Foster Grandparent Program

Grandparent Education Management System

Project Notebook

Version 2.0

April 17, 2023

Prepared by CS 471 Documentation Team

Table of Contents

[1.0 Introduction 3](#_Toc130401482)

[1.1 Purpose 3](#_Toc130401483)

[1.2 Scope 3](#_Toc130401484)

[1.3 Definitions and Acronyms 3](#_Toc130401485)

[1.4 Overview 6](#_Toc130401486)

[2.0 Description 7](#_Toc130401487)

[2.1 Product Perspective 7](#_Toc130401488)

[2.1.1 System Interfaces[B] 7](#_Toc130401489)

[2.1.2 User Interfaces[H] 7](#_Toc130401490)

[2.1.3 Hardware Interfaces 8](#_Toc130401491)

[2.1.4 Software Interfaces 8](#_Toc130401492)

[2.1.5 Communication Interfaces 9](#_Toc130401493)

[2.1.6 Memory (Primary & Secondary) 9](#_Toc130401494)

[2.1.7 Operations 10](#_Toc130401495)

[2.1.8 Site Adaptations Requirements 10](#_Toc130401496)

[2.2 Product Functions [D], [K] 11](#_Toc130401497)

[2.3 User Characteristics 11](#_Toc130401498)

[2.4 Constraints 12](#_Toc130401499)

[2.5 Assumptions and Dependencies 12](#_Toc130401500)

[2.6 Apportioning of Requirements 12](#_Toc130401501)

[General Requirements 13](#_Toc130401502)

[2.0 Volunteer Page Requirements 14](#_Toc130401503)

[3.0 Students Page Requirements 15](#_Toc130401504)

[4.0 Site Page Requirements 15](#_Toc130401505)

[5.0 Budget Page Requirements 18](#_Toc130401506)

[6.0 Nonfunctional Requirements 21](#_Toc130401507)

[7.0 Login Page Requirements 21](#_Toc130401508)

[8.0 Child Assignment Requirements 25](#_Toc130401509)

[9.0 Volunteer Information Requirements 25](#_Toc130401510)

[11.0 Report Page Requirements 27](#_Toc130401511)

[12.0 Report Builder Requirements 29](#_Toc130401512)

[13.0 Data Requirements 30](#_Toc130401513)

[4.0 Appendices 37](#_Toc130401514)

[A. Cost/Benefit Analysis 37](#_Toc130401515)

[B. Context Diagrams 39](#_Toc130401516)

[C. System Modeling 40](#_Toc130401517)

[D. Use Cases/Expanded Use Cases 55](#_Toc130401518)

[E. Requirements to Use Case Matrix 92](#_Toc130401519)

[F. Requirements to Iteration Matrix 93](#_Toc130401520)

[G. UI/Storyboards 94](#_Toc130401521)

[H. Domain Model 139](#_Toc130401522)

[I. ERD Diagrams/Data Dictionaries/Data Dictionary Notation 140](#_Toc130401523)

[J. UI Workflow Diagrams 143](#_Toc130401524)

[K. Battle Tests 145](#_Toc130401525)

[M. Recommended Implementation Architecture Diagram 150](#_Toc130401526)

[N. Image Citations 155](#_Toc130401527)

[Index 157](#_Toc130401528)

# Introduction

## Purpose

This specification document is intended to describe the proposed Volunteer and Student Management System – Grandparent Education Management (GEM) - designed and developed for internal use by the Director/Coordinator of the Foster Grandparents Program, Mrs. Tara Stinson. This description includes GEM's purpose, required functionality, proposed user interface designs, use cases of GEM, proposed database schema, architectural diagrams, and a recommendation on the implementation of GEM. The intended audience is directed to the Director/Coordinator of the Foster Grandparents Program.

## Scope

GEM is responsible for storing a list of volunteers and schools both active and inactive in the program, as well as serving as a budget/accounting management system. The end goal is to reduce bookkeeping and busy work, eliminate redundancies, and overall provide a centralized location for all necessary information to run the Foster Grandparents Program. GEM will bring all existing functions of the current system as well as new and improved functionalities into one singular application.

## Definitions and Acronyms

|  |  |
| --- | --- |
| Abused or Neglected - (AN) | A student's condition is identified and tracked to allow the  program and volunteers to facilitate greater assistance and understanding. |
| Active Volunteers | The set of volunteers actively volunteering in the current year. |
| Annual | The period of a year where data is gathered to be used in reporting. |
| Application  Programming Interfaces – (API) | A set of functions and procedures allowing the creation of  applications that access the features or data of an operating system, application, or other services. |
| Assist with Cognitive Activities - (I) | A student condition that is identified and tracked to allow the program and volunteers to facilitate greater assistance and  understanding. |
| Attention Difficulties - (AD) | A student condition that is identified and tracked to allow the  program and volunteers to facilitate greater assistance and understanding. |
| Behavior/Social Difficulties - (HS) | A student condition that is identified and tracked to allow the program and volunteers to facilitate greater assistance and understanding. |
| C# | An object-oriented, high-level, programing language created by Microsoft for building applications. |
| Child of Incarcerated Parent - (CI) | A student condition that is identified and tracked to allow the program and volunteers to facilitate greater assistance and  understanding. |

|  |  |
| --- | --- |
| Communication Skills - (C) | A student need formed from that student’s condition that the volunteer works to improve with the student. |
| Conversation Skills - (G) | A student need formed from that student’s condition that the volunteer works to improve with the student. |
| County Fiscal Year | The designated year of the county where financial information  is tracked from October 1st to September 30th. |
| CRUD | Stands for “create”, “read”, “update”, and “delete” data. Defines how one can operate on stored data. |
| Database | A structured set of data held in a computer that can be accessible in various ways. |
| Data source | The location where data that is being used originates from. |
| Developmental Disabilities - (DD) | A student condition that is identified and tracked to allow the program and volunteers to facilitate greater assistance and  understanding. |
| Emotional/ Social Difficulties - (ES) | A student condition that is identified and tracked to allow the  program and volunteers to facilitate greater assistance and understanding. |
| Former Volunteers | Volunteers who have left the program and were not active in the current grant or fiscal year. |
| Foster Grandparents – (FGP) | The official term for volunteers who operate within the program. |
| Grant Fiscal Year | The designated year of the county where financial information is tracked from July 1st to June 30th. |
| GEM | The name of the application that is being built to manage the Saginaw Counties Foster Grandparent program. |
| Help with Letter  Identification - (E) | A student need formed from that student’s condition that the  volunteer works to improve with the student. |
| Help with Numeracy Skills - (H) | A student need formed from that student’s condition that the volunteer works to improve with the student. |
| Homeless or Recently Displaced - (CH) | A student condition that is identified and tracked to allow the  program and volunteers to facilitate greater assistance and understanding. |
| Human Resource Management (HRM) | The strategic and coherent approach to the effective and efficient management of people in an organization. |
| In-Kind | Payments of goods or services instead of money or currency. |
| Inactive Volunteer | Volunteers that were active within the current year but had to stop for some reason. |
| Language/Literacy Barriers - (LB) | A student condition that is identified and tracked to allow the  program and volunteers to facilitate greater assistance and understanding. |
| Learning Disabilities - (LD) | A student condition that is identified and tracked to allow the program and volunteers to facilitate greater assistance and  understanding. |
| Local Machine | A computer that a user can access directly without using a communications line or device. |
| Microsoft Excel (Excel) | A spreadsheet software developed by Microsoft to create reports. |

|  |  |
| --- | --- |
| Microsoft Word (Word) | A word processing software developed by Microsoft to create documents and reports. |
| Mileage | The distance a volunteer has traveled to get from their home to their assigned school. |
| Monthly | The period of a month where data is gathered to be used in  reporting. |
| National Service  Criminal History Check (NCHC) | A baseline screening requirement established by law to protect the beneficiaries of national service. |
| Nurturing/Comfort - (A) | A student need formed from that student’s condition that the volunteer works to improve with the student. |
| Operating System | The software that supports a computer's basic functions. |
| Other – (J) | A student need formed from that student’s condition that the volunteer works to improve with the student. |
| Other needs - (ON) | A student condition that is identified and tracked to allow the  program and volunteers to facilitate greater assistance and understanding. |
| Paid time off (PTO) | A human resource management policy that provides employees with a pool of bankable hours that can be used to take time off from work. |
| Physical Disabilities - (PD) | A student condition that is identified and tracked to allow the program and volunteers to facilitate greater assistance and understanding. |
| Positive Reinforcement/Redirection - (F) | A student need formed from that student’s condition that the volunteer works to improve with the student. |
| Project Profile and Volunteer Activity – (PPVA) | This report contains all the volunteer’s information including, Name, Status, Start Date, Date of Birth, Birth Month, Age, Separation Date, Reason for Separation, Gender, Ethnicity,  Race, if they are a Veteran, and if they have family in the military. |
| Quarterly | The three-month period of a fiscal year where data is gathered,  to be used in reporting. There are generally four quarters within a fiscal year. |
| Random Access Memory – (RAM) | Short-term memory where data is stored to be quickly accessed by the device’s processor. |
| Reading - (D) | A student need formed from that student’s condition that the volunteer works to improve with the student. |
| RESTful APIs | An interface that two systems can use to exchange information. |
| Saginaw County Commission on Aging | A government organization that provides programs and services to meet the needs of older adults. |
| Social Skills - (B) | A student need formed from that student’s condition that the volunteer works to improve with the student. |
| Speech Impaired - (SP) | A student condition that is identified and tracked to allow the  program and volunteers to facilitate greater assistance and understanding. |
| Statement of Understanding - (SOU) | A signed document completed by an applicant or foster parent confirming the person has read, understands, and agrees to |

|  |  |
| --- | --- |
|  | comply with all applicable laws, rules, and regulations relating to the operation of a foster home. |
| Stipend | A fixed regular sum paid as a salary or allowance. |
| Structured Query Language (SQL) | A standardized programing language that is used to manage relational databases and perform various operations with the  data in them. |
| User Interface - (UI) | A platform allowing a user to interact with the software or the hardware in a natural and intuitive way |
| Visual Basic for Applications - (VBA) | An event-driven programming language implemented by Microsoft to develop office applications. |
| Year to Date - (YTD) | A term that covers the period between the beginning of the fiscal or calendar year and the current (present) date. |

## Overview

The remainder of this document consists of a high-level overview of the description of GEM (Section 2), a collection of formal system functions and requirements (Section 3), a cost/benefit analysis (Appendix A), context diagrams representing outside factors that will interact with GEM (Appendix B), system modeling (Appendix C), use case diagrams and extended use case diagrams representing how the end user will interact with GEM (Appendix D), a collection of formal requirements for GEM (Appendix E), requirements to use case matrix (Appendix F), requirements to iteration matrix (Appendix G), two proposed user interface storyboards (Appendix H), domain models (Appendix I), ERD Diagrams/Data Dictionaries/Data Dictionary Notation (Appendix J), UI Workflow diagrams (Appendix K), battle tests examining failures and recovery methods of GEM (Appendix L), implementation alternatives and final recommendation (Appendix M), and recommended implementation architectural diagram (Appendix N).

# Description

## Product Perspective

A Windows-based desktop application called GEM (Grandparent Education Management System) will be built as a completely new system.

### System Interfaces[B]

GEM will be connected to Microsoft Excel and Microsoft Word by using RESTful APIs. This will be required for exporting data. GEM will also be connected to a SQL database for performing CRUD operations.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Microsoft Word | Export  Data | GEM | Export  Data | Microsoft Excel |
|  |  |

CRUD Operations

Database

**Figure 1 General overview of the GEM system interfaces**

### User Interfaces[H]

The GEM will be built as a Windows 10 and 11 desktop application. A login page will allow access to users. Admins will have access to add/manage users, while secondary users can view/edit important program records. An individual volunteer's information can be viewed through their status page; while corresponding site and student assignment information can be found in the site and student tabs, respectively. Budgetary reports can be generated to Excel and printed through both a quarterly and annual view.

### Hardware Interfaces

The end user’s hardware will require an internet connection to connect to the database. The hardware will also be required to interact with a printer to be able to print all reports and forms necessary for the Foster Grandparents Program’s internal use.





Printer

Router

End User s Hardware

**Figure 2 General overview of the interfaces the end user's hardware will interact with**

### Software Interfaces

GEM is designed to be installed by the end-user on Windows 10 and 11 systems. GEM will be communicating with a Microsoft SQL Server database system to store, update, and delete records regarding volunteers, students, school sites, and pertinent bookkeeping based on need. Other outside interfaces GEM will communicate with include Microsoft Word to fill out the monthly site form and Microsoft Excel to generate the numerous required forms based on selected information by the end user. Both interfaces would be linked to the software through API calls.

### Communication Interfaces

It is expected that GEM will utilize RESTful APIs to communicate with the external tools used by the Foster Grandparent Program, including Microsoft Word and Excel.

Microsoft Word API



GEM System

Microsoft Excel API

**Figure 3 General overview of the communications the GEM system will interact with**

### Memory (Primary & Secondary)

It is to be assumed that every machine running GEM will be equipped with at least 8 GBs of RAM as primary memory. Meanwhile, secondary memory will be used as a permanent external memory to store GEM's data. It is estimated that the database will require approximately 20GB of local storage.

### Operations

GEM’s function is to store and maintain a roster of volunteers and their personal information. GEM also tracks individual and budgetary finances to assist The Saginaw County Commission on Aging in their government reporting. Operations include:

Storing information:

* A SQL database will be used to store the information.

Validating user credentials:

* GEM will hold the privileges of the user in the database in a variable.
* GEM will prompt the user to enter their username and password.

Read/Write Database:

* + - * Input from the user will be written to the database when information can be added, updated, or deleted.
      * Users will be able to retrieve the information they need from the database.

Export to Word:

* + - * Word handles each volunteer’s activity log.

Export to Excel:

* Excel handles all canned and customized reports that can be exported from the system.

### Site Adaptations Requirements

To use the software at the intended location, the following adaptations will need to be made:

* + - * To install the application on the user’s computer, personnel will be dispatched from the site.
      * To ensure that the computer being used meets specifications, personnel will verify the computer’s hardware and software.
      * A package for installing the software will need to be constructed by developers.
      * Installation packages will include executables that set environmental variables to obtain an IP address of the offsite database server.
      * Security checks may also be performed by the installation package on hardware properties, including firmware versions and Microsoft Windows versions.
      * Three flash drives containing GEM will be provided to the Director of the Commission on Aging’s Foster Grandparent Program for additional access when required.

## Product Functions [D], [K]

GEM will maintain information about volunteers and financial status for Saginaw County Commission on Aging’s Foster Grandparent Program.

Functions related to maintaining financial information include:

* Maintaining a record of each active volunteer’s expenses on a monthly and yearly basis.
* Maintaining a log of all donations and expenses that can be generated by desired grand or fiscal periods.
* Tracking paid time off for volunteers on a monthly and yearly basis.
* Tracking stipends paid to volunteers throughout each month. Functions related to maintaining volunteer information include:
* Adding new volunteers to the system.
* Changing a volunteer’s status.
* Updating a volunteer’s personal or contact information.
* Exporting reports to Microsoft Excel so that the data can be easily accessed by third parties.
* Using a single database for all volunteer information to reduce data redundancies.

For GEM to be configurable, the following items must be configurable:

* Changing the value for school meals, cost per mile, bus rides, and hourly stipend pay.
* Adding/Removing attributes for data entries related to both financial and volunteer records in the database.

## User Characteristics

The Director of Saginaw Commission’s Foster Grandparent Program will be GEM’s administrative user. The administrator account will have permission to create other GEM users. The administrative account can also manually back up the database to the local machine or assign a new administrative user, as well as have all permissions of secondary users. All secondary users will have permission to sign in and make changes to database entries.

## Constraints

GEM must run on Windows 10 and 11 as a desktop application. This was specifically requested by the Program's Director/Coordinator (Mrs. Tara Stinson) instead of a web-based alternative. It was also requested GEM can export reports to Excel. By focusing on a singular operating system, the software development team is constrained to languages and frameworks that can be compiled and run as a native Windows application. Developers must build a system that runs efficiently and keep in mind the overall technical knowledge of the end user.

## Assumptions and Dependencies

The desktop application software GEM is expected to use the below external components for the user interface design, implementation, and data storage.

* C#, an object-oriented high-level programing language that will be used to create the user interface and run the GEM application. This language is fully compatible with and supported by the Windows operating system.
* Microsoft Word, a word processing software that will be used to generate a site visit document or any other documents needed.
* Microsoft Excel, a spreadsheet software that will be used to export data to create various reports as needed.
* SQLite, a database engine that will be used to store all the data to run the GEM application.
* Saginaw County Servers, a server bank overseen by Saginaw County will host the database for the GEM application.

## Apportioning of Requirements

A future version of the GEM system should implement the use of concurrency control for when data is being changed or added to the system. The development team ran out of time during the implementation of the first version and was unable to implement this. Instead the development team implemented the refresh buttons to update when the data is change or additions are made. With a future update that included concurrency control, these buttons would be able to be removed from the system.

* 1. System Features and Requirements[M]
  2. External Interfaces[K]
     1. Hardware Interfaces

1. The hardware shall be attached to a router via Wi-Fi network or Ethernet cable to communicate with the county database.

2. The hardware shall have a USB 2.0 or 3.0 port.

* + 1. Software Interfaces
       1. GEM shall interact with Microsoft Word to generate the monthly site visit sheet.
       2. GEM shall interact with Microsoft Excel to generate the remaining reports.
       3. GEM shall interact with SVSU’s Software Distribution Center to install new copies of GEM.
    2. Communication Interfaces
       1. TCP (Transmission Control Protocol) /IP (Internet Protocol) – for database connection.
       2. SMTP (Simple Mail Transfer Protocol) – to send emails to user accounts.
       3. Word API – to communicate with Microsoft Word.
       4. Excel API – to communicate with Microsoft Excel.
  1. Functions[D],[E],[K]

### General Requirements

* 1. GEM shall run on Windows 10 and Windows 11.

Test: Attempt to start the application on Windows 10 and 11 computers.

* 1. GEM shall store and retrieve data from a database.

Test: Add data to one of the sections, close the application, start the application, and check if the previously stored data was retrieved.

* 1. GEM shall have a backup storage system when network errors occur. Test: Disconnect GEM from the network, attempt to save, and observe if changes persist.
  2. GEM shall allow multiple users to access it at once.

Test: Attempt to log multiple users into GEM at the same time.

* 1. GEM shall allow the user to navigate from grant year to grant year. Test: Navigate from one grant year to another.
  2. GEM shall display the number of active volunteers.

Test: Attempt to locate the display of active volunteers by visual inspection.

* 1. GEM shall display the total number of active schools.

Test: Attempt to locate the display of the total number of active schools by visual inspection.

1.7 The system shall have a graphical user interface for the user to interact with

Test: Run application and verify that the program can be interacted with via a graphical interface.

### 2.0 Volunteer Page Requirements

2.1 GEM shall have a page to manage volunteers. Test: Attempt to view the volunteer page.

2.2 GEM shall allow the storage of a volunteer profile.

Test: Attempt to store a volunteer profile and attempt to view the profile.

2.2.1 GEM shall allow a minimum of 150 volunteer profiles at any given time.

Test: Attempt to enter 150 unique volunteers into the storage.

1. The System shall be able to report expenditures in either fiscal or grant year.  
a. Test: When choosing what data is reported allow the user to choose between   
either grant year or fiscal year.  
b. Check the report to see if that data is between fiscal or grant year depending on   
which was picked.

2.3 GEM shall allow a user to switch from an individual volunteer profile to a view showing all volunteers.

Test: Select the all-volunteer option and visually inspect the view.

2.4 GEM shall allow multiple entries of volunteers at once. Test: Attempt to edit multiple volunteers at once.

2.5 GEM shall allow volunteers to be active or inactive.

Test: Attempt to switch the status of a volunteer from active to inactive. Attempt to switch the status of a volunteer from inactive to active.

2.6 GEM shall denote inactive volunteers with a specific color.

Test: Visually inspect that an inactive volunteer is denoted with a specific color.

2.7 GEM shall generate a report of volunteers that can be exported to Excel. Test: Attempt to generate a report and see if it opens in Excel.

2.7.1 GEM shall allow the user to select what data is generated in the volunteer report.

Test: Select the data to export and make sure the correct data is displayed in the Excel file.

2.7.2 GEM shall have an activity log for each user that is displayed in reverse chronological order (from newest to oldest).

Test: Inspect a volunteer's activity log and check if the log starts at the newest activity and check that proceeding activities are always older than the previous.

2.8 GEM shall generate a report of an activity log that the user can select the data returned and export it to Excel.

Test: Attempt to generate a report and make sure all the data is exported to Excel.

2.9 GEM shall generate a demographic report that the user can select the date returned and export to Excel.

Test: Attempt to generate a report and make sure all the data is exported to Excel.

### 3.0 Students Page Requirements

3.1 GEM shall have a page to manage students. Test: Visually inspect the page.

3.2 GEM shall only allow a student to be assigned to a single volunteer.

Test: Try to assign a student to multiple volunteers.

3.3 GEM shall generate a student report so that the user can select the data returned and export it to Excel.

Test: Attempt to generate a report and make sure all the data is exported to Excel.

### 4.0 Site Page Requirements

4.1 GEM shall generate a monthly site visit sheet.

Test: Attempt to generate a site visit sheet and visually inspect that it is formatted correctly.

4.2 GEM shall track the name of every school currently in the program. Test: Search GEM for schools in the system and ensure they appear.

4.3 GEM shall track the name of every school no longer in the program. Test: Search for inactive schools in the system and ensure they appear.

4.4 GEM shall track the principal’s name of every school currently in the program.

Test: Open the school tab and visually inspect that the principal’s name is showing and is associated with the correct school.

4.5 GEM shall track the principal's name of every school no longer in the program.

Test: Open the school tab and visually inspect that the principal’s name is showing and is associated with the correct school.

4.6 GEM shall track the primary contact phone number for every school currently in the program.

Test: Open the school tab and visually inspect that the primary contact phone is showing and is associated with the correct school.

4.7 GEM shall track the primary contact phone number for every school no longer in the program.

Test: Open the school tab and visually inspect that the primary contact phone is showing and is associated with the correct school.

4.8 GEM shall track the status of every school both currently in, and no longer in the program (active/inactive)

Test: Open the school tab and visually inspect that the status is showing and is correct for each school.

4.9 GEM shall generate a total sum of students within classrooms that are in the program.

Test: Open the school tab and visually inspect that the total sum of students is correct.

4.10 GEM shall generate a total sum of students assigned to a grandparent in the program.

Test: Open the school tab and visually inspect that the total sum of students per grandparent is correct.

4.11 GEM shall generate a total sum of students aged 0-5.

Test: Open the school tab and visually inspect that the total sum of students aged 0-5 is correct.

4.12 GEM shall generate a total sum of students aged 6-12.

Test: Open the school tab and visually inspect that the total sum of students aged 6-12 is correct.

4.13 GEM shall allow for a school’s status to be changed from active to inactive (and vice versa).

Test: Go to the edit school screen, change a school from active to inactive, go to the school's tab, and ensure the school is no longer showing as active.

4.14 GEM shall allow the manual addition of a new school into the system. Test: Open the school tab and attempt to add a new school. Ensure by visual inspection that a new school is successfully added.

4.15 GEM shall allow for the complete list of active and inactive schools in the system to be exported to Excel.

Test: Attempt to export a complete list of active and inactive schools to Excel and visually inspect the outcome.

4.16 GEM shall allow for the information of each school to be viewed separately from the rest.

Test: Open the school tab and visually inspect that the information for each school can be viewed independently and is correct for each school.

4.17 GEM shall track the address of each school in the program.

Test: Open the school tab and visually inspect that the address of each school is showing and is correct for each school.

4.18 GEM shall track the hours that each school in the program is open. Test: Open the school tab and visually inspect that the hours of operation for each school are correct.

4.19 GEM shall track the point of contact for each school in the program. Test: Open the school tab and visually inspect that the point of contact for each school is showing and is correct for each school.

4.20 GEM shall track the current volunteers at each school in the program. Test: Open the school tab and visually inspect that the current volunteers in a school are correct.

4.21 GEM shall track the teacher of each classroom the volunteer is assigned to.

Test: Open the school tab and visually inspect that the teacher which a volunteer is assigned to is showing and is correct for each school.

4.22 GEM shall track the grade level of each classroom the volunteer is assigned to.

Test: Open the school tab and visually inspect that the grade level a volunteer is assigned to is showing and is correct for each school.

4.23 GEM shall track the room numbers of each classroom the volunteer is assigned to.

Test: Open the school tab and visually inspect that the classroom room number a volunteer is assigned to is showing and is correct for each school.

4.24 GEM shall track the number of hours the volunteer spends in the classroom each week.

Test: Open the school tab and visually inspect that the number of hours a week a volunteer is working in a classroom is showing and is correct for each school.

4.25 GEM shall track the number of days the volunteer works in the classrooms he/she is assigned to.

Test: Open the school tab and visually inspect that the number of days a volunteer is working in a classroom is showing and is correct for each school.

4.26 GEM shall allow for manual editing of the school’s information. Test: Open the school tab and select edit. Visually inspect manual editing can be performed.

4.27 GEM shall allow for the individual school’s information to be sent to the printer for future viewing.

Test: Attempt to send an individual school’s information to a printer and visually inspect the outcome.

### 5.0 Budget Page Requirements

5.1 GEM shall track the grant year from July 1st to June 30th.

Test: Visually inspect that a report is generated for the grant year.

5.2 GEM shall track the county fiscal year from October 1st to September 30th.

Test: Visually inspect that a report is generated for the county fiscal year.

5.3 GEM shall record stipends and PTO information for volunteers. Test: Enter stipend and PTO information and check if the data is persistent upon closing and reopening.

5.4 GEM shall allow users to input the stipend and PTO rates for the month.

Test: Attempt to input data into the stipend and PTO rate sections.

5.4.1 The system shall allow meal and transport records to be created or updated by an authorized user.  
 Test: Have an authorized user attempt to create or edit meal and transport records and verify it completed successfully.

5.5 GEM shall flag volunteers who do not qualify for PTO.

Test: Visually inspect that there is a flag showing that the user does not qualify for PTO.

5.6 GEM shall calculate stipend and PTO totals monthly.

Test: Calculate the stipend and PTO totals for a month by hand and

compare them to GEM’s totals.

5.7 GEM shall calculate stipend and PTO totals quarterly.

Test: Calculate the stipend and PTO totals for a quarter by hand and compare them to GEM’s totals.

5.8 GEM shall calculate stipend and PTO totals yearly.

Test: Calculate the stipend and PTO totals for a year by hand. Compare to GEM’s totals.

5.9 GEM shall generate a stiped and PTO report that is exported to Excel.

Test: Attempt to generate a report and make sure all the data is exported to Excel.

5.10 GEM shall allow users to record timecard information turned in by volunteers each month.

Test: Attempt to add data to GEM. Restart GEM and see if the data persists.

5.10.1 GEM shall allow the user to change the timecard information at any time.

Test: Attempt to change timecard information. Visually inspect that the data has been added.

5.10.2 GEM shall track a volunteer's meal amount monthly.

Test: Attempt to add meal expenditure data. Visually inspect that the data has been added.

5.10.3 GEM shall allow users to input the meal rate quarterly.

Test: Attempt to add meal rate. Visually inspect that the data has been added.

5.10.4 GEM shall calculate the total meal cost for a single volunteer and display the total.

Test: Calculate the total meal cost by hand. Visually inspect to see if that is the total displayed.

5.10.5 GEM shall calculate the total meal cost of all volunteers combined and display the total.

Test: Calculate the total meal cost by hand. Visually inspect to see if that is the total displayed.

5.10.6 GEM shall track the volunteer's mileage monthly.

Test: Attempt to add mileage data. Visually inspect that the data has been added.

5.10.7 GEM shall allow users to input the mileage rate monthly. Test: Attempt to add mileage rate data. Visually inspect that the data has been added.

5.10.8 GEM shall calculate the total mileage cost for a single volunteer and display the total.

Test: Calculate the total mileage cost by hand. Visually inspect to see if that is the total displayed.

5.10.9 GEM shall calculate the total mileage cost for all volunteers combined and display the total.

Test: Calculate the total mileage cost by hand. Visually inspect to see if that is the total displayed.

5.10.10 GEM shall generate a meal expenditure report that is exported to Excel.

Test: Attempt to generate a report and make sure all the data is exported to Excel.

5.10.11 GEM shall generate a mileage report that is exported to Excel.

Test: Attempt to generate a report and make sure all the data is exported to Excel.

