CASA Donation Management Web Application

CASA
Court Appointed Special Advocates
FOR CHILDREN
RAPPAHANNOCK CASA

System Requirements

Presented by: Ethan Bostick, Joshua Byrne, James Heathcock, Garrett Mckenzie, Maximilian Redman, and Carter Walker

Rappahannock Area Court Appointed Special Advocates
509 C Lafayette Blvd, Fredericksburg, VA 22401
(540) 710-6199

Department of Computer Science, University of Mary Washington

CPSC 430: Software Engineering

Dr. Jennifer A. Polack

September 28, 2025

This Page Is Left Intentionally Blank

Table Of Contents

1 Introduction

1.1 Statement of Purpose

This document represents the cumulative effort of Rappahannock CASA along with students from the University of Mary Washington in planning for the production of the CASA Donation Management Web Application. The reason for making this document is to consolidate prior efforts in Requirements gathering, planning, and specifications into a single cohesive piece. A thorough reading of this document will enable the reader to fully understand the high level design and expectations for this system.

1.2 Intended Audience

This document is intended for the developers of the CASA Donation Management Web Application, as well as Edie Evans, the executive director of Rappahannock CASA. Other concerned parties who are considered in the document are the general staff of Rappahannock CASA, as well as professors of the UMW computer science department.

1.3 Scope

This system will be designed only for internal use by the staff of Rappahannock CASA. It will serve as an all-encompassing process for donation, fundraising, and information management. The system achieves this goal through many subsystems. One such subsystem is an interactive storage system accessible through the Web Application. The storage system will support the import of properly formatted Excel files via a drag-and-drop box, as well as the export of stored information filtered by date into an Excel file. The application will also serve as a system for creating and tracking fundraising events through a calendar application. In addition, the application will contain a data analysis suite which is capable of generating comprehensive reports dynamically on the bequest of the user. This analysis suite will produce extensive graphics as well as detailed statistics from information stored in the database. The final capability of the project is a sub-system for automatic email generation and sending, given an appropriate prompt from the user, in combination with a list of recipients for the email.

Upon completion of each of these systems, the application will serve as a significant force multiplier for CASA staff. Workers will be able to generate reports comparable to in-depth analysis by a professional on demand while seamlessly moving data through the storage system. Further, CASA staff will be able to rapidly write personalized messages to generous donors for any purpose deemed necessary.

1.4 References

The homepage for Rappahannock CASA can be found via the link below. This homepage is the primary source for reaching out to CASA for donations, more information, or contact.

This paper also references the work of Karl Eugene Wiegers in his work “Software Requirements Essentials” which can be referenced via the link below.

1.5 Overview

The remainder of the paper contains specifics on each function of the system in greater detail. There are several core sections to the piece. The first section is a background section that covers information such as constraints, target users, and security concerns. The next section is a full description of the project, both in terms of the webpages themselves and the anticipated workflows for users. The following section contains diagrams, specification forms, and other Requirements for the software as a whole. Following the Requirements section is a list of any other assumptions about users or hardware made during the planning for the application, as well as a glossary to enable quick reference of possibly unfamiliar terms. The final section is largely titular and contains contact information for the student developers participating in the development of the system.

2 Background

2.1 Key Constraints

This section identifies the key constraints for the CASA Donor Management System. Key constraints are defined as known or unknown factors that may impact the design and implementation of the solution. Additionally, key constraints may influence the development or usability of certain functionalities in the final solution by requiring certain conditions to be met.

- The CASA Donor Management System must ensure web accessibility compliance with the Americans with Disabilities Act (ADA).
- The development team must complete Requirements within the time constraint of the UMW Fall 2025 Semester, ending December 8th, 2025.
- The CASA Donor Management System must be developed using open source free software, both within the actual system and when referencing any exterior systems.

2.2 End User Characteristics

The end users of the system are the people who will be utilizing it after development is complete and the Web Application has been deployed. When considering end users, it is best to think of the users as roles that an individual might fill while using the system, instead of separate people using the site.

- Admins who manage the system's storage and other users' permissions.
- Analysts who use the system to generate reports and search through the stored information.
- Community managers use the system to automatically create and send thank-you or alert emails.
- Fundraising managers who use the system to create, track, and edit fundraising events. The fundraising manager also uses the system to import/export donation information.

2.3 Recommended Use

The system will be designed with the following recommendations for its best use. While the client will be able to access the system in non-recommended ways, certain stylistic features may not look as intended, and certain security measures may trip.

- Use of the system on the Chrome web browser.
- Use of the system on a laptop or desktop device.
- Use of the system over a secure internet, in other words, a non-public internet.

2.4 System Security

The main security of the system is implemented by the separation of the admin from other users. The admin and other profiles designated as admin will be the only actors capable of directly editing stored information and managing other profiles. Initially, there will only be one admin, created by the developers and granted to Edie Evans. From this point, all other Admins will be designated by the client with no further input from the developers.

Additional security features include validation of stored donation information against tracked totals for debits and credits to ensure accurate financial information within the system. Further, because the system will utilize an external application to assist in email generation, email outputs will be checked for offensive, graphic, or otherwise inappropriate language before being sent to donors.

3 Product Description And Functions

3.1 Description

The Rappahannock Area Court Appointed Special Advocates (CASA) Donor Management System is a user-friendly application designed to centralize and streamline the management of donor information and donations for non-profit organizations like Rappahannock CASA. This section details the project and the product functions.

3a.1 Login Page

The Login Page is where the user begins their interaction with the system. From this page, they simply input their username and password, allowing them to progress to the Home Page. If either username or password is incorrect, then the user is given an error message and remains on the Login Page.

A note about the Login Page, and all other pages for that matter, is their style. While this system remains a back-of-house operation, it is still important to keep a clean and coherent style throughout. Therefore, all pages of the system will aim to utilize the same style as the main Rappahannock CASA website. This includes the usage of dark blue, light blue, white, and orange in the color scheme, as well as the usage of the CASA logo wherever necessary.

3a.2 Home Page

The Home Page is the page the user is taken to immediately after logging into the site. On the page, there are buttons that redirect the user to each of the system's other pages. On the top left of the page, the CASA logo will be displayed and itself will act as a link to the homepage of the CASA website. Opposite the logo will be a user profile picture, which can be clicked on to activate a drop-down from which the user can select to log out or switch users. Any page accessible from the Home Page will contain a “back to home” button, which returns the user to this page.

