been identified as test items. All module interfaces will be tested and unit testing will be performed through black-box testing. The following browsers will have their external interfaces tested:

- Mozilla Firefox
- Google Chrome
- Safari

Compatibility testing will take place on the following external platforms:

- Windows 10
- Mac OSX
- Mobile Devices
- Android Pixel 3a or higher
- iPhone 10 or higher

The performance test between the interfaces will include:

- Validity of Login
- Email transition
- Permissions of user interactions
- Smooth performance
- Navigation through links
- User traffic

- Image optimization
- Creation of profile
- Event/ news content creation
- Account customization
- Purchasing of products
- Editing of content
- Payment portal
- Screen Resolution
- Donation portal
- Browser compatibility

The highest priority of testing will be given to:

- The validity of login and special permission
- The functionality of the major items
- Purchasing products
- Content creation
- The optimal transition between web pages, and resolution performance

## 7.6 Successful Criteria

- High-quality/priority testing must be performed and retested for any bugs.
- High quality/priority testing will ensure the client's need for the system to work properly.
- All phases should collaborate together appropriately to achieve success criteria.
- When testing each item we look for the following:
  - **Check out all the links:** Testing the outgoing links, testing all internal links, test links jumping on the same page, page to page, test links that are used for emails to admin users or other users, and finally check broken links.
  - **Test forms on all pages:** Testing for users to receive information to interact with them. Testing form for website input user's information, wrong inputs, options to create forms, form delete, view, and modify.
  - **Cookie Testing:** testing the system for enabling or disabling cookies in all browser options.
  - **Validate your System language:** Testing for syntax errors and testing for search engines.
  - **Database Testing:** Testing for data integrity and errors while editing, deleting, and modifying all the forms.

As a result of the success criteria, our testing items will result in the highest quality product we are capable of. Manual verification is required for all these validations.

## 7.7 Testing Methodologies

Module #	Module Name	Priority	Testing Approach	Objective	Tester
1	Login Registration	High	Black Box Testing	Verify a user can sign up an account through the website, and transition to a customized page with identified user permissions.	Nesreen Salah
2	Viewing Featured Pages	High	White Box Testing	The tester will be able to view all featured pages when clicking on the featured pages.	Nesreen Salah
3	Purchasing	High	White Box Testing	The object of purchasing is to verify all items ready for purchase can be added to the "Cart" page. This functionality is to ensure multiple items can be purchased.	Nesreen Salah
4	Cart Page	High	End to end testing	The objective of the cart page is to make sure valid products can go through the system and submit, edit, or delete an order.	Nesreen Salah
5	Create / Edit / Delete Events	High	Black Box Testing	The objective is to check if it allows the	Nesreen Salah

				administrators to create and delete events & show them on the events page under a particular category on the Events Page.	
6	Donation	High	Black Box Testing	The objective is to verify the functionality of the Donation buttons, the donation custom bar, the payment prompts, and the completion of a donation.	Nesreen Salah
7	Profile Customization	High	Black Box Testing	The objective is to verify the profile functionality can upload images, crop images, edit descriptions, change passwords, and view shipping information.	Nesreen Salah
8	Delete/Edit/Create Volunteer opportunity	High	Black Box Testing	The objective is to check if the administrator can create and delete volunteer opportunities from the Volunteer Page.	Nesreen Salah
9	Delete/Edit User Profile	High	Black Box Testing	The Objective is to check if the administrator & the assistant can edit the user profile.	Nesreen Salah
10	Payment Page	High	Black Box Testing	Verifying functionalities of the payment portal, making sure the order confirmation is	Nesreen Salah

				given, an order number is given, and cross-checking effects of the submodules information.	
--	--	--	--	--	--

## 7.8 Module Test Cases

### 7.8.1 Login Registration

Case #	Tester Permission Level	Page	Test Case	Expected Result	Actual Result	Result
1	0	Login Page	Tester clicks “Login” Tester then identifies and clicks the “Sign up” button. Tester fills out all entry information, and clicks submit.	A user account will be registered within the database, the system will display registration successful and the user will be able to Login into the homepage using the created credentials with email and password.	Registration Successful.	pass
2	0	Login Page	Tester clicks “Login” at the top-right hand corner. Tester then identifies and clicks the “Sign up” button. Tester fills out only username and password entry information, and clicks submit.	This will be an invalid registration and the system will display “Please fill out this field.”	Registration Unsuccessful	pass

3	0	Login Page	Tester clicks “Login” at the top-right hand corner. Tester then identifies and clicks the “Sign up” button. Tester fills out no entry	This will be an invalid registration and the system will display “Please fill out this field.”	Registration Unsuccessful	pass
4	0	Login Page	Tester clicks “Login” at the top-right hand corner. Tester then identifies and clicks the “Sign up” button. Tester fills out all required entry information indicated by the asterisks, but not optional tabs, and clicks submit.	A user account will be registered within the database, the system will display registration successfully and the user will be able to Login into the homepage using the created credentials with email and password.	Registration Unsuccessful	pass
5	0	Login Page	Tester enters a password that does not follow the password parameters indicated on the display.	The system will display in red font “One lowercase letter required One uppercase letter required One number required One special character required 8	Registration Unsuccessful	pass

				characters minimum” in the chat. The system will display “Please fill out this field.” when clicking submit.		
6	0	Login Page	Tester uses an existing email as user email and fills out all other required entries.	The system will display in red font “email already exists” in the chat. The system will display “Please fill out this field.” when clicking submit.	Registration Unsuccessful	pass
7	0	Login Page	Tester enters different password in “re-enter password(*)” than the “password (*)”. Password does not match.	The system will display in red font “Please enter the same password” in the chat. The system will display “Please fill out this field.” when clicking submit.	Registration Successful	pass

### 7.8.2 Viewing Featured Pages

Case #	Tester Permission Level	Page	Test Case	Expected Result	Actual Result	Result
1	0	Home Page	Tester clicks on News Page	The tester should be able to see the alumni association news listed in descending (Newest on the top) order.	Tester is able to see the alumni association news listed in descending (Newest on the top) order.	Pass
2	0	Home Page	Tester clicks on Events Page	The tester should see categories of events on the left and a list of events for that category listed on the right.	Tester sees categories of events on the left and a list of events for that category listed on the right.	Pass
3	0	Home Page	Tester Clicks on Shop Page	The tester should see the shop page with the list of items on sale, with price and availability.	Tester sees the shop page with the list of items on sale, with price and availability.	Pass
4	0	Home Page	Tester Clicks on the Cart page	If the tester has added any merchandise in the cart from the shop page,	Tester sees the message saying "Your cart is empty. Let's go shopping"	Pass

				It should be listed on the cart page with its price. Else, he should be displayed a message saying "Your cart is empty now. Let's go shopping"		
5	0	Home Page	Tester Clicks on My Maggotty	The tester should see the screen displaying his profile picture info, shipping info &	Tester sees the screen displaying her profile picture info, shipping info & description with options to edit them.	Pass
6	0	Home Page	Tester clicks on history	Tester should see the history page of the website with a few lines about how it came to be established.	Tester sees the history page of the website with a few lines about how it came to be established.	Pass

### 7.8.3 Purchasing

Case #	Tester Permission Level	Page	Test Case	Expected Result	Actual Result	Result
1	0	Shop	The tester will click the “Shop” from the main page or from clicking the “Shop” from the drop-down from the top right corner from the profile menu.	The tester will be navigated to the shop page from any portal listed.	The tester will be navigated to the shop page from any portal listed.	Pass
2	0	Shop	If there are no products for sale that are created by admin, then the tester clicks on the shop page.	The system will display “Market” with no available products for sale.	System displays “Market” with no available products for sale.	Pass
3	0	Shop	If there are products for sale that are created by admin, then the tester clicks on the shop page.	System displays an image of the product, a short description of the product, product name, unit price, and stock availability.	System displays an image of the product, a short description of the product, product name, unit price, and stock availability.	Pass
4	0	Shop	If there are products for sale, created by admin, the tester clicks on the shop page. The tester selects the product of	System displays a query “Enter how many do you want?”	System displays a query “Enter how many do you want?”	Pass

			choice by clicking the image.			
5	0	Shop	If there are products for sale, created by admin, the tester clicks on the shop page. Select the product of choice by clicking the image. Enters the amount in the “Enter how many do you want to box as “5”	System displays a query “Value must be less than or equal to 1”	System displays a query “Value must be less than or equal to 1”	Pass
6	0	Shop	If there are products for sale, created by admin, the tester clicks on the shop page. Select the product of choice by clicking the image. Enters the amount in the “Enter how many do you want to box as”1”.	“System displays a query “Item successfully added”. Items will be added to the cart.	“System displays a query “Item successfully added”. Items will be added to the cart.	Pass
7	0	Shop	If there are products for sale, created by admin, the tester clicks on the shop page. Select the product of choice by clicking the image. Enter the amount in the “Enter how many do you want box as “1”.	The system displays a query “Item is already added in your cart.”	The system displays a query “Item is already added in your cart.”	Pass