5.11 GEM shall allow users to add new expense types for in-kind items. Test: Attempt to add a new expense type and visually inspect that the item was added.

5.12 GEM shall generate a monthly financial report so that the user can select the data returned and export it to Excel.

Test: Attempt to generate a report and make sure all the data is exported to Excel.

5.13 GEM shall generate a quarterly financial report so that the user can select the data returned and export it to Excel.

Test: Attempt to generate a report and make sure all the data is exported to Excel.

5.14 GEM shall generate an annual financial report so that the user can select the data returned and export it to Excel.

Test: Attempt to generate a report and make sure all the data is exported to Excel.

### 6.0 Nonfunctional Requirements

6.1 GEM shall use the same logo graphics throughout all pages. Test: View that any logo graphics in GEM are the correct ones.

6.2 GEM shall start/open in less than five seconds.

Test: Attempt to start GEM and ensure that it opens in less than five seconds by measuring the time.

6.3 GEM shall maintain a uniform color scheme by using the same set of colors on every page.

Test: View that no pages of GEM have any colors that are unique to that page or deviate from the colors on the other pages.

### 7.0 Login Page Requirements

7.1 GEM shall have a login page.

Test: Attempt to run GEM and observe if a login page exists.

7.2 GEM shall have the ability to create an account. Test: Attempt to create a new user account.

7.3 GEM shall have the ability to recover a user account’s password. Test: Attempt to recover a user account's password.

7.4 GEM shall have the ability to recover a user's username/email. Test: Attempt to recover a user’s username/email.

7.5 GEM shall allow the ability to change the password of a user account. Test: Attempt to change a user account's password.

7.6 GEM shall have the ability to send verification emails.

Test: Attempt to change a password and observe if a verification email is

received.

7.7 GEM shall have a text field to input username/email. Test: Visual observation.

7.8 GEM shall be able to recognize a valid email address format of [xxx@xxx.xxx](mailto:xxx@xxx.xxx)

Test: Attempt to input an email address with an invalid format and observe if GEM allows it.

7.9 GEM shall have a text field to input the user's password. Test: Visual observation.

7.10 GEM shall be able to recognize invalid password formats.

Test: Attempt to input a password with an invalid format and observe if GEM allows it.

7.10.1 GEM shall not allow passwords with a length shorter than eight characters.

Test: Attempt to enter a password with a length of six and observe if GEM allows it.

7.10.2 GEM shall not allow passwords that do not contain a capital letter.

Test: Attempt to enter a password that does not contain a capital letter and observe if GEM allows it.

7.10.3 GEM shall not allow passwords that do not contain a number. Test: Attempt to enter a password that does not contain a number and observe if GEM allows it.

7.10.4 GEM shall not allow passwords that do not contain a special character.

Test: Attempt to enter a password that does not contain a special character and observe if GEM allows it.

7.11 GEM shall have an initial login screen in the user interface. Test: Open the application and verify there is a login screen.

7.12 GEM shall prompt the user for their username and password on the login screen.

Test: Visually verify that username and password are prompted when entering the login screen.

7.13 GEM shall allow only authorized users access to GEM. Test: Have an authorized user attempt to access GEM.

Test: Have an unauthorized user attempt to access GEM.

7.14 GEM shall direct the user to the dashboard after a successful login attempt.

Test: Have an authorized user log in and verify they land on the dashboard.

7.15 GEM shall inform the user the credentials they entered are invalid in the case of an unsuccessful login.

Test: Have a user enter invalid credentials and verify the application informs them of the invalid credentials.

7.16 GEM shall have an administrative account.

Test: Attempt to sign into the administrative account.

7.17 GEM shall have the ability to create a new basic user account. Test: Attempt to have the administrative account create a new basic user account and then attempt to sign in with the newly created basic user account.

7.18 GEM shall have basic user accounts that will be authorized to log in to GEM.

Test: Attempt to sign in with a basic user account.

7.19 GEM shall prevent basic users from creating authorized accounts. Test: Attempt to create a new authorized account while signed in as a basic user.

7.20 GEM shall have the ability to assign roles to a generic account. Test: Have an administrative account attempt to assign a role to a generic account.

7.21 GEM shall allow traversal to all necessary interfaces from the starting interface.

Test: Attempt to traverse to all interfaces from the starting interface

7.22 GEM shall have an account management interface connected to the starting interface.

Test: Attempt to enter the account management interface from the starting interface.

7.23 GEM shall only allow an account with administrator privileges to access the account management interface.

Test: Attempt to enter the account management interface with an account without administrator privileges.

7.24 GEM shall prompt the user to either create a user account, delete a user account, reset a password, or list the usernames of the user accounts and their privileges in the account management interface. Test: Navigate to the account management page and visually validate that the above-mentioned options appear.

7.25 GEM shall prompt the user for the credentials of the new account if the user responds with create in the account management interface.

Test: Visibly verify that a prompt is displayed after the user chooses the ‘create new account’ action.

7.26 GEM shall deny duplicate usernames and inform the user of the failure to create the account.

Test: Attempt to create an account with an already taken username, and visibly inspect that GEM responds with the appropriate message.

7.27 GEM shall create a new account if it is not a duplicate username. Test: Attempt to create a new account with a non-duplicate username and ensure that the new account is successfully created. Then attempt to create a new account with a duplicate username and ensure that the new account is unable to be created.

7.28 GEM shall prompt the user for the username of the account they wish to delete if following a deletion request.

Test: Attempt to delete an account and verify that a prompt for a username is shown.

7.29 GEM shall prompt the user for the (administrative) password to confirm the deletion.

Test: Attempt to delete an account and verify that GEM prompts for the administrative password.

7.30 GEM shall delete the user from GEM if the administrative password was correct.

Test: Verify that the user is deleted from GEM after the administrative password is correctly input.

7.31 GEM shall deny the deletion of the administrative account.

Test: Attempt to delete the administrative account and verify that the administrative account is not deleted.

7.32 GEM shall have a secure developer account that may only reset the password of the administrator in an emergency.

Test: Login to the developer account and attempt to change the administrator password.

7.33 GEM shall prompt the user for the username of the target account in the case of a password reset.

Test: Verify that a prompt for the target account username is displayed when using the change user password interface.

7.34 GEM shall prompt the administrator for a new password for the target account if the response matches a user.

Test: Verify that a new password prompt is displayed after the username prompt in the change password interface.

7.35 GEM shall have a screen that lists all users sorted by username. Test: Navigate to the user list screen and verify that all users appear in the list of accounts and appear in sorted order.

7.36 GEM shall allow the user to select another task in the account management interface in two clicks or fewer.

Test: Verify that each task in the account management interface can be navigated to in two or fewer clicks.

7.37 GEM shall allow the user to traverse to another interface from the account management interface.

Test: Attempt to navigate to another interface from the account management interface.

7.38 The login screen shall display the FGPMS logo on the page.

Test: Visually look and ensure the FGPMS logo is displayed on the screen.

### 8.0 Child Assignment Requirements

8.1 GEM shall have a user interface screen for child assignment.

Test: Verify GEM contains a child assignment screen by visual inspection of the system.

8.2 GEM shall allow the student's needs/conditions to be user definable Test: Attempt to create a student with needs and conditions that are not currently in GEM.

8.3 GEM shall sum up the total number of children assigned to each volunteer.

Test: Visually check that the total children assigned to each volunteer are present in the report.

8.4 The child assignment page shall display each child assigned to a selected   
volunteer and their respective conditions, student needs and desired outcomes.  
Test: Visually check that the total children for each volunteer are present in   
the report and show their conditions, student needs and desired outcomes.

8.5 The system shall allow the user to create, read, update, and delete any of the child assignments to a volunteer.  
a. Test: Attempt to create, read, update, and delete child assignments for a   
volunteer.

### 9.0 Volunteer Information Requirements

9.1 GEM shall contain a volunteer general information screen.

Test: Verify GEM contains a volunteer information screen by visual inspection of the system.

9.2 GEM shall be able to display the currently active and inactive volunteers at the start of the grant or fiscal year.

Test: Navigate to the volunteer’s page and verify the statuses of inactive and active can be seen.

9.3 GEM shall allow all volunteers to be viewed and edited within a single location.

Test: Have a user navigate to the volunteer’s page and attempt to view and edit a volunteer and verify it was completed successfully.

9.4 GEM shall allow a user to change the active status of a volunteer. Test: attempt to change the active status of a volunteer that is in GEM. Validate that the volunteer no longer shows up when active volunteers are listed.

9.5 GEM shall contain volunteer contact information including the following: Name

Site Phone Address Email

Alternate Phone Active/Current Status Start Date

End Date

Test: Navigate to a volunteer and verify the listed information is seen.

9.6 GEM shall allow a user to add a new volunteer to the volunteer page. Test: Have a user navigate to the volunteer page and attempt to add a new volunteer and verify it successfully added them.

9.7 GEM shall allow a user to edit an existing volunteer on the volunteer page.

Test: Have a user navigate to the volunteer page and edit a volunteer and verify it updated the record.

9.8 GEM shall allow a user to delete a volunteer on the volunteer page. Test: Have a user navigate to the volunteer page and delete a volunteer and verify it deleted the record.

9.9 GEM shall display the status of volunteer requirements for employment including items such as:

File Photo

Service Description Orientation Training Sheet

Confidential Statement of Understanding Scheduled Photo Release

Emergency Beneficiary Form

Test: Navigate to the volunteer information page and verify this information is displayed.

9.10 GEM shall indicate if each requirement is annual or one-time.

Test: Visually verify that each required field indicates if a requirement is a one-time or annual field.

9.11 GEM shall use a ‘details’ field to handle any ancillary information the agency would want to record about a volunteer.

Test: When editing a volunteer’s information visually inspect that the details field is present.

9.12 GEM shall allow the service start date to potentially contain multiple values (at least 2).

Test: Attempt to add a second start date for a volunteer.

9.13 GEM shall allow the user to navigate between the child profile and the assigned volunteer profile.

Test: While on a child profile, validate that there is a means that allows the user to view the assigned volunteer information.

10.0 Volunteer Activity Log Requirements

10.1 GEM shall contain an activity log interface screen.

Test: Verify GEM contains an activity log screen by visual inspection of the system.

10.2 GEM shall display the following information about a volunteer and their absence from work:

Date Absent Initials Reason

Test: Navigate to the activity log and visually verify that the above fields are present.

10.3 GEM shall allow a user to filter by volunteer name and year.

Test: Have a user attempt to filter the activity log by volunteer name and year and verify it shows proper data.

10.4 GEM shall allow a user to create a logged activity for a specific volunteer.

Test: Have a user attempt to create a logged activity and verify it works successfully.

10.5 GEM shall allow a user to edit a logged activity for a specific volunteer Test: Have a user attempt to edit a logged activity and verify it works successfully.

10.6 GEM shall not allow the activity log to be cleared out by a secondary user.

Test: Attempt to clear out the activity log by a secondary user and ensure that the log is not cleared out.

10.7 GEM shall allow an administrator to delete a logged activity for a specific volunteer.

Test: Have an administrative account attempt to delete a logged activity and verify it works successfully.

### 11.0 Report Page Requirements

11.1 GEM shall track each volunteer’s one-time checks to be exported as a report.

Test: Attempt to view the one-time check for a volunteer on the reports page.

11.2 GEM shall track each volunteer’s annual checks to be exported as a report.

Test: Attempt to view annual checks for a volunteer on the reports page.

11.3 GEM shall filter the report page by current volunteers.

Test: Attempt to filter the report page by current volunteers.

11.4 GEM shall filter the report page alphabetically.

Test: Attempt to filter the report page alphabetically.

11.5 GEM shall display checks in a color-coded manner defined by the program director.

Test: Visually inspect checks to make sure they match the color code.

11.6 GEM shall allow sending reports directly to the printer for future viewing.

Test: Attempt to print a report.

11.7 GEM shall display expandable rows to display all volunteers’ check information.

Test: Attempt to select a volunteer to view expanded information.

11.8 GEM shall track all volunteers’ personal information to be exported as a report.

Test: Attempt to view volunteers’ information.

11.8.1 GEM shall track a volunteer’s name.

Test: Attempt to check for the volunteer’s name.

11.8.2 GEM shall track a volunteer’s status(active/inactive).

Test: Attempt to check for volunteers’ active status.

11.8.3 GEM shall track a volunteer’s start date.

Test: Attempt to check for the volunteer’s start date.

11.8.4 GEM shall track a volunteer’s end date (only for those whose field applies).

Test: Attempt to check the volunteer’s end date if they have left the program.

11.8.5 GEM shall track a volunteer’s date of birth.

Test: Attempt to check the volunteer’s date of birth.

11.8.6 GEM shall track a volunteer’s gender.

Test: Attempt to check the volunteer’s gender.

11.8.7 GEM shall track a volunteer’s preferred pronouns.

Test: Attempt to check the volunteer’s preferred pronouns.

11.8.8 GEM shall track a volunteer’s ethnicity.

Test: Attempt to check the volunteer’s ethnicity.

11.8.9 GEM shall track a volunteer’s race.

Test: Attempt to check the volunteer’s race.

11.8.10 GEM shall track a volunteer’s veteran status.

Test: Attempt to check the volunteer’s veteran status.

11.8.11 GEM shall track whether a volunteer is family of military.

Test: Attempt to check whether the volunteer is part of a family of someone who served or is serving in the military.

11.9 GEM shall allow for the editing of volunteers’ information directly from the reports page.

Test: Attempt to edit the information of a volunteer directly from the reports page.

### 12.0 Report Builder Requirements

12.1 GEM shall have a report builder for volunteer information.

Test: Navigate to GEM’s report builder subsystem.

12.2 GEM shall allow the user to select which set of volunteers (active/inactive) should be displayed in the report.

Test: Validate that both active and inactive volunteer options are present when trying to generate a report.

12.3 GEM shall allow a user to select which fields should be displayed in a generated report.

Test: Attempt to generate a report and verify that the user can select the fields to be included in the report.

12.4 GEM shall allow the user to save a generated report.

Test: Have a user generate a report and attempt to save it as a local copy and verify the save was successful.

12.5 GEM shall generate a report containing volunteers and the information of all their assigned children.

Test: Attempt to generate the above report.

12.6 The system shall all have an intelligent report builder that allows the end user to specify any valid data field in the builder into the report.

Test: Inspect the ‘Reports’ interface and view the ‘Report Builder’ tab. Inspect the customizable options. Make the desired customizable selections. Select the data fields to generate. Select the ‘Generate Report’ button. Verify that the report has been correctly generated.

12.7 GEM shall allow the end user to save the report’s builder options as a preset to be used anytime again.

Test: Inspect the ‘Reports’ interface and view the ‘Report Builder’ tab. Inspect the customizable options. Make the desired customizable selections. Select the data fields to generate. Select the ‘Generate Report’ button. Select the option to save the report as a preset. Name the preset and confirm addition. Verify in the ‘Presets’ drop- down list that the report preset has been saved.

12.8 GEM shall allow the end user to overwrite an existing preset.

Test: Inspect the ‘Reports’ interface and view the ‘Report Builder’ tab. Inspect the customizable options. Make the desired customizable selections. Select the data fields to generate. Select the ‘Generate Report’ button. Select the option to save the report as a preset. Give the preset the same name as to the desired preset to overwrite and confirm addition. Verify in the ‘Presets’ drop-down list that the

newly report preset has been saved.

12.9 The system shall allow the end user to delete any existing preset.

Test: Inspect the ‘Reports’ interface and view the ‘Report Builder’ tab. Inspect the ‘Presets’ drop-down list and select option to delete the selected preset item. Confirm deletion. Verify in the ‘Presets’ drop-down list that the deleted preset no longer is found within the list.

12.10 GEM shall allow the end user to select a new time range for any existing preset.

Test: Inspect the ‘Reports’ interface and view the ‘Report Builder’ tab. Select any preset in the ‘Presets’ drop-down list. Confirm the start and end date for the report. Generate the report. Verify that the report has the desired time range.

### 13.0 Data Requirements

13.1 GEM shall allow a user to delete a volunteer on the volunteer page and the volunteer will be marked as inactive in the database.

Test: Let the user navigate to the volunteer page and delete a volunteer and verify the record is marked as inactive in the database.

13.2 GEM shall allow a user to change the active status of a volunteer. Test: attempt to change the active status of a volunteer that is in GEM. Validate that the volunteer no longer shows up when active volunteers are listed.

13.3 GEM shall allow an authorized user to add a foster grandparent's demographic information to GEM’s database.

Test: Use the system functionality to add demographic information for a Foster Grandparent and add a new Foster Grandparent. Verify that the new Grandparent is in the database.

13.4 GEM shall allow data to be exported to Microsoft Excel spreadsheets. Test: Attempt to have GEM export data to Excel. Check the exported file and ensure that the data is the same data that was meant to be exported.

13.5 GEM shall allow a user to record that a gift was received by the Foster Grandparent Program.

Test: attempt to record a gift in GEM, and then enter the view for gifts and validate that the new gift is present.

13.6 GEM shall allow a user to record that a donation was received by the Foster Grandparent Program.

Test: attempt to record a donation in GEM, and then enter the view for donations and validate that the new donation is present.

* 1. Performance Requirements

1.0 GEM shall start in less than 5 seconds.

2.0 The database shall be normalized to reduce query times.

3.0 Queries shall time out after 30 seconds of no response.

4.0 GEM shall not force a timed logout.

* 1. Logical Database Requirements[I],[J]

1.0 The database shall store data about a volunteer.

This data shall include volunteer id, address id, first name, last name, age, phone number, alternate phone number, date of birth, separation date, separation reason, gender, ethnicity, race, identifies as, is family of military, is veteran, email address, and active status.

2.0 The database shall store data about a school.

This data shall include school id, address id, name, school hours, principal, secretary, contact number, start time, and end time.

3.0 The database shall store data about a student.

This data shall include student id, and student identifier.

4.0 The database shall store data about an assignment of a volunteer to a school.

This data shall include assignment id, volunteer id, school id, days, start time, end time, hour per day, number of days, hour per week, teacher, grade, room, total classroom, total assigned, total ages 5 to 12, and total ages to 5.

5.0 The database shall store data about a user of the system.

This data shall include user id, username, password, role, and active status.

6.0 The database shall store data about the PTO stipend rate.

This data shall include PTO stipend rate id, stipend rate, PTO rate, and date.

7.0 The database shall store data about the meal transport rate.

This data shall include meal transport rate id, meal value, mileage rate, and date.

8.0 The database shall store data about the school cost share.

This data shall include school cost share id, name, cost, and date.

9.0 The database shall store data about in-kind items.

This data shall include in-kind id, volunteer id, expense type id, date, and value.

10.0 The database shall store data about expense types.

This data shall include expense type id, name, and description.

11.0 The database shall store data about school and volunteer addresses.

This data shall include address id, address, city, state, and zip.

12.0 The database shall store data about volunteer one-time checks.

This data shall include volunteer check id, volunteer id, has file photo, has service description, has training sheet, confidence SOU date, service start date, has NSCHC, has background check, has id copy, NSOPW date, iChat date, true screen date, alias fingerprint date, field print date, DHS date, and TB shot date.

13.0 The database shall store data about volunteer annual checks.

This data shall include volunteer annual check id, volunteer id, photo release date, emergency beneficiary date, HIPPA release date, physical date, car insurance date, covid shot date, and year.

14.0 The database shall store data about volunteer meal mileage.

This data shall include meal mileage id, volunteer id, meal count, total meal value, bus ride count, mileage, total mileage value, and date.

15.0 The database shall store data about volunteer PTO and stipends.

This data shall include PTO stipend id, volunteer id, regular hours, PTO start, PTO end, PTO used, PTO earned, stipend paid, year to date hours, date.

16.0 The database shall store data about the separation reason for a volunteer.

This data shall include reason separated id, name, and description.

17.0 The database shall store data about the students’ volunteers are assigned to.

This data shall include assignment student id, assignment id, student id, desired outcome, and date.

18.0 The database shall store data about the condition a student has.

This data shall include student condition id, student id, and condition id.

19.0 The database shall store data about student conditions.

This data shall include condition id, acronym, and description.

20.0 The database shall store data about needs a student has.

This data shall include student need id, student id, and need id.

21.0 The database shall store data about student needs.

This data shall include need id, acronym, and description.

22.0 The database shall store data about report presets.

This data shall include preset name, sort by, volunteer status, volunteer id, start date, and end date.

23.0 The database shall store data about what information should be gathered by a preset.

This data shall include report preset id, and report file explorer id.

24.0 The database shall store data about the preset selections.

This data shall include report file explorer id, and name.

25.0 The Data Repository shall be able to delete an authorized user from the data repository

Test: An authorized user deletes an authorized user from the Data Repository

26.0 The Data Repository shall be able to edit an authorized user in the data repository.

a. Test: The authorized user changes their password.  
b. The Data Repository changes the value stored for their password  
c. The system now allows them to login with their new password.

27.0 The database shall store a physical for each volunteer, including the data and price.

28.0 The database shall be able to store each donation.

Test: Confirm that each donation is stored in the data storage medium.

29.0 The database shall be able to store each donation category.

Test: Confirm that each donation category is stored in the database.

* 1. Design Constraints

1.0 GEM is a Windows 10 and 11 desktop application.

2.0 Reports must be able to be generated in Excel.

3.0 Monthly site forms must be able to be generated and filled out in Word.

GEM must be able to support multiple users on the system at any given time.

4.0 The system shall have a dialog to always display a confirmation when deleting a data   
record.  
 Test: Visually verify that the delete dialog is showing up when any save action   
 involves removing any data. Verify that clicking cancel does not move forward   
 with the save. Verify that canceling the save reverts the data deleted to its   
 original state.

5.0 The system’s UI shall allow a Program Coordinator to access a specific volunteer’s   
 information in no more than 20 seconds.  
 Test: Time the program coordinator to see how long it will take to find a   
 volunteer’s information.

6.0 The system’s UI shall allow a Program Coordinator to access a specific site’s information   
 in no more than 20 seconds.  
 Test: Time the program coordinator to see how long it will take to find a site's   
 information.

7.0 The system’s UI shall contain a visual indication when a volunteer requires annual   
 checks to be turned in for the program in both the annual checks report and the   
 individual volunteer pages.  
 Test: Visually verify that a particular annual check has not been satisfied and that  
 is stands out when compared to an annual check that has been satisfied when   
 reviewing the volunteer annual checks report or a specific volunteer page.

8.0 The system’s UI shall contain the Saginaw Area Foster Grandparent Program Logo on   
 every screen  
 Test: Visually verify that the Logo is on every screen of the program.  
9.0 The system’s UI shall be in a 16:9 aspect ratio  
 Test: Visually verify that the UI looks correct on multiple different sized monitors  
 10.0 The system’s UI shall follow the three clicks rule (No more than three clicks to find any   
 information)  
 Test: Starting from the dashboard, the user should be able to find volunteer   
 information and school information in under three clicks.  
11.0 The system shall be split into six different sections for an administration account. (Home or Dashboard, Volunteer, Schools, Financial, Users, and Reports)  
 Test: The Program Coordinator shall go into each of the six sections of the   
 program making sure each section is working correctly while logged into an administration account.

11.0 The system shall be split into five different sections for a non-administration account. (Home or Dashboard, Volunteer, Schools, Financial, and Reports)  
 Test: The Program Coordinator shall go into each of the five sections of the   
 program making sure each section is working correctly while logged into a non-administration account.

* 1. Software System Attributes

1.0 GEM shall be available 99.98 percent of the time to users every month during business hours.

Test: Take the total downtime for a month and divide it by the total time in a month. Subtract the result from one and multiply by 100 to get the percent uptime. Check if it meets the availability requirement.

2.0 The mean time to restore GEM shall not exceed fifteen minutes.

Test: Add up the total downtime for a month and divide it by the number of incidents. Check to see if it exceeds fifteen minutes.

3.0 GEM shall check that all inputs are valid concerning their fields.

Test: Enter incorrect data into fields. Check that the system does not allow data to be inputted.

4.0 GEM shall notify the user of which fields are incorrect when invalid data is entered.

Test: Enter incorrectly formatted data and visually inspect the notification.

5.0 User passwords shall be encrypted before being stored in the database. Test: Create a user and check if the decrypted password matches up with the encrypted password stored in the database.

6.0 GEM shall retain field information after a user is redirected back to a page after a failure.

Test: Attempt to fill out data and enter invalid data into one of the fields. Attempt to save. Visually inspect that data is retained after being redirected back.

7.0 GEM shall notify the users when they are missing the required form fields. Test: Attempt to fill out data and save. Visually inspect that the page redirects back and displays a missing field error.

8.0 GEM shall timeout on a data request that has reached sixty seconds.

Test: Shut down the database and attempt to retrieve data. Visually inspect that GEM stops the request at sixty seconds.

9.0 GEM shall notify the user when the database cannot be reached.

Test: Shut down the database and attempt to retrieve data. Visually inspect that GEM displays an error message.

10.0 GEM shall notify the user if the Excel API cannot be reached.

Test: Uninstall Excel. Attempt to export any data to Excel. Visually inspect that GEM displays an error message.

11.0 GEM shall notify the user if the Word API cannot be reached.

Test: Uninstall Word. Attempt to export any data to Word. Visually inspect that GEM displays an error message.

12.0 GEM shall be able to be downloaded onto Windows 10 and Windows 11 systems via an executable.

Test: Attempt to download software onto a system with Windows 10 and Windows 11. Inspect that the software is working.

13.0 GEM shall be able to be updated to different versions while it is deployed on multiple computers.

Test: Attempt to push out a software update. Inspect that the users can download the new update.

14.0 GEM shall provide the use of navigation bars to jump from view to view.

Test: Visually inspect that there are navigation bars and that they link to different pages.

15.0 GEM shall use standard icons for toolbars and functions.

Test: Visually inspect that there are clickable icons that perform actions relevant to the icons.

16.0 GEM shall have icons for functions (updated, save, delete, print, etc..) placed in the same place from page to page for consistency.

Test: Visually inspect that icons are placed in the same place on each page.

17.0 GEM shall prompt the user for confirmation when attempting to delete records. Test: Attempt to delete a record and visually inspect that a delete confirmation window appears.

3.7 HIPAA Compliance Requirements

1.0 The FGPMS shall be HIPAA compliant.

Test: Evaluate the application by auditing the organization using the HHS Office

of Civil Rights, HIPAA Audit Protocol. Expected policies and procedures are

outlined for HIPAA compliance. Verify that the policies and procedures meet the

HIPAA requirements and that these policies have been implemented.

2.0 The FGPMS shall enforce the HIPAA Privacy Rule.

Test: Any volunteer that requests their own personal and medical information that is

on file will not be denied. Volunteers will be denied access to other

volunteer’s personal information. Next of kin is restricted from access of

information unless with explicit consent. Verify that every volunteer request is

upheld for their personal information and any other volunteer information is denied.

3.0 The FGPMS shall enforce the HIPAA Security Rule.

Test: Verify that the system is only accessible by authenticated users within a

closed space.

4.0 The FGPMS shall enforce the HIPAA Breach Notification Rule.

Test: On the event of a system data breach, verify that the Department of Health

and Human Services are notified and that every volunteer is informed of the

data breach.

