

# P03 – T2 – Management System for Nonprofits

Final Report

CS 4850-04, Fall 2025

Evan Conley, Tyler Stroud, Ella Goode, Fatih Kurt

Professor Sharon Perry

December 5<sup>th</sup>, 2025

Project Website: <https://p03-t2mgmtsysnp.github.io/>

Project GitHub: <https://github.com/P03-T2MgmtSysNP> (for the website)

STATS AND STATUS	
Project is under NDA	
Lines Of Code	1,139 lines of .CSV data from HubSpot
Components/Tools	1 - HubSpot
Hours Estimate	350
Hours Actual	362
Status	Project is 100% complete and under consideration by sponsors

# Table of Contents

1. Introduction.....	4
<b>1.1.</b> Background .....	4
<b>1.2.</b> Objectives .....	4
<b>1.3.</b> Scope .....	4
2. Challenges .....	5
3. Requirements.....	5
3.1.    Functional Requirements.....	5
3.1.1.    Platform Assessment.....	5
3.1.2.    POC System .....	6
3.1.3.    Documentation .....	6
3.2.    Non-Functional Requirements .....	7
3.2.1.    Security .....	7
3.2.2.    Capacity.....	7
3.2.3.    Usability .....	8
4. Analysis/Design.....	8
4.1.    Assumptions and Dependencies.....	8
The software will have integration with Google workspace .....	8
The user will access the software using either Windows 10 or 11 operating systems	9
The system will be regularly updated and logged into by the End User .....	9
4.2.    General Constraints.....	9
4.3.    Detailed System Design .....	11
4.3.1.    Platform Evaluation & Selection Subsystem .....	11
4.3.2.    Data Model & Configuration Subsystem .....	11
4.3.3.    Integration & Automation Subsystem .....	12
4.3.4.    Architecture Drawing .....	13
5. Development.....	14
5.1.    Development Methods.....	14

5.2.	Architectural Strategies.....	14
5.3.	Database Connection .....	15
5.4.	Project Setup .....	15
<b>6.</b>	Testing .....	18
6.1.	Scope of Testing.....	18
6.1.1.	In Scope .....	18
<b>6.1.2.</b>	Out-of-scope .....	18
6.2.	Test Cases .....	19
6.3.	Test Procedures .....	19
	<i>For Test Cases TC-SI-01 to TC-SI-03.....</i>	19
6.4.	Test Environment .....	20
6.5.	Test Data .....	21
6.6.	Software Test Report (STR) .....	21
7.	Version Control .....	22
8.	Summary.....	22
9.	Appendix .....	23
	Project Plan .....	23
	Deliverables .....	23
	Expected Deliverables: .....	23
	Group Meeting Schedule Date/Time .....	23
	Collaboration and Communication Plan .....	24
	Risk Assessment .....	24
	Source Code.....	24

# 1. Introduction

## 1.1. Background

Georgia Laws of Life Essay Contest is a 25-year-old 501(c)(3) nonprofit that facilitates a statewide character education program in partnership with high schools and Rotary clubs. The program engages students through an annual essay contest, having collected over 800,000 essays to date. However, the organization currently manages its operations using disconnected spreadsheets, making it difficult to track student participants, school contacts, Rotary partners, donors, and overlapping events. They are uncertain whether a CRM, donor management, or project management tool best suits their needs. This project will evaluate available free/freemium tools, propose a suitable solution, and deliver a proof-of-concept system tailored to their operations.

## 1.2. Objectives

The primary objective of this project is to analyze the nonprofit's operational workflows and stakeholder relationships and to identify, evaluate, and prototype a suitable centralized management solution. This will be achieved by assessing freemium Software-as-a-Service (SaaS) platforms such as Zoho CRM, HubSpot CRM, Airtable, and Trello. These platforms will be utilized for CRM, donor management, and project/event management functionality. The project will deliver a proof-of-concept system configured to demonstrate centralized tracking of students, schools, Rotary clubs, and donors along with management of entity relationships and communications and coordination of event workflows. These functionalities will be integrated with the organization's existing Google Workspace environment.