#### 7.8.4 Cart Page

Case #	Tester Permission Level	Page	Test Case	Expected Result	Actual Result	Result
1	0	Cart Page	Tester clicks on the top right corner with username ID and selects “Cart” from the drop-down menu from any page.	Tester is navigated to the cart page.	Tester is navigated to the cart page.	Pass
2	0	Cart Page	With no items in the cart, Tester clicks on the top right corner with username ID and selects “Cart” from the drop-down menu from any page.	The system displays “Your cart is empty now, Let’s go shopping.”	The system displays “Your cart is empty now, Let’s go shopping.”	Pass
3	0	Cart Page	With 1 item in the cart, Tester clicks on the top right corner with username ID and selects “Cart” from the drop-down menu from any page.	The system displays an image of the item, the item name, the unit price, and select quantity.	The system displays an image of the item, the item name, the unit price, and the chosen amount.	Pass
4	0	Cart Page	With 1 item in the cart, Tester clicks on the top right corner with username ID and selects “Cart” from the drop-down menu from any page. The tester chooses “remove this item from cart”.	The system displays a query, “Do you want to remove this item from your shopping cart?” With an option of OK or Cancel.	The system displays a query, “Do you want to remove this item from your shopping cart?” With an option of OK or Cancel	Pass
5	0	Cart Page	With 1 item in the cart, Tester clicks on the top right corner with username ID and selects “Cart” from the drop-down menu from	The system navigates to an empty cart and displays “Your cart is	The system navigates to an empty cart and displays “Your cart is	Pass

			any page. The tester chooses “remove this item from cart”. Tester selects “Ok”.	empty now, Let’s go shopping.”	empty now, Let’s go shopping.”	
6	0	Cart Page	With 1 item in the cart, Tester clicks on the top right corner with username ID and selects “Cart” from the drop-down menu from any page. The tester chooses “remove this item from cart”. Tester selects “Cancel”.	The system will return to the previous interface to either check out, change the quantity, or decide to remove this item.	The system will return to the previous interface to either check out, change the quantity, or decide to remove this item.	Pass
7	0	Cart Page	With 1 item in the cart, Tester clicks on the top right corner with username ID and selects “Cart” from the drop-down menu from any page.	The system will display “Please Confirm your order!” with the selected product name and Total price tag with the option to “confirm” or “cancel”.	The system will display “Please Confirm your order!” with the selected product name and Total price tag with the option to “confirm” or “cancel”.	Pass

### 7.8.5 Create / Edit / Delete / Events

Case #	Tester Permission Level	Page	Test Case	Expected Result	Actual Result	Result
1	1&4	Admin > Events	The tester will click the “Events” from the admin page.	The tester should be navigated to the page “Add Events” with the option to delete existing events at the bottom	The tester is navigated to the page “Add Events” with the option to delete existing events at the bottom	Pass

2	1&4	Add Events	The tester leaves all the fields empty and clicks the “Add Event” button.	The system should show the message “Please fill out the mandatory fields”	The system shows the message “Please fill out all mandatory fields”	Pass
3	1&4	Add Events	Tester tries to fill the mandatory fields with incorrect input types.	The system shouldn't do anything.	The system doesn't do anything.	Pass
4	1&4	Add Events	If there is an existing event and the tester tries to edit the event by clicking the Change button under the event.	The system should show the user all the fields (editable)	The system shows the user all the fields (editable)	Pass
5	1&4	Add Events	Tester fills up all the mandatory fields with the correct input data types and clicks the submit button.	The system should show, “event created successfully” and show the event on the “Events” page.	System shows, “event created successfully” and shows the event on the “Events” page.	Pass
6	1&4	Add Events	Tester clicks the “delete” button under the existing event.	The system should ask the tester “Are you sure?” and Clicking yes should delete the event from the Events page.	System asks the tester “Are you sure?” and Clicking “Ok” deletes the event from the Events page.	Pass
7	1&4	Add Events	Tester clicks the “delete” button under the existing event.	The system should ask the tester “Are you sure?” and Clicking “No” should not delete the event from the Events page.	System asks the tester “Are you sure?” and Clicking “Cancel” doesn't delete the event from the Events page	Pass

## 7.8.6 Event Registration

Case #	Tester Permission Level	Page	Test Case	Expected Result	Actual Result	Result
1	0	Donation	The tester will access the donation page by selecting the dropdown menu from the top right corner and selecting “Donate”.	The system will display the Donation Page.	The system will display the Donation Page.	Pass
2	0	Donation	On the donation page, the tester will select \$5 and click the “Donate \$5” button.	The system will display the Donation payment page.	The system will display the Donation payment page.	Pass
3	0	Donation	On the donation page, the tester will select the circular button, and click and hold, then drag it to the right.	The system will change values in the Donation amount in increasing value as you drag the bar further towards the right.	The system will change values in the Donation amount in increasing value as you drag the bar further towards the right.	Pass
4	0	Donation	On the donation page, the tester will select \$5 and click the “Donate \$5” button. Tester fills out all the information and selects confirm.	The system will display “Thank you for your Donation! Every penny counts! Thank you for helping out your Alumni Association! We are glad to have you”.	The system will display “Thank you for your Donation! Every penny counts! Thank you for helping out your Alumni Association! We are glad to have you”.	Pass

					to have you".	
5	0	Donation	On the donation page, the tester will select \$10 and click the “Donate \$10” button. Tester fills out all required information, and selects confirm.	System will display “Thank you for your Donation! Every penny counts! Thank you for helping out your Alumni Association! We are glad to have you”.	System will display “Thank you for your Donation! Every penny counts! Thank you for helping out your Alumni Association! We are glad to have you”.	Pass
6	0	Donation	On the donation page, the tester will select \$10 and click the “Donate \$10” button. Tester fills out all required information, except the card type, and selects confirm.	The system will prompt the tester “Please select an item in the list.”, indicating nothing was selected in the dropdown menu for “Select card type.”	The system will prompt the tester “Please select an item in the list.”, indicating nothing was selected in the dropdown menu for “Select card type.”	Pass
7	0	Donation	On the donation page, the tester will select \$50 and click the “Donate \$50” button. Tester fills out only the “select card type” as “Visa”, and selects confirm.	The system will prompt the tester “Please fill out this field.”, in the empty box.	The system will prompt the tester “Please fill out this field.”, in the empty box.	Pass

### 7.8.7 Profile Customization

Case #	Tester Permission Level	Page	Test Case	Expected Result	Actual Result	Result
1	0	My Profile	The tester will click the top right corner of the interface and select my profile.	The system will display the “My Maggotty” Navigation Menu which includes User, Edit Shipping, Description & Portrait, Change password, and Orders. The profile information will be visible on this page.	The system will display the “My Maggotty” Navigation Menu which includes User, Edit Shipping, Description & Portrait, Change password, and Orders. The profile information will be visible on this page.	Pass
2	0	My Profile	The tester will click the top right corner of the interface and select my profile. The tester will select “Edit shipping”.	The system will display editable shipping information including name, cell phone number, and address registered to the account.	The system will display editable shipping information including name, cell phone number, and address registered to the account.	Pass
3	0	My Profile	The tester will click the top right corner of the interface and select my profile. The tester will select “Edit shipping”. The tester will change entries and click submit.	The system will change entries registered to the account. For example, If the name is changed, the top right corner will display a new name.	The system will change entries registered to the account to match what was in the system in all areas in the backend.	Pass
4	0	My	The tester will click	The system will	The system	Pass

		Profile	the top right corner of the interface and select my profile. The tester will select "Description & Portrait".	display a page where the tester can change the avatar picture with a cropping tool and change the description of the profile.	displays a page where the tester can change the avatar picture with a cropping tool and change the description of the profile	
5	0	My Profile	The tester will click the top right corner of the interface and select my profile. The tester will select "Description & Portrait". The tester uploads a picture, resizes the image and clicks crop.	The system displays the newly cropped image. As the avatar and the image on the top right corner.	The system displays the newly cropped image. As the avatar and the image on the top right corner.	Pass
6	0	My Profile	The tester will click the top right corner of the interface and select my profile. The tester will select "Change Password".	The system brings the tester to a page where the tester can change the login password. Original Password is required for entry.	The system brings the tester to a page where the tester can change the login password. Original Password is required for entry.	Pass
7	0	My Profile	The tester will click the top right corner of the interface and select my profile. The tester will select "Change Password". Tester types tester's unique password credential to the account.	The system displays a page where a new password can be registered with existing password parameters as the account registration and a "Submit" button that allows for	The system displays a page where a new password can be registered with existing password parameters as the account registration and a "Submit" button that Change allows	Pass

				similar functionality as password registration in account registration.	for similar functionality as password registration in account registration.	
--	--	--	--	---	---	--

### 7.8.8 Delete / Edit/ Create Volunteer Opportunity

Case #	Tester Permission Level	Page	Test Case	Expected Result	Actual Result	Result
1	4	Admin Page	Tester can access the admin page by typing “localhost:8080/admin/” in the URL of the browser.	System displays admin pages with specific features for News, Volunteer, Events, Shopping, User, and Order tabs from the menu.	System displays an admin page with specific features for News, Volunteer, Events, Shopping, User, and Order tabs from the menu.	Pass
2	4	Admin Page	On the admin page, the tester clicks “Volunteer” from the menu.	System displays a Volunteer page with specific features for adding Volunteer Opportunity, searching for opportunities, and a volunteer list.	System displays a Volunteer page with specific features for adding Volunteer Opportunity, searching for opportunities, and a volunteer list.	Pass

3	4	Admin Page/Home Page	<p>On the admin page, the tester clicks “Volunteer” from the menu. Tester enters title, content, date, and category and clicks “Add Volunteer Opportunity”.</p> <p>Tester then goes to Homepage, clicks the image for “New Volunteer Opportunity”, selects the category that was created.</p>	<p>System displays the title of the volunteering opportunity, the date, and the content of what was created.</p>	<p>System displays the title of the volunteering opportunity, the date, and the content of what was created.</p>	Pass
4	4	Admin Page/Home Page	<p>On the admin page, the tester clicks “Volunteer” from the menu. The tester clicks “change” on a current event from the volunteer list.</p> <p>Tester will change any entries, and click submit change. . Tester then goes to Homepage, clicks the image for “New Volunteer Opportunity”, selects the category that was edited/unedited.</p>	<p>System displays the title of the volunteering opportunity, the date, and the content of what was edited.</p>	<p>System displays the title of the volunteering opportunity, the date, and the content of what was edited.</p>	Pass
5	4	Admin Page/Home Page	<p>On the admin page, the tester clicks “Volunteer” from the menu. The tester clicks “delete” on a current event from the volunteer list.</p> <p>Tester will click “Yes” when prompted “Are you sure?”</p> <p>Tester then goes to Homepage, clicks the image for “New Volunteer Opportunity”,</p>	<p>System displays a page where the event that was “deleted” no longer exists.</p>	<p>System displays a page where the event that was “deleted” no longer exists.</p>	Pass

			selects the category that was deleted.			
6	4	Admin Page/Home Page	<p>On the admin page, the tester clicks “Volunteer” from the menu. The tester clicks “delete” on a current event from the volunteer list.</p> <p>Tester will click “Cancel” when prompted “Are you sure?”</p> <p>Tester then goes to Homepage, clicks the image for “New Volunteer Opportunity”, selects the category that was prompted to be deleted.</p>	<p>System displays a page where the event that was prompted to be “deleted”, still exists.</p>	<p>System displays a page where the event that was prompted to be “deleted”, still exists.</p>	Pass
7	4	Admin Page/Home Page	<p>On the admin page, the tester clicks “Volunteer” from the menu. Tester enters title, content, date, and category and clicks “Add Volunteer Opportunity”. This will be repeated four times within the same category, to create four events. Tester then selects “change” on the second category that was created and checks the box “move to front”.</p> <p>Tester then goes to Homepage, clicks the image for “New Volunteer Opportunity”, selects the category that was edited.</p>	<p>System displays the category that was moved to the top of the volunteering category page.</p>	<p>System displays the category that was moved to the top of the volunteering category page.</p>	Pass

