P03-T2 Mgmt Sys NP

SOFTWARE REQUIREMENTS SPECIFICATION (SRS)

CS 4850

Fall 2025

Professor Perry

9/14/2025

P03-T2 Mgmt Sys NP

 Evan Conley Team Leader	 Tyler Stroud Programmer / Tester	 Ella Goode Product/Documentation	 Fatih Kurt Documentation/Programmer

Table of Contents

1.0 Introduction

1.1 Overview

The Georgia Laws of Life Essay Contest is a nonprofit organization that is currently dealing with many disconnected spreadsheets. These spreadsheets can give important information such as student involvement, school contacts, donors, and events that are taking place.

They are looking for a technological solution that will be both cost effective and user friendly, to bridge the gap between the disconnected spreadsheets. Our team will look at numerous customer relationship management (CRM) and project management tools and present a solution that is both viable for everyday use to the company and cost effective, given the budget constraints imposed by being a nonprofit organization. By the end, the Georgia Laws of Life Essay Contest will not only be given an implementation and cost analysis for the best CRM or project management tool but will also have a working and functional proof-of-concept design, showing that the result they expect is achievable.

1.2 Project Scope

The project will seek to find a management platform suitable for the needs of Georgia Laws of Life. The scope of the project will include:

- Assessment of each software as a service (SaaS) platform for fit
- Mapping of stakeholders and entities relevant to the nonprofit's operations
- Creating a proof of concept (POC) system with simulated data
- Demonstrating common workflows on the POC
- Providing or creating documentation for platform options
- Search for wanted, but not required features, namely dashboarding, bulk import, and reminders

1.3 Definitions and Acronyms

- **CRM (Customer Relationship Management)** - system for managing a company's interactions with current and potential customers
- **SaaS (Software as a Service)** - software licensing and delivery model where software is accessed online via a subscription
- **Freemium** - business model where basic features are provided for free, while advanced features must be paid for

- **POC (Proof-of-Concept)** - prototype designed to demonstrate the feasibility and potential of a proposed solution

- **API (Application Programming Interface)** - set of rules that allow different software applications to communicate with each other

- **Google Workspace** - suite of cloud-based productivity and collaboration tools developed by Google

1.4 Assumptions

The following assumptions will be made about the requirements. The first assumption is that the user has a computer that can run modern software. Most computers in use today can run the software that is required to run the software that is being used, but it will be assumed that the user has the computer. Another assumption is that the user will have a steady internet connection, and most of the time they use the software will be in a steady internet connection. Since the industry partner does want to use Software as a Service product, it will be assumed that they will have an internet connection to connect them to use the product at most times.

Another important assumption is security. From a security perspective, the assumption will be that they will use a login and password to enter the software. We will assume that the software that we choose for the industry partner will have basic password requirements. These requirements will be used by the industry partner, and it will be assumed that the industry partner will follow the password guidelines. Data that the industry partner enters will also be valid data. While mistakes happen and there will be corrections and error control in place for an error in the data, we will assume that most of the data being entered into the database will be in the correct format as requested by the industry partner.

1.5 References

Georgia Laws of Life Essay Contest. (n.d.). Retrieved from https://georgialawsoflife.org/

HubSpot CRM. (n.d.). Retrieved from https://www.hubspot.com/products/crm

Zoho CRM. (n.d.). Retrieved from https://www.zoho.com/crm/

Airtable. (n.d.). Retrieved from https://www.airtable.com/

Google Workspace API Documentation. (n.d.). Retrieved from https://developers.google.com/workspace

2.0 Constraints

2.1 Environment

The Environment for the project must be done in accordance with the industry partners' guidelines. They have requested that the environment be a free version of Software as a Service (SaaS). Only those SaaS approved by the industry partners will be considered for further development towards the solution they recommended. The preexisting environment limits what the service can provide to us, rather than creating a custom application to try and fit all the needs of our industry partners. One major environmental constraint will be the ability to use SaaS on a device other than a computer. Some of the SaaS that are being researched have some compatibility with IOS and mobile devices, some software does not have this functionality. Since the industry partner has not mentioned any need for a mobile application or use on IOS devices or Mac, all software will be looked at will be to run on Windows.

The environment will also need to consider the cost of the software. The environment will be directly impacted by the budget of the industry partners. The total cost of the software will impact on the decision to choose the software that we choose. Being a nonprofit organization, there is not much money that can be used for this software. This is a constraint that can be worked around. Many of the SaaS offer free software or basic versions which are at a lower cost. These will work for the non-profit as many of these SaaS base their cost on the number of users. Many of the free SaaS versions only allow a maximum of 5 users. This will work for the entire organization currently as they expect to only have a maximum of three users. So currently, we can take advantage of the free software. However, if the organization expects to grow, this will have to be revisited and reevaluated for the new cost that more users will incur.

2.2 System

There are limited system constraints as most of the SaaS can run on any Windows system. The System constraints that will be imposed is that any system using the Software as a Service must be able to run Windows 11 operating systems. This is the optimal operating system in terms of both security and performance for industry partners. We also will need a computer that has available memory space for holding the spreadsheets and any downloadable data that they might need. Any current laptop or desktop computer will be sufficient to run the product.

Another system constraint will be the use of certain software used currently by the industry partner. The software that is chosen needs to have the ability to integrate with the current software

used by the industry partner. One example of this is Google workspace. The industry partner currently uses Google workspace for documents and spreadsheets. The spreadsheet that will be used to create the database needs to have functionality with the software chosen so it can be easily implemented. If there is not any functionality with Google workspace, the software cannot be used as it will create problems in the future when trying to update the database.

3.0 Functional Requirements

3.1 Platform Assessment

3.1.1 The platform assessment shall define a means of platform evaluation.

3.1.1.1 The platform evaluation shall be weighed by client requests and priorities.

3.1.1.2 The platform evaluation shall consider the impact of inclusion of optional features, namely dashboarding, bulk import, and follow-up reminders.

3.1.1.3 The platform evaluation shall be quantitative in nature.

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

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

3.2 POC System

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

3.2.2 The POC shall use simulated data.

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

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

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

3.2.4 The POC shall be interactable live.

3.3 Documentation

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

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

3.3.2 Documentation shall contain the platform assessment.

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

3.3.4 Documentation shall be provided to the company.

3.3.5 Documentation shall be readable and relevant to the project.

4.0 Non-Functional Requirements

4.1 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.

4.2 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.

4.3 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.