## 1.3. Scope

The project scope includes requirements analysis and entity mapping, comparative evaluation of SaaS platforms, and the development of a proof-of-concept utilizing a selected tool. The implementation will demonstrate core functionalities, including data import/export via the Google Sheets API and event tracking. Final deliverables will include a platform evaluation matrix, a configured proof-of-concept, and a complete implementation of the roadmap. The project explicitly excludes any custom software development or the use of paid software

licenses. With that in mind, the focus will solely be on the evaluation and configuration of existing freemium SaaS solutions.

## 2. Challenges

The major challenges associated with this project, prior to even beginning development of requirements, were largely related to the nature of the project as consultancy work. Given that a computer science senior design course expects certain things of a project that ours would simply not include, it was a challenge to understand how to demonstrate our work effectively, and meet the criteria required to show proficiency in this course. Our work did not include the same stack selection that other projects did, for example. Instead, we were searching for a pre-built solution and conducting analysis, rather than designing our own from scratch.

A challenge that was encountered during the process of requirements development was uncertain and shifting requirements from the industry partner. We had received initial documents before our first meeting that explained their wishes for the project, but later found these documents to be inaccurate. The way we handled this challenge was by maintaining consistent contact with the sponsors, and routinely checking in with them, both during our biweekly meetings and over email. This allowed us to catch issues and inconsistencies by having sponsor review before moving between stages.

## 3. Requirements

### 3.1. Functional Requirements

#### 3.1.1. Platform Assessment

- 3.1.1.1. The platform assessment shall define a means of platform evaluation.
- 3.1.1.2. The platform evaluation shall be weighed by client requests and priorities.
- 3.1.1.3. The platform evaluation shall consider the impact of inclusion of optional features, namely dashboarding, bulk import, and follow-up reminders.

3.1.1.4. The platform evaluation shall be quantitative in nature.

3.1.1.5. The platform assessment shall contain an analysis of all evaluated platforms based on their quantitative assessment.

3.1.1.6. The platform assessment shall contain a final recommendation for a specific platform and justification for its selection.

### 3.1.2. POC System

3.1.2.1. The POC shall utilize the platform selected from the platform assessment.

3.1.2.2. The POC shall use simulated data.

3.1.2.2.1. Simulated data shall follow the same format as current data used by the company.

3.1.2.2.2. Simulated data shall be placeholders with no real human data used.

3.1.2.3. The POC shall demonstrate the workflows of event tracking, communication, and data import/export.

3.1.2.4. The POC shall be interactable live.

### 3.1.3. Documentation

3.1.3.1. Documentation shall contain guidance on the usage of platform options.

3.1.3.1.1. The final recommendation shall have a more extensive usage guide provided.

3.1.3.2. Documentation shall contain the platform assessment.

3.1.3.3. Documentation shall include information on upgrade options for reviewed platforms.

3.1.3.4. Documentation shall be provided to the company.

3.1.3.5. Documentation shall be readable and relevant to the project.

## 3.2. Non-Functional Requirements

### 3.2.1. Security

Most of the security requirements will be based on the SaaS that we choose. Certain security requirements will be wanted, such as industry standards like SOC 2 and ISO 27001. This shows that they are active in trying to make their software as secure as possible. Since our product does not require much in terms of us to develop security protocols, we need to ensure that the products that we choose have security measures in place. Since the data that the industry partners do have sensitive information, it is important to consider the need for best security practices. Information like a donor list and personal contact information can have major implications if it is to be involved in a data breach. This is why it is important that we find a software product that is reliable and meets certain standards set by industry. The best practice ensures that the industry partner's data is in trusted in safe hands and meets the requirements set by the industry so that in the case of a potential breach, the partner can be at ease knowing that everything has been done on the software companies end to prevent security breach.

### 3.2.2. Capacity

The system needs to be able to handle a limited number of concurrent users and to be able to handle the data being stored in the database from the spreadsheets that were given to the group. According to the industry partners, the max number of users who would be using the product is currently at 3. The software as a service needs to handle the capabilities of being able to let 3 users update the database and use the service at the same time, in the same environment. The software will need to have room to grow for the industry partners. The industry partners have expressed interest in the potential for them to increase staff, which will mean the need for the software to be able to handle the growing number of users is also an important consideration to be made.