### 7.8.9 Delete / Edit User Profile

Case #	Tester Permission Level	Page	Test Case	Expected Result	Actual Result	Result
1	4	Admin Page	Tester will access the admin page by typing "localhost:8080/admin/" in the URL of the browser.	System displays an admin page with specific features for News, Volunteer, Events, Shopping, User, and Order tabs from the menu. User permission level is also displayed on this page.	System displays an admin page with specific features for News, Volunteer, Events, Shopping, User, and Order tabs from the menu. User permission level is also displayed on this page.	Pass
2	4	Admin Page	On the admin page, the tester will select "User" from the menu.	The system will display a "Find User" header with an entry box for email of the user that the tester would like to search.	The system will display a "Find User" header with an entry box for email of the user that the tester would like to search.	Pass
3	4	Admin Page	On the admin page, the tester will select "User" from the menu. Tester types a nonexistent email, and selects "Find User".	The system will prompt with a message "Can not find", indicating the user does not exist.	The system will prompt with a message "Can not find", indicating the user does not exist.	Pass

4	4	Admin Page	On the admin page, the tester will select “User” from the menu. Tester types an existing email, and selects “Find User”.	The system will display proprietary information about the user which includes First Name, Last Name, Student ID, Email, Password, Q1, A1, Q2, A2, and permission level.	The system will display proprietary information about the user which includes First Name, Last Name, Student ID, Email, Password, Q1, A1, Q2, A2, and permission level.	Pass
5	4	Admin Page	On the admin page, the tester will select “User” from the menu. Tester types an existing email, and selects “Find User”. Tester selects “change”.	The system will display proprietary information about the user which includes First Name, Last Name, Student ID, Email, Password, Q1, A1, Q2, A2, and permission level that can be modified by changing the fields in the entry boxes.	The system will display proprietary information about the user which includes First Name, Last Name, Student ID, Email, Password, Q1, A1, Q2, A2, and permission level that can be modified by changing the fields in the entry boxes.	Pass
6	4	Admin Page	On the admin page, the tester will select “User” from the menu. Tester types an existing email, and selects “Find User”. Tester selects “change”. Tester	System displays edited/modified user permission when searched.	System displays edited/modified user permission when searched.	Pass

			changes “isAdmin” to Normal Account. Tester searches the same user profile.			
7	4	Admin Page	On the admin page, the tester will select “User” from the menu. Tester types an existing email, and selects “Find User”. Tester selects “delete”. Tested selects “Yes” when prompted. Tester searches the same user profile.	System no longer displays the user profile when searched.	System no longer displays the user profile when searched.	Pass

### 7.8.10 Payment Page

Case #	Tester Permission Level	Page	Test Case	Expected Result	Actual Result	Result
1	0	Payment Page	Tester clicks on the top right corner with username ID, and selects “Cart” from the drop down menu from any page. Selects the product, checks out, and confirms to enter the payment page.	Tester is navigated to the payment page.	Tester is navigated to the payment page.	Pass
2	0	Payment Page	The tester fills out no entries in the payment page and clicking confirm	The system displays “Please fill out this field.”, in the empty fields.	The system displays “Please fill out this field.”, in the empty fields.	Pass

3	0		The tester fills out all entries in the payment page and clicks confirm.	The system displays, "Congratulations, Your order is confirmed, you can check it in your profile. Thanks for shopping with us."	The system displays, "Congratulations, Your order is confirmed, you can check it in your profile. Thanks for shopping with us."	Pass
4	0	Payment Page	The tester fills out all entries in the payment page and clicks confirm. Tester then clicks cart.	The system displays a query, "Your cart is empty now, Let's go shopping."	The system displays a query, "Your cart is empty now, Let's go shopping."	Pass
5	0	Payment Page	The tester fills out all entries in the payment page and clicks confirm. Tester then clicks shop from top right corner or other shop portal methods..	The system displays product items but the available amount will be reduced by 1 prior to purchase.	The system displays product items but the available amount will be reduced by 1 prior to purchase.	Pass
6	0	Payment Page	The tester fills out only required entries in the payment page and clicks confirm.	The system displays, "Congratulations, Your order is confirmed, you can check it in your profile. Thanks for shopping with us."	The system displays, "Congratulations, Your order is confirmed, you can check it in your profile. Thanks for shopping with us."	Pass
7	0	Payment Page	The tester fills out all entries in the payment	The system displays a	The system displays a	Pass

			page and clicks confirm. Tester clicks profile from dropdown menu in top right corner. Tester then selects “Orders”.	unique Order Number, Status, and more details regarding the order.	unique Order Number, Status, and more details regarding the order.	
--	--	--	---	--	--	--

## 7.9 Testing Conclusion

Testing was completed by testers of The Mondays internally. Any bugs that were found in the high priority modules were found and fixed to obtain a “Pass” result. Multiple test cases were used for various scenarios to find all possible errors. The Mondays did its best to prioritize testing functionality of major modules due to the time constraint of deliverable products. Any additional flaws that can be found, clients may contact The Mondays. The next section will provide a user manual that will allow training of client members.

## 8 Appendix A

Attach any additional information that supplements the design specification

### 8.1 User Manual

#### Installation of necessary software

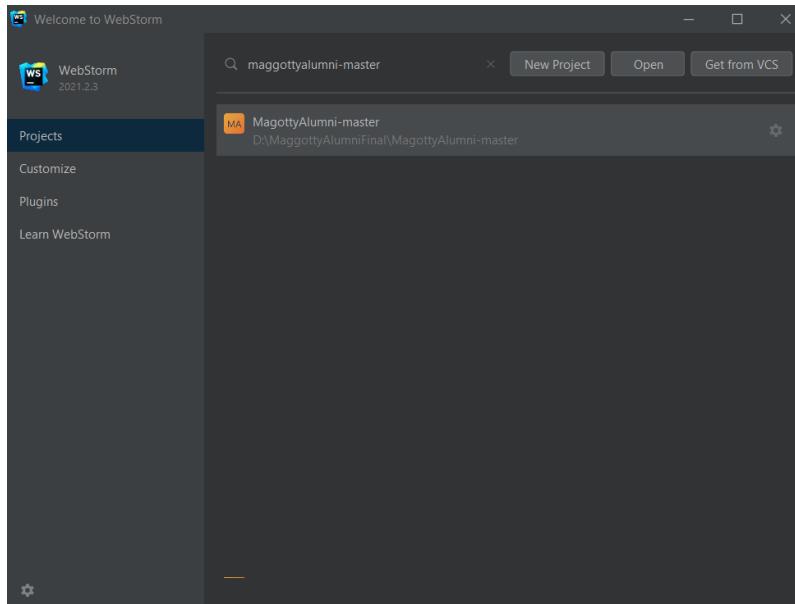
Please see links below to download and install all necessary components.

- Nodejs:
  - Download Link:  
<https://nodejs.org/en/download/>
  - Youtube Instructions for windows:  
[https://www.youtube.com/watch?v=qYwLOXjAiwM&ab\\_channel=Simplilearn](https://www.youtube.com/watch?v=qYwLOXjAiwM&ab_channel=Simplilearn)
  - Youtube Instructions for Mac:  
[https://www.youtube.com/watch?v=TQks1p7xjdl&ab\\_channel=ProgrammingKnowledge](https://www.youtube.com/watch?v=TQks1p7xjdl&ab_channel=ProgrammingKnowledge)
- MongoDB:
  - Download Link:  
<https://www.mongodb.com/try/download/community>
  - Youtube Instructions for windows:  
[https://www.youtube.com/watch?v=QTZmA1kCEEw&ab\\_channel=GridoWit](https://www.youtube.com/watch?v=QTZmA1kCEEw&ab_channel=GridoWit)
  - Youtube Instructions for Mac:  
[https://www.youtube.com/watch?v=DX15WbKidXY&ab\\_channel=ProgrammingKnowledge](https://www.youtube.com/watch?v=DX15WbKidXY&ab_channel=ProgrammingKnowledge)
- WebStorm:
  - Download Link:  
<https://www.jetbrains.com/webstorm/download/#section=windows>
  - Youtube Instructions for windows:  
[https://www.youtube.com/watch?v=gwoOeRJfv9s&ab\\_channel=GeekyScript](https://www.youtube.com/watch?v=gwoOeRJfv9s&ab_channel=GeekyScript)

- Youtube Instructions for Mac:
  - [https://www.youtube.com/watch?v=l6iOCaXFGyA&ab\\_channel=DevChannel](https://www.youtube.com/watch?v=l6iOCaXFGyA&ab_channel=DevChannel)
- Website Code:
  - In provided flash drive or.
  - Download Link:
    - <https://github.com/omnidox/MaggottyAlumni-master>

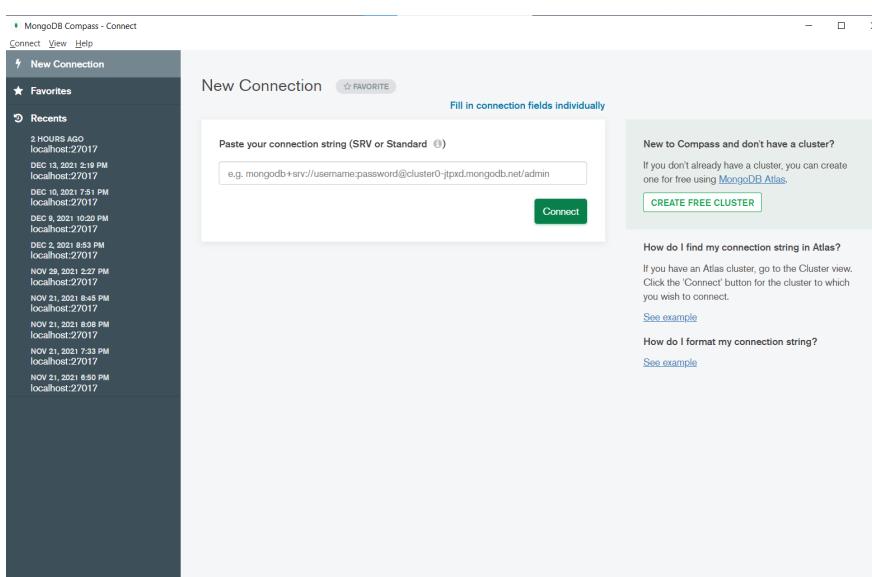
## Getting the Website to Run

Once everything is downloaded, please use WebStorm to open up the project (**Install Figure 1**).