3a.3 Analyst Page

The Analyst Page is where the user goes if they would like to generate automatic reports using the information in the database. There are two kinds of reports the user can generate. One is a qualitative report which focuses on non-numeric data like who is donating, where they are donating from, why they donated, and frequent/infrequent donors. The other kind of report is a quantitative report, which focuses on numeric data such as total donations over a period and donations scattered by date of donation. When generating a report, the user can select to generate either or both of the report types via buttons on the Analyst Page. Once the report types have been selected, a form will be displayed allowing the user to select what kind of graphs and statistics they would like included in the report. In addition, the user will be able to select the date range over which they would like the report to reference and the style of the graphs the report creates. Once selected, the forms are sent to a separate program, which creates the report and stores it as a PDF in the database. If the generation is successful, the PDFs are displayed for the user to view in a separate tab. If the generation fails, then the user is given an alert informing them of the issue. In addition to the report generation form at the top of the page, there is also a table at the bottom of the page from which the user can scroll through to select prior generated reports. Again, once selected, the reports are displayed in a separate tab.

3a.4 Import/Export Page

The import export page is where the user goes if they would like to move data into the system via Excel files or pull data from the database into Excel files. When exporting from the system, the user simply selects what category of information they would like to export, for example, financial or donor. The relevant information is then retrieved and exported to the user's computer via download.

When importing into the system, there is a button that the user can select to open up their computer's filesystem and allow them to drag and drop their Excel file into a field. There will be a table-like list showing the files to be imported, the size of the files, as well as an option to remove the file. Once the added files are submitted, the files are moved into the system and validated. If all files are properly formatted, the information will be added to the database. Otherwise, the user will be notified about which files failed and roughly why. Finally, at the bottom of the page is an example layout for the proper formatting of the Excel file.

3a.5 Storage Page
The Storage Page is the page the user goes to if they would like to interact with the database in an easy manner. If the user is not an admin, they will only be able to search through the database on this page. To search through the database, the user will fill out a form on the Storage Page, which details each of the fields and includes an option to display entire categories of data, such as displaying all donors stored in the database. If the user is an admin, they will further have the ability to delete or otherwise edit data that is pulled during a search. Further, the admin will have an additional set of buttons at the bottom, each displaying a category of data the admin can submit. When a category is selected, the page reloads, displaying a custom form for that category of data, which the admin can fill out in order to add data to the database. For any edit, delete, or add action taken by the admin, confirmation will be requested before the change is pushed to the database. Further, at any time, the user can cancel their operation and reset all data.

3a.6 Outreach Page
The outreach page is where the user goes to if they would like to automatically generate and send emails. The main content of the page is a form the user can fill out specifying the purpose of the message, key points the message must contain, who the message should be sent to, and an optional image the user would like included in the email. Once the form has been filled out, the information is sent to a "Chat-GPT" like model, which uses it to generate an email. If the user opted to review the email before it is sent, the outreach page displays each generated email before it is sent out. Otherwise, the email is automatically sent. If the process is successful, the user is given a success message on the outreach page. If there is an error, then the detected issue is given to the user through an alert.

3a.7 Calendar Page
The calendar page is where the user goes to view a calendar with all existing fundraising events included. To add a new event, the user can press the add event button on the upper right of the page, which takes the user to a separate page containing a form with both required and optional fields. Once filled in, the user can submit the new event. If successful, the user is redirected to the calendar and can view the event on the calendar. If the addition fails, the user remains on the add event page and is given a warning indicating that there was an issue. Also on the calendar page, the user can select an event to view. Selecting an event displays all relevant information, including but not limited to fundraising target, start date, end date, and event description. While viewing an event, the user has the ability to manually edit some fields by clicking the edit button. This will make certain aspects of the page editable, and the changes will be made permanent upon successful submission of any changes. If the edit fails, or the user would like to cancel the edit by pressing the cancel button, the fundraising event is returned to its original state.

3b.1 Analyst Workflow
The Analyst workflow focuses on creating actionable insights from the donor and fundraising data collected by the system. Analysts begin by accessing the Analyst Page, where they are presented with options to generate either quantitative or qualitative reports. Quantitative reports focus on numerical data such as donation totals, average donation size, peak giving periods, and year-over-year comparisons. Qualitative reports emphasize trends such as donor motivations, repeat vs. first-time givers, and campaign engagement patterns.

The system provides analysts with a configurable form where they can select date ranges, choose filters, and specify the type of graphs or visualizations they would like to include. Once a report request is submitted, the system generates a professional PDF file and stores it in the database for future retrieval. Analysts can view newly generated reports in a separate tab, review historical reports in a table, and export results for presentation or strategy meetings. This workflow enables CASA to transform raw data into meaningful metrics that inform outreach strategies, donor recognition, and event planning.

3b.2 Donor Search Workflow
The Donor Search workflow allows staff to quickly locate donor records across a wide range of parameters. Accessible through the Storage Page, the search function provides filters for donor name, donation amount or range, campaign association, timeframe (e.g., last donation within six months), and geographical indicators such as zip code. These filters enable both quick lookups and advanced queries that support more detailed analysis.

When a user executes a search, the system retrieves matching records and displays them in an organized results table. Analysts and Admins can drill into donor details to better understand individual giving patterns, confirm contact information, or prepare lists for communication campaigns. If no search criteria are provided, the system defaults to displaying the full donor database. Additionally, the workflow ensures that failed searches return clear messages, such as "No results found," to avoid confusion. This workflow plays a critical role in identifying lapsed donors, recognizing consistent supporters, and preparing segmented lists for outreach.

3b.3 Admin Storage Management Workflow
The Admin Storage Management workflow is designed for administrators who maintain the accuracy and integrity of the donor database. While general users may only perform searches, Admins gain enhanced permissions through the Storage Page to add new records, edit existing entries, or delete inaccurate information. Each action taken by an admin requires confirmation before it is finalized, reducing the chance of accidental errors.

Admins can also use structured forms to add new categories of information into the database, such as creating a new donor profile, logging a donation, or uploading campaign-related records. The system allows Admins to cancel or reset any operation at any point, ensuring that partial or mistaken entries do not clutter the database. By giving administrators full oversight of data while enforcing careful review of each action, this workflow ensures the donor management system remains reliable, accurate, and ready for reporting or outreach.