Another capacity requirement is the need for enough memory to handle the growing database. Once the current spreadsheets are turned into a database to be

used, memory for the database will be an important requirement. We want to make sure that there is enough memory not only for the current database that we plan to implement in the software but also must be able to handle updates from the users. The last thing we want to have is a situation where the database has run out of memory and no longer can be updated. The database will be at the core of what the industry partner's main goal is, and importance needs to be placed on current and future memory considerations.

### 3.2.3. Usability

Since the users of the finished product are not technically advanced, the product needs to be usable for everyone, no matter what the level of technical skills. It will be important to consider the different SaaS not only in terms of what it provides in terms of software but is the user able to achieve the same result with the limited technical background they have. The product should be able to be used after some time and training and should not be overcomplicated for the user. After a week, the user should be able to start using the software on their own, with minimal help from the developers of the product. Another thing that is important is the ease of returning to the system after a break. The user should be able to take a break, like a vacation, and be able to return to the system with limited issues with navigating the system.

Another usability requirement will be that the system should be able to run at a steady rate. The loading times and speed should be reasonable and accepted to the end user. An important factor in using software is the speed at which the user can complete the task they want to complete. If the loading and downloading times take too long, the user will not want to use the software as they feel like it is delaying them from achieving the work that they want to get finished.

## 4. Analysis/Design

### 4.1. Assumptions and Dependencies

*The following assumptions can be made about the software:*

**The software will have integration with Google workspace**

The industry partner uses Google workspace to do all their current work. The software that is chosen as the best option for their project will be under the assumption that Google workspace has been integrated into the software and there will be a way to grab data from Google Spreadsheets and be able to transfer this



data into the software for them to see. Any method of data transferring will be allowed, but the preferred method will be the Google Docs API to transfer data to the CRM or Project Management Software.

#### The user will access the software using either Windows 10 or 11 operating systems

The industry partners do not use Apple Mac and have up to date Windows upgrades. All the software that is looked at for this solution will be able to run on Windows 10 or Windows 11. This will ensure quality control as if the software works on one Windows system during testing, then replicating it to work on other Windows systems will be the goal to determine if the software is functional.

#### The system will be regularly updated and logged into by the End User

The chosen software product will require that updates and changes be made by the End User. To deem the software as the optimal solution, the End User will be made aware that the success of the software is determined by the ability to be active and response to it. For example, for every change in the database that the customer will want to make, they will need to be active and update the database to reflect the change they want to make. Not doing so will harm their experience as the software will no longer have the most up to date database that the industry partner is looking for

## 4.2. General Constraints

### *Hardware or software environment*

One of the constraints that we faced on the project is the software environment. Since the industry partner has specificized what software they expect to use, we cannot create a custom product that fully meets the needs of the partner. We instead need to rely on the listed software as a service product that they are expecting to see. These services are limited to what they provide and may not provide everything that the partner needs. The final product may be split across multiple services, to fulfill the needs of the industry partners. We also may be limited in the hardware as some of these services may only be available on Windows operating systems and will have trouble running on Linux or Mac operating systems.

### *End-user environment*

The end-user environment needs to be designed in a way that is user friendly so that anyone is able to use the product. It needs to have a user interface that can be easily learned and trained to be used by anyone.

### *Security requirements (or other such regulations)*

Certain security standards like SOC 2 and ISO 27001 need to be considered when looking at Software as a Service. Since we cannot implement security measures on our end, we need to rely on the Software we are using to have this security. This is important for the nonprofit as they have sensitive data that will be an issue if it gets leaked from a breach. By focusing on finding software that is designed with security in mind, we can ensure that we have software that practices the best methods to ensure that the data stored by the industry partner is secure.

### *Performance requirements*

The software needs to be able to perform for the maximum number of users who use the software at the same time. The maximum number of users according to industry partners is currently 3. The software needs to have performance to be able to run at the busiest time, when all 3 users are on the software at the same time trying to do all the work on the software. There should not be any major performance delay when users log on to the software when others are working on the software. New screens should be loaded promptly and when they do not, they should display a message saying that it is taking longer than usual.