# Appendices

## Cost/Benefit Analysis

A cost-benefit analysis was conducted using the estimated data provided by Mrs. Tara Stinson. Although we cannot calculate ROI without costs, the NPV is positive, and the payback starts during the first year of use. Therefore, we can conclude that this project would be worth pursuing.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  | Low | High | Average |
| Current time spend weekly |  |  |  | 20 hrs. | 25 hrs. | 22.5 hrs. |
| Estimated time spent with software weekly |  |  |  | 7 hrs. | 10 hrs. | 8.5 hrs. |
| Hourly Cost |  |  |  | 32.20/Hr. | 48.73/hr. | 40.47/hr. |
|  |  |  |  |  |  |  |
| Per Year |  |  |  | Low | High | Average |
| Original Cost |  |  |  | 33488.00 | 63349.00 | 48418.50 |
| With Software |  |  |  | 11720.80 | 25339.60 | 18530.20 |
| Savings |  |  |  | 22604.40 | 38009.40 | 29888.30 |
|  |  |  |  |  |  |  |
|  | Year 1 | Year 2 | Year 3 | Year 4 | Year 5 | Totals |
| Benefits |  |  |  |  |  |  |
| Savings | 29888.30 | 29888.30 | 29888.30 | 29888.30 | 29888.30 | 149441.50 |
| Total Benefits | 29888.30 | 29888.30 | 29888.30 | 29888.30 | 29888.30 | 149441.50 |
|  |  |  |  |  |  |  |
| Total Costs | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Year 1 | Year 2 | Year 3 | Year 4 | Year 5 | Totals |
| Total Benefits | 29888.30 | 29888.30 | 29888.30 | 29888.30 | 29888.30 | 149441.50 |
| Total Costs | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| Ben - Costs | 29888.30 | 29888.30 | 29888.30 | 29888.30 | 29888.30 | 29888.30 |
|  |  |  |  |  |  |  |
| Discount Rate | 0.00 |  |  |  |  |  |
| Discount Factor | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |  |
| Discounted Benefits | 29888.30 | 29888.30 | 29888.30 | 29888.30 | 29888.30 | 149441.50 |
| Discounted Costs | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
|  |  |  |  |  |  |  |
| Disc Ben - Cost | 29888.30 | 29888.30 | 29888.30 | 29888.30 | 29888.30 | 149,441.50 |
| Cumulative Ben - Cost | 29888.30 | 59,776.60 | 89,664.90 | 119,553.20 | 149,441.50 | ^NPV |
|  | ^ Payback |  |  |  |  |  |
| Non-Disc ROI | N/A |  |  |  |  |  |
| Disc ROI | N/A |  |  |  |  |  |

## Context Diagrams

Internet

Foster Grandparent Program Manager

Windows

10/11 Users

Inputs Mouse

Keyboard Excel

Volunteer Data

Site Data

Student Data

Budget Data

Report Data

Word

External Storage

Internal Storage

Stakeholders

Database System

Login

NSCHC Accountant

Directors

Developers

Credentials Coordinators

**Figure 4 General overview of outside factors that GEM system will interact with**

## System Modeling

|  |  |  |
| --- | --- | --- |
| User Volunteer Transpor Windows  Manual Timecards Scheduling Budget -tation Time 10/11  ( )  Login credentials Word Formatted Data  Volunteer data Manage Foster  Managed Data  Site data Grandparent Program  Student data  Managed Report  Budget data Data  Report data A0 Excel Formatted Data  ( ) NO: 2 ( )  Input Database Word Excel Windows 10/11  Devices | | |
| NODE: A-0 | TITLE: Manage Foster Grandparent Program Data | NO.: 1 |

**Figure 5 Highest-level overview of the GEM system**

User

Manual

Volunteer

Timecards Scheduling Budget

Transport

-tation Time

Login

credentials

Authenticated User

Word Formatted Data

Volunteer data

Site data

Student data

Budget data

Word Formatted

Data

Managed Data

Managed Reports

Excel Formatted

Report

Report data

Database

Word

Manage Foster Grandparent Program Data

Excel

TITLE:

2

NO.:

A0

NODE:

3

Manage

Reports

Managed Data

2

Manage

Data

1

Log In

**Figure 6 Overview of the 3 main functions of the GEM system**



|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | GEM  System | |  | | | |
|  | | | | |  | | | | |
|  | | | | User Management Subsystem | |  | | | |
|  | | | | |  | | | | |
|  | | | | CRUD  Subsystem | |  | | | |
|  | | | | |  | | | | |
|  |  | |  | |  | |  | |  |
| Add User Func | | Delete User Func | | Give Permissions Func | | Update Data Func | | View Users Func | |

**Figure 7 System model for user management subsystem**



|  |  |
| --- | --- |
| GEM  System | |
|  |  |
| Dashboard Management Subsystem | |
|  |  |
| Display Charts Func | |

**Figure 8 System model for dashboard management subsystem**



|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | | | | GEM  System | |  | | | | | | | | | | | |
|  | | | | | | | | | | | | |  | | | | | | | | | | | | |
|  | | | | | | | | | | | | Report Management Subsystem | |  | | | | | | | | | | | |
|  | | | | | | | | | | | | |  | | | | | | | | | | | | |
|  | | | |  | | | | | | | | |  | | | | | | | | |  | | | |
|  | | | Report Builder Subsystem | |  | | | | | | | Annual Check Subsystem | |  | | | | | | | Volunteer Info Subsystem | |  | | |
|  | | | |  | | | | | | | | |  | | | | | | | | |  | | | |
|  |  | | |  | | |  | | |  | | |  | | |  | | |  | | |  | | |  |
| Build Report Func | |  | Generate Report Func | |  | Save Preset Func | |  | Filter Func | |  | Export to Excel Func | |  | Print Func | |  | Filter Func | |  | Edit Volunteers Func | |  | Print Func | |

**Figure 9 System model for report management subsystem**

GEM



System

Budget Management Subsystem

Expenses Management Subsystem

Update Cost Share Func

Generate Report Subsystem

Calculate Annual Totals Subsystem

Add New Expense Type Func

Add New Expense Record Func

Update Existing Expense Record Func

Delete Existing Expense Record Func

Calculate Expense Totals Func

Generate Fiscal Year Report Func

Generate Grant Year Report Func

Display Report Func

Export Report Func

Calculate Fiscal Year Totals Func

Calculate Grant Year Totals Func

**Figure 10 System model for budget management subsystem**

GEM



System

Budget Management Subsystem

Stipend/PTO Management Subsystem

Generate Monthly Report Subsystem

CRUD

Subsystem

Flag Non- eligible Volunteers Func

Calculate Data Subsystem

Generate Monthly Report Func

Print Report

Func

Update Data Func

Display Data Func

Create Data Func

Delete Data Func

Calculate Period Totals Subsystem

Calculate Hours Data Subsystem

Calculate Stipend Data Subsystem

Calculate PTO Data Subsystem

Calculate Quarterly Totals Func

Calculate Annual Totals Func

Calculate PTO Earned Func

Calculate PTO Start Func

Calculate PTO End Func

Calculate YTD Stipends Paid Func

Calculate Stipends Paid This Period Func

Calculate Beginning Stipends

Calculate Stipend Pd Func

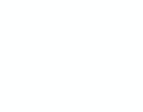
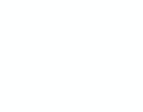
Calculate Current Month Remaining Stipends Func

Calculate YTD Reg Hours Func

Calculate YTD Hours Func

**Figure 11 System model for budget management subsystem**

GEM



System

Budget Management Subsystem

Filter

Subsystem

Monthly In- Kind Subsystem

Navigate To Single Volunteer Budget Func

Reset Filter

Func

Apply Filter

Func

Calculate Data

Subsystem

Flag Non- inactive Volunteers Func

Calculate Period Totals Subsystem

Calculate Meal Data Subsystem

Calculate Mileage Data Subsystem

Calculate Quarterly Totals Func

Calculate Annual Totals Func

Calculate Monthly Total Meal Value Func

Calculate Monthly Total Meals Func

Calculate YTD Site Meals Func

Calculate YTD Total Meal Value Func

Calculate Monthly Total Bus Rides Func

Calculate Monthly Total Mileage Func

Calculate Monthly Total Mileage Cost Func

Calculate YTD Total Bus Rides Func

Calculate YTD Total Mileage Func

Calculate YTD Total Mileage Cost Func

**Figure 12 System model for budget management subsystem**



Log Out Func

Credential Verification Func

Account Recovery Subsystem

Log In Subsystem

GEM

System

Send Email Func

**Figure 13 System model for login subsystem**

Send Email Func

Password Validation Func

Account Creation Subsystem



|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | | | GEM  System | |  | | | | |
|  | | | | | | | | | | | |  | | | | | |
|  | | | | | | | | | | | Volunteer Management Subsystem | |  | | | | |
|  | | | | | | | | | | | |  | | | | | |
|  | | | | | | | |  | | | | | | | |  | |
|  | | | | | | | Activity Log Management Subsystem | |  | | | | | | Filter Subsystem | |  |
|  | | | | | | | |  | | | | | | | |  | |
|  |  | | |  | | |  | |  | | |  | | |  | |  |
| Delete Activity Log Func | |  | Display Activity Log Func | |  | Add New Entry Func | | Update Entry Func | |  | Print Activity Log Func | |  | Reset Filter | | Apply Filter | |

**Figure 14 System model for volunteer management subsystem**

GEM



System

Volunteer Management Subsystem

All Volunteer Management Subsystem

Individual Volunteer Management Subsystem

Display Volunteer Data Func

Add New Volunteer Func

Generate Report Subsystem

Update Volunteer Row Func

Go To Individual Volunteer Page Func

Go To Assigned Students Page Func

Go To Assigned Site Page Func

Update Volunteer Data Func

View Volunteer Data Func

Generate Report Func

Display Report Func

Print Report Func

Export Report To Excel Func

**FIGURE 15 SYSTEM MODEL FOR VOLUNTEER MANAGEMENT SUBSYSTEM**

GEM



System

Student Management Subsystem

CRUD

Subsystem

Filter Subsystem

Create Data Subsystem

Display Data Subsystem

Delete Data Func

Update Data Func

Filter By School Func

Filter By Name Func

Filter By Student

Information Func

Create General Data Func

Create Student Conditions Func

Create Student Needs func

Display Table Func

Display Student Conditions Func

Display Student Needs Func

**Figure 16 System model for student management subsystem**



Student Management Subsystem

GEM

System

Age Total check Func

Age 6-12 Total Func

Age 0-5 Total Func

Age Total

Subsystem

Assignment

Total Func

Classroom

Total Func

Totals Subsystem

**FIGURE 17 SYSTEM MODEL FOR STUDENT MANAGEMENT SUBSYSTEM**

Go To Site

Page Func

Go To Student

Page Func

Send To Report

Func

Navigation Subsystem



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | GEM  System | |  | |
|  | | |  | | |
|  | | Saving  Subsystem | |  | |
|  | | |  | | |
|  |  | |  | |  |
| Local Save Func | | Cloud Save Func | | Sync Func | |

**Figure 18 System model for saving subsystem**

GEM



System

Site Management Subsystem

CRUD

Subsystem

Filter Subsystem

Generate Monthly Visit List Func

Create Data

Func

Display Data

Func

Delete Data

Func

Update Data

Subsystem

Filter By School Func

Filter By Name Func

Filter By Days Func

Filter By Hours Per Day Func

Filter By Hours Per Week Func

Filter By User Defined Func

One

Grandparent Per Teacher Func

Active / Inactive Func

General Data Update Func

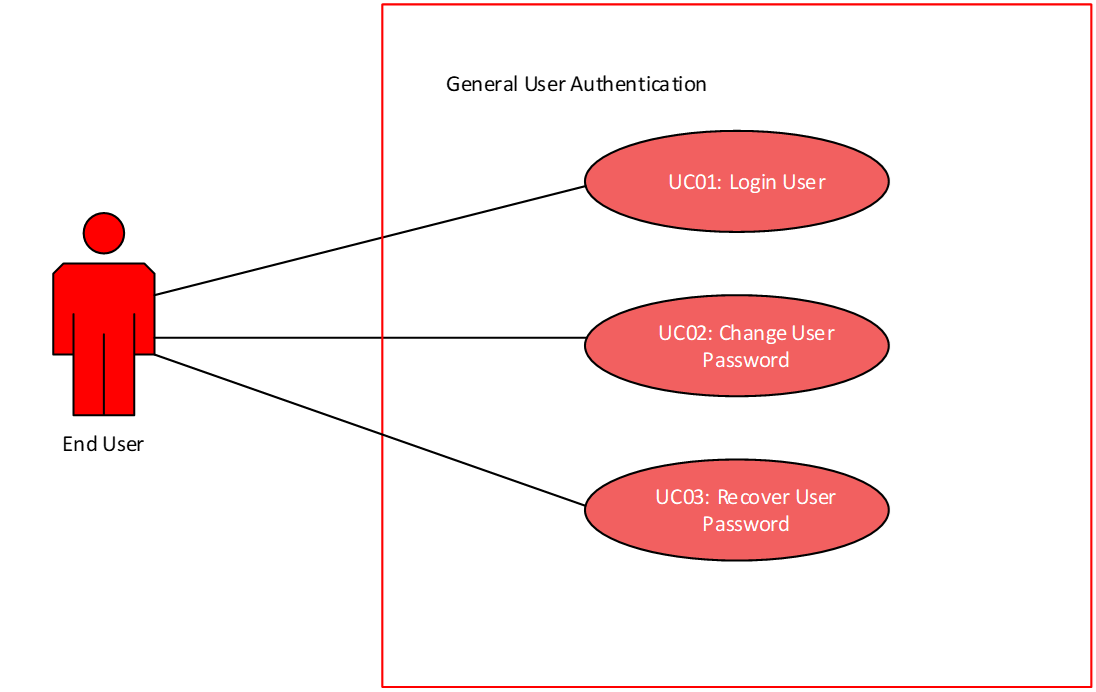
**Figure 19 System model for site management subsystem**

## Use Cases/Expanded Use Cases

|  |  |  |  |
| --- | --- | --- | --- |
| UC01 | | Adding a volunteer | |
| UC02 | | Adding a student | |
| UC03 | | Assigning volunteers to school | |
| UC04 | | Assigning volunteer to student | |
| UC05 | | Logging into GEM | |
| UC06 | | Viewing reports from previous grant years | |
| UC07 | | Viewing student profile | |
| UC08 | | Viewing a volunteer’s profile | |
| UC09 | | Generating volunteer report to Excel | |
| UC10 | | Generating activity log report to Excel | |
| UC11 | | Generating demographic report to Excel | |
| UC12 | | Generating stipend and PTO report to Excel | |
| UC13 | | Generating meal expenditure report to Excel | |
| UC14 | | Generating milage report top Excel | |
| UC15 | | Generating student report to Excel | |
| UC16 | | Generating annual financial report to Excel | |
| UC17 | | Adding to activity log | |
| UC18 | | Recording stipends | |
| UC19 | | Changing timecard | |
| UC20 | | Generating monthly site form | |
| UC21 | | Recording budgetary values | |
| UC22 | | Entering volunteers milage | |
| UC23 | | Entering volunteers’ meal expenditure | |
| UC24 | | Retrieving timecard information | |
| UC25 | | Mass entering of volunteers | |
| UC26 | | Mass entering of students | |
| UC27 | | Entering school cost share | |
| UC28 | | Entering meal/mileage rates | |
| UC29 | | Entering Stipend/PTO rates | |
| UC30 | | Viewing reports from previous fiscal years | |
| UC31 | | Recording PTO | |
| UC32 | | Updating volunteer information | |
| UC33 | | Adding Schools | |
| UC34 | | Deleting Students | |
| UC35 | | Changing a Volunteer’s Status | |
| UC36 | | Add new user | |

**Figure 20 Table listing all use cases of the system**

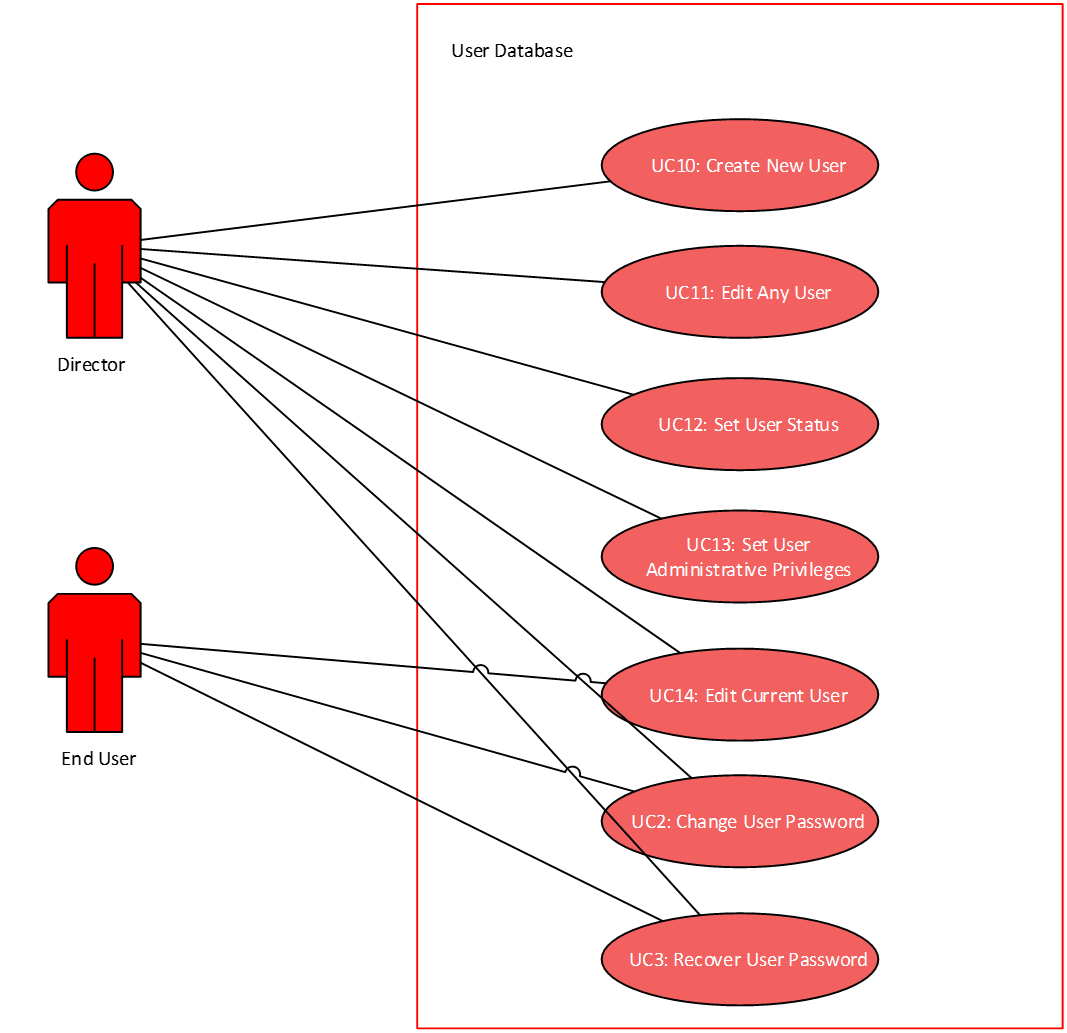
UC05: Logging into GEM



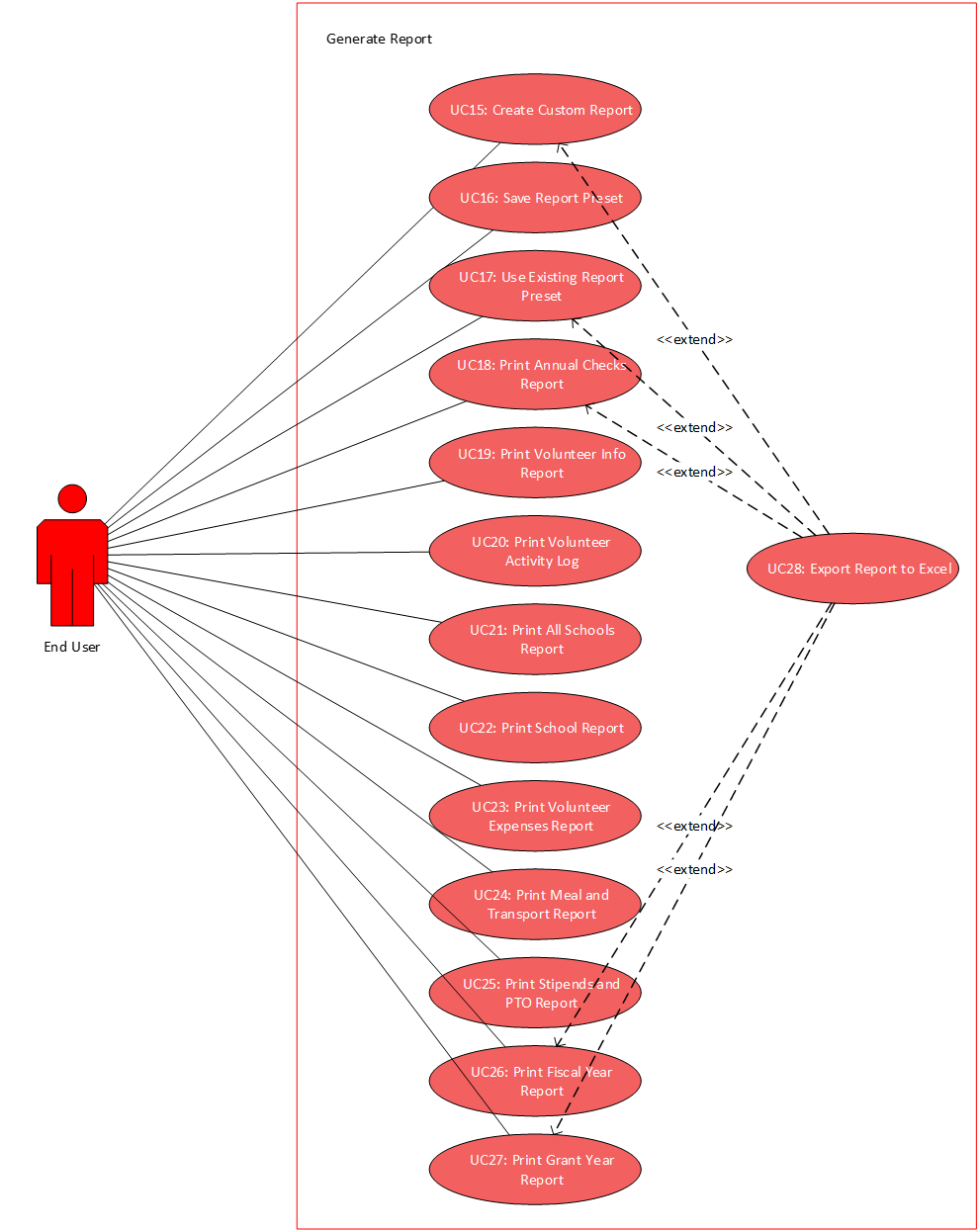
**Figure 21: UML Case Diagram representing the interactions between the end user and the FGPMS when attempting to login.**



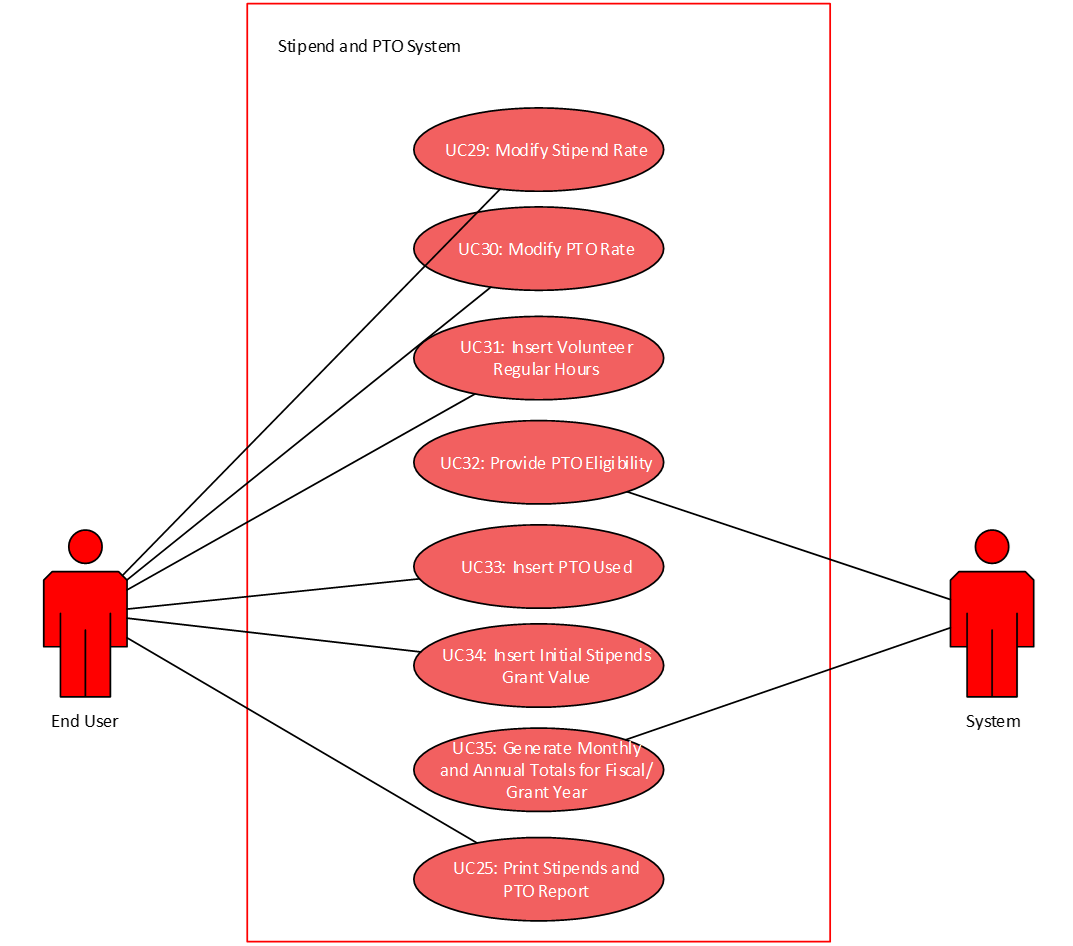
**Figure 22: UML Case Diagram representing the interactions between the Director and the staff members on the volunteer's database.**



**Figure 23: UML Case Diagram representing the interactions between the end user or director when attempting to add/modify users.**



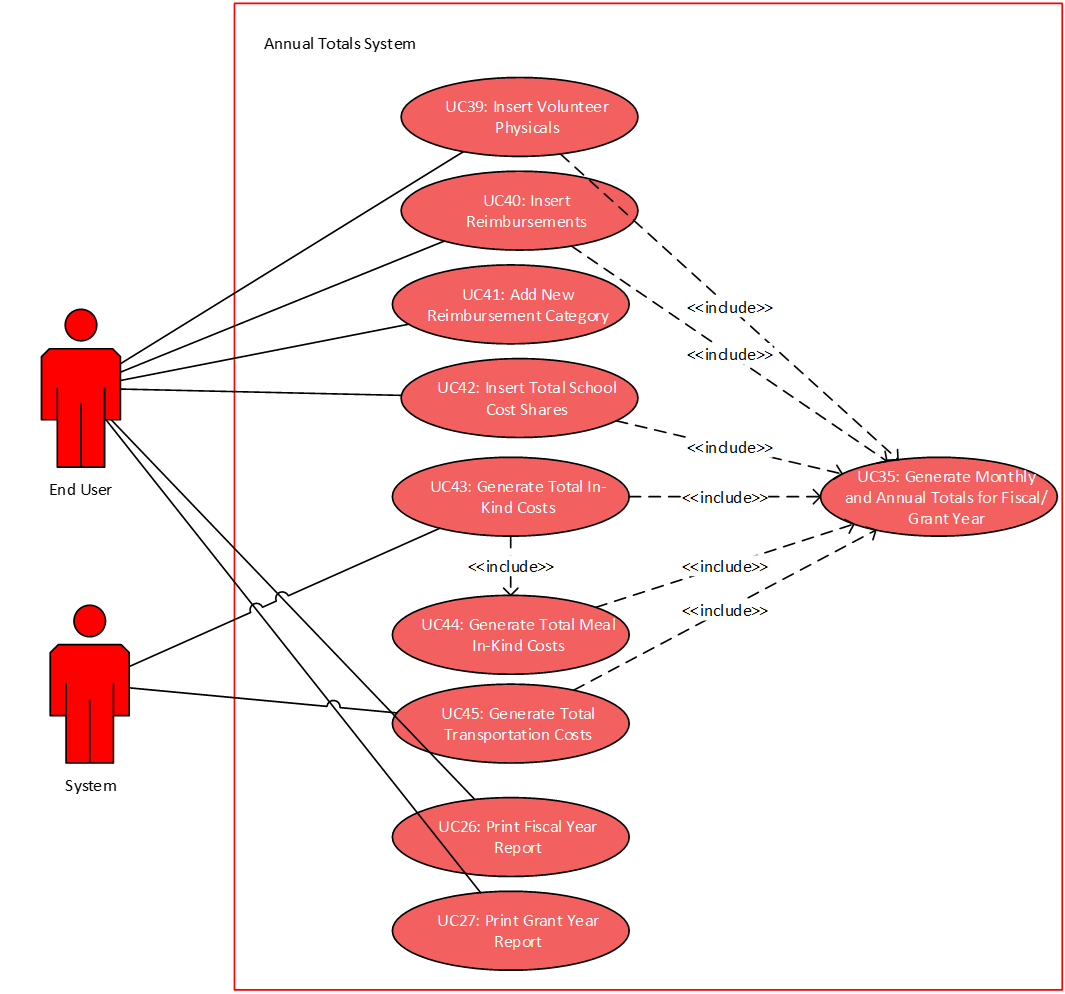
**Figure 24: UML Case Diagram representing the interactions between the end user when generating a report on the application.**