3b.4 Community Outreach Workflow
The Community Outreach workflow streamlines CASA's ability to engage with donors, volunteers, and community partners. Using the Outreach Page, staff members complete a structured form where they specify the purpose of the message (e.g., donor thank-you, campaign launch, community update), define key points, select recipients, and optionally attach images or media. The system then uses an integrated text generation model to produce professional, customized emails that align with the organization's tone and branding.

Once drafted, messages can either be automatically sent or routed for manual review, depending on staff preferences. Successful deliveries trigger confirmation messages, while any errors (such as invalid email addresses) generate clear alerts. By automating the drafting and sending process, this workflow reduces the administrative burden on staff while ensuring consistency in donor and community communications. The ability to quickly generate tailored outreach materials helps CASA strengthen relationships, increase campaign participation, and build greater awareness in the community.

3b.5 Import Date Workflow
The Importing Data workflow provides Admins with the ability to integrate external records, such as QuickBooks spreadsheets or other donation logs, into the system. Through the Import/Export Page, users can upload Excel files by dragging and dropping them into the interface or by selecting files through a dialog box. A preview table then displays the files to be imported, including their sizes and an option to remove them before submission.

Once submitted, the system validates the file format and structure against a set of expected templates. If the validation succeeds, the records are added to the donor database; if errors are found, the system identifies which files failed and explains why (e.g., incorrect column headings or missing fields). This workflow allows CASA to avoid time-consuming manual entry while ensuring accuracy and consistency across its financial and donor records. By maintaining compatibility with QuickBooks exports, it also supports smoother audits and reconciliations.

3b.6 Exporting Data Workflow
The Exporting Data workflow ensures that CASA staff and auditors can generate compliant, export-ready files of donor and financial data. From the Import/Export Page, users select the type of information they wish to export, such as donor lists, campaign results, or financial summaries. Filters may be applied to refine the dataset, enabling exports such as "Donors who contributed over $100 in the past year" or "All donations tied to Giving Tuesday."

Once selected, the system compiles the relevant records and generates an Excel file, which is then provided as a download to the user's computer. These exports serve multiple purposes: preparing data for board meetings, supplying auditors with fiscal details, or creating donor reports for fundraising reviews. By offering structured, filterable export options, the workflow supports both compliance requirements and internal strategic planning.

3b.7 Fundraising Event Creation Workflow
The Fundraising Event Creation workflow governs the setup of new fundraising events in the system. From the Calendar Page, Admins or fundraising coordinators can click the "Add Event" button, which opens a detailed form. Required fields include the fundraising target, start date, and end date, while optional fields allow for descriptions, photos, locations, and automated alerts. Once submitted, the system validates all inputs and inserts the event into the database.

Successful creation triggers a confirmation screen, often with celebratory visual feedback (e.g., confetti popup), while errors highlight the problematic fields for correction. The workflow ensures that both required and optional details are properly captured, allowing for comprehensive event tracking. By standardizing event creation, CASA ensures consistency across its campaigns, whether they involve Giving Tuesday, community races, or holiday drives.

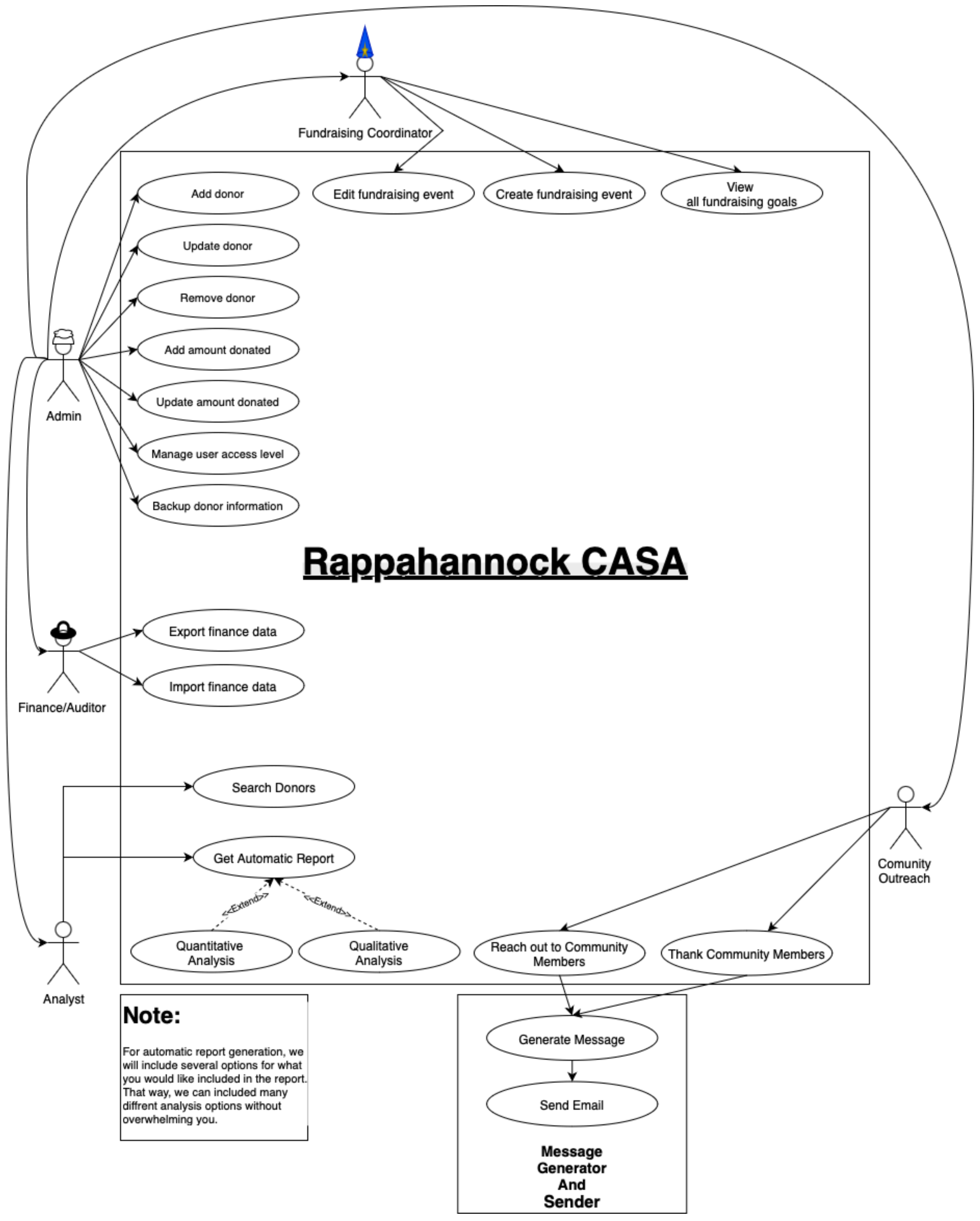
3b.8 Fundraising Event View/Edit Workflow
The Fundraising Event View/Edit workflow enables staff to maintain accurate and up-to-date event information. Through the Calendar Page, users can click on an event to review its details, including the fundraising goal, timeline, and description. Volunteers can view event details, while Admins and fundraising coordinators have the ability to make edits.