When it comes to updating the database that is planned, small changes to the database should happen quickly and updates should show in a timely manner. Larger changes should take longer but should be scalable to smaller changes. A larger change can take longer to update, but it should be done in a reasonable timeframe that is linear in time to making a small change. The software should be able to identify when a small change is happening and when a large change is happening and be able to allocate any resources necessary to make the change in an appropriate timeframe.

### *Network communications*

Using software as a service means that we need a consistent and reliable network connection. Whereas some software that is custom made could be used offline, these services are often connected to the internet and use some sort of cloud computing, which is why internet connection is important. Without a constant internet connection, the service may not be available or lack performance due to a poor internet connection.

## 4.3. Detailed System Design

### 4.3.1. Platform Evaluation & Selection Subsystem

- **Classification:** Process / Methodology
- **Definition:** A structured framework for comparing and scoring shortlisted SaaS platforms against a weighted set of criteria derived from sponsor requirements.
- **Constraints:** Evaluation is limited to features available in freemium tiers and publicly documented API capabilities.
- **Composition:** Criteria include Cost (30% weight), Google Workspace Integration (25%), Entity Relationship Flexibility (20%), Ease of Use (15%), and Feature Set (10%).
- **Processing:**
  - Research and shortlist platforms (Zoho CRM, HubSpot CRM, Airtable).
  - Score each platform (0-5) on each criterion.
  - Calculate total weighted score for each platform.
  - Select the platform with the highest score for POC development.
- **Interface/Exports:** Platform Evaluation Matrix (Deliverable: A spreadsheet documenting scores and rationale).

### 4.3.2. Data Model & Configuration Subsystem

- **Classification:** Configured SaaS Application Modules

- **Definition:** The designed and implemented data structure within the selected SaaS platform that represents the sponsor's entities and processes.
- **Composition:** Configured modules will include:
  - Contacts: To manage School Contacts, Rotary Members, Donors.
  - Companies: To represent Schools and Rotary Clubs.
  - Deals/Pipelines: To represent the Essay Contest lifecycle (e.g., Submitted, Judged, Won).
  - Projects/Events: To manage contest cycles and special events.
- **Resources:** Freemium account on the selected SaaS platform. Simulated data sets.
- **Uses/Interactions:** Users will interact with these modules via the platform's UI to view, create, edit, and relate records.
- **Interface/Exports:** A configured, live POC environment. User Guide documenting the data model and navigation.

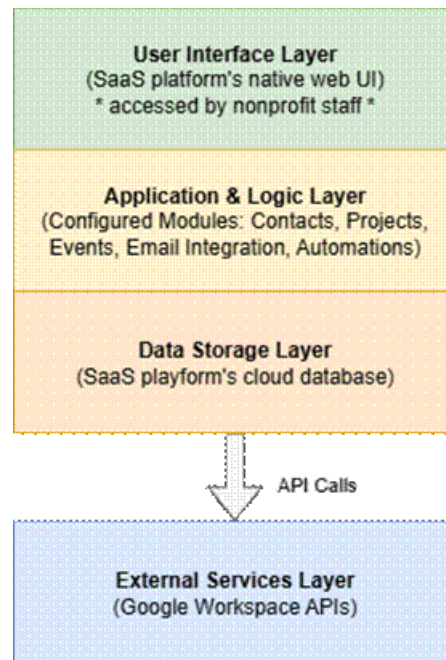
### 4.3.3. Integration & Automation Subsystem

- **Classification:** API-based Design Specification
- **Definition:** The blueprint for connecting the SaaS platform to Google Workspace to facilitate data import/export.
- **Constraints:** Limited by the API call limits and features of the freemium tier.
- **Processing:**
  - Design a standardized Google Sheet template for bulk data entry.
  - Document the manual process for exporting data from the SaaS platform to Sheets for reporting.
  - If supported, design a simple automation (e.g., using native platform automation or Zapier) to add a new Contact from a Google Form submission.