**Figure 25: UML Case Diagram representing the interactions between the end user and the system automation when calculating the stipends and PTO for each volunteer on a monthly basis.**



**Figure 26: UML Case Diagram representing the interactions between the end user and the system automation when calculating the meal and transport costs for each volunteer on a monthly basis.**



**Figure 27: UML Case Diagram representing the interactions between the end user and the system automation when calculating the annual totals for the fiscal and grant year.**

Expanded Use Cases

UC01: Adding a volunteer

|  |  |
| --- | --- |
| Actor: User | System: GEM |
|  | 0. GEM displays home page (unpopulated Volunteers tab) |
| 1. TUCBW User selects the drop down box under volunteer name | 2. The list drops down, populated with all users. At the bottom of the list exists an -add new user- option |
| 3. Tara selects -add new user- | 4. The page populates with empty text boxes to fill with the new volunteers demographics (name, dob, etc…). A Save and Cancel button appear in the bottom right corner |
| 5. Tara: a) Populates the open fields and clicks Save -or- b) Selects Cancel -> returns to Step 0 | 6. A popup window confirms if the entry is to be recorded |
| 7. Tara: a) TUCEW Tara selects Yes -or- b) Tara selects No -> returns to Step 4 |  |

**Figure 27 Expanded Use case for adding a volunteer**

UC02: Adding a student

|  |  |
| --- | --- |
| Actor: User | System: GEM |
|  | 0. GEM displays home page (unpopulated Volunteers tab) |
| 1. TUCBW User selects Students tab | 2. GEM displays unpopulated Students page |
| 3. User selects drop down box under Volunteer Name | 4. Box populates list with all active volunteers. At the bottom of the list exists an -unassigned- option |
| 5. User selects -unassigned- | 6. A table appears in the blank area below and:  a) If Results Exist: populates table and adds one blank row following the last entry -or- b) If No Results: populates table with one blank row |
| 7. User double clicks table to enter edit mode | 8. The table becomes editable and a Save and Cancel button appear in the bottom right |
| 9. User: a) Populates the open fields and clicks Save -or- b) Selects Cancel -> returns to Step 0 | 10. A popup window confirms if the entry is to be recorded |
| 11. User: a) TUCEW User selects Yes -or- b) User selects No -> returns to Step 6 |  |

**Figure 28 Expanded Use Case for adding a student**

UC03: Assigning volunteers to school

|  |  |
| --- | --- |
| Actor: User | System: GEM |
|  | 0. GEM displays home page (unpopulated Volunteers tab) |
| 1. TUCBW User selects the Sites tab | 2. GEM displays an unpopulated Sites tab |
| 3. User selects desired school from drop down menu and presses the Search By School button | 4. A table appears in the blank area below and:  a) If Results Exist: populates table with all volunteers at that school and adds one blank row following the last entry -or- b) If No Results: populates table with one blank row |
| 5. User double clicks table to enter edit mode | 6. The table becomes editable and a Save and Cancel button appear in the bottom right |
| 7. User: a) Populates the open fields and clicks Save -or- b) Selects Cancel -> returns to Step 0 | 8. A popup window confirms if the entry is to be recorded |
| 9. User: a) TUCEW User selects Yes -or- b) User selects No -> returns to Step 4 |  |

**Figure 29 Expanded Use Case for assigning volunteer to school**

UC04: Assigning volunteer to student

|  |  |
| --- | --- |
| Actor: User | System: GEM |
|  | 0. GEM displays home page (unpopulated Volunteers tab) |
| 1. TUCBW User selects the Students tab | 2. GEM displays an unpopulated Students tab |
| 3. User selects the desired volunteer from the Volunteer Name dropdown menu and presses Search By Name | 4. A table appears in the blank area below and:  a) If Results Exist: populates table with all students paired to that volunteer and adds one blank row following the last entry -or- b) If No Results: populates table with one blank row |
| 5. User double clicks table to enter edit mode | 6. The table becomes editable and a Save and Cancel button appear in the bottom right |
| 7. User: a) Populates the open fields and clicks Save -or- b) Selects Cancel -> returns to Step 0 | 8. A popup window confirms if the entry is to be recorded |
| 9. User: a) TUCEW User selects Yes -or- b) User selects No -> returns to Step 4 |  |

**Figure 30 Expanded Use Case for assigning volunteer to student**

UC05???

UC06: Viewing reports from previous grant years

|  |  |
| --- | --- |
| Actor: User | System: GEM |
|  | 0. GEM displays home page (unpopulated Volunteers tab) |
| 1. TUCBW User selects Reporting Tab | 2. GEM displays the Reporting screen |
| 3. User selects View Grant Report button | 4. GEM displays the Reports for the current Grant Year |
| 5. The User: a) Wanted the current Year -> Skip to Step 7 -or- b) Double Clicks the Date Range header (is a drop down menu) and selects the desired year | 6. GEM updates the reports for the selected Grant Year |
| 7. TUCEW The User viewing financial reports from the selected Grant Year |  |

**Figure 32 Expanded Use Case for viewing reports from previous fiscal years**

UC07: Viewing student profile

|  |  |
| --- | --- |
| Actor: User | System: GEM |
|  | 0. GEM displays home page (unpopulated Volunteers tab) |
| 1. TUCBW User selects Students tab | 2. GEM displays the Student page |
| 3. User enters drop down menu under Volunteer Name: a) If associated volunteer is known -> select volunteers name and presses the Search By Name button -or- b) If unknown -> selects -SEARCH ALL- option and presses the Search By Name button | 4. A table appears in the blank area below and:  a) If Results Exist: populates table with all students -or- b) If No Results: populates table with one blank row |
| 5. User \*single\* clicks on the desired student in the table | 6. The row containing that student's info is highlighted |
| 7. User presses Select Student button (bottom right corner) | 8. The selected student's profile is displayed in a popup window |
| 9. TUCEW The User viewing the Student Profile and closing the popup window |  |

**Figure 33 Expanded Use Case for viewing a student's profile**

UC08: Viewing a volunteer’s profile

|  |  |
| --- | --- |
| Actor: User | System: GEM |
|  | 0. GEM displays home page (unpopulated Volunteers tab) |
| 1. TUCBW User selects the drop down box under volunteer name | 2. The list drops down, populated with all users |
| 3. User selects the desired volunteer name and presses the Search By Name button | 4. The page populates with the selected volunteer's information |
| 5. TUCEW The User viewing the Volunteer's profile |  |

**Figure 34 Expanded Use Case for viewing a volunteer’s profile**

UC09: Generating a volunteer report to Excel

|  |  |
| --- | --- |
| Actor: User | System: GEM |
|  | 0. GEM displays home page (unpopulated Volunteers tab) |
| 1. TUCBW The User presses the View All Active Volunteers button | 2. GEM displays a table of all volunteers classified as "current" and their corresponding contact information |
| 3. User double clicks the table to enter Edit Mode | 4. At the top right of the screen an export button becomes visible |
| 5. User presses export | 6. A popup window appears and asks what to name the file being created |
| 7. The User: a) Names the document and clicks ok -or- b) Closes the popup window -> Returns to Step 2 | 8. The Excel file downloads |
| 9. TUCEW The user opens the downloaded file |  |

**Figure 35 Expanded Use Case for generating volunteer reports**

UC10: Generating activity log report to Excel

|  |  |
| --- | --- |
| Actor: User | System: GEM |
|  | 0. GEM displays home page (unpopulated Volunteers tab) |
| 1. TUCBW The User presses the View All Active Volunteers button | 2. GEM displays a table of all volunteers classified as "current" and their corresponding contact information |
| 3. User presses the View Activity Log button | 4. GEM displays the main unpopulated Activity Log page |
| 5. The User: a) Searches by Volunteer Name -> Selects desired volunteer from the Volunteer Name drop down box and presses the Search by Name button -or- b) Searches by Site Location -> Selects desired school from the Site Location drop down box and presses Search by Site button -or- c) Searches by Date Range -> Selects all reports in the date range entered in the Date Range drop down box and presses Search by Date button -or- d) Searches All -> Presses View All Volunteers button -or- e) Wants to return to page -> Selects Volunteers tab, Return to Step 0. | 6. GEM populates the table based on search criteria |
| 7. User double clicks table to enter edit mode | 8. The table becomes editable and a Save and Cancel button appear in the bottom right |
| 9. User: a) Populates the open fields and clicks Save -or- b) Selects Cancel -> returns to Step 0 | 10. A popup window confirms if the entry is to be recorded |
| 11. User: a) TUCEW User selects Yes -or- b) User selects No -> returns to Step 6 |  |

**Figure 36 Expanded Use Case for generating the activity log report**

UC11: Generating demographic report to Excel

|  |  |
| --- | --- |
| Actor: User | System: GEM |
|  | 0. GEM displays home page (unpopulated Volunteers tab) |
| 1. TUCBW User presses the View All Volunteers button | 2. GEM displays a table of all the Volunteers who have ever been entered into the system |
| 3. User double clicks the table to enter Edit Mode | 4. At the top right of the screen an export button becomes visible |
| 5. User presses export | 6. A popup window appears and asks what to name the file being created |
| 7. The User: a) Names the document and clicks ok -or- b) Closes the popup window -> Returns to Step 2 | 8. The Excel file downloads |
| 9. TUCEW The user opens the downloaded file |  |

**Figure 37 Expanded Use Case for generating the demographic report**

UC12: Generating stipend and PTO report to Excel

|  |  |
| --- | --- |
| Actor: User | System: GEM |
|  | 0. GEM displays home page (unpopulated Volunteers tab) |
| 1. TUCBW User selects the Tracking tab | 2. GEM displays the main budget page |
| 3. User double clicks on the table at the bottom of the page to enter Edit Mode | 4. The table opens in a new window. An export button appears at the top right. |
| 5. User presses export | 6. A popup window appears and asks what to name the file being created |
| 7. The User: a) Names the document and clicks ok -or- b) Closes the popup window -> Returns to Step 4 | 8. The Excel file downloads |
| 9. TUCEW The user opens the downloaded file |  |

**Figure 38 Expanded Use Case for generating stipends and pto**

UC13: Generating meal expenditure report to Excel

|  |  |
| --- | --- |
| Actor: User | System: GEM |
|  | 0. GEM displays home page (unpopulated Volunteers tab) |
| 1. TUCBW User selects the Reporting tab | 2. GEM displays the main reporting page |
| 3. The User: a) Presses the View Grant Report button -or- b) Presses the View Fiscal Report button | 4. GEM loads the Current Date Range in the Selected Reporting Style |
| 5. User double clicks the table under Meal-In-Kind to enter Edit mode | 6. The table opens in a new screen. An export button appears in the top right corner. |
| 7. User presses export | 8. A popup window appears and asks what to name the file being created |
| 9. The User: a) Names the document and clicks ok -or- b) Closes the popup window -> Returns to Step 4 | 10. The Excel file downloads |
| 11. TUCEW The user opens the downloaded file |  |

**Figure 39 Expanded Use Case for generating the meal reports**

UC14: Generating milage report to Excel

|  |  |
| --- | --- |
| Actor: User | System: GEM |
|  | 0. GEM displays home page (unpopulated Volunteers tab) |
| 1. TUCBW User selects the Reporting tab | 2. GEM displays the main reporting page |
| 3. The User: a) Presses the View Grant Report button -or- b) Presses the View Fiscal Report button | 4. GEM loads the Current Date Range in the Selected Reporting Style |
| 5. User double clicks the table under Volunteer Mileage to enter Edit mode | 6. The table opens in a new screen. An export button appears in the top right corner. |
| 7. User presses export | 8. A popup window appears and asks what to name the file being created |
| 9. The User: a) Names the document and clicks ok -or- b) Closes the popup window -> Returns to Step 4 | 10. The Excel file downloads |
| 11. TUCEW The user opens the downloaded file |  |

**Figure 40 Expanded Use Case for generating the mileage reports**

UC15: Generating student report to Excel

|  |  |
| --- | --- |
| Actor: User | System: GEM |
|  | 0. GEM displays home page (unpopulated Volunteers tab) |
| 1. TUCBW User selects Students tab | 2. GEM displays the Student page |
| 3. User enters drop down menu under Volunteer Name: a) If the desired Report is for one volunteer -> Selects volunteer and presses the Search By Name button -or- b) If all results desired -> selects -SEARCH ALL- option and presses the Search By Name button | 4. A table appears in the blank area below and:  a) If Results Exist: populates table with all students -or- b) If No Results: populates table with one blank row |
| 5. User double clicks the table under Volunteer Mileage to enter Edit mode | 6. The table opens in a new screen. An export button appears in the top right corner. |
| 7. User presses export | 8. A popup window appears and asks what to name the file being created |
| 9. The User: a) Names the document and clicks ok -or- b) Closes the popup window -> Returns to Step 4 | 10. The Excel file downloads |
| 11. TUCEW The user opens the downloaded file |  |

**Figure 41 Expanded Use Case for generating the student reports**

UC16: Generating annual financial report to Excel

|  |  |
| --- | --- |
| Actor: User | System: GEM |
|  | 0. GEM displays home page (unpopulated Volunteers tab) |
| 1. TUCBW User presses the Reporting Tab at the top of the screen | 2. GEM displays reporting home page |
| 3. User double clicks the table under Annual Total Inputs to enter Edit Mode | 4. At the top right of the screen an export button becomes visible |
| 5. User presses export | 6. A popup window appears and asks what to name the file being created |
| 7. The User: a) Names the document and clicks ok -or- b) Closes the popup window -> Returns to Step 2 | 8. The Excel file downloads |
| 9. TUCEW The user opens the downloaded file |  |

**Figure 42 Expanded Use Case for generating annual reports**

UC18: Recording stipends

|  |  |
| --- | --- |
| Actor: User | System: GEM |
|  | 0. GEM displays home page (unpopulated Volunteers tab) |
| 1. TUCBW User selects the Tracking tab | 2. GEM displays the home budget page |
| 3. User locates the desired volunteer in the table and \*single\* clicks on the row containing their entry | 4. GEM highlights the selected row/individual's information in the table |
| 5. User enters the Stipend amount in the Current Stipends input box and presses the Save button | 6. A popup box appears asking to confirm the update on the table |
| 7. The User: a) Presses Yes -or- b) Presses No -> Returns to Step 4 | 8. The row containing the Volunteer's information is still highlighted, and the new totals are shown |
| 9. TUCEW User viewing updated totals |  |

**Figure 43 Expanded Use Case for recording stipends**

UC19: Changing timecard

|  |  |
| --- | --- |
| Actor: User | System: GEM |
|  | 0. GEM displays home page (unpopulated Volunteers tab) |
| 1. TUCBW User selects the Sites tab | 2. GEM displays an unpopulated Sites tab |
| 3. User enters drop down menu under School: a) If associated volunteer's site is known -> select School and press the Search By School button -or- b) If unknown -> Presses the View All button | 4. A table appears in the blank area below and:  a) If Results Exist: populates table with filtered volunteers -or- b) If No Results: populates table with one blank row |
| 5. User double clicks on the table to enter Edit Mode | 6. The table becomes editable. A Save and Cancel button appear in the top right corner. |
| 7. User edits the the Days and Hours column of the Row of the desired volunteer to edit their total calculated timecard. The User: a) Presses Save -or- b) Presses Cancel -> Return to step 4 | 8. The table saves and displays the new values |
| 9. TUCEW The User viewing the newly calculated timecard totals |  |

**Figure 45 Expanded Use Case for changing timecard**

UC21: Recording budgetary values

|  |  |
| --- | --- |
| Actor: User | System: GEM |
|  | 0. GEM displays home page (unpopulated Volunteers tab) |
| 1. TUCBW User selects the Tracking tab | 2. GEM displays the home budget page |
| 3. User locates the desired volunteer in the table and \*single\* clicks on the row containing their entry | 4. GEM highlights the selected row/individual's information in the table |
| 5. User double clicks the highlighted row | 6. GEM displays the Budget page belonging to that individual volunteer |
| 7. The User: a) Enters whatever needs to be added to the budget and presses Save -or- b) Presses Cancel -> Returns to Step 4 | 8. GEM returns to the home budget page. The selected volunteer's row is still highlighted, and shows the updated totals |
| 9. TUCEW User viewing updated budgetary totals |  |

**Figure 46 Expanded Use Case for recording budgetary values**

UC22: Entering volunteers’ mileage

|  |  |
| --- | --- |
| Actor: User | System: GEM |
|  | 0. GEM displays home page (unpopulated Volunteers tab) |
| 1. TUCBW User selects the Tracking tab | 2. GEM displays the home budget page |
| 3. User locates the desired volunteer in the table and \*single\* clicks on the row containing their entry | 4. GEM highlights the selected row/individual's information in the table |
| 5. User double clicks the highlighted row | 6. GEM displays the Budget page belonging to that individual volunteer |
| 7. The User: a) Enters the Bus Ride value in the Bus Rides input and presses the Save button -or- b) Enters the Mileage value in the Mileage input and presses the Save button -or- c) Presses Cancel -> Returns to Step 4 | 8. GEM returns to the home budget page. The selected volunteer's row is still highlighted, and shows the updated value in the Bus Rides or Mileage column |
| 9. TUCEW User viewing updated Mileage/Bus totals |  |

**Figure 47 Expanded Use Case for recording volunteer mileage**

UC23: Entering volunteers’ meal expenditure

|  |  |
| --- | --- |
| Actor: User | System: GEM |
|  | 0. GEM displays home page (unpopulated Volunteers tab) |
| 1. TUCBW User selects the Tracking tab | 2. GEM displays the home budget page |
| 3. User locates the desired volunteer in the table and \*single\* clicks on the row containing their entry | 4. GEM highlights the selected row/individual's information in the table |
| 5. User double clicks the highlighted row | 6. GEM displays the Budget page belonging to that individual volunteer |
| 7. The User: a) Enters the Meals value in the Site Meals Input box and presses the Save button -or- b) Enters an updated Meal Rate in the Total Meal Value (Rate) input box and presses the Save button -or- c) Presses Cancel -> Returns to Step 4 | 8. GEM returns to the home budget page. The selected volunteer's row is still highlighted, and shows the updated value in the Site Meals or Total Meal Value column |
| 9. TUCEW User viewing updated Meal Cost totals |  |

**Figure 48 Expanded Use Case for recording volunteer meals**

UC24: Retrieving timecard information

|  |  |
| --- | --- |
| Actor: User | System: GEM |
|  | 0. GEM displays home page |
| 1. TUCBW User selects the Volunteers tab | 2. The Volunteers page loads |
| 3. User selects the Financials tab from the top of the page | 4. GEM loads the Financials page |
| 5. User Selects the desired User from the drop down box | 6. GEM populates the selected user's information |
| 7. TUCEW User viewing a Volunteers accrued hours and YTD  hours |  |

**Figure 73 Expanded Use Case for retrieving timecard information**

UC25: Mass entering of volunteers

|  |  |
| --- | --- |
| Actor: User | System: GEM |
|  | 0. GEM displays home page (unpopulated Volunteers tab) |
| 1. TUCBW The User presses the View All Active Volunteers button | 2. GEM displays a table of all volunteers classified as "current" and their corresponding contact information |
| 3. User double clicks the table to enter Edit Mode | 4. At the top right of the screen a Save and Cancel button become visible |
| 5. User enters the contact information of new volunteer in blank row and hits enter on their keyboard | 6. GEM adds the entered row to the table, along with a new blank row below it |
| 7. The User: a) Wants to do another entry -> Return to step 5. -or- b) Presses Save -or- c) Presses Cancel | 8. A popup window asks to confirm whether to Save(or cancel) your changes |
| 9. TUCEW The User views changes made to the table |  |

**Figure 49 Expanded Use Case for mass entry of volunteers**

UC26: Mass entering of students

|  |  |
| --- | --- |
| Actor: User | System: GEM |
|  | 0. GEM displays home page (unpopulated Volunteers tab) |
| 1. TUCBW User selects Students tab | 2. GEM displays unpopulated Students page |
| 3. User selects drop down box under Volunteer Name | 4. Box populates list with all active volunteers. At the bottom of the list exists an -unassigned- option |
| 5. The User: a) Selects the Volunteer to add students to in the Volunteer Name drop down menu -or- b) Selects the -Unassigned- option at the bottom | 6. A table appears in the blank area below and:  a) If Results Exist: populates table and adds one blank row following the last entry -or- b) If No Results: populates table with one blank row |
| 7. User double clicks table to enter edit mode | 8. The table becomes editable and a Save and Cancel button appear in the bottom right |
| 9. User populates the fields in blank row and hits enter on their keyboard | 10. GEM adds the entered row to the table, along with a new blank row below it |
| 11. The User: a) Wants to do another entry -> Return to step 5. -or- b) Presses Save -or- c) Presses Cancel | 12. A popup window asks to confirm whether to Save(or cancel) your changes |
| 13. TUCEW The User views changes made to the table |  |

**Figure 50 Expanded Use Case for mass entry of students**

UC27: Entering school cost share

|  |  |
| --- | --- |
| Actor: User | System: GEM |
|  | 0. GEM displays home page (unpopulated Volunteers tab) |
| 1. TUCBW User selects the reporting tab | 2. GEM displays the reporting screen |
| 3. User fills out the input boxes next to the desired billing | 4. GEM displays the inputted values |
| 5. The User: a) TUCEW User presses save -or- b) User presses cancel -> Return to step 2. |  |

**Figure 51 Expanded Use Case for entry of school cost share**

UC28: Entering meal/mileage rates

|  |  |
| --- | --- |
| Actor: User | System: GEM |
|  | 0. GEM displays home page (unpopulated Volunteers tab) |
| 1. TUCBW User selects the tracking tab | 2. GEM displays the budget screen |
| 3. User enters the new desired value in the Meal Cost or Mileage Cost input boxes | 4. GEM displays the inputted values in their boxes |
| 5. The User: a) TUCEW the user presses Save -or- b) User presses Cancel -> Return to step 2 |  |

**Figure 52 Expanded Use Case for entry of meal and mileage rates**

UC29: Entering stipend/PTO rates

|  |  |
| --- | --- |
| Actor: User | System: GEM |
|  | 0. GEM displays home page (unpopulated Volunteers tab) |
| 1. TUCBW User selects the tracking tab | 2. GEM displays the budget screen |
| 3. User enters the new desired value in the Meal Cost or Mileage Cost input boxes | 4. GEM displays the inputted values in their boxes |
| 5. The User: a) TUCEW the user presses Save -or- b) User presses Cancel -> Return to step 2 |  |

**Figure 53 Expanded Use Case for entry of stipend and pto rates**

UC30: Viewing reports from previous fiscal years

|  |  |
| --- | --- |
| Actor: User | System: GEM |
|  | 0. GEM displays home page (unpopulated Volunteers tab) |
| 1. TUCBW User selects Reporting Tab | 2. GEM displays the Reporting screen |
| 3. User selects View Fiscal Report button | 4. GEM displays the Reports for the current Fiscal Year |
| 5. The User: a) Wanted the current Year -> Skip to Step 7 -or- b) Double Clicks the Date Range header (is a drop down menu) and selects the desired year | 6. GEM updates the reports for the selected Fiscal Year |
| 7. TUCEW The User viewing financial reports from the selected Fiscal Year |  |

**Figure 32 Expanded Use Case for viewing reports from previous fiscal years**

UC31: Recording PTO

|  |  |
| --- | --- |
| Actor: User | System: GEM |
|  | 0. GEM displays home page (unpopulated Volunteers tab) |
| 1. TUCBW User selects the Tracking tab | 2. GEM displays the home budget page |
| 3. User locates the desired volunteer in the table and \*single\* clicks on the row containing their entry | 4. GEM highlights the selected row/individual's information in the table |
| 5. User enters the PTO amount used in the PTO input box and presses the Save button | 6. A popup box appears asking to confirm the update on the table |
| 7. The User: a) Presses Yes -or- b) Presses No -> Returns to Step 4 | 8. The row containing the Volunteer's information is still highlighted, and the new totals are shown |
| 9. TUCEW User viewing updated totals |  |

**Figure 44 Expanded Use Case for recording pto**

UC32: Updating volunteer information

|  |  |
| --- | --- |
| Actor: User | System: GEM |
|  | 0. GEM displays home page (unpopulated Volunteers tab) |
| 1. TUCBW User selects desired volunteer from drop down menu | 2. GEM displays the selected volunteer's name in the box |
| 3. User presses the Search By Name button | 4. GEM populates the Volunteers page with the selected volunteer's information |
| 5. User presses the View/Edit Forms button | 6. Volunteer information on the left side of the screen becomes editable. The View/Edit Forms button is replaced by a Save and Cancel button |
| 7. The User: a) TUCEW the user presses Save -or- b) User presses Cancel -> Return to step 4 |  |

**Figure 54 Expanded Use Case for updating volunteer information**

UC33: Adding schools

|  |  |
| --- | --- |
| Actor: User | System: GEM |
|  | 0. GEM displays home page (unpopulated Volunteers tab) |
| 1. User selects the sites tab at the top | 2. GEM populates with an empty sites tab |
| 3. User clicks into the School drop down menu and selects the -add new- option. | 4. The School information labels become editable on a blank screen |
| 5. User inputs data for the new school. The user: a) TUCEW user selects save -or- b) User selects cancel -> return to Step 2 |  |

**Figure 55 Expanded Use Case for adding a new school**

Still need to find UC 34-36 / Don’t know what consolidated ones mean

|  |  |
| --- | --- |
| Actor: User | System: GEM |
|  | 0. GEM displays home page (unpopulated Volunteers tab) |
| 1. TUCBW User selects desired volunteer from drop down menu | 2. GEM displays the selected volunteer's name in the box |
| 3. User presses the Search By Name button | 4. GEM populates the Volunteers page with the selected volunteer's information |
| 5. User presses the View/Edit Forms button | 6. Volunteer information on the left side of the screen becomes editable. The View/Edit Forms button is replaced by a Save and Cancel button |
| 7. The User changes the volunteer's status and: a) TUCEW the user presses Save -or- b) User presses Cancel -> Return to step 4 |  |

**Figure 56 Expanded Use Case for changing a volunteer’s status**

|  |  |
| --- | --- |
| Actor: Admin | System: GEM |
|  | 0. GEM displays home page (unpopulated Volunteers tab). Admins have 6 tabs across the top of the screen instead of the secondary user's 5 -> the last tab being "management". |
| 1. TUCBW Admin selects the management tab at the top | 2. GEM populates the admin tab with a list of all system users |
| 3. Admin clicks the add new user button in the top right corner | 4. GEM adds a new row with fields to populate with the new User's email and assigned password (new users will be prompted to change their password upon first login) |
| 5. User inputs data for the new school. The user: a) TUCEW user selects save -or- b) User selects cancel -> return to Step 2 |  |

**Figure 57 Expanded Use Case for the admin to add a new user**

|  |  |
| --- | --- |
| Actor: User | System: GEM |
|  | 0. GEM displays home page |
| 1. TUCBW User selects the Volunteers tab | 2. The Volunteers page loads |
| 3. User presses the Add New Volunteer button | 4. The fields normally populated with volunteer information  become editable to allow for the entry of a new Volunteer |
| 5. The User:   1. TUCEW User fills in the new volunteer's information and presses Save New Volunteer   -or-   1. User Presses the X to cancel entry -> Return to step 2. |  |

Consolidated UI Expanded Use Cases

**Figure 58 Expanded Use case for adding a volunteer**

|  |  |
| --- | --- |
| Actor: User | System: GEM |
|  | 0. GEM displays home page |
| 1. TUCBW User selects Volunteers tab | 2. GEM displays the main Volunteers page |
| 3. User selects the Child Assignment tab at the top of the  page | 4. GEM displays an unpopulated Child Assignment page |
| 1. User clicks on the drop down box titled Select Volunteer and:    1. Selects desired volunteer (if known)   -or-   * 1. Selects -unassigned- | 6. GEM will display the students belonging to the selected volunteer (or unassigned) |
| 7. User clicks the edit pen in the top right corner | 8. GEM displays the editable version of the page |
| 9. User:   1. Clicks the pen icon next to New Student   -or-   1. Selects the X in the top corner to cancel edit mode -> returns to Step 6. | 10. GEM allows the box containing the student's tracked information to become editable |
| 11. User:   1. TUCEW User fills out the appropriate boxes and presses Save   -or-   1. Selects the X in the top corner to cancel edit mode and   delete changes -> returns to Step 6. |  |