Editing is initiated by clicking the "Edit" button, which makes specific fields editable within the interface. Changes are validated upon submission, and if successful, the database is updated and a confirmation message is displayed. If an edit fails or is canceled, the event reverts to its original state. This workflow ensures that event information remains dynamic, supporting last-minute changes to goals, locations, or schedules, and providing all stakeholders with current, reliable information.

3.2 Conclusion
The overall goal of this system is to make the tasks of donation gathering, tracking, and analysis easier and more effective for Rappahannock CASA. Further, this system aims to make the tasks of tracking and reaching out to the donors themselves easier while also enabling quick and personal responses to each donor. The described pieces of the system will achieve all of these goals, each in its own manner, to combine into a comprehensive tool for Rappahannock CASA donation management. Through the proper use of this tool, Rappahannock CASA will enable itself to reach financial goals and support its volunteers. Most importantly, this tool will assist Rappahannock CASA in helping the children in our society who are most in need.

4 Project Requirements

4.1 System Use-Case Diagram:



Note:

For automatic report generation, we will include several options for what you would like included in the report. That way, we can included many diffrent analysis options without overwhelming you.

4.2 Use Case description

In the use case diagram, the boxes represent the different systems included in the software, each labeled in bold. Within these systems, the bubbles illustrate the functions that can be performed. Stick figures are used to represent the various actors or roles involved. Solid arrows indicate the actions that each actor is able to perform, while dotted arrows labeled with <<extends>> denote optional actions that may occur during the referenced action. The administrator role is shown with arrows pointing to every other actor, signifying that the admin has the ability to perform all of the actions available to those roles.

4.3 About Use Case Specifications

The following use case specifications describe the key interactions between users and the system, outlining how various functions are carried out to meet organizational needs. Each specification details the purpose of the use case, the actors involved, preconditions that must be met, the typical sequence of steps in the process, as well as alternate and exception flows. Use case specifications serve to clarify requirements, reduce ambiguity, and ensure that both technical and non-technical stakeholders share a clear understanding of how the system is expected to operate. By capturing these scenarios, the specifications provide a structured foundation for system design, implementation, and validation.

4.4 Use Case Specifications

UC1 – Add Donor Description: Adds a donor to the list of donors. On success, the system redirects to a confirmation page. If an error occurs, the system displays an error message.

Actors: Admin

Preconditions:

- Database is accessible and operational
- Required fields: donor name, contact information, donation amount

Basic Flow:

1. Admin accesses the donor form.
2. Admin enters required information.
3. Admin optionally enters additional information.
4. Admin submits the form.
5. System processes the form.
6. System adds the donor to the database.
7. System redirects to a “Form Complete” page.
8. “Form Complete” page allows navigation back to home or calendar.

Alternate Flows:

- AF1: If required information is missing, the system highlights missing fields and displays an error above the submit button. Flow resumes at step 4.

Exception Flows:

- EF1: Admin exits the program.
- EF2: Database is inaccessible.

Postconditions: Donor and related information are added to the system.

UC2 – Update Donor Description: Updates donor information to correct errors. Name, phone number, and email cannot be changed; a new entry must be created if those fields require modification.

Actors: Admin

Preconditions: Database is accessible and operational.

Basic Flow:

1. Admin opens the donor list.
2. Admin selects a donor.
3. Admin updates editable fields.
4. Admin submits modifications.
5. System validates fields.
6. System saves the updates.
7. System displays an “Updated!” popup with a “Done” button.

Exception Flows:

- EF1: Admin exits the program.
- EF2: Database is inaccessible.
- EF3: Admin attempts to edit a protected field.

Postconditions: Donor information is updated in the system.

UC3 – Remove Donor Description: Removes donor information from the system, typically when data is no longer valid.

Actors: Admin

Preconditions: Database is accessible and operational.

Basic Flow:

1. Admin opens the donor list.
2. Admin selects a donor.
3. Admin submits removal request.
4. System removes donor record(s).
5. System displays a “Removed: [List of Names]” popup with a “Done” button.

Exception Flows:

- EF1: Admin exits the program.
- EF2: Database is inaccessible.
- EF3: Donor does not exist.

Postconditions: Donor is removed from the system.

UC4 – Update Amount Donated Description: Updates the donation amount associated with a donor.

Actors: Admin

Preconditions:

- Database is accessible and operational
- Fundraising events page and form are available

Basic Flow:

1. Admin opens the fundraising events page.
2. Admin selects "Update Donation."
3. System displays a form.
4. Admin updates required fields.
5. Admin optionally updates additional fields.
6. Admin submits the form.
7. System validates all fields.
8. System updates donation data in the database.
9. System displays confirmation with success notification.

Alternate Flows:

- AF1: If required fields are missing, the system highlights them in red and prompts correction. Flow resumes at step 4.
- AF2: If any field is invalid, the system highlights errors and prompts correction. Flow resumes at step 4.

Exception Flows:

- EF1: Admin exits the page before submission.
- EF2: Database update fails; system displays an error page.

Postconditions: Donation amount is updated in the system.

UC5 – Manage User Access Level Description: Modifies the access level of an existing user.

Actors: Admin

Preconditions:

- Database is accessible and operational
- User database with access levels exists

Basic Flow:

1. Admin opens the user management dashboard.
2. System displays all users with current access levels.
3. Admin changes a user's access level via dropdown.
4. System updates the database.