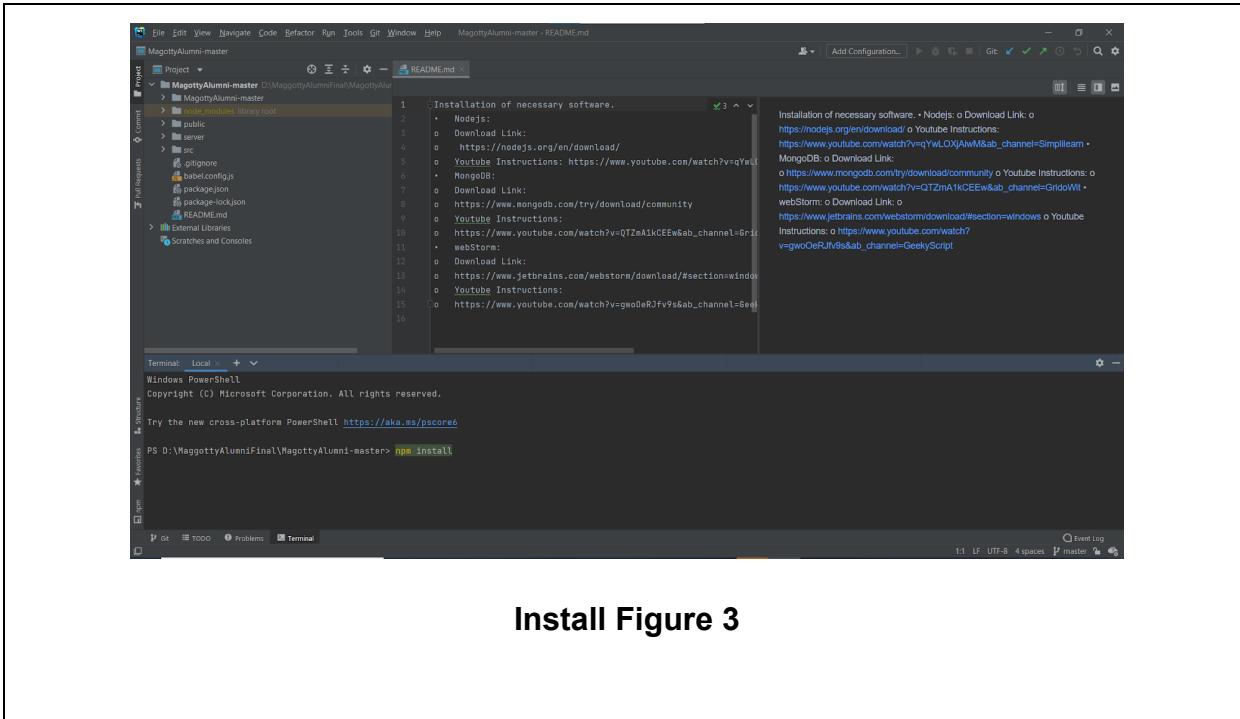
**Install Figure 1**

Ensure mongoDB is connected by pressing the green connect button (**see Install Figure 2**).

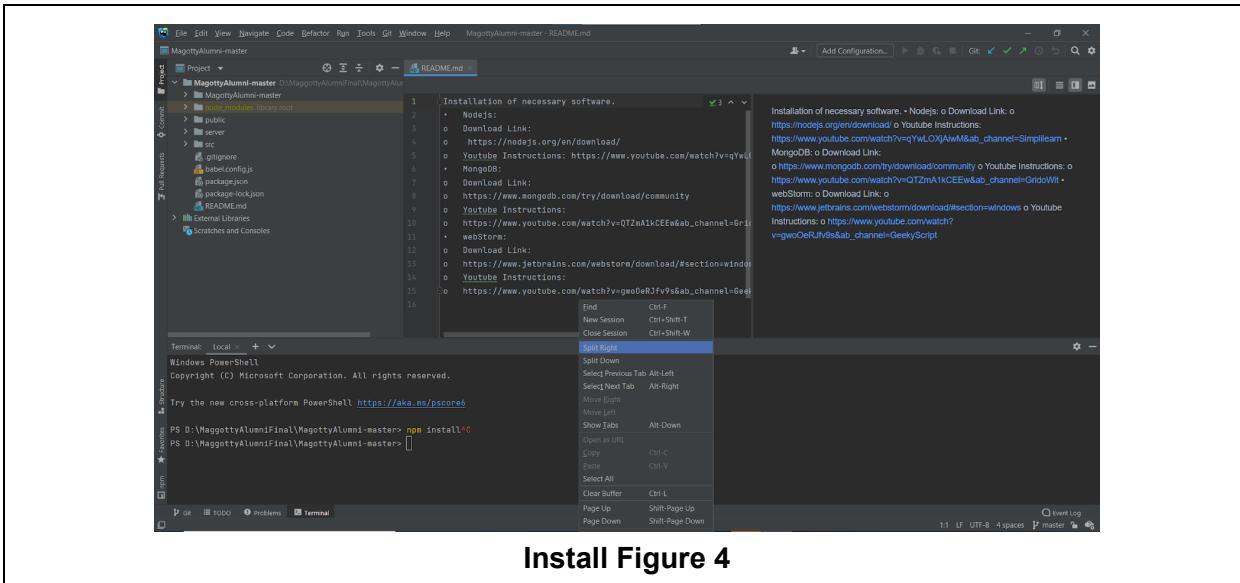


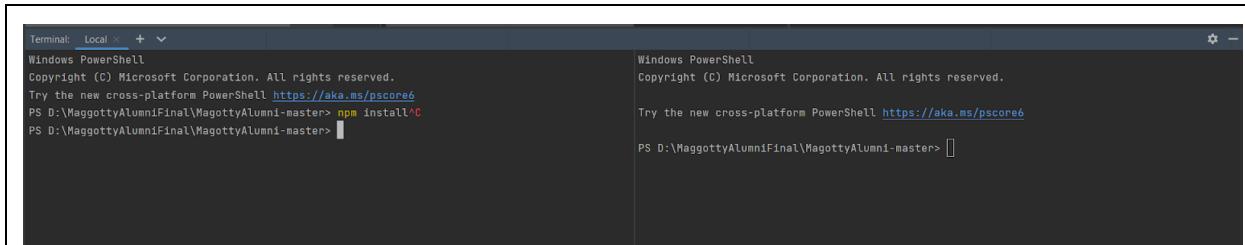
**Install Figure 2**

In Webstorm, click on the bottom tab where it says terminal. Then type in the following command “npm install” and then press enter. This will download and install all of the necessary node modules for the project. See (**Install Figure 3**)

**Install Figure 3**

Once the npm install is finished, right click on the terminal window (**Install Figure 4**) and choose the split right option. This will split the terminal into two windows. (**Install Figure 5**).

**Install Figure 4**



The image shows two adjacent terminal windows. The left window is titled 'Terminal' and has a tab labeled 'Local'. It contains a PowerShell session with the following command:

```
PS D:\MaggottyAlumniFinal\MaggottyAlumni-master> npm install^C
```

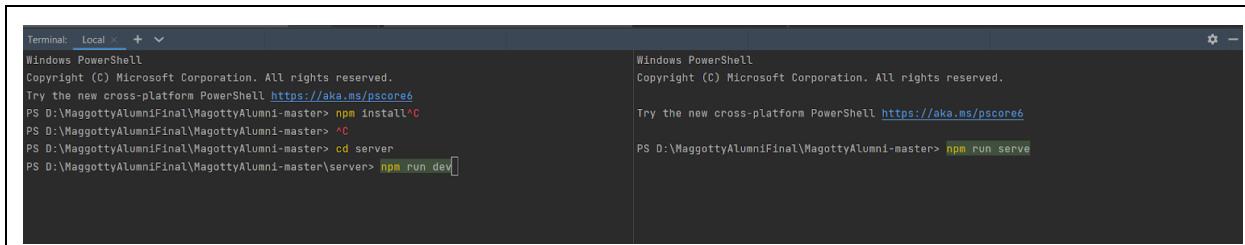
The right window is also a PowerShell session with the following command:

```
PS D:\MaggottyAlumniFinal\MaggottyAlumni-master>
```

**Install Figure 5**

In the leftmost terminal window we run our back end, while in the rightmost window we will run our front end. In the leftmost window we will type `cd server` to go to the server file. Then, while in the server file we will type “`npm run dev`” and then press enter to get the back end server running.

In the rightmost window we will type “`npm run serve`” and then press enter to get the front end server running. (**See Install Figure 6**)



The image shows two adjacent terminal windows. The left window is titled 'Terminal' and has a tab labeled 'Local'. It contains a PowerShell session with the following commands:

```
PS D:\MaggottyAlumniFinal\MaggottyAlumni-master> npm install^C
PS D:\MaggottyAlumniFinal\MaggottyAlumni-master> cd server
PS D:\MaggottyAlumniFinal\MaggottyAlumni-master> npm run dev
```

The right window is also a PowerShell session with the following command:

```
PS D:\MaggottyAlumniFinal\MaggottyAlumni-master> npm run serve
```

**Install Figure 6**

With the frontend and backend servers running, and with the database connected, we are now ready to access our website homepage by typing “localhost: 8080” in the browser window.