- **Interface/Exports:** Designed Google Sheet Templates. Integration Guide documenting the step-by-step processes for data transfer.

#### 4.3.4. Architecture Drawing

- The system architecture is conceptualized as a layered SaaS model, centered around the configured platform:



- The overall project is decomposed into three logical subsystems:
  1. **Platform Evaluation & Selection:** The process for choosing the right SaaS tool
  2. **Data Model & Configuration:** The design and setup of data structures within the chosen platform
  3. **Integration & Automation:** The design of data flow between the platform and Google Workspace

## 5. Development

### 5.1. Development Methods

The Development Methods for the project will be an Agile approach with UML-based modeling. When looking at and determining the best approach for development, this is the best method for the project given that we are looking at multiple CRM or Project Management tools and need a methodology that would allow us to shift from one SaaS to another if something was not working. Agile approach of doing development in increments and then looking at feedback will allow us to work on a SaaS for a while and if it is not working for the industry partners' standards, quickly shift to another approach. UML modeling is important because we need to clearly define who is doing what to the system and how they should interact with the system. By creating a proper UML model, it will help visualize the roles that each End User will have and how they interact with the system.

Other methods that were not considered for this project include the Waterfall and Spiral Model. The Waterfall method was not used as it was outdated and does not give the flexibility needed to do this project. For example, the waterfall does not allow for an easy way of going back once you have completed a phase, which means that we could have run into the issue of choosing a software that was not going to work and have no easy way of returning back to the previous phase.

The Spiral Model had the opposite issue when looking at it. Since the spiral model places an emphasis on risk analysis, we felt that we would spend too much time researching the best Software as a Service to use and not have enough time to implement it. This prevented us from choosing this model as the best option and led us to choose the Agile approach as the ideal approach to doing this project.

### 5.2. Architectural Strategies

The primary architectural strategy is to leverage configured SaaS over a custom build.

**Decision:** To utilize an existing, cloud-based SaaS platform for CRM, donor, and project management functionality.

**Reasoning:** This strategy directly satisfies the core constraints of \$0 cost and no custom development. It provides an immediately available, supported, and scalable foundation that can be tailored to the sponsor's needs through configuration.

#### Alternatives Considered:

**Custom-Built Database Application:** Rejected due to prohibitive development cost, long-term maintenance burden, and explicit project scope exclusion.

**Enhanced Google Sheets Setup:** Rejected as it does not fundamentally solve the problem of disconnected data and lacks native support for complex entity relationships, robust access controls, and automated workflows.

**Key Mechanisms:** The architecture will rely on the native data models, relationship modules, automation rules, and API endpoints of the chosen SaaS platform.

### 5.3. Database Connection

A database connection is not applicable to our project. Given that we are analyzing extant solutions which come with their own data solutions, we did not create this for any of the products we reviewed.

### 5.4. Project Setup

1. Create an account on HubSpot.com
2. In the home menu navigate to the side bar on the left side of the screen
3. Midway down on the side bar, you should see a section titled Data Management, stay in this menu
4. Click on the tab in Data Management titled Data Integration
5. In the Data Integration menu, we want to click on the option to Import a File. You should see a tab named "Import a File" with a button at the bottom titled "Import Data". Click on Import Data

6. In the new screen that pull up, click on the black button on the page titled “Start import.” Optionally, if you have completed these instructions before, you can also opt to use the “Repeat a past import” button to redo an import of a previous database.

7. HubSpot should bring you to a new screen titled “What kind of data is in your file?” There are many options for different applications based on the needs of every company. For our projects, select the box “Contacts”

8. After selecting the “Contacts” box, you should see at the bottom of the screen the “Next” button lights up. Click on the “Next” button

9. This should bring you to the “Upload your files” menu with many different options. The following are the setting needed to properly set up the project

- On Choose how to import Contacts keep the current option “Create and update Contacts” in the drop-down menu
- In Contact file click “choose a file” and enter either a .csv, .xlsx, or .xls file to download
- Leave the box uncheck under the Same-object associations
- Under Select the language of the column headers in your file, keep “English” as the selected option

10. Click on the “Next” Button to advance to the next screen

11. You should now be in the “Map columns in your file to contact properties” The following options are present:

- Mapping Guide: This will give you an overview of what is needed to import the data into HubSpot based on options provided from step 7. Since we chose contacts, we either need to map one of three required properties which are: Email, First Name, Last Name
- Below is the number of errors that are found. This is provided by HubSpot so that it ensures smooth translation and includes errors such as improper format and missing values. You must resolve these errors to ensure you can import the data.