5. System displays confirmation of the update.

Alternate Flows:

- AF1: If elevating a user to Admin, the system prompts for confirmation ("CONFIRM"). Flow resumes at step 4.

Exception Flows:

- EF1: Admin exits the process.
- EF2: Database update fails; system displays an error page.

Postconditions: User access level is updated in the system.

UC6 – Backup Donor Information Description: Creates a backup of donor information for recovery purposes.

Actors: Admin

Preconditions:

- Database is accessible and operational
- Backup button is available

Basic Flow:

1. Admin logs in.
2. System displays a "Backup" button.
3. Admin clicks the button.
4. System saves the database state.
5. System displays backup status (success or failure).

Alternate Flows:

- AF1: If there is no data to save, the system displays "No data available."
- AF2: If admin cancels the operation, the system deletes the partial save and shows cancellation message.

Exception Flows:

- EF1: Database query fails; system displays an error page.

Postconditions: A backup file is created and stored.

UC7 – Add Amount Donated Description: Adds a donation amount for a donor.

Actors: Admin

Preconditions:

- Database is accessible and operational
- Fundraising events page and form are available

Basic Flow:

1. Admin opens fundraising events page.
2. Admin selects “Add Donation.”
3. System displays a donation form.
4. Admin enters required information.
5. Admin optionally enters additional details.
6. Admin submits the form.
7. System validates input.
8. System inserts data into the database.
9. System displays confirmation with success notification.

Alternate Flows:

- AF1: Missing required fields prompt correction. Flow resumes at step 4.
- AF2: Invalid fields prompt correction. Flow resumes at step 4.

Exception Flows:

- EF1: Admin exits the page before submission.
- EF2: Database insertion fails; system displays an error page.

Postconditions: Donation record is added to the database.

UC8 – Export Finance Data Description: Generates an exportable Excel file for auditing purposes.

Actors: Admin, Financial Auditor

Preconditions:

- Database is accessible and operational
- Actor has permission to access financial data

Basic Flow:

1. Actor opens the financial statistics page.
2. Actor selects “Generate Report.”
3. System displays filtering options.
4. Actor applies filters or selects all data.
5. System retrieves filtered data.
6. System exports data to an Excel file.
7. System displays the file location.

Alternate Flows:

- AF1: If filters are selected, only filtered data is included.

Exception Flows:

- EF1: Actor exits the process.
- EF2: Database query fails; system displays error page.

Postconditions: Excel report is generated and stored.

UC9 – Import Finance Data Description: Imports financial data from a valid Excel file into the system.

Actors: Admin, Financial Auditor

Preconditions:

- Database is accessible and operational
- Actor has permission to import financial data

Basic Flow:

1. Admin opens the financial statistics page.
2. Admin clicks “Import” button.
3. System displays a file dialog.
4. Admin selects an Excel file.
5. System validates file format.
6. System imports file data.
7. System displays completion message.

Alternate Flows:

- AF1: If a non-Excel file is selected, the system prompts for re-selection.

Exception Flows:

- EF1: Admin exits the process.
- EF2: File scan fails; system displays error page.

Postconditions: Financial data is added to the system.

UC10 – Search Donor Description: Allows an Admin or Analyst to search for donors using filters or search criteria.

Actors: Admin, Analyst

Preconditions:

- Actor is logged in as Admin or Analyst
- Donors exist in the system

Basic Flow:

1. Actor opens the search page.

2. System displays available search fields and filters.
3. Actor enters search criteria.
4. System retrieves matching donors.
5. Actor selects a donor to view details.

Alternate Flows:

- AF1: If unauthorized, the system denies access.
- AF2: If no search options are provided, all donors are displayed.
- AF3: If no matches are found, the system displays “No results.”

Exception Flows:

- EF1: Actor leaves or reloads page; search resets.

Postconditions: Search results are displayed or reset if interrupted.

UC11 – Get Automated Report Description: Generates a donor or fundraising report with filters and analysis options, output as a PDF.

Actors: Admin, Analyst

Preconditions:

- Actor logged in as Admin or Analyst
- Donor/fundraising data exists in the system
- Database and reporting module available

Basic Flow:

1. Actor opens “Generate Report” page.
2. System displays available options (filters, ranges, analysis type).
3. Actor selects options.
4. System validates selections.
5. System retrieves data and generates report.
6. System presents the PDF report.
7. Actor may view, download, or print.
8. System saves report history.

Alternate Flows:

- AF1: Unauthorized actor denied access.
- AF2: Missing required selections prompt correction.
- AF3: Empty data results in “No Data Available.”

Exception Flows:

- EF1: Actor exits before completion; selections discarded.
- EF2: Report generation fails; system displays error.

Postconditions: PDF report is generated and stored, or process fails with no report created.

UC12 – Quantitative Analysis Description: Performs quantitative analysis based on user-selected parameters.

Actors: Admin, Analyst

Preconditions:

- Database is accessible and operational
- Analysis options form is available
- Report output folder exists

Basic Flow:

1. System receives analysis request.
2. System generates graphs as specified.
3. System calculates statistics.
4. System saves results to output file.
5. System notifies interface of success.
6. System provides metadata about the report.

Exception Flows:

- EF1: Insufficient data; execution stops.
- EF2: Output folder missing; execution stops.

Postconditions: Quantitative report is generated and stored.

UC13 – Qualitative Analysis Description: Performs qualitative analysis based on user-selected parameters.

Actors: Admin, Analyst

Preconditions:

- Database is accessible and operational
- Analysis options form is available
- Report output folder exists

Basic Flow:

1. System receives analysis request.
2. System generates graphs as specified.
3. System calculates statistics.
4. System saves results to output file.
5. System notifies interface of success.
6. System provides metadata about the report.

Exception Flows:

- EF1: Insufficient data; execution stops.
- EF2: Output folder missing; execution stops.

Postconditions: Qualitative report is generated and stored.

UC14 – Edit Fundraising Event Description: Allows editing of existing fundraising events.

Actors: Admin, Fundraising Coordinator

Preconditions:

- Event already exists
- Database is accessible and operational
- Fundraising events page and form are available

Basic Flow:

1. Actor opens fundraising events page.
2. Actor selects "Edit Event."
3. System displays editable form.
4. Actor updates required fields.
5. Actor optionally updates additional details.
6. Actor submits changes.
7. System validates inputs.
8. System updates event in database.
9. System displays confirmation with success notification.