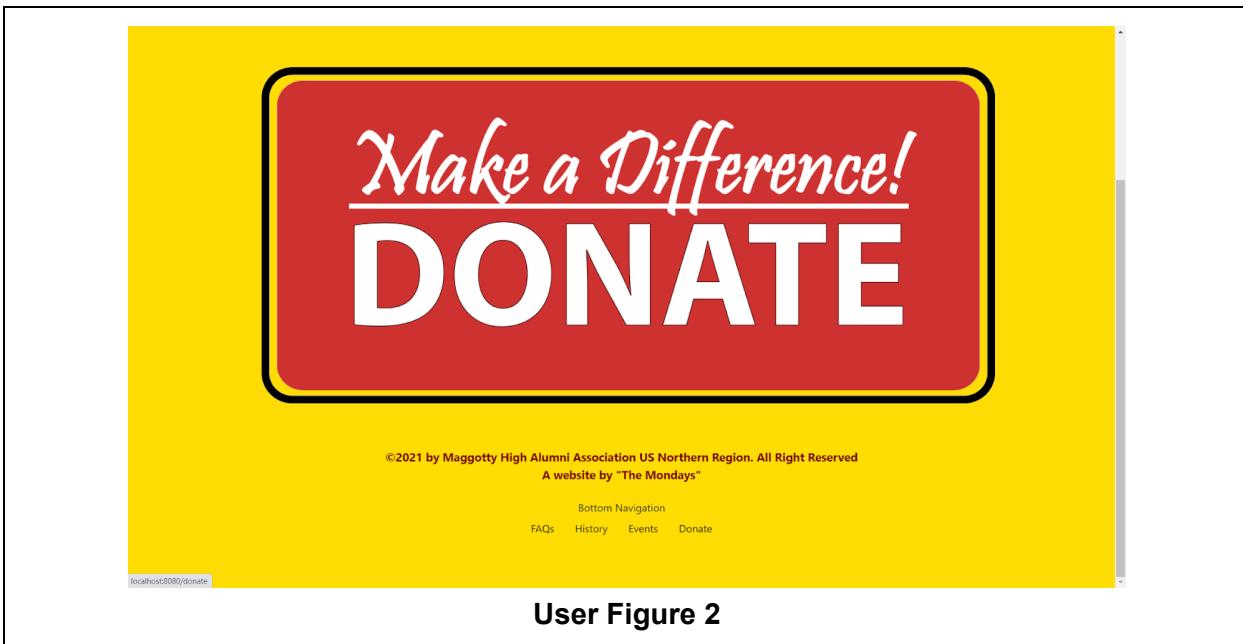
## Website Navigation Instructions

Homepage:

The first time the homepage is visited, you are greeted with a top and bottom UI as outlined in **User Figures 1 and 2**.



User Figure 1



User Figure 2

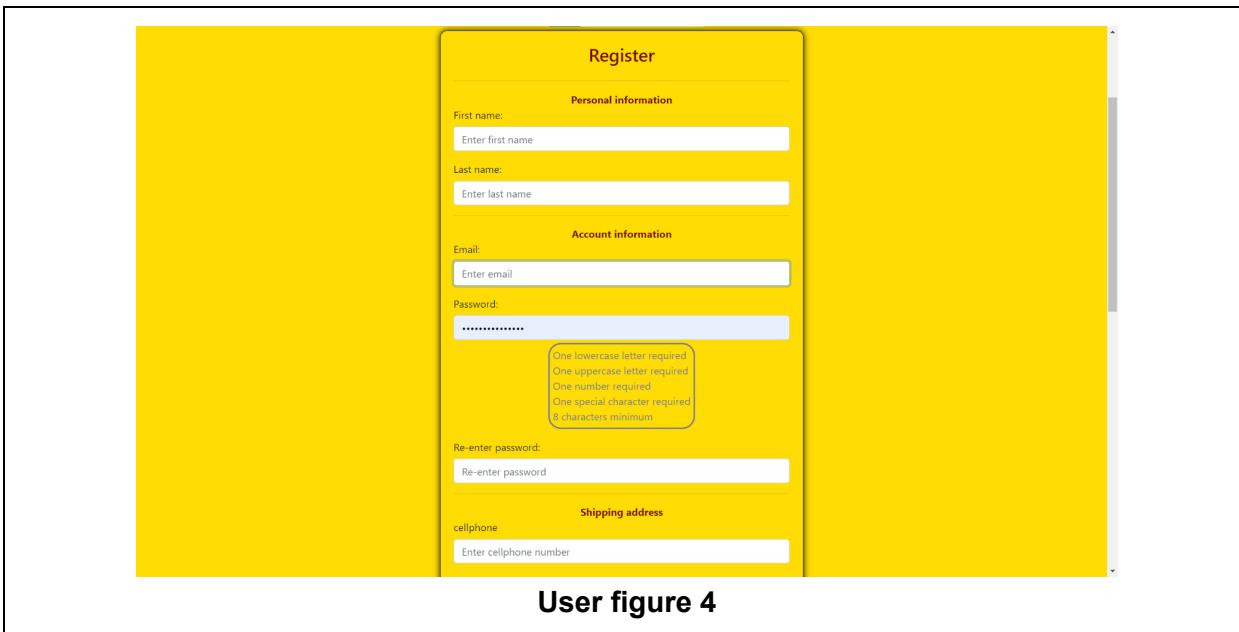
## Registration:

To register click on the upper left icon that says login, as shown in **User Figure 3**. You will then be redirected to the login page. Users who wish to make an account will click on the sign up button to enter the registration page.



**User Figure 3**

Once in the registration page (**User Figure 4**) fill out all of the relevant forms and then click on submit. Once registered, you will be redirected to the login page (**User Figure 3**). There you fill your relevant credentials in the login box and then click login.



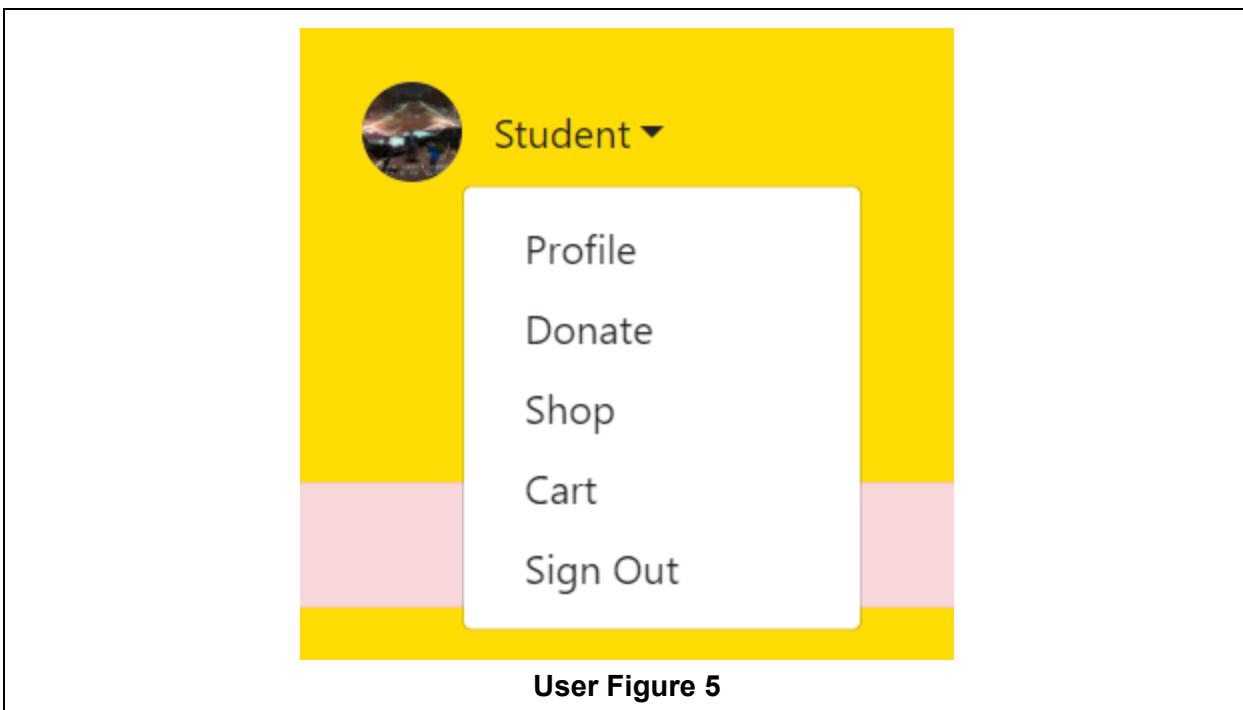
**User figure 4**

### User Account:

Once you are logged in, you will see your profile name appear in the upper left of the screen where the login icon was. Next to the icon you will see a dropdown box appear. Click on this box to gain access to the dropdown menu as shown in **User Figure 5**.