**Figure 59 Expanded Use Case for adding a student**

|  |  |
| --- | --- |
| Actor: User | System: GEM |
|  | 0. GEM displays home page |
| 1. TUCBW User selects the Volunteers tab | 2. GEM displays the main volunteers page |
| 3. User selects the desired volunteer from the drop down  menu | 4. GEM retrieves and displays that user's information |
| 5. User selects the Pen in the top right corner to enter Edit  Mode | 6. The selected volunteer's information becomes editable |
| 7. User:   1. TUCEW User changes or enters the desired site and presses save   -or-   1. User presses the X to cancel -> Return to step 4. |  |

**Figure 60 Expanded Use Case for assigning volunteer to school**

|  |  |
| --- | --- |
| Actor: User | System: GEM |
|  | 0. GEM displays home page |
| 1. TUCBW User selects Volunteers tab | 2. GEM displays the main Volunteers page |
| 3. User selects the Child Assignment tab at the top of the  page | 4. GEM displays an unpopulated Child Assignment page |
| 5. User clicks on the drop down box titled Select Volunteer  and selects desired volunteer | 6. GEM will display the students belonging to the selected  volunteer |
| 7. User clicks the edit pen in the top right corner | 8. GEM displays the editable version of the page |
| 9. User:   1. Clicks the pen icon next to New Student   -or-   1. Selects the X in the top corner to cancel edit mode -> returns to Step 6. | 10. GEM allows the box containing the student's tracked information to become editable |
| 11. User:   1. TUCEW User fills out the appropriate boxes and presses Save   -or-   1. Selects the X in the top corner to cancel edit mode and delete changes -> returns to Step 6. |  |

**Figure 61 Expanded Use Case for assigning volunteer to student**

|  |  |
| --- | --- |
| Actor: User | System: GEM |
|  | 0. GEM displays home page |
| 1. TUCBW User selects Finance Tab | 2. GEM displays the main financial page |
| 3. User selects Grant Year tab at the top of the page | 4. GEM displays the report for the current grant year |
| 5. TUCEW the user selects the desired date range to view  from the drop down boxes |  |

**Figure 62 Expanded use case for viewing reports from grant years**

|  |  |
| --- | --- |
| Actor: User | System: GEM |
|  | 0. GEM displays home page |
| 1. TUCBW User selects Finance Tab | 2. GEM displays the main financial page |
| 3. User selects Fiscal Year tab at the top of the page | 4. GEM displays the report for the current fiscal year |
| 5. TUCEW User selects the desired date range to view from  the drop down boxes |  |

**Figure 63 Expanded Use Case for Viewing reports from fiscal years**

|  |  |
| --- | --- |
| Actor: User | System: GEM |
|  | 0. GEM displays home page |
| 1. TUCBW User selects Volunteers tab | 2. GEM displays the main Volunteers page |
| 3. User selects the Child Assignment tab at the top of the  page | 4. GEM displays an unpopulated Child Assignment page |
| 5. User clicks on the drop down box titled Select Volunteer  and selects the volunteer the desired student belongs to | 6. GEM will display the students belonging to the selected  volunteer along with their associated information |
| 7. TUCEW User clicks on row containing the desired student  to highlight it |  |
|  |  |

**Figure 64 Expanded use Case for viewing student profile**

|  |  |
| --- | --- |
| Actor: User | System: GEM |
|  | 0. GEM displays home page |
| 1. TUCBW User selects the Volunteers tab | 2. The Volunteers page loads |
| 3. User selects the desired volunteer from the drop down  menu | 4. The volunteer's information loads |
| 5. TUCEW User selects the demographics tab at the top |  |

**Figure 65 Expanded Use Case for viewing volunteer profile**

|  |  |
| --- | --- |
| Actor: User | System: GEM |
|  | 0. GEM displays home page |
| 1. TUCBW User selects Reports tab | 2. GEM displays the report builder |
| 3. User selects the desired volunteer(s) to build the report on and the desired date range | 4. GEM displays a check mark next to all selected volunteers and the entry boxes for date range contain the entered  values |
| 5. User clicks the files corresponding to the desired information (middle box) and moves them to the rightmost  box using the right arrow button (->). If something is mistakenly transferred to the report it can be undone by selecting the information from the rightmost box and  pressing the left arrow button (<-) to remove it. | 6. GEM displays the selected/moved information in the rightmost box |
| 7. TUCEW User presses Generate Report and GEM exports  the selected data to Excel |  |

**Figure 66 Expanded Use Case for generating reports**

|  |  |
| --- | --- |
| Actor: User | System: GEM |
|  | 0. GEM displays home page |
| 1. TUCBW User selects the Volunteers tab | 2. The Volunteers page loads |
| 3. User selects the Activity Log tab from the top of the page | 4. GEM loads the Activity log page |
| 5. User Selects the desired volunteer and date range from  their corresponding drop down boxes | 6. GEM populates table with the corresponding activity log  entries |
| 7. User selects pen in top corner to enter Edit Mode | 8. GEM displays the editable Activity Log table |
| 9. User:   1. TUCEW user selects the + and adds the entry in the new row   -or-   1. User presses the X to cancel -> Return to step 6. |  |

**Figure 67 Expanded use Case for adding to the activity log**

|  |  |
| --- | --- |
| Actor: User | System: GEM |
|  | 0. GEM displays home page |
| 1. TUCBW User selects the Volunteers tab | 2. The Volunteers page loads |
| 3. User selects the Financials tab from the top of the page | 4. GEM loads the Financials page |
| 5. User selects the desired volunteer to add to | 6. GEM displays the finances related to that volunteer |
| 7. User Selects the pen to enter Edit Mode | 8. GEM allows the fields on the page to become editable |
| 7. User:   1. TUCEW user Adjust the desired field (PTO, Meals, Mileage, etc.) and presses save   -or-   1. User presses the X to cancel -> Return to step 6. |  |

**Figure 68 Expanded Use Case for editing a volunteer’s information**

|  |  |
| --- | --- |
| Actor: User | System: GEM |
|  | 0. GEM displays home page |
| 1. TUCBW User selects the Volunteers tab | 2. The Volunteers page loads |
| 3. User selects the Financials tab from the top of the page | 4. GEM loads the Financials page |
| 5. User selects the desired volunteer to add to | 6. GEM displays the finances related to that volunteer |
| 7. User Selects the pen to enter Edit Mode | 8. GEM allows the fields on the page to become editable |
| 7. User:   1. TUCEW user adjusts the Regular Hours column to adjust that volunteer's timecard and presses save   -or-   1. user presses X to cancel -> return to step 6. |  |

**Figure 69 Expanded Use case for changing timecard**

|  |  |
| --- | --- |
| Actor: User | System: GEM |
|  | 0. GEM displays home page |
| 1. TUCBW User selects the Volunteers tab | 2. The Volunteers page loads |
| 3. User selects the Financials tab from the top of the page | 4. GEM loads the Financials page |
| 5. User Selects the pen to enter Edit Mode | 6. GEM allows the fields on the page to become editable |
| 7. User:   1. TUCEW user enters the new desired rates   -or-   1. User presses the X to cancel -> Return to step 4 |  |

**Figure 70 Expanded Use Case for changing meal, stipend, and PTO rates**

|  |  |
| --- | --- |
| Actor: User | System: GEM |
|  | 0. GEM displays home page |
| 1. TUCBW User selects the Schools tab | 2. The Schools page loads populated with a list of all schools  that have ever been associated with the program |
| 3. User clicks on the school they desire to generate a report  on | 4. GEM loads that school and the corresponding  volunteers/information |
| 5. User clicks the print button in the top right corner | 6. GEM displays a prompt asking if the current page is to be  printed, or the corresponding site form |
| 7. TUCEW User selects Get Site Form option |  |

**Figure 71 Expanded Use Case for generating monthly site form**

|  |  |
| --- | --- |
| Actor: User | System: GEM |
|  | 0. GEM displays home page |
| 1. TUCBW User selects the Finance Tab | 2. GEM displays the general finance page |
| 3. User selects the pen in the top right corner to enter Edit  Mode | 4. GEM displays a blank, editable row added to the table |
| 5. User:   1. TUCEW User fills in the new expense and presses save   -or-   1. User presses the X to cancel -> Return to step 2. |  |

**Figure 72 Expanded Use Case for recording budgetary values**

|  |  |
| --- | --- |
| Actor: User | System: GEM |
|  | 0. GEM displays home page |
| 1. TUCBW User selects the Finance Tab | 2. GEM displays the general finance page |
| 3. User selects the pen in the top right corner to enter Edit  Mode | 4. GEM allows the top table - School Cost Share - to become  editable |
| 5. User:   1. TUCEW User fills in the new cost value at the designated billing   -or-   1. User presses the X to cancel -> Return to step 2. |  |

**Figure 74 Expanded Use Case for filling out school cost share**

UC32: Updating volunteer information

|  |  |
| --- | --- |
| Actor: User | System: GEM |
|  | 0. GEM displays home page |
| 1. TUCBW User selects the Volunteers tab | 2. The Volunteers page loads |
| 3. User selects the desired volunteer from the drop down list | 4. The fields populate with information belonging to the  selected volunteer |
| 5. The User clicks on the pen in the top right corner to enter  Edit Mode | 6. GEM allows the displayed fields to become editable |
| 7. The User:   1. TUCEW User changes the desired information and presses Save   -or-   1. User Presses the X to cancel entry -> Return to step 4. |  |

**Figure 75 Expanded Use Case for updating volunteer information**

UC33: Adding schools

|  |  |
| --- | --- |
| Actor: User | System: GEM |
|  | 0. GEM displays home page |
| 1. TUCBW User selects the Schools tab | 2. GEM loads the main schools page, populated with a list of  all schools ever involved in the program |
| 3. User selects the + in the top right corner | 4. GEM displays the Enter New School page - an unpopulated,  editable Schools page |
| 5. User:   1. TUCEW User inputs the desired information for the new school   -or-   1. User presses the X to cancel -> Return to step 2. |  |

**Figure 76 Expanded Use Case for adding schools**

UC34: Deleting students

|  |  |
| --- | --- |
| Actor: User | System: GEM |
|  | 0. GEM displays home page |
| 1. TUCBW User selects Volunteers tab | 2. GEM displays the main Volunteers page |
| 3. User selects the Child Assignment tab at the top of the  page | 4. GEM displays an unpopulated Child Assignment page |
| 5. User clicks on the drop down box titled Select Volunteer  and selects desired volunteer | 6. GEM will display the students belonging to the selected  volunteer |
| 7. User clicks the edit pen in the top right corner | 8. GEM displays the editable version of the page |
| 9. User:   1. Clicks the X icon next to the desired Student   -or-   1. Selects the X in the top corner to cancel edit mode -> returns to Step 6. | 10. GEM launches a popup window to confirm deletion of that student |
| 11. User:   1. TUCEW User selects yes   -or-   1. Selects no -> returns to step 8 (but still populated with the   selected choices) |  |

**Figure 77 Expanded Use Case for deleting A Student**

UC35: Changing a volunteer’s status

|  |  |
| --- | --- |
| Actor: User | System: GEM |
|  | 0. GEM displays home page |
| 1. TUCBW User selects the Volunteers tab | 2. The Volunteers page loads |
| 3. User selects the desired volunteer from the drop down list | 4. The fields populate with information belonging to the  selected volunteer |
| 5. The User clicks on the pen in the top right corner to enter  Edit Mode | 6. GEM allows the displayed fields to become editable |
| 7. The User:   1. TUCEW User changes the status of the volunteer and presses Save   -or-   1. User Presses the X to cancel entry -> Return to step 4. |  |

**Figure 78 Expanded Use Case for changing a volunteer's status**

UC36: Add new user

|  |  |
| --- | --- |
| Actor: Admin | System: GEM |
|  | 0. GEM displays home page |
| 1. TUCBW An Administrative account selects Users tab | 2. The Users page loads populated with all users in the system |
| 3. Admin presses the + button to add a new user | 4. Add new user page loads with fields for name, email,  password, and phone number. |
| 5. Admin:   1. TUCEW Admin Presses the Save icon in the top right corner (replaces the pen icon)   -or-   1. Admin Presses the X to Cancel -> Return to step 2. |  |

**Figure 79 Expanded Use Case for adding a new user**

## Requirements to Use Case Matrix

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** | **19** | **20** | **21** | **22** | **23** | **24** | **25** | **26** | **27** | **28** | **29** | **30** | **31** | **32** | **33** | **34** | **35** | **36** |
| **1.0** | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x |
| **2.0** | x |  | x | x |  |  |  | x | x | x | x | x |  | x |  |  | x | x | x |  |  | x | x | x | x |  |  |  |  |  | x | x |  | x | x |  |
| **3.0** |  | x |  | x |  |  | x |  |  |  |  |  |  |  | x |  |  |  |  |  |  |  |  |  |  | x |  |  |  |  |  |  |  | x |  |  |
| **4.0** |  | x | x | x |  |  | x |  |  |  |  |  |  |  |  |  |  |  | x |  |  |  |  |  |  | x | x |  |  |  |  |  | x |  |  |  |
| **5.0** |  |  |  |  |  | x |  |  | x | x |  | x | x | x |  | x |  | x |  | x | x | x | x | x |  |  | x | x | x | x | x |  |  |  |  |  |
| **6.0** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **7.0** |  |  |  |  | x |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | x |
| **8.0** |  | x | x | x |  |  | x |  |  |  |  |  |  |  | x |  |  |  | x |  |  |  |  |  |  | x |  |  |  |  |  |  |  |  |  |  |
| **9.0** | x |  | x | x |  |  |  | x | x | x | x |  |  |  |  |  | x | x |  |  |  | x | x |  | x |  |  |  |  |  | x | x |  | x | x |  |
| **10.0** | x |  |  |  |  |  |  | x | x | x | x |  |  |  |  |  | x |  |  |  |  |  |  |  |  |  |  |  |  |  |  | x |  |  | x |  |
| **11.0** | x | x | x | x |  | x |  |  | x | x | x | x | x | x | x | x |  |  |  |  |  |  |  |  |  |  |  |  |  | x | x |  |  | x | x |  |
| **12.0** | x | x | x | x |  | x |  |  | x | x | x | x | x | x | x | x |  |  |  |  |  |  |  |  |  |  |  |  |  | x | x |  |  | x | x |  |
| **13.0** | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x |

**Figure 80 Matrix to show the relations between the requirements and Use Cases**

X-axis: Use Cases

Y-axis: Requirements grouped by section

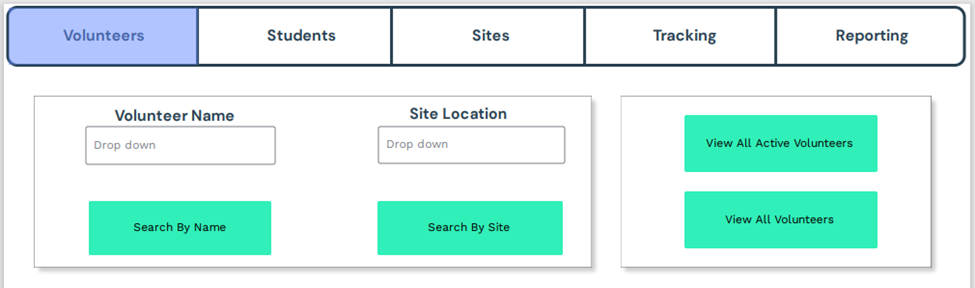
## Requirements to Iteration Matrix

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Requirements** | **Priority** | **Effort** | **Dependency** | **Iteration 1** | **Iteration 2** | **Iteration 3** | **Iteration 4** |
| **1.0** | 5 | 1 | None | 1 |  |  |  |
| **2.0** | 5 | 5 | None | 5 |  |  |  |
| **3.0** | 3 | 2 | 2 |  | 2 |  |  |
| **4.0** | 4 | 2 | 3 |  | 2 |  |  |
| **5.0** | 5 | 5 | 2 |  | 5 |  |  |
| **6.0** | 1 | 2 | None |  |  |  | 2 |
| **7.0** | 2 | 2 | None | 2 |  |  |  |
| **8.0** | 3 | 1 | 3,4 |  |  |  | 1 |
| **9.0** | 5 | 2 | 2 |  |  |  | 2 |
| **10.0** | 5 | 2 | 2,9 |  | 2 |  |  |
| **11.0** | 4 | 5 | 2,3,4,5 |  |  | 5 |  |
| **12.0** | 4 | 5 | 2,3,4,5 |  |  | 5 |  |
| **13.0** | 3 | 1 | None | 1 |  |  |  |
| **Total Effort** |  | **35** |  | **9** | **11** | **10** | **5** |

**Figure 81 Matrix to show which requirements will be implemented in each iteration of the development process**

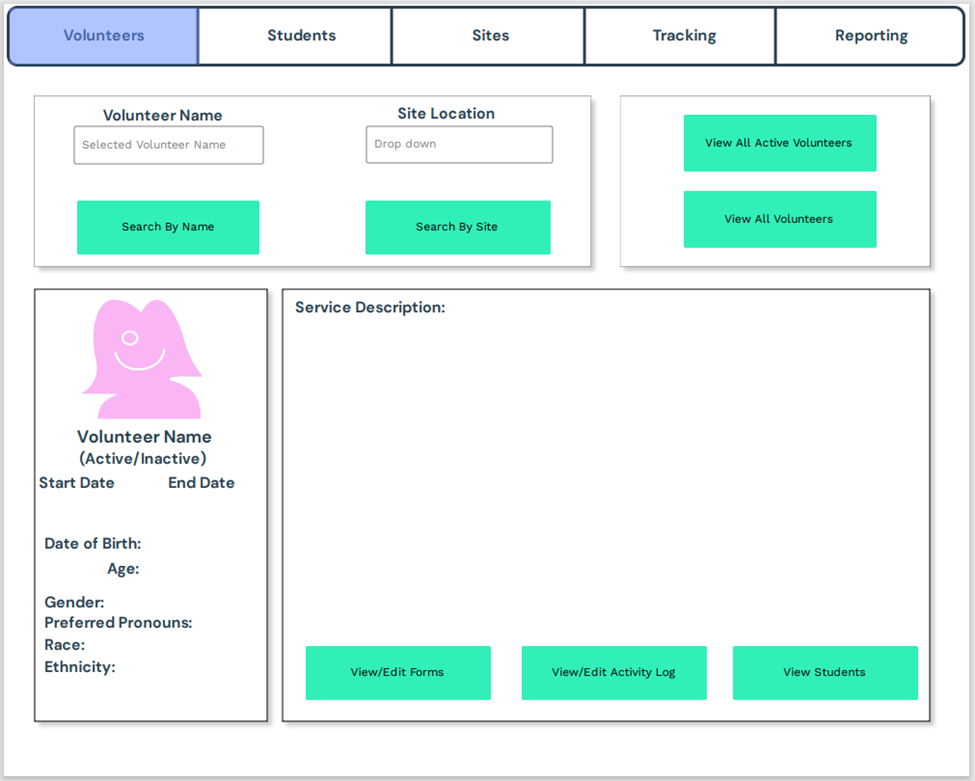
## UI/Storyboards

Group UI Proposal



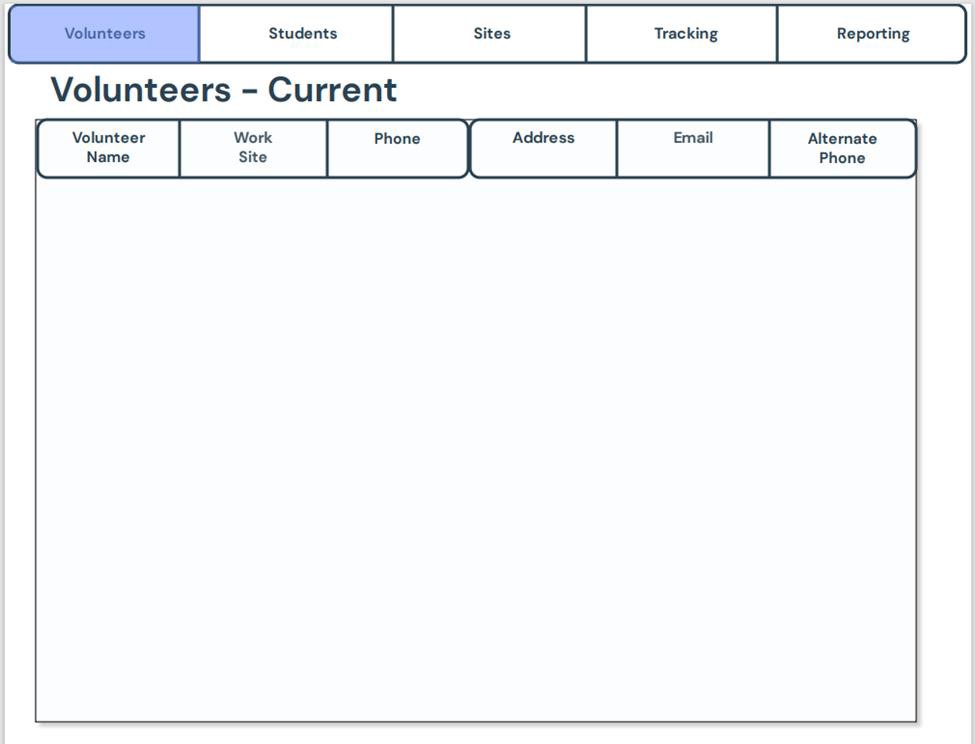
**Figure 82 Volunteers – Search/Individual Information**

The main/landing screen – an unpopulated Volunteers tab. From here individual volunteer information can be retrieved by name or site, or a list of all volunteers may be viewed.



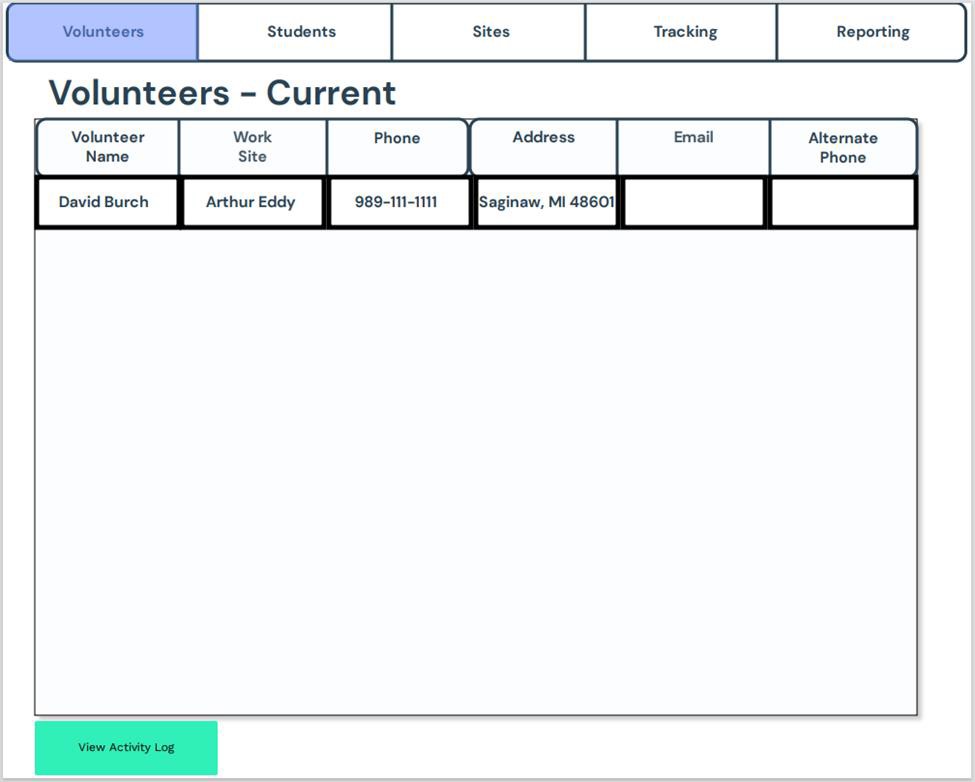
**Figure 83 Volunteers – Search/Individual Information (Populated)**

An example of a populated volunteers page. The volunteer’s information is listed on the left hand side, along with the picture of the volunteer. From this screen a user can edit and upload relevant forms, update that volunteer’s activity log, or see all students associated with that volunteer.



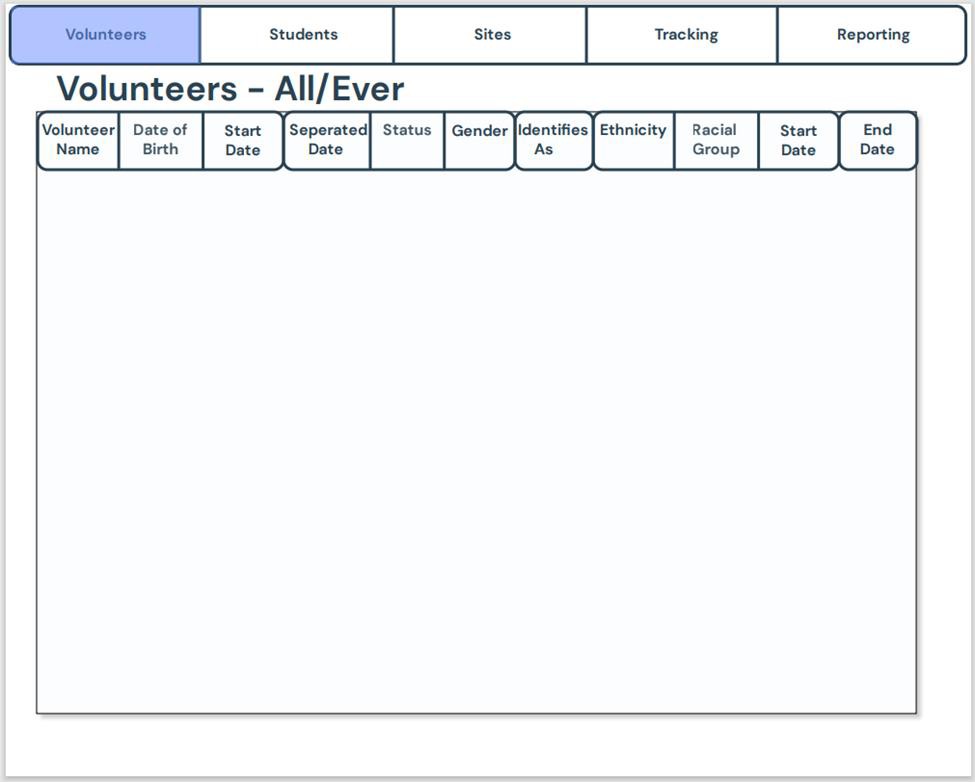
**Figure 84 Volunteers – List of Current**

A list of all current volunteers and their relevant contact information.



**Figure 85 Volunteers – List of Current (Populated)**

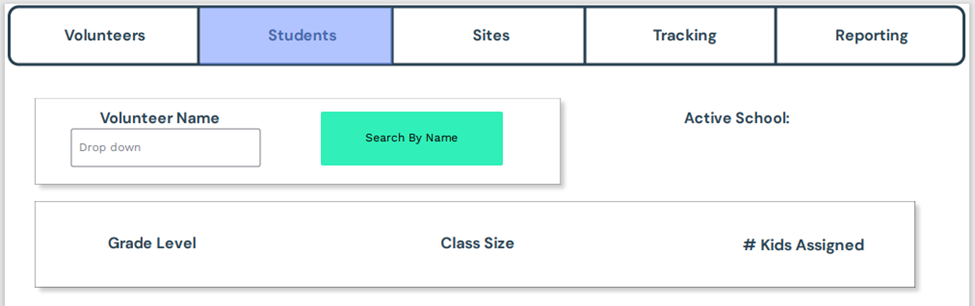
A populated example on the volunteers screen.



**Figure 86 Volunteers – List of All/Ever**

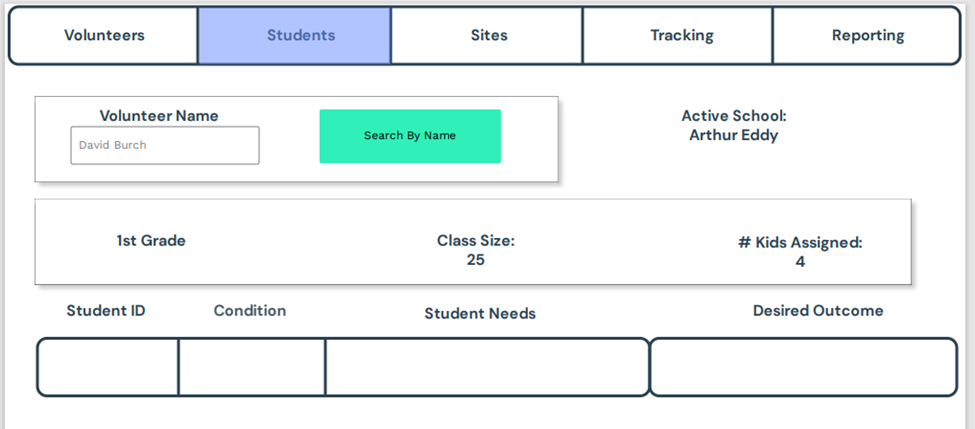
This would be populated similarly to the example shown in Current Volunteers – This list holds all fields of all volunteers whereas Current maintains a contact list of current volunteers. We can think of Current holding the information in Spreadsheet 2 Tab 1, and All/Ever holding the information in Spreadsheet 4 Tab 2.