12. Next is the All-Mapped Status menu. All the rows must have a green check mark under the Mapped column to be able to move to the next screen. The columns are as followed:

- Column Header from File: This is automatically filled in by HubSpot. On your spreadsheet, it contains and reads the list of headers from each column and stores them in this field. To update these, you need to make the change in the original spreadsheet and reupload the sheet into HubSpot

- Preview Information: This is automatically filled in by HubSpot. It matches the Column header with the data from the spreadsheet and allows the user to view the data that will be entered under the fields in HubSpot

- Mapped: This column shows three options. Green means the column is good and ready to be entered. Yellow means there is an error and includes an error option to click on to directly resolve the issues that are at the bottom. Last is blank, which means the column is not mapped, but no errors are present.

- Import As: This section allows the user to exclude any information from HubSpot that they do not feel is necessary. By selecting the option “Don’t import Column” the row is automatically marked as mapped, and the next two options are greyed out.

- HubSpot Property: This is how a user can find the field once it has entered HubSpot. This is the column name that will be found in HubSpot. You can create this in either of the two ways. You can use a preset option that is in HubSpot or create your own property using the “Create new property” to better fit the needs of your column names.

- Manage Existing Values: This is only if you have existing data; you can either check the box if you do not want the new data replacing the old data.

12. Once all rows have been successfully mapped, click the “Next” Button at the bottom right of the screen. This will import the data into HubSpot.

13. Return to the home screen of HubSpot

14. On the left side menu, navigate to the section label “CRM”. Click on the first option in this menu named “Contacts”

15. You should see the data from the spreadsheets entered HubSpot. You can rearrange the columns and add columns as needed from the data you have entered. This is the end of the instructions on how to set up the project so that you can start the HubSpot experience and set it up to create a successful CRM.

## 6. Testing

### 6.1. Scope of Testing

#### 6.1.1. In Scope

When it comes to HubSpot, the test plan will involve the tools of HubSpot that are most useful for the industry partner. The test plan will include testing on the spreadsheet integration. The test will include examining the results of trying to put in spreadsheets from both Microsoft Excel and Google Spreadsheets as well as testing the ability to use .csv files, as this is another common method for data entry into a database.

Another feature that will be tested is email integration. The industry partner currently uses Mailchimp for marketing emails and sending out mass emails. This is important to be featured when looking at HubSpot. The test will include not only the downloading and installation of the Mailchimp extension in HubSpot, but also the ability to use Mailchimp while in HubSpot and the ensuring that it works to the standards of the group and the industry partners.

The last feature that we plan on testing is the addition of new contacts in HubSpot. Currently, when the partner has a new contact, they need to put them on the spreadsheet. HubSpot has a place to add new contacts to which testing will be done. Testing it for edge cases and situations where not all information is present to see if there is anything that needs to be modified or updated or can HubSpot handle these issues.

#### 6.1.2. Out-of-scope

The features that will not be tested are anything in HubSpot that does not directly relate to the needs of industry partners. There are many features that HubSpot has for all different sorts of businesses. Some of these features are of no use to industry partners. For example, HubSpot has a full section dedicated to marketing. This is something that industry partners have said they have no interest

in exploring as they already have a dedicated resource for marketing. These features will not be tested or explored at this time. The features that will not be tested are to be assumed not useful to handing off the project to the industry partner.