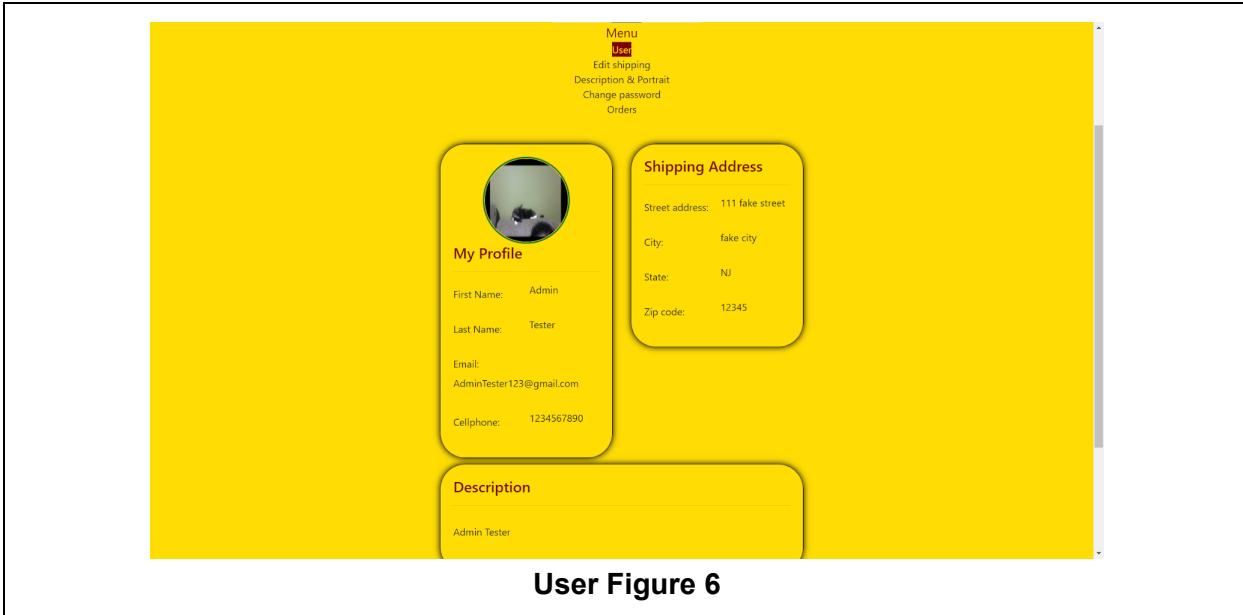
In the dropdown menu, you will see the following options

- Profile
- Donate
- Shop
- Cart
- Sign out (this simply signs the user out)



## Profile:

In the profile page (**User Figure 6**) the user can then edit their shipping information, edit their profile description and Portrait, change their passwords, and review their orders.



## Donate:

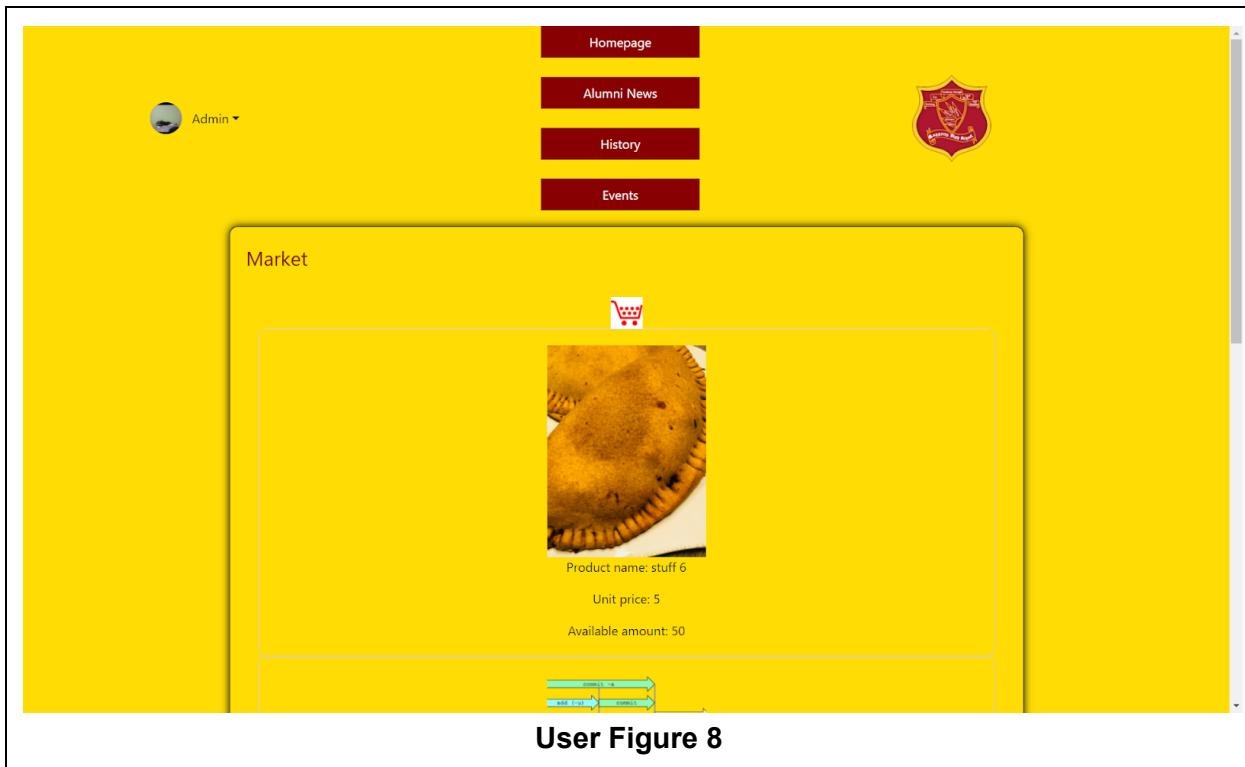
In the donate page (**User Figure 7**) the user can Donate a set amount as determined by the top four radio buttons or they can enter a custom amount.

The screenshot shows a donation page with a yellow header and footer. The main content area has a white background. At the top center is a small image of four plants growing from soil. Below it is a caption: "From sapling to mighty trees, your donations will help ensure the growth of all Maggotty Alumni." Underneath is a call-to-action: "Please donate today!". A section titled "Donate options:" lists four radio buttons: \$ 5 (selected), \$ 10, \$ 20, and \$ 50. Below this is another section titled "Or add a custom amount!" with a radio button selected for \$ 1 and a text input field containing the value 1. A "Donate \$ 5" button is located at the bottom of this section. At the very bottom of the page is a footer with copyright information: "©2021 by Maggotty High Alumni Association US Northern Region. All Right Reserved" and "A website by 'The Mondays'". Below the footer is a "Bottom Navigation" bar with links for "FAQs", "History", "Events", and "Donate".

**User Figure 7**

## Shop:

In the shop page (**User Figure 8**) the user can view the different wares the alumni association has to offer and can purchase a limited amount of them. The amount is determined by the number of products available.



Cart:

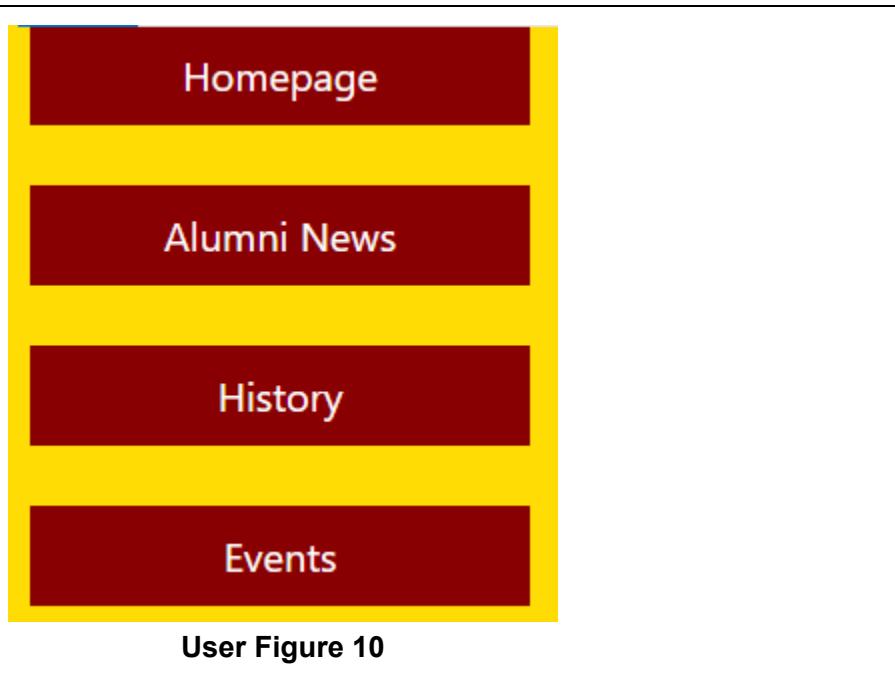
In the shopping cart page (**User Figure 9**) the user can view what was put into his/ her cart. Then the user can check out and pay for the items at their leisure.

The screenshot shows a website interface for the Maggotty High Alumni Association. At the top, there is a navigation bar with links for 'Alumni NEWS', 'History', and 'Events'. On the left, there is a user profile section labeled 'Admin' with a dropdown arrow. On the right, there is a logo featuring a crest with text. Below the navigation bar, a large yellow banner spans the width of the page. In the center of the yellow banner, there is a 'Shopping Cart' section. This section contains an image of a pie, labeled 'stuff 6', with a unit price of '\$5/ea'. Below the image, there is a text input field labeled 'Choose amount:' containing the number '5'. To the right of the input field are two buttons: 'Remove this item from cart' and 'Check out!'. At the bottom of the yellow banner, there is a copyright notice: '©2021 by Maggotty High Alumni Association US Northern Region. All Right Reserved'. Below the yellow banner, the text 'User Figure 9' is centered.

## Navigation Menu:

In the navigation menu (**User Figure 10**) you will see the following options.

- Homepage
- Alumni News
- History
- Events



## Alumni News:

When the Alumni News bar on the Navigation menu is chosen, the user is brought to the alumni news page (**User Figure 11**), where the user can see any news the Alumni Association has posted.

The screenshot shows a web page titled "Alumni News". It displays four news items, each with a thumbnail image, the title, a brief description, the date, and a "Read More" button. The news items are arranged vertically.

- News 4**  
News 4  
2021-12-19  
[Read More](#)
- news 3**  
news 3  
2021-12-19  
[Read More](#)
- random news 2**  
random news 2  
2021-12-18  
[Read More](#)
- random news 1**  
random news 1  
2021-12-18  
[Read More](#)

**User Figure 11**

## History:

The History bar will bring the user to the history page of the Alumni Association (**User Figure 12**).

Homepage

Alumni News

History

Events

Admin ▾

The Organization's history

The inaugural meeting was held at the elegant Jamaican Restaurant, The Door, located at 163-67 Baisley Blvd, Jamaica Queens NY 11434.

©2021 by Maggotty High Alumni Association US Northern Region. All Right Reserved  
A website by "The Mondays"

Bottom Navigation

FAQs History Events Donate

**User Figure 12**

## Events:

When the Events bar on the Navigation menu is chosen, the user is brought to the Events page (**User Figure 13**), where the user can see any events the Alumni Association has posted. Events are organized according to volunteer and alumni events.