Between these two tabs, we encompass all information on Spreadsheets 2 and 4 and can ignore the rest of the tabs in those sheets.



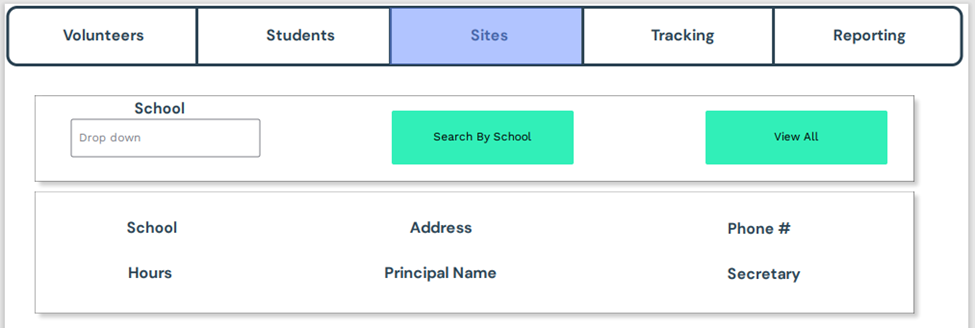
**Figure 87 Student – Populate per volunteer**

An unpopulated Students tab. From here we can search our list of volunteers and view all associated students, as well as details concerning class size and grade level.



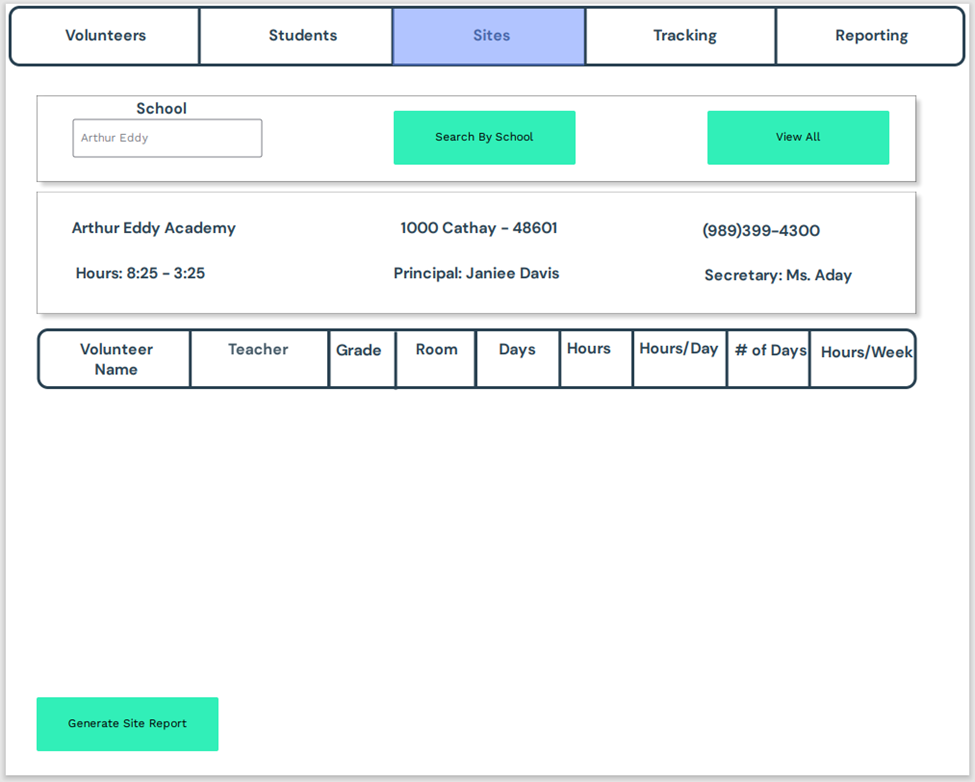
**Figure 88 Student – Populate per volunteer (Populated)**

An example of what a student’s tab would look like once a volunteer is selected.



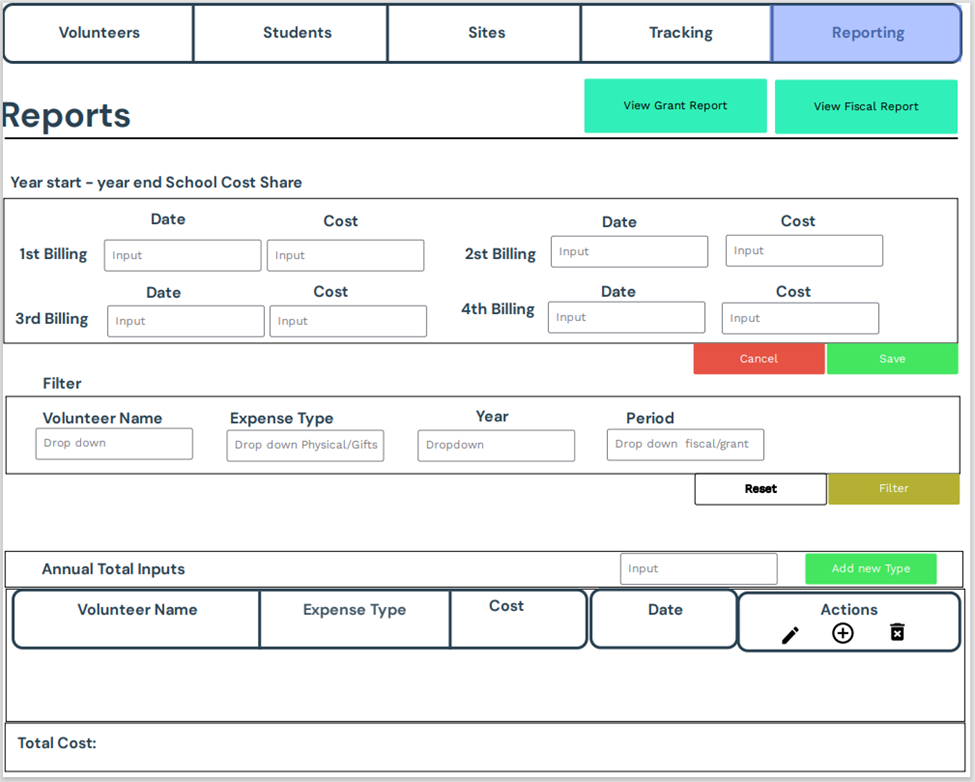
**Figure 89 Site – Populate List of Volunteers by location**

An unpopulated Sites tab. From here a list of all schools will be able to be displayed, or one may be selected to view its specific details.



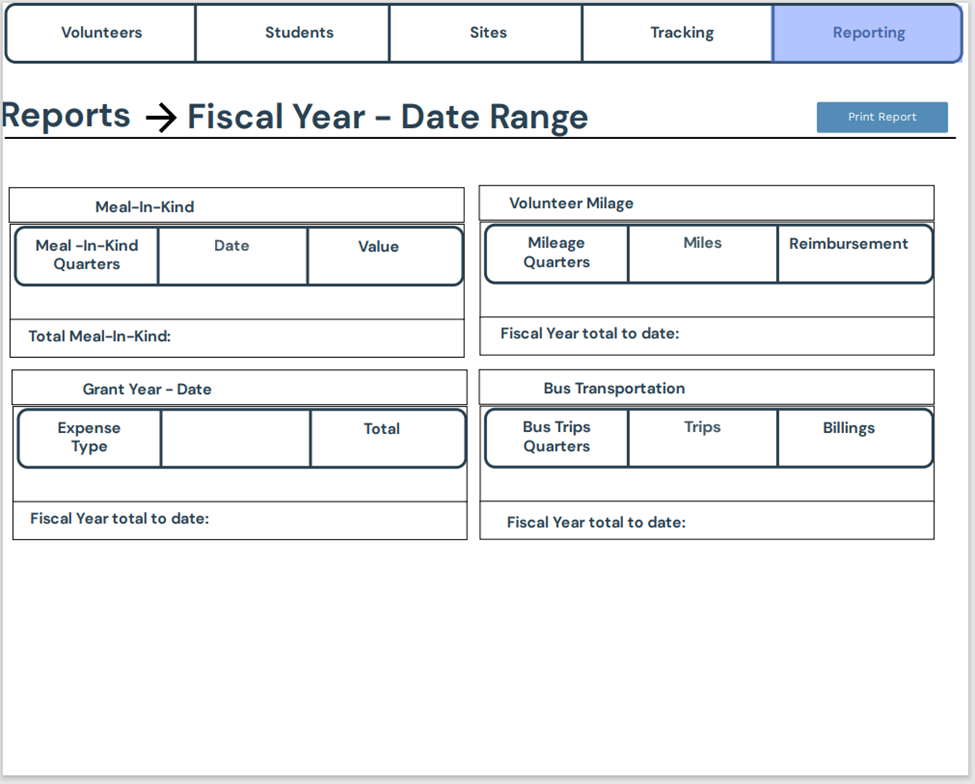
**Figure 90 Site – Populate List of Volunteers by location (Populated)**

An example of what a populated Sites tab may look like for the school Arthur Eddy Academy.



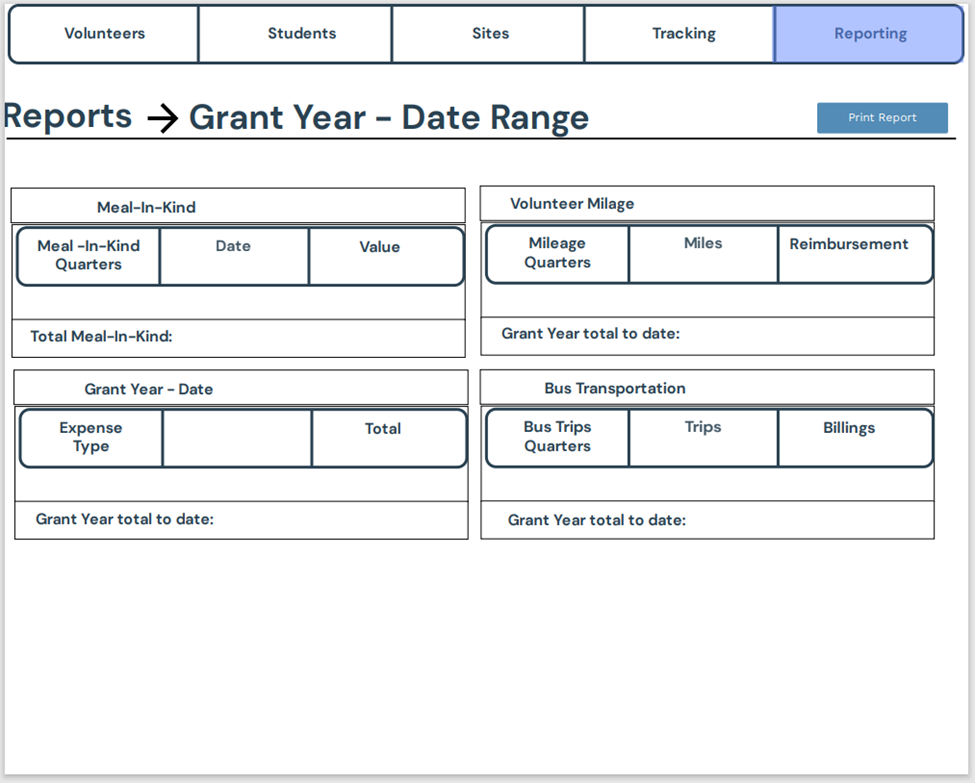
**Figure 91 Reporting – Main**

The main Reporting Screen. From here many budgetary values may be entered, and a table displays all current expenses. These expenses can be arranged and viewed by grant year and by fiscal year.



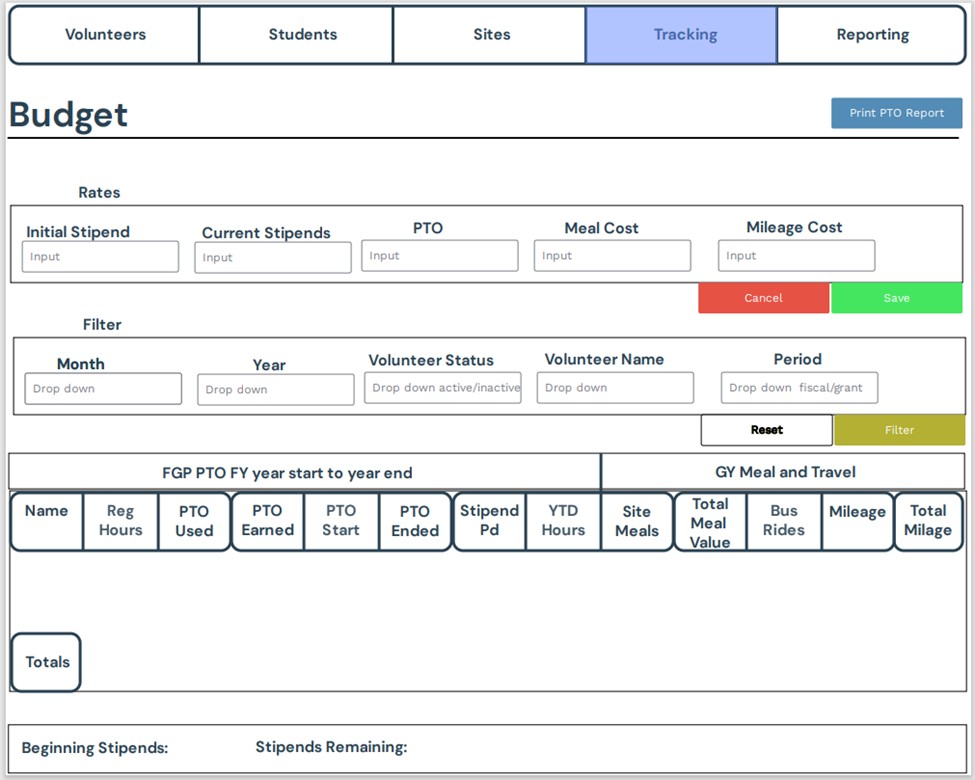
**Figure 92 Reporting – By Fiscal Year**

This is an example of viewing the reporting according to fiscal year.



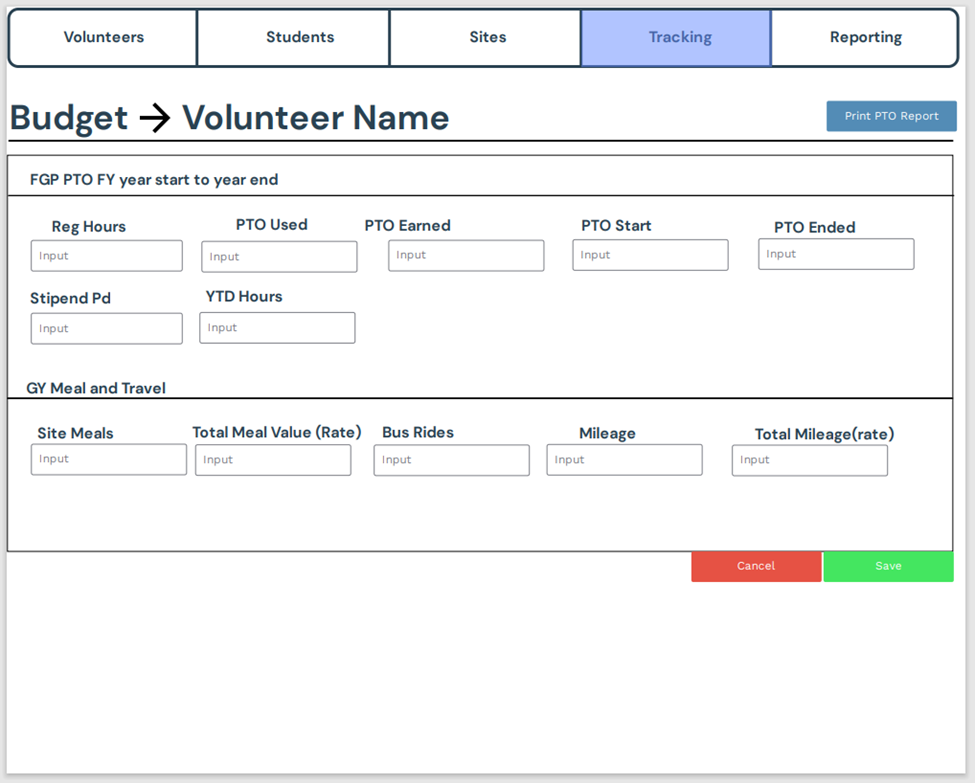
**Figure 93 Reporting – By Grant Year**

This is an example of viewing the reporting according to grant year.



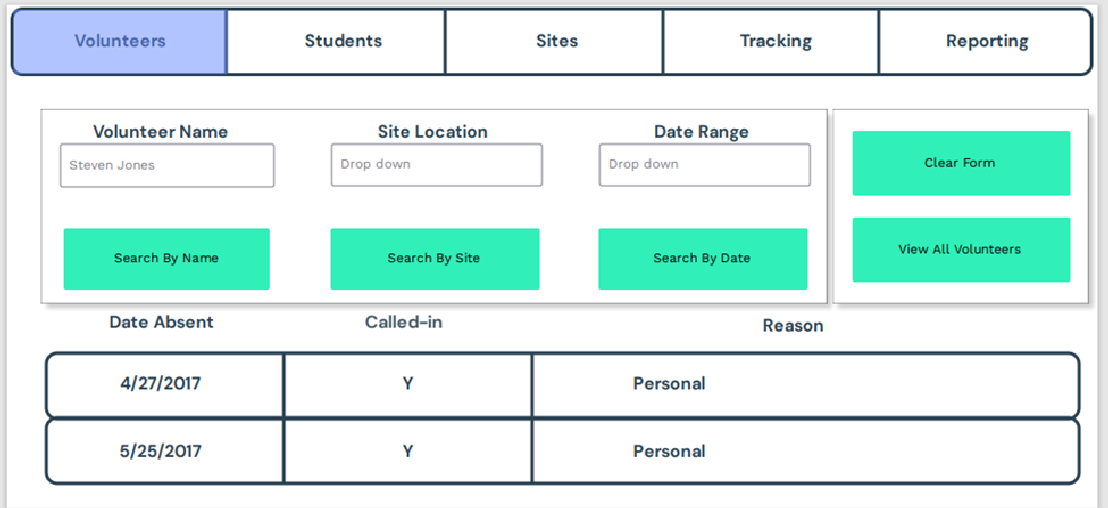
**Figure 94 Budget – Main**

The main Budget Tracking screen. Here rates may be adjusted and a list of all expenses is populated. The list can be filtered and searched for convenience.



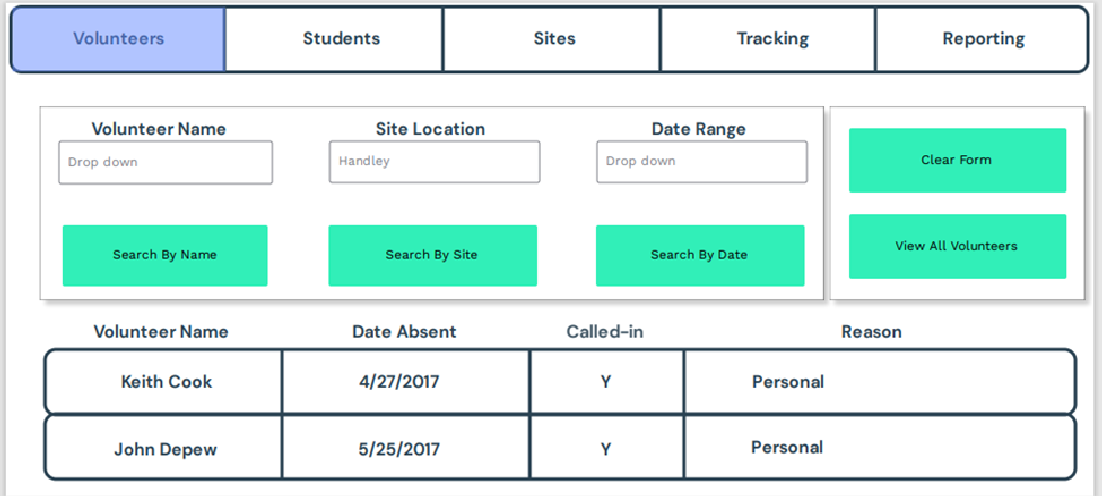
**Figure 95 Budget by Volunteer**

A budget screen to adjust values for an individual volunteer.



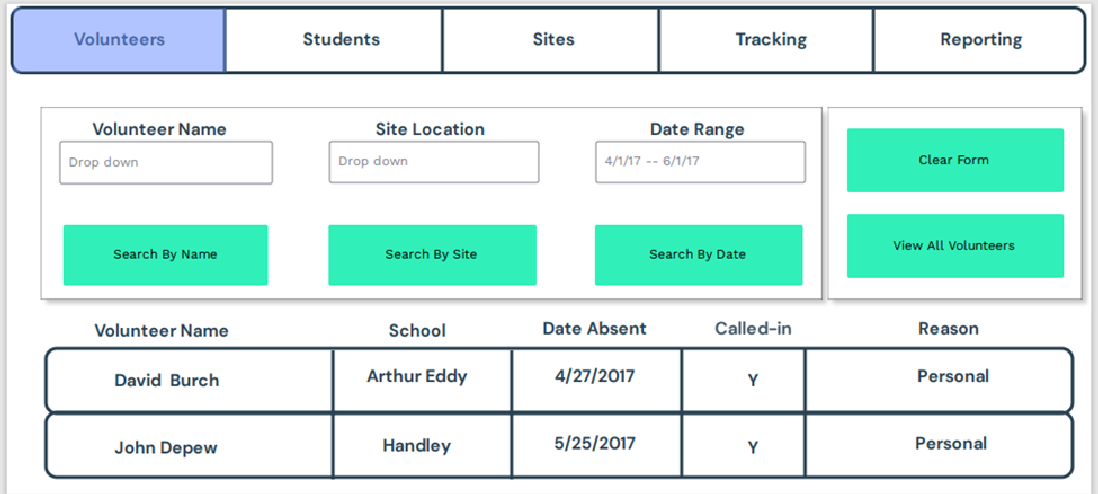
**Figure 96 Activity Log (Populated by Volunteer Name)**

This shows the activity log for an individual volunteer



**Figure 97 Activity Log (Populated by Site)**

This shows the activity log populated by each school location.



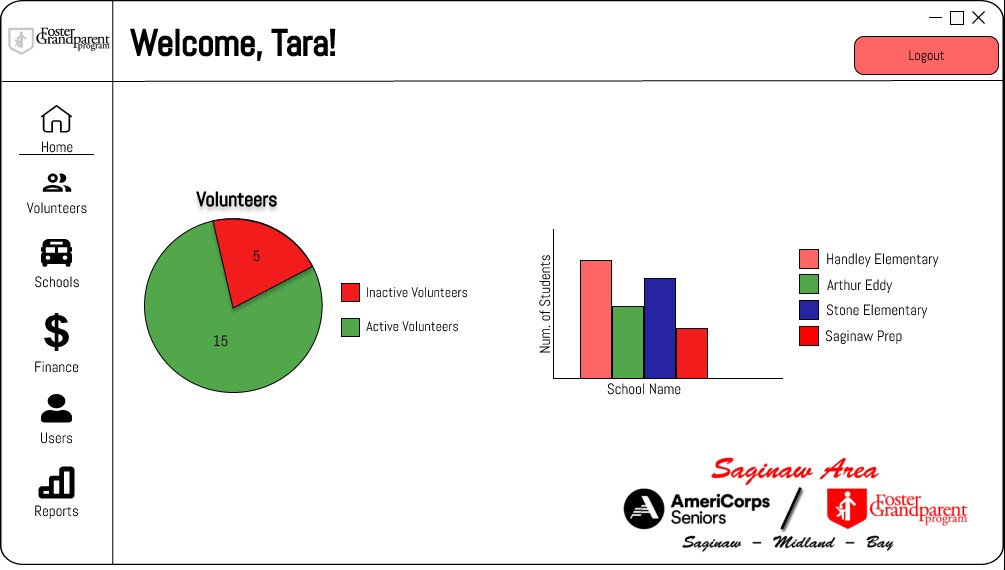
**Figure 98 Activity Log (Populated by date range)**

This shows the activity log populated by date range. View all volunteers would be populated in similar nature, but without the date restriction.

Class UI Proposal

**Figure 99 Login screen**

The initial screen upon entering GEM. Hosts a login and a retrieve password option.



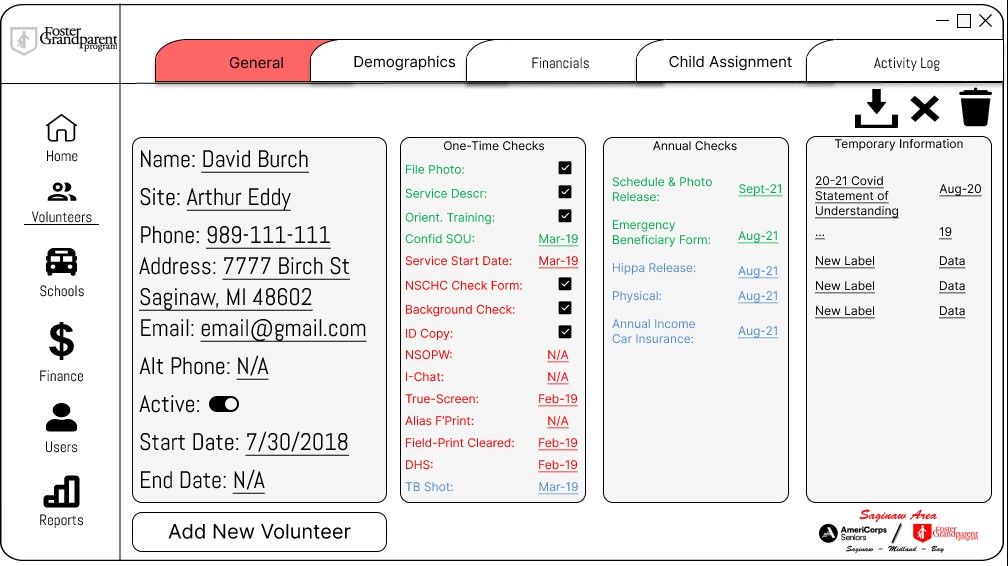
**Figure 100 Home screen**

The Home Screen is what will be landed on upon logging in. This hosts a couple graphs displaying current volunteer deployment.



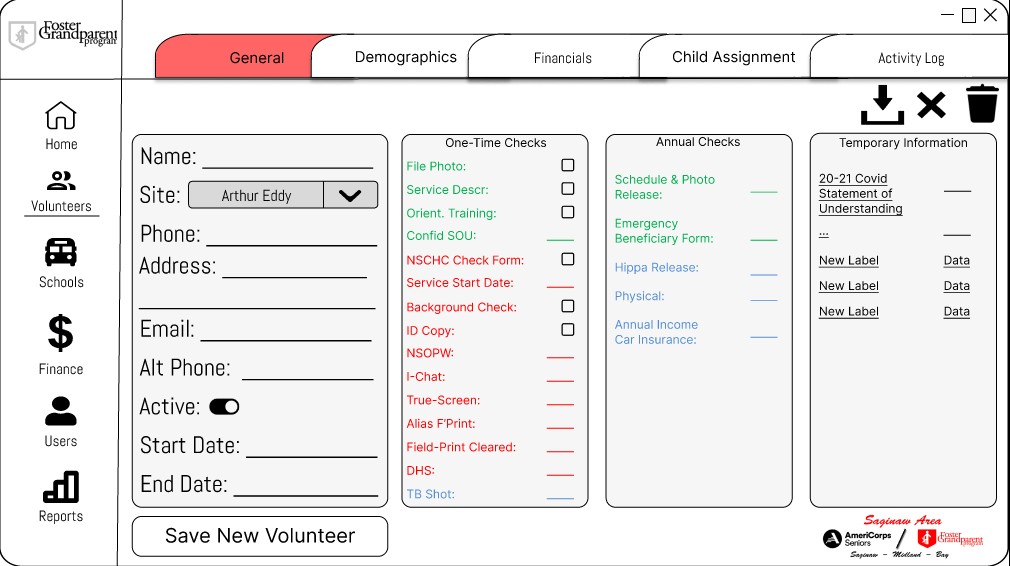
**Figure 101 General Volunteer page**

The landing page for the volunteers tab. This is populated with an example volunteer.