Alternate Flows:

- AF1: Missing required fields prompt correction. Flow resumes at step 4.
- AF2: Invalid fields prompt correction. Flow resumes at step 4.

Exception Flows:

- EF1: Actor exits page before submission.
- EF2: Database update fails; system displays error page.

Postconditions: Fundraising event details are updated in the system.

UC15 – Create Fundraising Event Description: Allows creation of a new fundraising event.

Actors: Admin, Fundraising Coordinator

Preconditions:

- Database is accessible and operational
- Fundraising events page and form are available

Basic Flow:

1. Actor opens fundraising events page.
2. Actor selects “Create Event.”
3. System displays event form.
4. Actor completes required fields.
5. Actor optionally fills additional details.
6. Actor submits the form.
7. System validates inputs.
8. System saves event to database.
9. System displays confirmation with success notification.
10. Actor returns to previous page.

Alternate Flows:

- AF1: Missing required fields prompt correction. Flow resumes at step 4.
- AF2: Invalid fields prompt correction. Flow resumes at step 4.

Exception Flows:

- EF1: Actor exits page before submission.
- EF2: Database insertion fails; system displays error page.

Postconditions: Fundraising event is created and saved in the system.

UC16 – View Fundraising Events Description: Allows viewing of all fundraising events.

Actors: Admin, Fundraising Coordinator

Preconditions:

- Database is accessible and operational
- Fundraising events page is available

Basic Flow:

1. Actor opens fundraising events page.
2. System retrieves and displays list of all fundraising events.
3. Actor selects an event to view details.

Exception Flows:

- EF1: Actor exits page before viewing.
- EF2: Database query fails; system displays error page.

Postconditions: Fundraising events are displayed to the actor.

UC17 – Thank Community Members Description: Generates and sends thank-you communications to donors or supporters.

Actors: Admin, Community Outreach Staff

Preconditions:

- Database is accessible and operational
- Donor/community member contact information exists
- Valid donation or engagement records exist

Basic Flow:

1. Actor opens “Community Outreach” page.
2. Actor selects “Generate Thank You.”
3. System retrieves eligible recipients.
4. Actor selects recipients.
5. System generates thank-you messages (document or email).
6. Actor reviews and confirms.
7. System sends messages and logs activity.

Alternate Flows:

- AF1: If no recipients exist, the system displays “No recipients available.”

Exception Flows:

- EF1: Database error occurs.
- EF2: Message sending fails; system logs error.

Postconditions: Thank-you messages are sent and logged.

UC18 – Alert Community Members Description: Sends alerts or announcements to donors/community members.

Actors: Admin, Community Outreach Staff

Preconditions:

- Database is accessible and operational
- Message templates or content are available

Basic Flow:

1. Actor opens “Alerts” page.
2. Actor creates or selects template.
3. Actor selects recipients or groups.
4. Actor sends or schedules message.
5. System validates contact details.
6. System delivers messages and logs activity.

Alternate Flows:

- AF1: Actor schedules alert for later delivery.

Exception Flows:

- EF1: Invalid contact info skipped with error log.
- EF2: Sending fails; system retries or notifies actor.

Postconditions: Alerts are delivered and logged.

UC19 – Generate Message Description: Drafts or auto-generates custom messages using donor/event data.

Actors: Admin, Community Outreach Staff

Preconditions:

- Database is accessible and operational
- Donor/event data available
- Message templates exist in the system

Basic Flow:

1. Actor selects “Generate Message.”
2. Actor chooses message type (email, letter, announcement).
3. Actor selects variables (e.g., donor name, donation amount).
4. System generates draft message.
5. Actor reviews draft.
6. Actor confirms and saves message.

Alternate Flows:

- AF1: Actor manually edits draft before saving.

Exception Flows:

- EF1: Missing data results in placeholders flagged for manual input.

Postconditions: Completed message draft is saved and ready to send.

UC20 – Send Email: Send finalized emails (thank-you, alerts, announcements) to recipients selected in UC17–UC19.

Actors: Admin or Community Outreach Staff

Preconditions:

- Outbound email service configured (SMTP)
- Recipient list validated and available
- A generated message is ready to send

Basic Flow:

1. Actor opens “Send Email” function.

2. Actor selects a saved/generated message.
3. Actor confirms recipients.
4. Actor clicks “Send.”
5. System validates recipients and message format.
6. System transmits email(s) through email service.
7. System marks messages as sent and logs activity.

Alternate Flows:

- AF1: Actor schedules the email to be sent at a future time..

Exception Flows:

- EF1: Invalid or unreachable email addresses are skipped and logged.
- EF2: If email server is unavailable, system queues email until service is restored.

Postconditions:

- Selected recipients receive the email.
- Email status recorded for tracking and reporting.

4.5 About Functional And Non-Functional Requirements

The functional and non-functional requirements outlined in this document define both what the system must do and how it must perform under various conditions. Functional requirements describe the specific behaviors, features, and processes the system must support, such as donor management, fundraising event handling, reporting, and communication with community members. Non-functional requirements, on the other hand, capture the quality attributes of the system, including performance, usability, accessibility, and reliability. Together, these requirements provide a comprehensive view of the system’s expected capabilities and ensure that development efforts address not only the essential functions but also the standards of efficiency and user experience necessary for effective operation.

4.6 Functional And Non-Functional Requirements

UC1: Add Donor

Functional Requirements

- The system shall provide a manual data-entry form for new records.
- The system shall validate required fields, formats, and uniqueness.

Non-Functional Requirements

- The system should provide inline validation errors.
- The system should save within 2 seconds.

UC2: Update Donor

Functional Requirements

- The system shall allow editing donor fields (email, phone, address, preferences).
- The system shall allow users with proper access to change records via a form.
- The system shall validate that updated data does not cause errors or overlaps.

Non-Functional Requirements

- The system should display a tooltip warning if required fields are missing.
- The system should report success or failure on the update.

UC3: Remove Donor

Functional Requirements

- The system shall allow Admins to locate and delete specific records.
- The system shall present a pre-delete review/confirmation step.
- The system shall block deletes that violate referential integrity unless cascades are confirmed.

Non-Functional Requirements

- The system should complete confirmed deletes within 2 seconds for typical records.
- The system should display clear risk warnings and the count of affected relations.