The screenshot shows a yellow-themed web page titled "Upcoming Events". At the top, there's a navigation bar with links for "Events", "Up-Coming Events", "Volunteer Events", and "Alumni Events". Below the navigation is a large, stylized title "UPCOMING" where each letter is a different color (U-pink, P-orange, O-green, C-blue, M-yellow, I-blue, N-blue, G-blue). A small watermark "shutterstock.com 147920380" is visible below the title. The main content area lists four event entries, each with a title, date, and a brief description. The events are separated by horizontal lines.

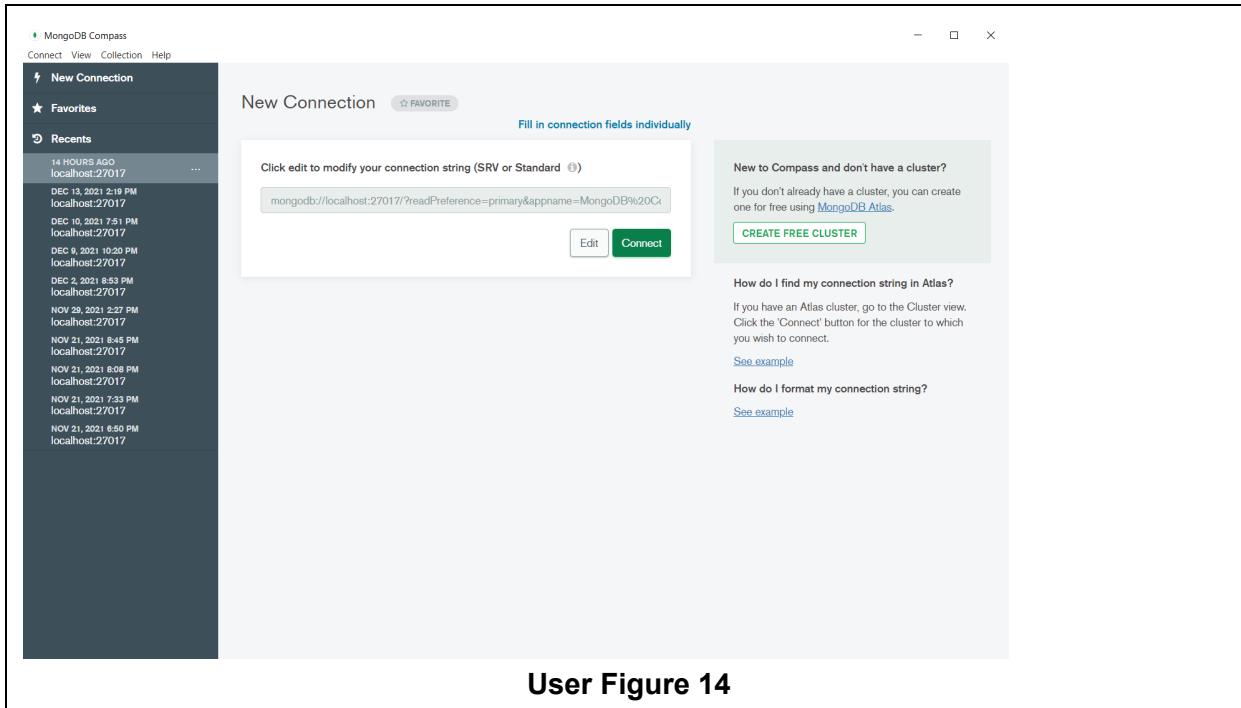
Event Type	Title	Date	Description
alumni event	alumni event 2	2022-02-04	alumni event 2
Alumni	Alumni event 1	2022-01-06	Alumni event 1
volunteer	volunteer 2	2022-01-21	volunteer 2
volunteer	volunteer 1	2021-12-30	volunteer 1
random	random event 2	2022-01-31	random event 2

**User Figure 13**

Admin:

To enable admin privileges, the Admin needs to make a user account as stipulated in the steps above. Once the Admin has created an account, he/she will then need to find his/her account inside the mongoDB database. This is done by opening the MongoDB Compass program. (**User Figure 14**)

In the MongoDB program, click on connect.



User Figure 14

Once connected you will be brought to a menu of available Databases (**User Figure 15**). The alumni association website is programmed to make a database called maggottyAlumni. Click and enter this database.

The screenshot shows the MongoDB Compass interface with the 'Databases' tab selected. On the left sidebar, it shows 'Local' with 'HOST localhost:27017', 'CLUSTER Standalone', and 'EDITION MongoDB 5.0.4 Community'. Below this is a search bar and a list of databases: admin, config, local, maggottyAlumni, and mevn\_full\_stack. The main area displays a table with columns: Database Name, Storage Size, Collections, and Indexes. The 'maggottyAlumni' database is listed with 208.9KB storage, 6 collections, and 6 indexes.

Database Name	Storage Size	Collections	Indexes
admin	20.5KB	0	1
config	36.9KB	0	2
local	36.9KB	1	1
maggottyAlumni	208.9KB	6	6
mevn_full_stack	36.9KB	1	1

**User Figure 15**

Once in the maggottyAlumni database, you will see one or several collections (**User Figure 16**). Click on the collection called users.

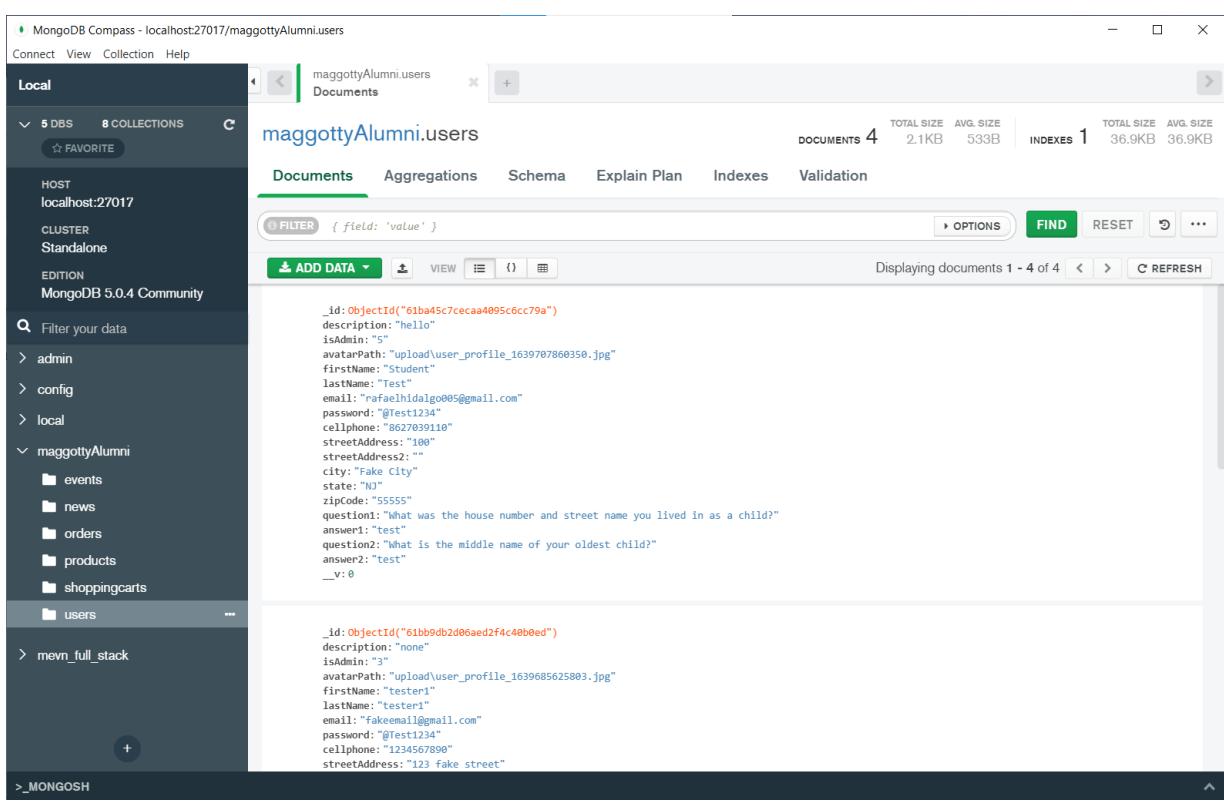
The screenshot shows the MongoDB Compass interface with the 'Collections' tab selected for the 'maggottyAlumni' database. The left sidebar shows the database structure with 'maggottyAlumni' expanded to show collections: events, news, orders, products, shoppingcarts, and users. The main area displays a table with columns: Collection Name, Documents, Avg. Document Size, Total Document Size, Num. Indexes, Total Index Size, and Properties. The 'users' collection is highlighted with 4 documents, 532.5B avg size, 2.1KB total size, and 1 index.

Collection Name	Documents	Avg. Document Size	Total Document Size	Num. Indexes	Total Index Size	Properties
events	6	151.7 B	910.0 B	1	36.9 KB	
news	4	155.0 B	620.0 B	1	36.9 KB	
orders	3	329.7 B	989.0 B	1	36.9 KB	
products	3	175.3 B	526.0 B	1	36.9 KB	
shoppingcarts	1	166.0 B	166.0 B	1	24.6 KB	
users	4	532.5 B	2.1 KB	1	36.9 KB	

**User Figure 16**

Inside the user's collection, you will see several attributes related to the user (**User Figure 17**). In order to enable admin privileges, you will have to change isAdmin value to 5. It is important to note here that privileges can be assigned to users via the isAdmin value. A number will correlate to a user's role. See the correlation below.