**Figure 102 General volunteer page (editing)**

This shows what the page will look like when editing a volunteer’s information or updating his/her yearly checks.



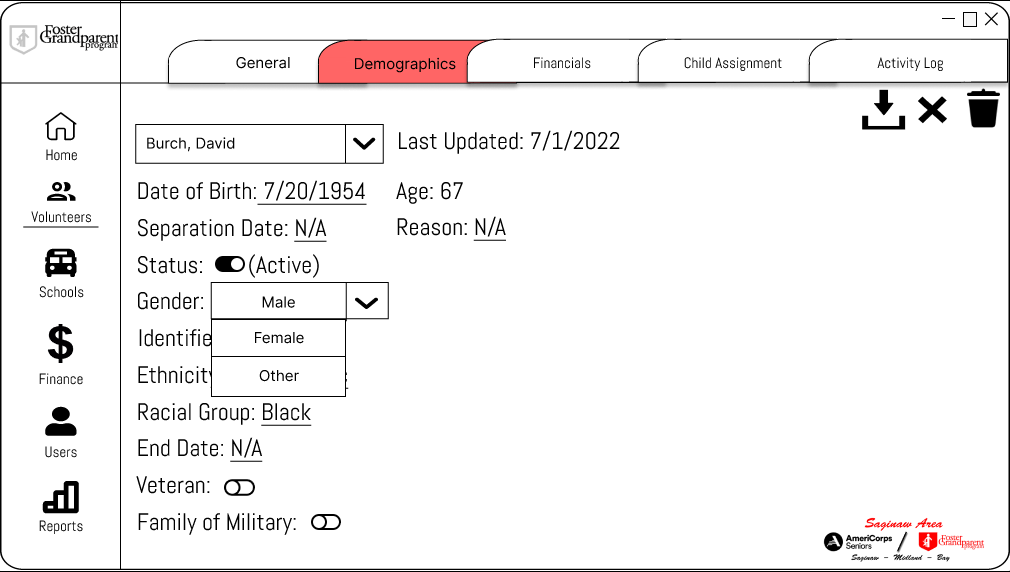
**Figure 103 General volunteer page (adding)**

An example of what adding a new volunteer will look like.



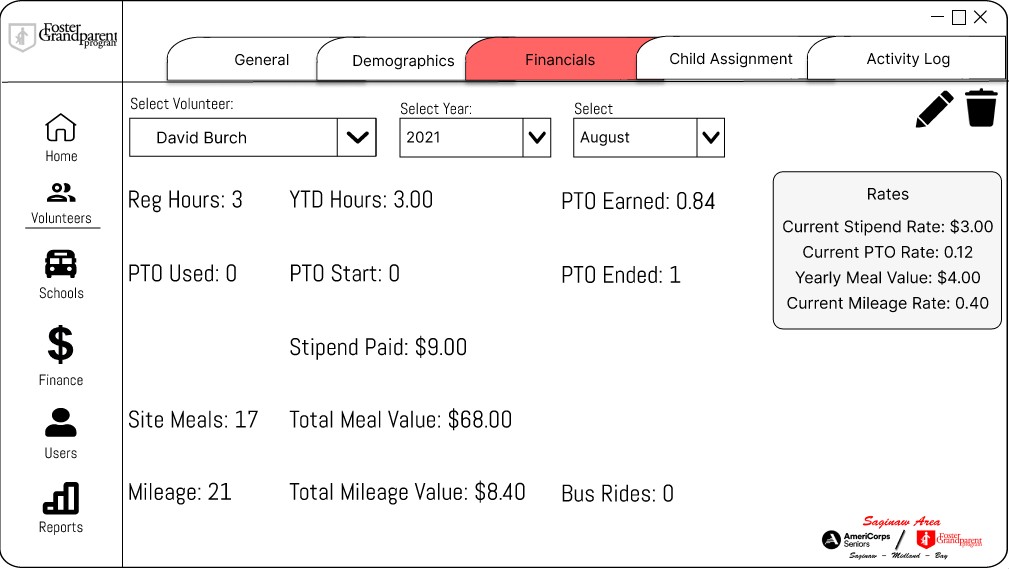
**Figure 104 Volunteer demographic page**

The demographics tab hosts an individual volunteer’s demographic information.



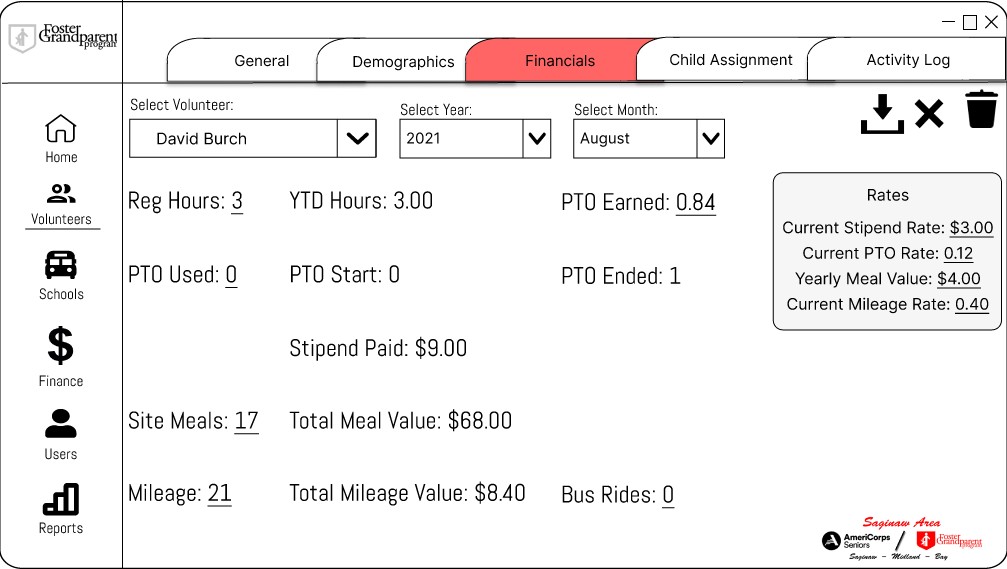
**Figure 105 Volunteer demographic page (Adding/editing)**

An example of editing the demographics page.



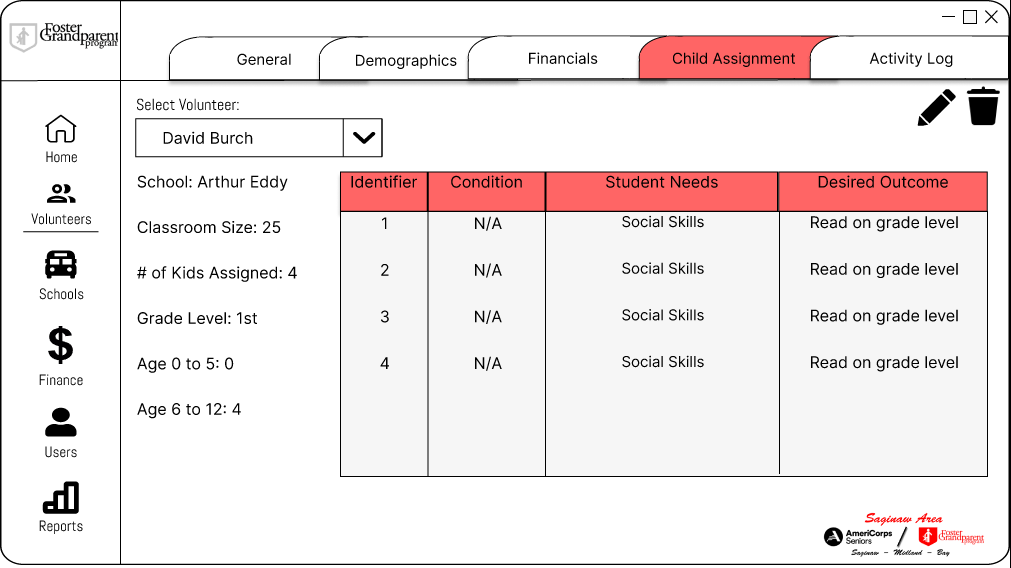
**Figure 106 Volunteer financials page**

The financials tab hosts a volunteer’s related financial information.



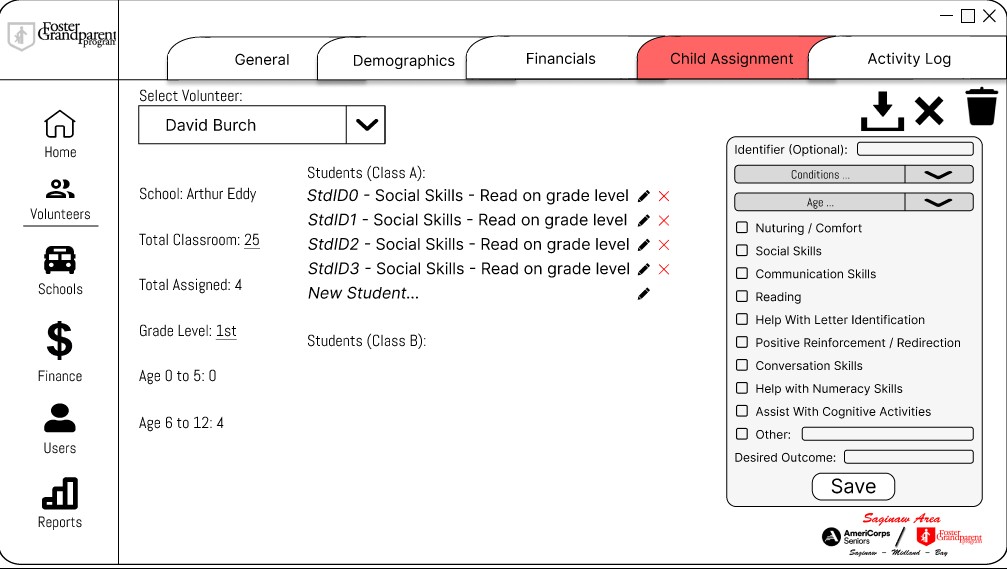
**Figure 107 Volunteer financials page (editing)**

An example of editing a volunteer’s financial page. The rates on the right-hand side are editable and changing those values will change other values calculated using those numbers.



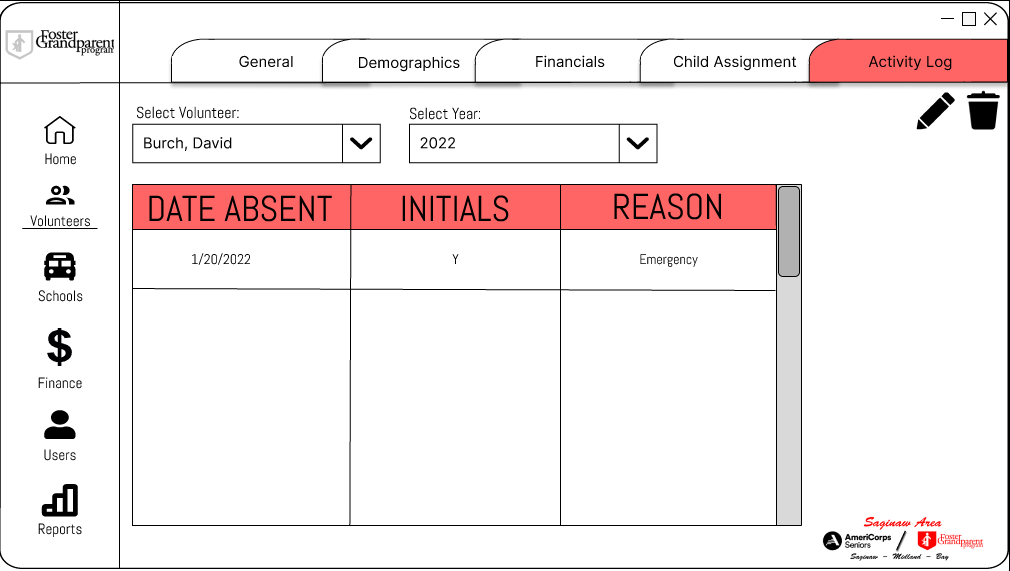
**Figure 108 Volunteer child assignment page**

The child assignment tab will show all students a volunteer is currently working with.



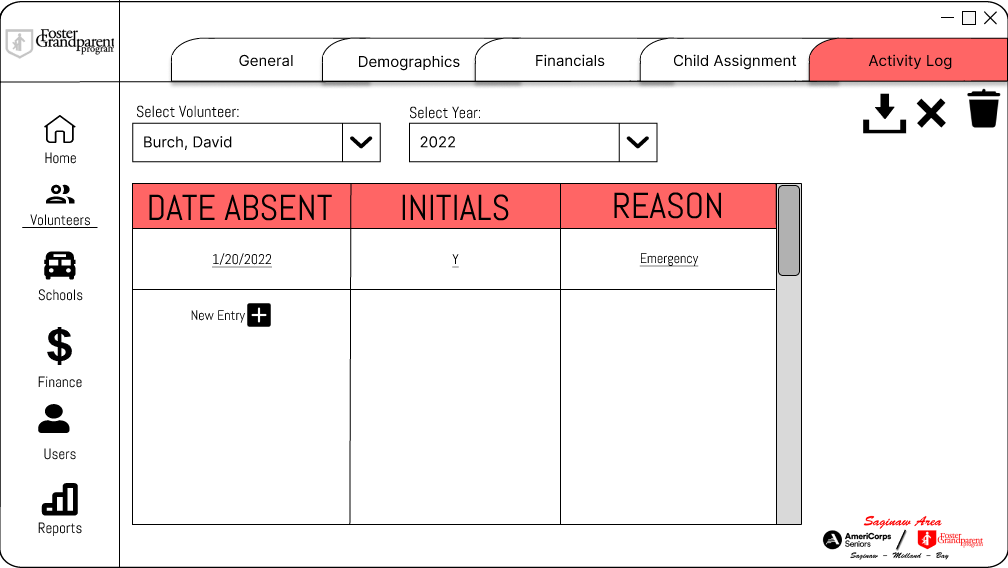
**Figure 109 Volunteer child assignment page (editing)**

An example of what editing a volunteer’s child assignment tab looks like.



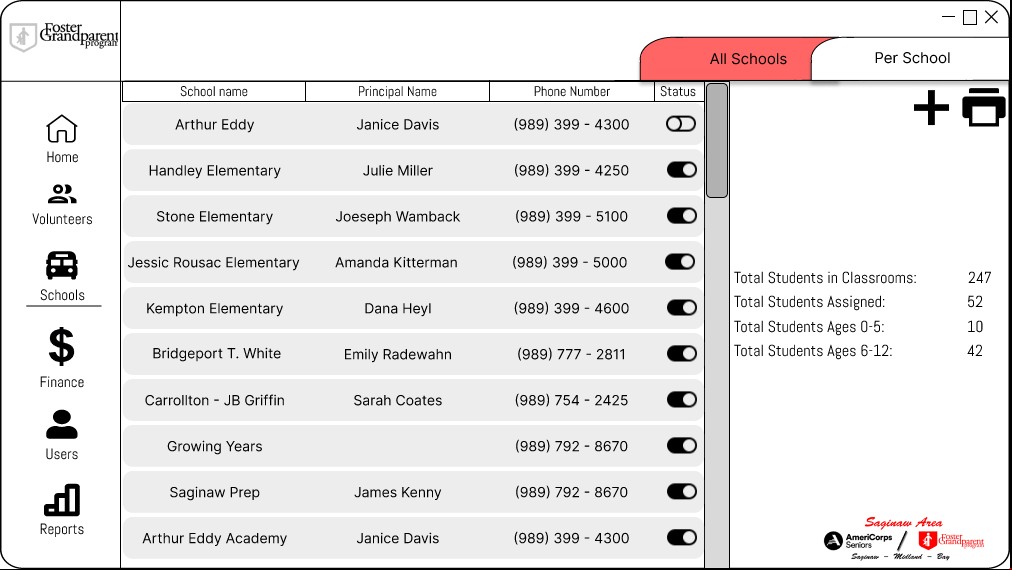
**Figure 110 Volunteer Activity log page**

A volunteer’s activity log. Data can be viewed by month, year, or the entire set.



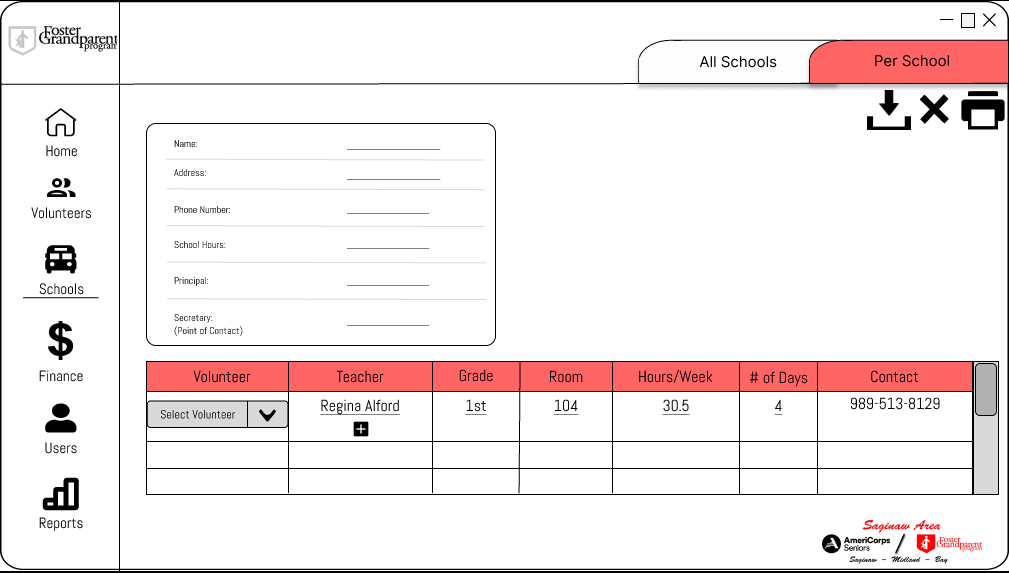
**Figure 111 Volunteer activity log page (editing)**

An example of editing a volunteer’s activity log.



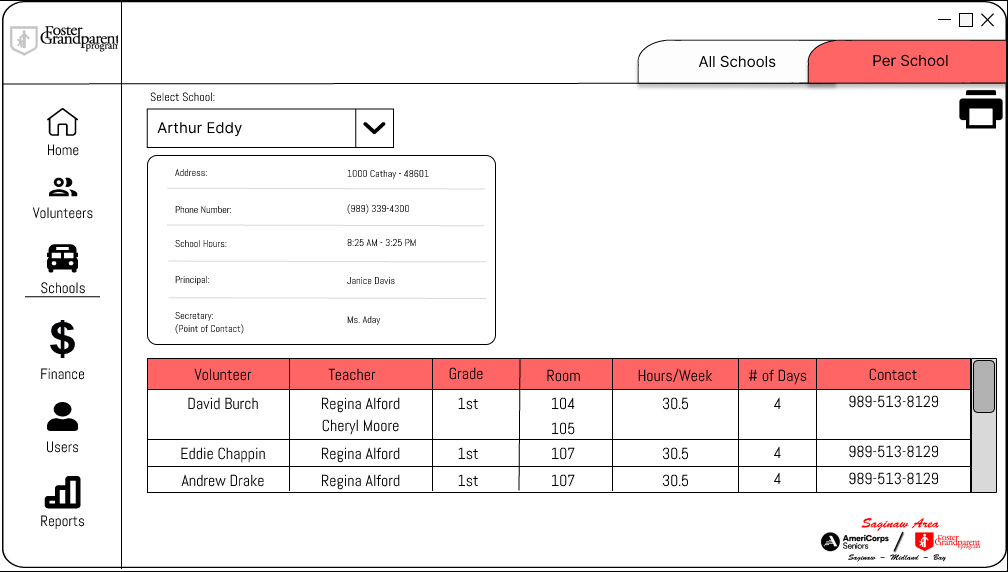
**Figure 112 All schools page**

The landing page for the schools tab. This shows a list of all schools ever associated with the program. The sliders on the right allow editing of a school’s current involvement with the program. A row can be selected to view information on that school.



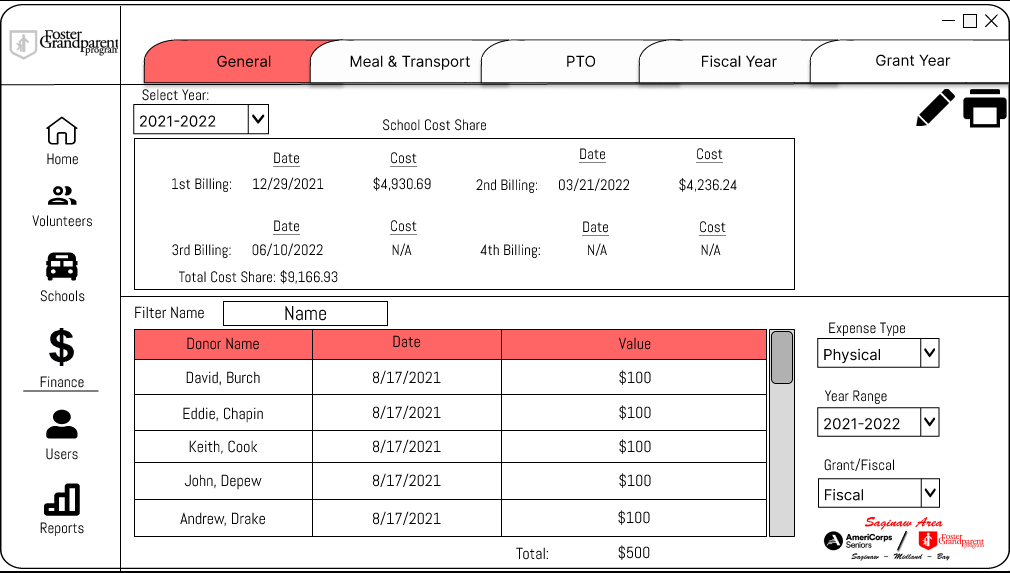
**Figure 113 Editing of a school’s information**

An example of editing a particular school’s associated volunteers and information. This is what adding a new school will also look like.



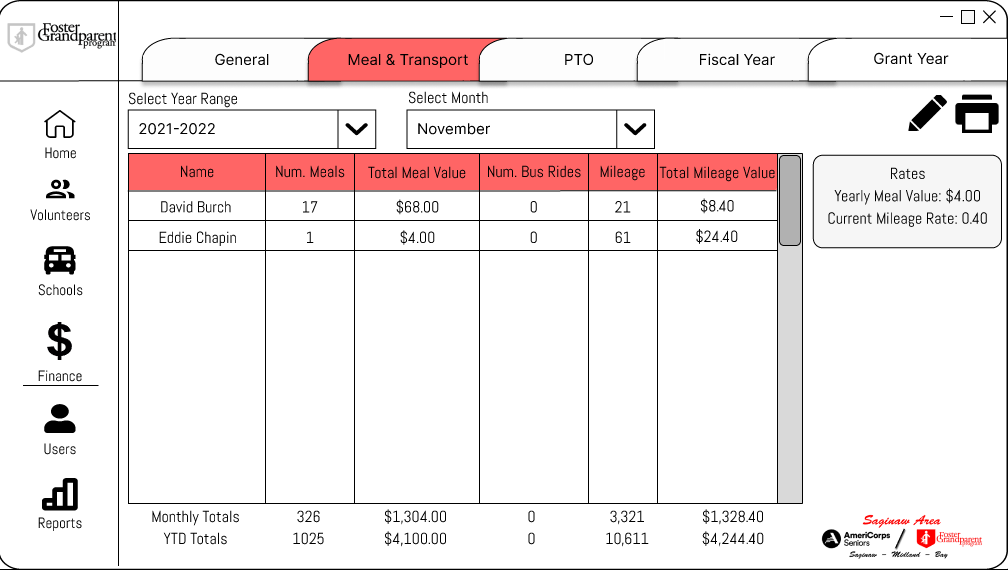
**Figure 114 Viewing of a school’s information**

An example of what a user might see when selecting a participating school from the list.



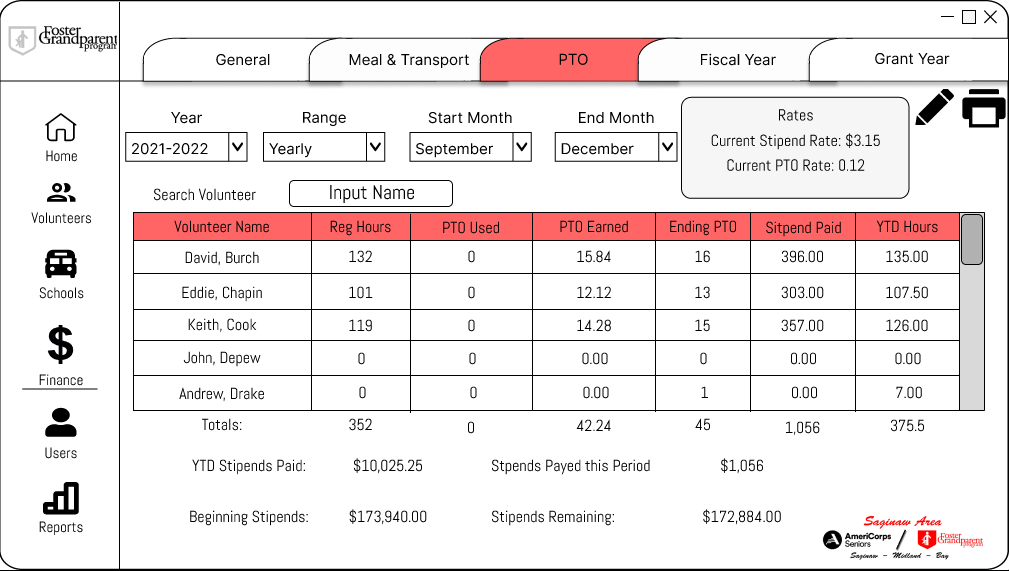
**Figure 115 General Finance page**

The main financial page. Expenses can be viewed and sorted by type or year.



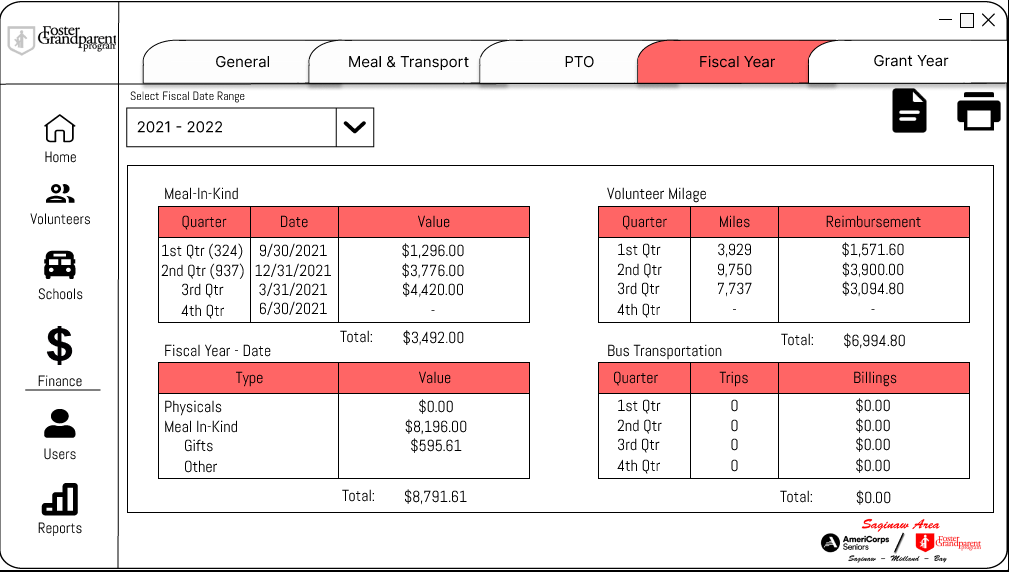
**Figure 116 Meal & Transport financial page**

Tracks all the expenses affiliated with meals and transportation.