## 6.2. Test Cases

Test Case ID	Description	Expected Result
TC-SI-01	Import a small (10-row) excel spreadsheet	All 10 elements are accurately imported and visible in their respective HubSpot list
TC-SI-02	Import industry partner's provided test data in a .csv format	All partner contacts are accurately imported with correct data mapping
TC-SI-03	Import a large (2000-row) CSV file	All 200 contacts are processed and imported into HubSpot
TC-MC-01	Install the Mailchimp extension from HubSpot Marketplace	Extension installs successfully without errors and appears in the HubSpot navigation menu
TC-CR-01	Manually create new contact with all required fields populated (First Name, Last Name, Email, etc.)	Contact is saved successfully and appears in the contact list with all provided information

## 6.3. Test Procedures

*For Test Cases TC-SI-01 to TC-SI-03*

- I) Log in to HubSpot portal
- II) Navigate to Contacts > Import
- III) Click Start and import and select the appropriate file (Excel, or CSV)
- IV) Map the spreadsheet columns to the correct HubSpot token properties

- V) Click 'Finish Import'
- VI) Verify results in contact list

#### *For Test Case TC-MC-01*

- I) Log in to HubSpot portal
- II) Navigate to Marketplace > App Marketplace
- III) Search for "Mailchimp" and click install
- IV) After installation, navigate to the new Mailchimp section
- V) Click 'Create Email' and draft a test email

#### *For Test Case TC-CR-01*

- I) Log in to HubSpot portal
- II) Navigate to Contacts > Create Contact
- III) Enter the information as specified in the test case
- IV) Click save
- V) Verify the new contact was created

## 6.4. Test Environment

The test environment that will be used for testing is one that can be easily tested across multiple devices in different environments. For the hardware requirements for the testing, we will be using a mixture of PCs and Laptops to run the testing. Since HubSpot is an online only service, any hardware can run HubSpot. For the software requirements, the only one would be to have the latest version of either Windows or IOS downloaded. HubSpot being in the cloud means that no software must be downloaded to use and run HubSpot. This also means that the way to start HubSpot is by simply going to the website and logging in. There is no software setup or any running of code on the host machine to

get HubSpot working, all you need in terms of software is an OS and a Browser to view HubSpot in. Since HubSpot is in the cloud, a constant internet connection is a must. This can be really any sort of internet connection, either commercial or residential. The connection will need to be able to allow access to the internet and provide a constant connection during the testing of HubSpot.

## 6.5. Test Data

The only data that is required for testing is a spreadsheet file or a .csv file. The data that is needed has to be between roughly 1 to 2,000 rows. The free version of HubSpot will only allow a maximum of 2,000 rows before it requires you to pay for the feature. For the testing, three different spreadsheets will be used. One will be the actual spreadsheet that the industry partner has sent to us. Another is a simple 10 row spreadsheet that will be used to start testing to ensure that the software of HubSpot is able to pick up the information of our spreadsheet accurately. The last spreadsheet will have the maximum number of rows, to test whether HubSpot is able to deal with a large spreadsheet set and how efficiently it handles the large data set. Lastly, there will be a .csv file that is used to test the functionality of being able to insert .csv into HubSpot.

## 6.6. Software Test Report (STR)

Test Case ID	Requirement	Pass/Fail	Severity	Notes & Observations
TC-SI-01	Small spreadsheet import	Pass	-	10-row import was successful and accurate
TC-SI-02	Partner's data import	Pass	-	Partner's provided test data was imported correctly; all data mapped as expected
TC-SI-03	Large spreadsheet import	Fail	Moderate	First import attempt timed out. A second

				attempt was successful
TC-MC-01	Mailchimp integration	Pass	-	Installation was straightforward and successful
TC-CR-01	Contact creation	Pass	-	Contact created with all info saved correctly

## 7. Version Control

Version Control in this project was done by saving progress .CSV files that were given to us by HubSpot. Because HubSpot free accounts only last for two weeks on the enterprise plans which were used, we version controlled by saving .CSV files that HubSpot allowed us to download. These .CSV files could be downloaded and imported into the new HubSpot account we created and allowed us to pick up where we left off.