- Assistant Treasury Account = “1”
- Treasurer account = “2”
- Vice President Account = “3”
- President Account = “4”
- Admin Account = “5”



**User Figure 17**

The screenshot shows the MongoDB Compass interface connected to the 'maggottyAlumni' database at localhost:27017. The 'users' collection is selected, displaying four documents. The first document is expanded, showing its fields:

```

_id: ObjectId("61ba45c7ecaa4095c6cc79a")
description: "hello"
isAdmin: "5"
avatarPath: "upload\user_profile_1639707868350.jpg"
firstName: "Student"
lastName: "Test"
email: "rafaelhidalg005@gmail.com"
password: "#Test1234"
cellphone: "8627899110"
streetAddress: "100"
streetAddress2: ""
city: "Fake City"
state: "ND"
zipCode: "55555"
question1: "what was the house number and street name you lived in as a child?"
answer1: "test"
question2: "what is the middle name of your oldest child?"
answer2: "test"
__v: 0

```

The second document is partially visible:

```

_id: ObjectId("61bb9db2d06aed2f4c40b0ed")
description: "none"
isAdmin: "3"
avatarPath: "upload\user_profile_1639685625803.jpg"
firstName: "tester1"
lastName: "tester1"
email: "fakemail@gmail.com"
password: "#Test1234"
cellphone: "1234567890"
streetAddress: "123 fake street"

```

## AdminPage:

Now that we have changed isAdmin to the appropriate value, the admin page can now be accessed. This is done via <http://localhost:8080/admin/>. Keep in mind once a domain name is assigned to the website admin will then be accessed as “domain\_name”.com/admin/. Please see **User Figure 18** for a display of the Admin homepage.

In the Admin navigation menu (**User Figure 18**) you will see the following options.

- News
- Alumni and Volunteer Events
- Shopping
- User
- Order

Through these options the admins can add, edit, find, and delete News, Alumni and Volunteer Events, available products for sale, User accounts, and User orders respectively.

The screenshot shows the Admin homepage with a yellow background. At the top left is a user profile icon labeled "Student ▾". On the right is a school crest. A vertical navigation bar on the left contains five dark red buttons: "Homepage", "Alumni News", "History", and "Events". Below this is a larger menu with the following items:  
Menu  
News  
Alumni and Volunteer Events  
Shopping  
User  
Order

In the center, a white rounded rectangle contains the text:  
Welcome to the admin page  
Account type: Administrator Account

At the bottom, there is a copyright notice: ©2021 by Maggotty High Alumni Association US Northern Region. All Right Reserved  
A website by "The Mondays"  
Bottom Navigation  
FAQs History Events Donate

**User Figure 18**

## 8.2 Data Dictionary

### 8.2.1 Overview

The Data Dictionary is an evolving document and will be continually refined throughout the functional design, technical design, and build stages of the project.

#### Field Definitions

##### Key

**PK** = primary key (concatenated key has more than one attribute)

**I** = index or non-primary key

Blank for other attributes

**Name:** as per Logical Data Model

**Description:** Textual description of the Name field

**Type** – choose one of the following

**ID** – for an ID field, usually a primary key – specify a reasonable length if known, only for the first primary key of the entity.

**Lookup** – for a field that is validated from a master table of valid values.

**Num** – for any numeric field.

**Char** – for any short text attribute likely to have a specifically defined length. Specify the length, if known.

**Text** – for any long text attribute likely to have a large or undefined length.

**Date** – if only date without a timestamp.

**Date/time** – if time stamp, e.g. for audit trail.

**Len:** See above for the types for which you should specify reasonable or actual field lengths.

**M:** M if mandatory field; A if auto-calculated field; otherwise blank.

##### Validation

Where the validation is for the attribute value to exist on a master table, designate the field type as “Lookup”. In the Validation column, state the entity name and attribute against which it is to be validated.

Otherwise, describe specific limits or business rules against which the attribute values are to be validated.

**Input/update:** Name the application/screens/processes that create, calculate or update stored values for this attribute.

**Where used:** Name the application/screens/processes/reports that read or use this attribute.

## 8.2.2 Alumni

Key	Name	Description	Type, length	M	Validation	Input / update
<b>PK</b>	universityId	Unique identifier for each Alumni	ID	M	Unique	
I	firstName	Alumni's Forename	Char(30)	M		/registerInfo
I	lastName	Alumni's Surname	Char(30)	M		/registerInfo
I	email	Alumni's Email	String	M	must contain "@"	/registerInfo
	password	Password used to access the system	String	M	One uppercase letter required One number required One special character required Eight characters minimum	/registerInfo  /changePassword
	cellphone	Mobile Number	Char(20)		3-digit prefix, hyphen, 7-digit num	/registerInfo  /shipping
	streetAddress	Street Address	String	M		/registerInfo  /shipping
	streetAddress2	Apt number if necessary	String			
	city	City of residence	String	M		/registerInfo  /shipping
	state	State residence	String	M		/registerInfo  /shipping

	zipCode	postal code	NUM	M	Must only contain numbers	/registerInfo /shipping
	question1	Required Security question for account access	String	M		/registerInfo
	answer1	Answer is mapped to Security question for validation of identity	String	M		/registerInfo
	question2	Required Security question for account access	String	M		/registerInfo
	answer2	Answer is mapped to Security question for validation of identity	String	M		/registerInfo
	description		String			/changeDescription /addCareer /deleteCareer /changeCareer
	isAdmin		NUM	M		
	avatarPath		String			/changePhoto

### 8.2.3 Cart

Key	Name	Description	Type, length	M	Validation	Input / update
<b>PK</b>	path	Unique identifier for each Cart	String	M	Unique	
I	item	Item's name	String	M		
I	email		String	M	must contain "@"	/addToCart
	price	total price in the cart	NUM	M	Must be a numerical value	
	amount	Current Item amount in cart	NUM	M	Must be a numerical value	/changeAmount /deleteItems /clearCart
	maxAmount	Total possible purchase amount	Lookup		amount cannot be greater than maxAmount	

## 8.2.4 Events

Key	Name	Description	Type, length	M	Validation	Input / update
<b>PK</b>	title	Unique identifier for each event	String	M	Unique	/addEvents /deleteEvent /changeEvent
<b>I</b>	content	Event's content	String	M		/addEvents /deleteEvent /changeEvent
<b>I</b>	date	The date the event will be held	Date	M	Must fall within a certain range of 100 years from the creation date.	/addEvents /deleteEvent /changeEvent
	category	What is the event?	String	M	Must fall under the following: <ul style="list-style-type: none"> <li>• homecoming</li> <li>• class reunions</li> <li>• fundraisers</li> <li>• athletics</li> <li>• webinars</li> </ul>	/addEvents /deleteEvent /changeEvent
	sortDate	sorts events from most recent to oldest.	NUM	A	Must be a numerical value	

## 8.2.5 Orders

Key	Name	Description	Type, length	M	Validation	Input / update
<b>PK</b>	orderNumber	Unique identifier for each Order	String	M	Unique	/addOrder
I	firstName	Forename	Char(30 )	M		
I	lastName	Surname	Char(30 )	M		
	email	Email	String	M	must contain "@"	
	cellphone	Mobile Number	Char(20 )		3-digit prefix, hyphen, 7-digit num	/shipping
	streetAddress	Street Address	String	M		/shipping
	streetAddress 2	Apt number if necessary	String			
	city	City of residence	String	M		/shipping
	state	State residence	String	M		/shipping
	zipCode	postal code	NUM	M	Must only contain numbers	/shipping
	amount	Current item amount in cart	NUM	M	Must be a numerical value	/changeAmount /deleteItems /clearCart
	status	Status of the Order	String	A	Default: 'Pending'	/deleted /delivered /success /reset
	track	Tracking status of the order	String	A	Default: 'Hasn't shipped'	/addTracking /success
	product		String	M		/detail /isEnough

## 8.2.6 News

Key	Name	Description	Type, length	M	Validation	Input / update
<b>PK</b>	path	Unique identifier for each news article	String	M	Unique	
<b>I</b>	content	news content	String	M		
<b>I</b>	date	The date the news was published	Date	M	Must fall within a certain range of 100 years from the creation date.	
	sortDate	sorts news from most recent to oldest.	NUM	A	Must be a numerical value	
	title		String	M	Unique	

## 8.2.7 Product

Key	Name	Description	Type, length	M	Validation	Input / update
<b>PK</b>	path	Unique identifier for each Cart	String	M	Unique	/addProduct /deleteProduct
I	name	Product's name	String	M		/changeProduct
I	size	Product's size	String	M	must contain “@”	/changeProduct
	price	Product's price	NUM	M	Must be a numerical value	/changeProduct
	amount	Current Item amount in cart	NUM	M	Must be a numerical value	/addProduct /deleteProduct
	description		String			/changeProduct
	category		String	M		/changeProduct

### 8.2.8 Volunteer

Key	Name	Description	Type, length	M	Validation	Input / update
<b>PK</b>	title	Unique identifier for each event	String	M	Unique	/addVolunteer /deleteVolunteer /changeVolunteer
<b>I</b>	content	Event's content	String	M		/addVolunteer /deleteVolunteer /changeVolunteer
<b>I</b>	date	The date the event will be held	Date	M	Must fall within a certain range of 100 years from the creation date.	/addVolunteer /deleteVolunteer /changeVolunteer
	category	What is the event?	String	M		/addVolunteer /deleteVolunteer /changeVolunteer
	sortDate	sorts events from most recent to oldest.	NUM	A	Must be a numerical value	

## 8.3 Glossary of Terms

Identify all terms that establish meaning within the context of the plan.

Term	Meaning
Front end	relating to or denoting the part of a computer system or application with which the user interacts directly
Back End	relating to or denoting the part of a computer system or application that is not directly accessed by the user, typically responsible for storing and manipulating data
Processor	A processor is an integrated electronic circuit that performs the calculations that run a computer
RAM	Random-access memory is a form of computer memory that can be read and changed in any order, typically used to store working data and machine code.
SSD	A solid-state drive (SSD) is a solid-state storage device that uses integrated circuit assemblies to store data persistently, typically using flash memory, and functioning as secondary storage in the hierarchy of computer storage.
Modem	A modem is a device that connects your home, usually through a coax cable connection, to your Internet service provider (ISP), like Xfinity. The modem takes signals from your ISP and translates them into signals your local devices can use, and vice versa.
Router	A router connects your devices to each other and, in hard-wired connection setups, to the modem. The router connects to your modem and then to your devices (laptops, smart TVs, printers, etc.) via either an Ethernet cable or, in the case of a wireless router, WiFi signal.
Firewall	A firewall is a network security device that monitors incoming and outgoing network traffic and decides whether to allow or block specific traffic based on a defined set of security rules.
Human-Machine Interface	A Human-Machine Interface (HMI) is a user interface or dashboard that connects a person to a machine, system, or device.

Table 1 — Glossary of Terms