UC4: Update Amount Donated

Functional Requirements

- The system shall track consent/opt-in status and update downstream communication lists.

Non-Functional Requirements

- The system should propagate updates to integrated systems within 5 minutes.
- The system should enforce field-level access controls.

UC5: Manage User Access Level

Functional Requirements

- The system shall allow admin users to control other users' access to the system.
- The system shall allow admin users to have full access to the system.

Non-Functional Requirements

- The system should allow admin users to view the access level of all users.
- The system should provide an intuitive interface for modifying access levels.

UC6: Backup Donor Information

Functional Requirements

- The system shall provide on-demand full backup initiation by Admins.
- The system shall log all backup and restore actions.

Non-Functional Requirements

- The system should ensure data integrity during backup processes.
- The system should notify users of success or failure.

UC7: Add Amount Donated

Functional Requirements

- The system shall allow authorized users to add a new donation to a donor.
- The system shall properly associate donations with the correct donor.

Non-Functional Requirements

- The system should provide a form to select a donor and input a donation.
- The system should request confirmation via a popup showing changes before completion.

UC8: Export Finance Data

Functional Requirements

- The system shall provide a page to manage exporting files.
- The system shall provide a display page containing possible filters.
- The system shall provide an audit/generate report button.
- The export page shall be the same as, or integrated with, the import page.
- The system shall save the file to the user's computer at a chosen location.
- The system shall ensure the integrity of exported data.

Non-Functional Requirements

- The system should notify the user where the file is saved.
- The system should provide a cancel button.
- The system should display a confirmation before export.

UC9: Import Finance Data

Functional Requirements

- The system shall provide a page to manage importing files.
- The import page shall be the same as, or integrated with, the export page.
- The system shall allow multiple files to be imported.
- The system shall allow Admins to remove files from import.
- The system shall ensure the integrity of imported information.

Non-Functional Requirements

- The system should have acceptable import speed.
- The system should display a popup tracking import progress.
- The system should provide smooth transitions between popups.
- The system shall store imported information appropriately.

UC10: Search Donor

Functional Requirements

- The system shall allow users to search for specific donors.
- The system shall display search results in a legible format.

Non-Functional Requirements

- The system should retain the last search query to improve user experience.
- The system should provide autocomplete with tab confirmation.

UC11: Get Automatic Report

Functional Requirements

- The system shall provide a button on the main page linking to the report generation page.
- The system shall provide buttons to generate either a quantitative or qualitative report.

Non-Functional Requirements

- The system should provide a table of past reports that users can view/export.
- The system should include accessibility features such as zoom and high-contrast mode.
- The system should provide a quick report option for faster, less detailed summaries.

UC12: Quantitative Analysis

Functional Requirements

- The system shall provide a form for users to select analysis options.
- The system shall calculate descriptive statistics of donation amounts and produce graphs over time.
- The system shall display the report on the website.
- The system shall allow users to export the report to a Word/PDF document.

Non-Functional Requirements

- The system should provide a recommended report option without requiring manual selection.
- The system should generate reports within 2 minutes.
- The system should allow users to select graph style palettes.

UC13: Qualitative Analysis

Functional Requirements

- The system shall provide a form for users to select analysis options.
- The system shall identify the most frequent donors and those who have not donated recently.
- The system shall determine top donating locations.
- The system shall display the report on the website.
- The system shall allow users to export the report to a Word/PDF document.

Non-Functional Requirements

- The system should provide a recommended report option without requiring manual selection.

- The system should generate reports within 2 minutes.
- The system should allow users to select graph style palettes.

UC14: Edit Fundraising Event

Functional Requirements

- The system shall allow editing of event fields (name, start date, end date, goal, description).
- The system shall revalidate constraints on save.
- The system shall not impact users who have already donated.

Non-Functional Requirements

- The system should display an impact preview (e.g., timeline changes).
- The system should save within 2 seconds.

UC15: Create Fundraising Event

Functional Requirements

- The system shall allow coordinators to create events with name, start date, end date, and goal.
- The system shall validate time ranges and nonnegative goals.

Non-Functional Requirements

- The system should provide timezone-aware date/time pickers.
- The system should support both military and standard time formats.
- The system should prevent duplicate overlapping events with gentle prompts.

UC16: View All Fundraising Events

Functional Requirements

- The system shall provide a button linking to a calendar page.
- The system shall display a dynamic calendar showing stored fundraising events.
- The system shall allow users to select an event on the calendar and view details on a new page.

Non-Functional Requirements

- The system should provide zoom or font-size adjustments for accessibility.
- The system should retain a record of past events accessible through the calendar.

UC17: Thank Community Members

Functional Requirements

- The system shall provide a button to generate thank-you notes for donors.
- The system shall use an LLM to generate customizable notes.

Non-Functional Requirements

- The system should allow users to proofread notes before sending.
- The system should support generating notes per event or per donor.

UC18: Alert Community Members

Functional Requirements

- The system shall provide a button to send donation requests to community members.
- The system shall provide an automated protocol for event-related outreach.

Non-Functional Requirements

- The system should prevent duplicate outreach emails within a defined timeframe.
- The system should produce emails with clear structure, concise language, and appropriate formatting.

UC19: Generate Message

Functional Requirements

- The system shall receive either a “thank you” or “alert” prompt from a user form.
- The system shall use a model (e.g., ChatGPT) to generate a message from the prompt.
- If “review message” is selected, the system shall pause and display the message for approval.
- The system shall allow users to cancel generation or sending if the message is unsatisfactory.

Non-Functional Requirements

- The system should include zoom and high-contrast accessibility features.
- The system should include content filters to prevent offensive or inappropriate messages.
- The system should provide a quality score highlighting issues such as grammar, tone mismatch, or sensitivity.

UC20: Send Email

Functional Requirements

- The system shall use an LLM to generate emails to community members.
- The system shall allow users to select an email purpose (e.g., thank-you, outreach).

Non-Functional Requirements

- The system should allow users to proofread emails before sending.
- The system should allow regeneration of unsatisfactory emails.

5 Assumptions

5.1 List of Assumptions Made By The Project

- The system will be used exclusively by Rappahannock CASA staff and authorized personnel; public access is not intended or supported.
- All users will have basic computer literacy and familiarity with web-based applications.