Since this project has no code, we do not have a way to show the list of programs and files that were used to build the project. Each .CSV we had was named the same and kept in a OneDrive folder so that if we needed to revert to a previous spreadsheet, we could. We labeled each new spreadsheet we saved with an increasing number so that we could keep track of how many versions of the spreadsheets we had created. For example, when we generated the first .CSV file it was named `companies_for_hubspot1.csv` and during our next iteration after updating and adding to the first spreadsheet, the next one became `companies_for_hubspot2.csv`. Using this method and the timestamp provided on when it was last updated, we were about to create a version control that allowed us to roll back to previous spreadsheet patterns if needed in case the current one that was worked on was corrupted and became unusable from trying to improve upon it.

## 8. Summary

This project successfully implemented and validated HubSpot CRM as a solution to streamline our industry partner's fragmented customer management workflow. Through systematic testing, we confirmed that HubSpot effectively meets their core requirements: it seamlessly imports contact data from various spreadsheet formats, integrates their existing Mailchimp email marketing tool directly within the platform, and provides a flexible

system for managing contacts even with incomplete information. All critical functions passed testing, demonstrating the platform's readiness for adoption.

The implementation was highly successful, with only a minor, non-blocking issue encountered during stress testing of large data imports. The HubSpot configuration not only consolidates the partner's manual processes into a single, efficient interface but also requires minimal change to their established workflows. Based on our comprehensive test results, the system is fully functional, user-friendly, and recommended for immediate handoff and operational use by the industry partner.

## 9. Appendix

### Project Plan

#### Deliverables

##### *Expected Deliverables:*

- Requirements and stakeholder/entity mapping document
- Platform evaluation matrix
- Proposed solution and configuration rationale
- Working proof-of-concept demonstrating student tracking, related entities (school, Rotary, lifecycle), and event workflows
- Implementation roadmap
- Cost breakdown and upgrade options
- User guide and documentation

#### Group Meeting Schedule Date/Time

The group will meet with the sponsor, Georgia Laws of Life Essay, on a biweekly basis on Wednesdays at 9am EST. The first meeting date will be on August 27th on Microsoft teams. Every meeting will take place with Microsoft Teams following the first meeting and will be done to ensure that accurate information and updates are given to the sponsors on how development is coming on the project.

Outside of meeting with the sponsor the group will also plan to meet with each other on the weeks that the meetings with the sponsors are not taking place through Discord. This is to ensure that everyone is staying on task and is able to complete their part

of the project on time and to facilitate any group discussions that may arise during the time of working on the project.

## Collaboration and Communication Plan

The plan to communicate as a team is to use GroupMe for reminders and things that do not require a physical meeting but require that the whole group is aware of important information. When it comes to meeting as a group, the group will use Discord to meet when it comes to collaboration that requires face-to-face meetings. This gives us more flexibility when it comes to scheduling and gives us a chance to meet and talk through things that otherwise would not be possible without Teams.

For communication from person to person we plan to use phone numbers as a way to send messages to each other if an individual question arises that does not require the whole group to be notified. If phone numbers are not available, the backup plan is to direct messages from GroupMe. In the case that a group member has something happen where they cannot use their number for text purposes, the ability to use GroupMe on mobile or desktop will provide us a failsafe to contact someone directly.

## Risk Assessment

With budget concerns being an issue for Georgia Laws of Life Essay, the biggest risk that is associated with this project is unknown and unaware cost of the software. Because of the nature of Software as a Service, some products do carry hidden cost or pricing models that do not always give a clear idea of the total cost of ownership that comes with using the product.

To combat this issue certain practices will be implemented to ensure that cost overrun does not occur. First, when looking at the Software as a Service models, not only will the cost of the model be taking into account and how they are charging, but also looking at terms and conditions that these companies are required to provide, to get a better understanding of how pricing will work. Second, working with the non-profit, we will get a better understanding of their plans and growth. If they are looking to expand and get more employees in the next years, growth and scaling will also be used to determine the best option. Many Software as a Service accounts restrict the number of users for lower cost tiers, so growth of the non-profit will be taken into consideration.

## Source Code

Since our project did not include any actual coding, the source code of the project consists of the final .csv files that we downloaded from HubSpot. These files can be reloaded into any new HubSpot account and will allow the user to pick up from where the



last person left off. They include all the contacts and information that was gathered, and the tokens needed to set up the next contact list in HubSpot. In total, there are five .csv files that are included in the zip file along with the report.