- The organization will provide timely feedback and access to sample data during development and testing phases.
- The system will be hosted on secure infrastructure with regular backups and maintenance handled by CASA or its designated IT provider.
- The initial admin account will be created by the development team and handed off to CASA leadership for future access control.
- All imported Excel files will follow the formatting guidelines provided in the Import/Export Page to ensure compatibility.
- The system will be accessed primarily via desktop or laptop devices using the Chrome browser.
- CASA will maintain responsibility for ongoing user training, data entry accuracy, and content moderation after deployment.
- External services used for email generation or reporting (e.g., LLMs) will remain available and functional during system operation.
- The system will not be integrated with third-party financial platforms (e.g., QuickBooks) beyond file import/export compatibility.
- The development team will complete all deliverables by the end of the Fall 2025 semester, with no post-semester support guaranteed.
- The system will comply with ADA accessibility standards, but advanced accessibility features (e.g., screen reader optimization) may require future enhancements.
- CASA will designate at least one staff member to serve as the internal system administrator post-deployment.

5.2 Glossary of Terms Used Throughout Document

Access Control: A mechanism that restricts or grants permissions to users based on their role or authorization level.

Actor: An external entity (such as a user or system) that interacts with the system.

Admin (Administrator): A user role with full or elevated privileges to manage system features, data, and access.

Admin Dashboard: A secure interface where system administrators can manage users, view system metrics, and configure settings.

Audit: A process of reviewing system activity, often including logging, verification, and reporting for compliance or debugging.

Authentication: The process of verifying a user's identity before granting access to the system.

Authorization: The process of determining what actions a verified user is allowed to perform within the system.

Autocomplete: A feature that predicts and suggests the remainder of a user's input (e.g., search queries) as they type.

Backend: The part of the system that handles data processing, storage, and logic behind the scenes. Users don't interact with it directly.

Backup: A copy of system data stored to prevent loss in case of system failure, corruption, or deletion.

Bug: An error or flaw in the software that causes it to behave unexpectedly or incorrectly.

Calendar (Dynamic Calendar): An interactive interface displaying events, often with the ability to navigate, zoom, and select items for more information.

Cascade Delete: A database operation where deleting a record automatically removes related records to maintain consistency.

Confirmation Popup: A dialog box requiring the user to confirm or cancel an action before it proceeds.

Consent/Opt-In Status: A user's explicit permission to receive communications or have their data processed.

Content Filters: Automated checks that block or flag text containing inappropriate, sensitive, or restricted material.

CRUD Operations: Acronym for Create, Read, Update, and Delete — the four basic functions of data management in software systems.

Database: A structured collection of data that the system uses to store and retrieve information efficiently.

Data Integrity: The accuracy, consistency, and reliability of data throughout its lifecycle.

Deployment: The process of making the system live and accessible to users.

Descriptive Statistics: Statistical measures (e.g., mean, median, variance) that summarize and describe a dataset.

Downstream Systems: Systems that receive or depend on updates or outputs from another system.

Dynamic Calendar: A calendar interface that updates automatically based on stored or incoming data.

Export: The process of transferring data out of the system into a file format usable externally.

Field-Level Access Controls: Restrictions applied at the individual data-field level, determining who can view or edit each piece of information.

Filters (Data Filters): Criteria applied to limit or refine the data displayed or exported.

Form (Data Entry Form): A structured interface where users enter information into predefined fields.

Frontend: The part of the system users interact with directly — includes buttons, forms, and visual elements.

Functional Requirement: A specification of what the system must do (features, actions, or behaviors).

High-Contrast Mode: An accessibility feature that increases color contrast for readability by users with visual impairments.

Import: The process of bringing external data files into the system for storage or processing.

Inline Validation: Immediate feedback provided within a form to indicate errors in entered data.

LLM (Large Language Model): An AI model trained on vast amounts of text to generate or process natural language (e.g., ChatGPT).

Module: A self-contained component of the system that performs a specific function (e.g., Volunteer Management, Case Tracking).

Non-Functional Requirement: A specification of how the system should perform or behave (e.g., speed, usability, accessibility).

Opt-In: The act of a user actively agreeing to participate in or receive communications/services.

PDF Generation: The ability to create downloadable PDF documents from system data, such as reports or summaries.

Preconditions: Conditions that must be satisfied before a use case or function can start.

Qualitative Analysis: Examination of non-numeric data to identify themes, patterns, or trends (e.g., donor behavior by region).

Quantitative Analysis: Examination of numeric data using statistics, often resulting in charts or numerical reports.

Referential Integrity: A database property ensuring that relationships between records remain consistent (e.g., foreign key constraints).

Report Generation: The automated process of creating structured documents summarizing data (quantitative or qualitative).

Role-Based Access Control (RBAC): A security feature that restricts system access based on a user's assigned role (e.g., Admin, Volunteer).

Search Query: A string of keywords or text entered by the user to search for information in the system.

Success/Fail Report: A notification indicating whether a system operation was successfully completed or failed.

System Requirements Document: A formal outline of what the system must do, how it should behave, and what constraints it must operate under.

Time Zone–Aware Picker: A date/time input control that adjusts for and supports different time zones.

Tooltip: A small pop-up message providing context or guidance when hovering over or interacting with an element.

Uniqueness (Field Uniqueness): A validation rule ensuring that a value (e.g., email address) is not duplicated in the database.

User Interface (UI): The visual part of the system that users interact with, including menus, buttons, and forms.

Validation: The process of checking that user input meets specified rules (format, completeness, uniqueness, etc.).

Zoom (Interface Zoom): An accessibility feature allowing users to enlarge or scale parts of the interface for better visibility.

Workflow: A defined sequence of steps or tasks that users follow to complete a process in the system.

5.3 Other Documents

At this point there are no other documents to reference.

6 Author Contact Information

6.1 Main Contact

Ethan Bostick: ebostick@mail.umw.edu (main)
ethan.bostick@gmail.com (personal)
(+1) 571-314-0071 (mobile)

6.2 Secondary Contact

Garrett McKenzie: gmckenzi@mail.umw.edu (main)
garrett.david.mckenzie@gmail.com (personal)
(+1) 571-332-5220 (mobile)

6.3 Other Contacts

Maximilian Redman: mredman@mail.umw.edu
Carter Walker: cwalke28@mail.umw.edu
James Heathcock: jheathco@mail.umw.edu
Joshua Byrne: jbyrne@mail.umw.edu