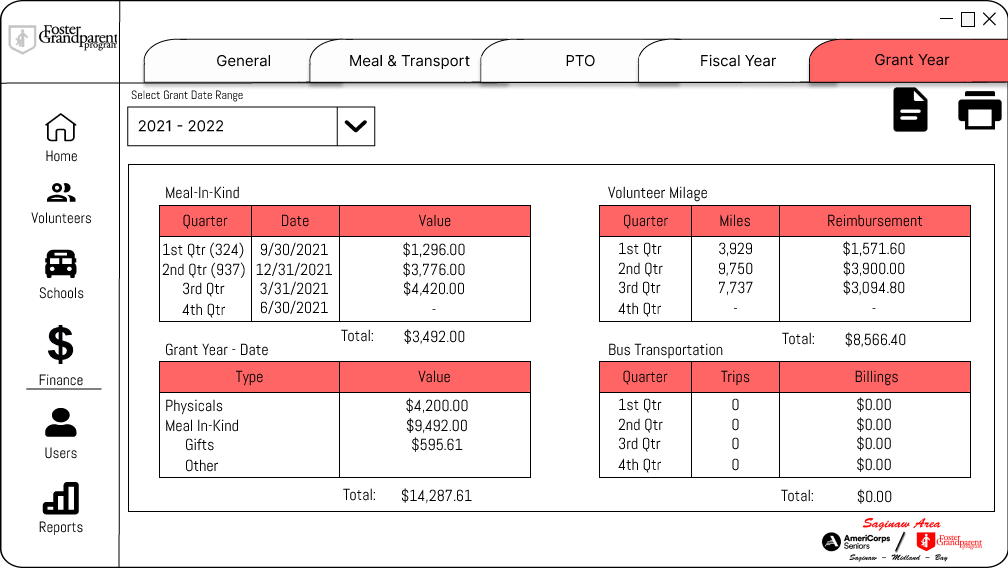
**Figure 117 PTO financial page**

Tracks each volunteer’s accrued and used paid time off.



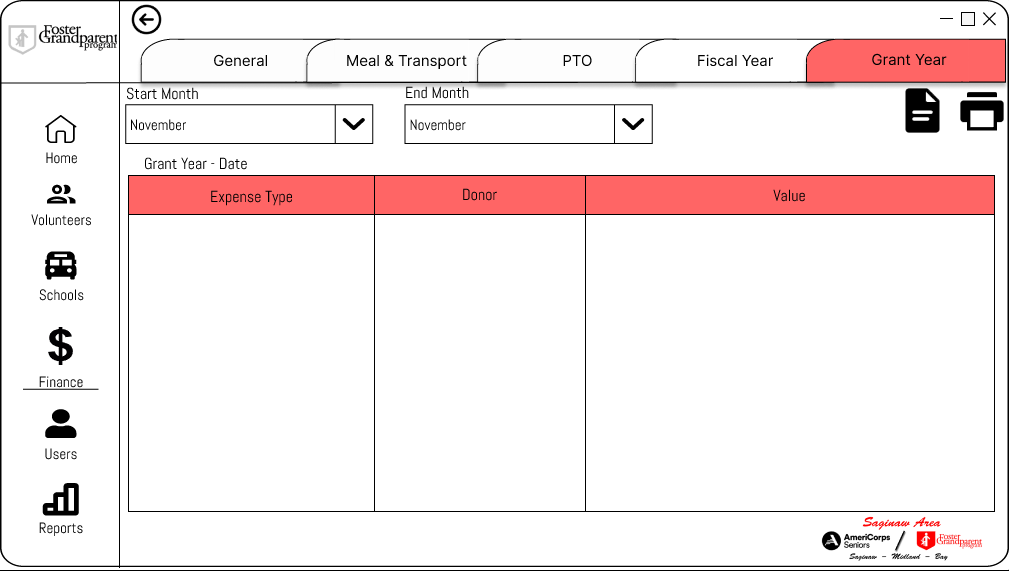
**Figure 118 Fiscal Year financial page**

A quick overview of the finances in the chosen fiscal year. Selecting a table opens a new screen containing that table in a further searchable view.



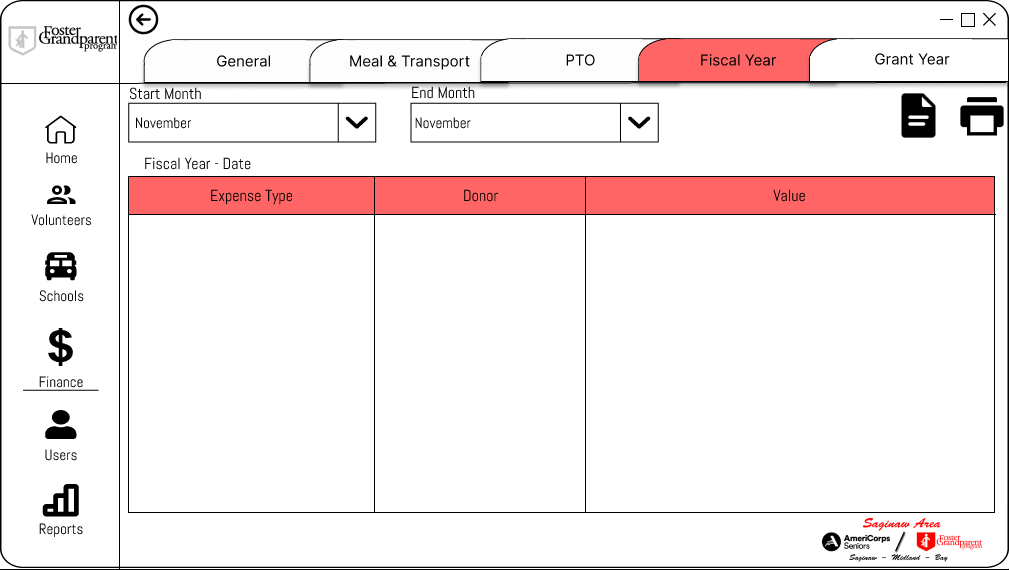
**Figure 119 Grant Year financial page**

A quick overview of the finances in the chosen grant year. Selecting a table opens a new screen containing that table in a further searchable view.



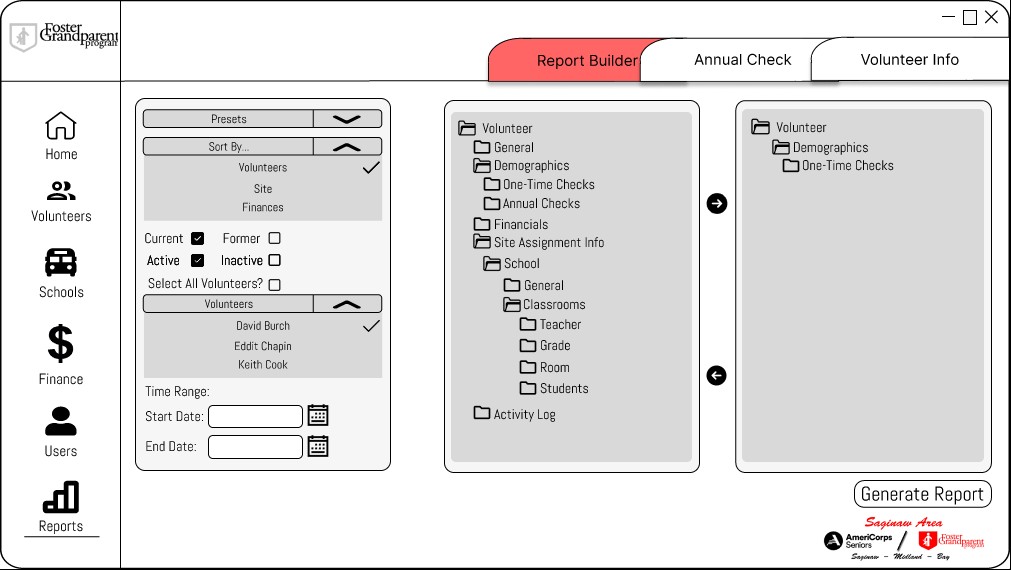
**Figure 120 Grant Year expanded table view**

What clicking on one of the tables would look like in grant year view.



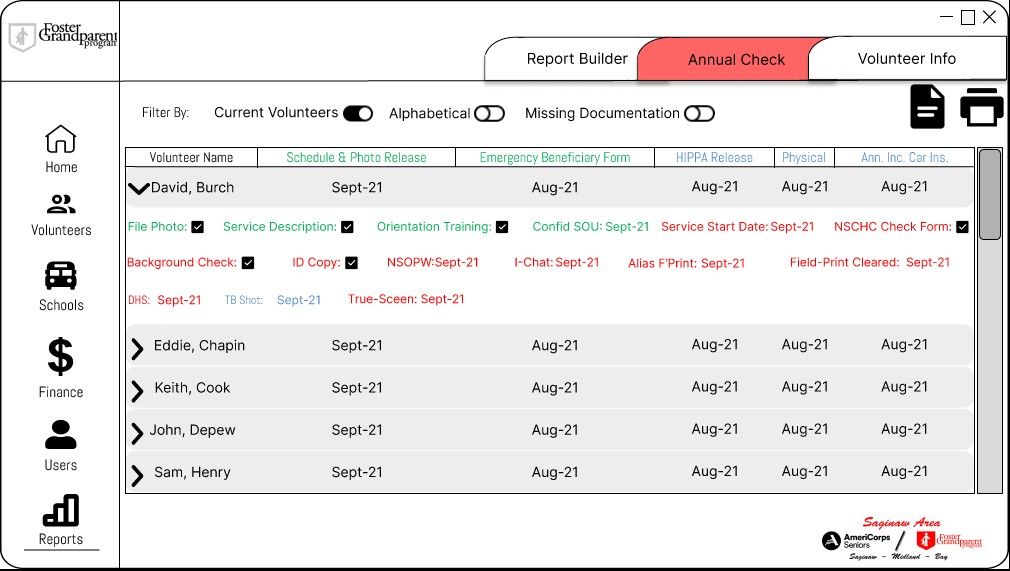
**Figure 121 Fiscal Year expanded table view**

What clicking on one of the tables would look like in fiscal year view.



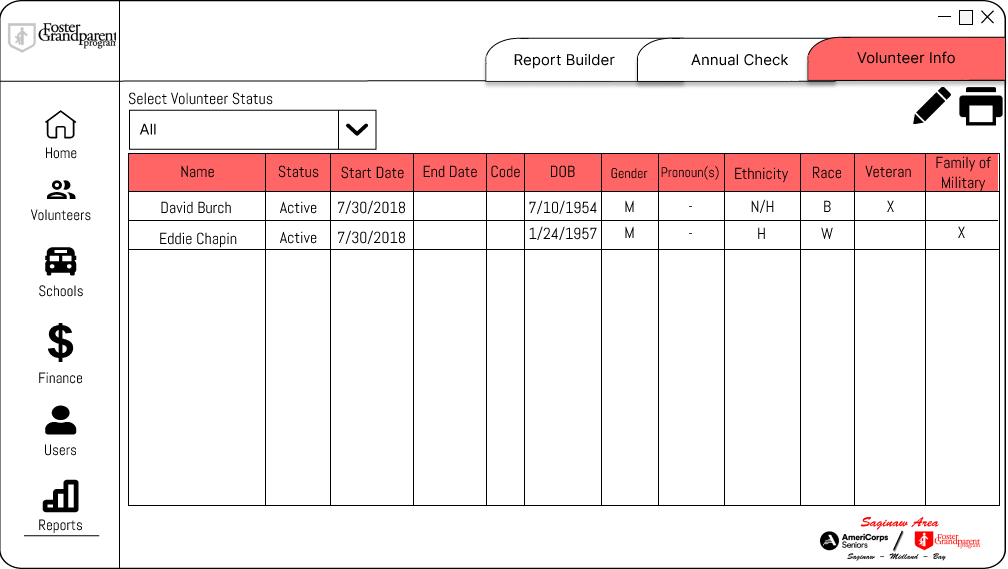
**Figure 122 Report Builder page**

The report builder screen. The user selects the desired information and the fields are compiled to an Excel document.



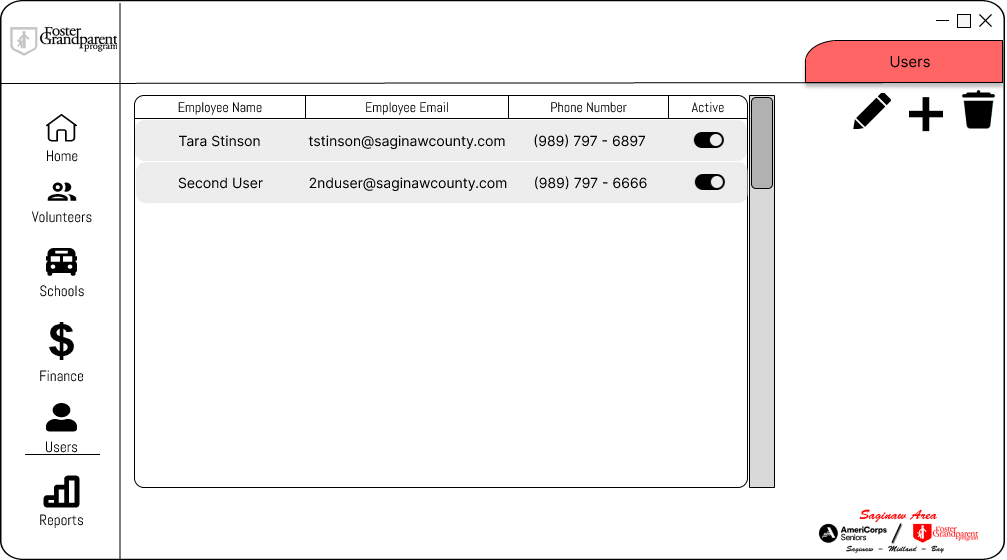
**Figure 123 Annual Check Report page**

The report builder for the annual checks. This provides one location where users can view all volunteers and their yearly required forms. An option is provided to only display volunteers who are missing information.



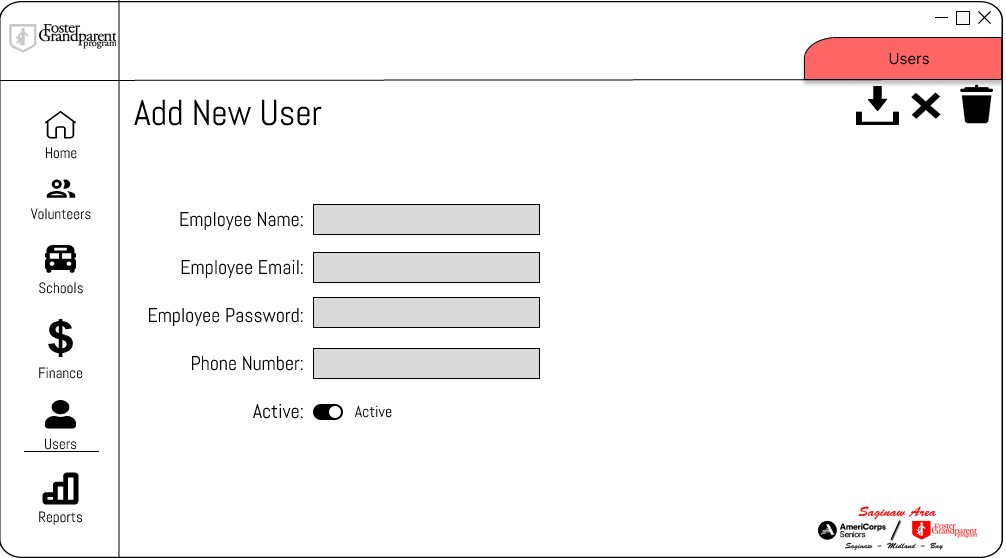
**Figure 124 Volunteer Information Report page**

A consolidated location where a user can view all current and former volunteers and their associated information.



**Figure 125 User page**

An example of the users page. Only the administrative account can add, edit, and delete users.



**Figure 126 User page (adding)**

An example of what adding a new user will look like.

## Domain Model

Volunteer

First Name Last Name Age

Phone Number

Alt Phone Number

Email 1

Address

Date

Hours Worked Meal Count Mileage

Bus Ride Count

In Kind Expenses

Timesheet

Second Address

City State Zip

Birthday

Separated Date Reason Separated Gender

Identifies As

Ethnicity 1

Race

Family of Military: boolean Veteran: boolean

Active: boolean

0..1

\*

Days

Start Time End Time

Hours Per Day Number of Days Hours Per Week Teacher

Grade Room

Total Classroom Total Assigned

Total Ages 5 to 12 Total ages to 5

Assignment

1

\*

Identifier

Student

1 \*

Reports to

Coordinator

Creates

\*

Need

1

\*

Condition

First Name

Last Name

Acronym

Description

Acronym

Description

Generates

Sets

Sets \*

Rate

School Cost Share

1st Billing 2nd Billing 3rd Billing

\* Current Stipend Rate 1

Current PTO Rate

Meal Rate Mileage Rate Date

Determines

4th Billing Date

1 1

1

One Time Checklist

File Photo

Service Description Training Sheet Confidence SOU Date Service Start Date

1 NSCHC

Background Check

ID Copy

NSOPW Date Ichat Date

True Screen Date Alias Fingerprint Date Field Print Date

DHS Date

Tb Shot Date

Annual Checklist

PTO Stipend Report 1

Regular Hours

PTO Start

\* PTO End \*

PTO Used PTO Earned YTD Hours Stipend Paid Date

1

Meal Report

\* \*

Meal Count

Total Meal Value

Date

1

Grant Year Report

\* Meal In Kind Total

In Kind Total

Volunteer Mileage Total Bus Transportation Total Date

1

Fiscal Year Report

\* Meal In Kind Total

In Kind Total

Volunteer Mileage Total Bus Transportation Total Date

Photo Release Date

\* Emergency Beneficiary Date \*

HIPPA Release Date

Physical Date

Car Insurance Date Covid SOU Date Year

In Kind

\* \*

Expense Type Value

Date

Transport Report 1

\* Mileage \*

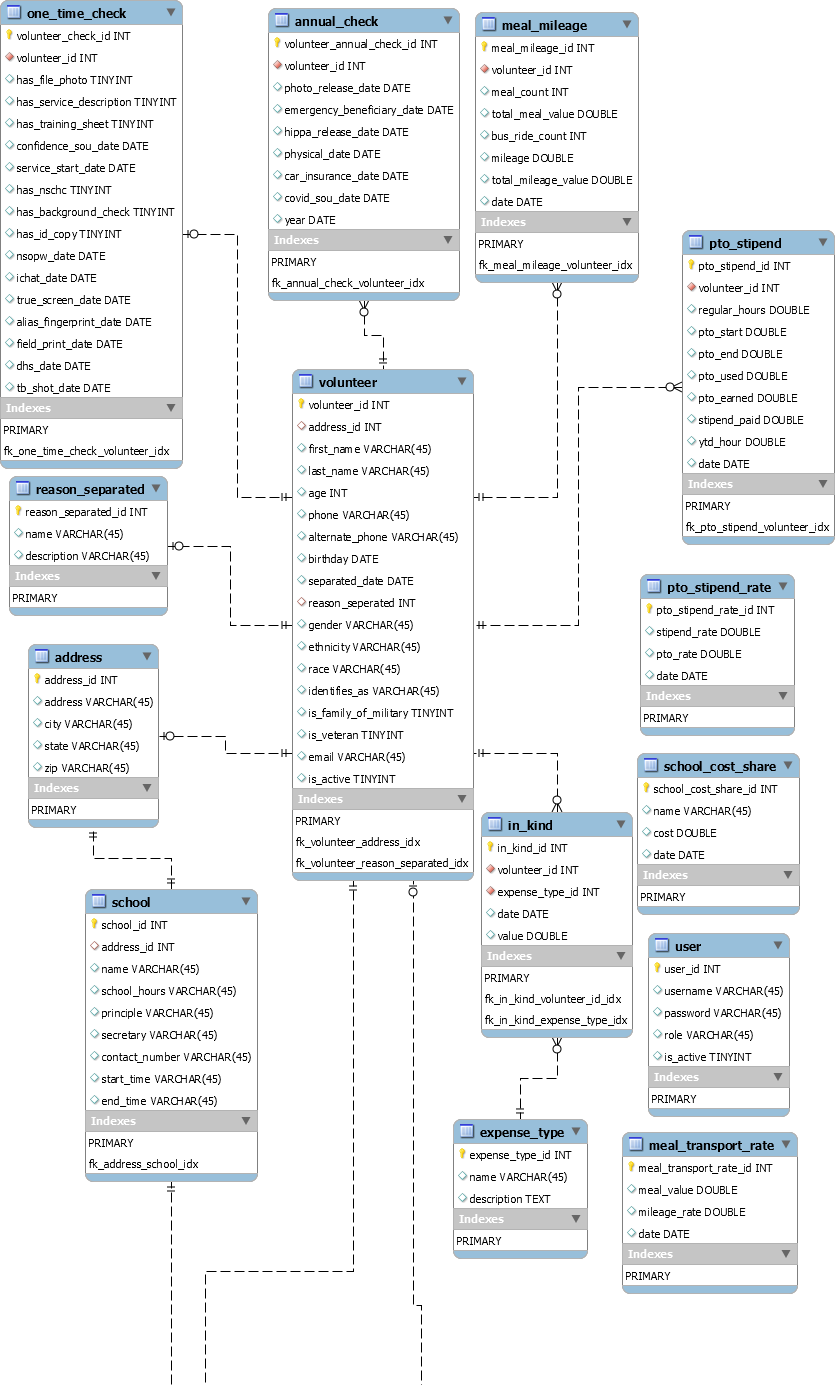
Total Mileage Value

Bus Ride Count

Total Bus Ride Value Date

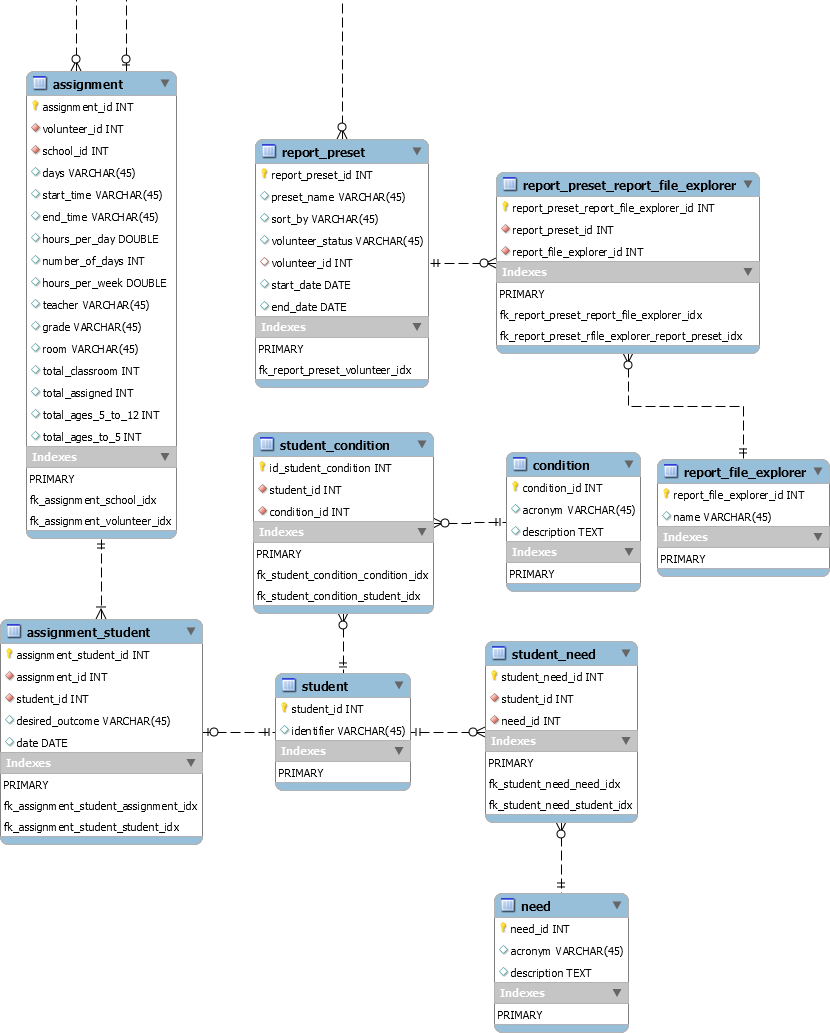
**Figure 127 Domain model of the GEM system**

## ERD Diagrams/Data Dictionaries/Data Dictionary Notation



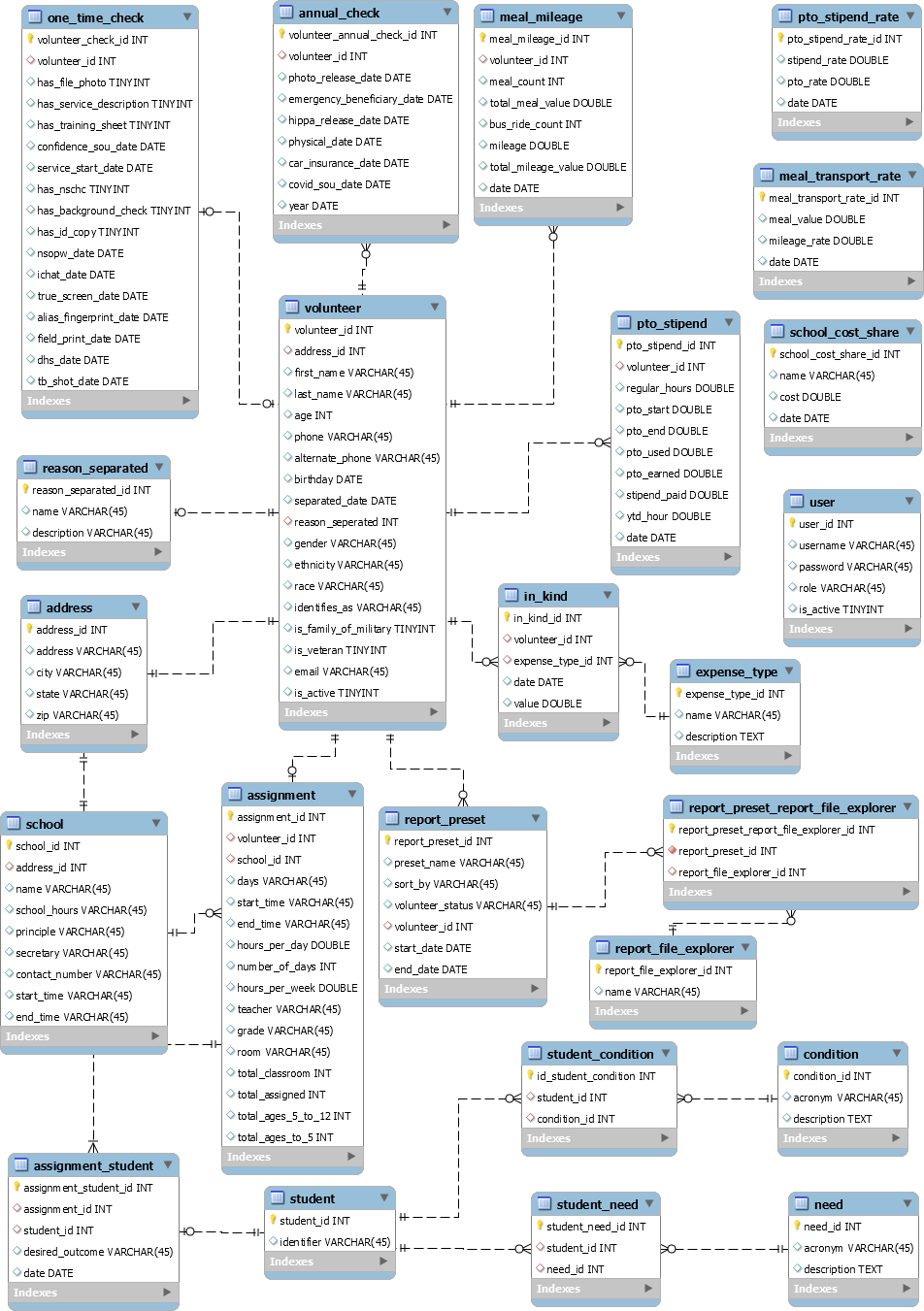
Assignment Table Report\_Preset Table

**FIGURE 128 ERD DIAGRAM DESCRIBING THE SCHEMA OF THE DATABASE**



**FIGURE 129 ERD DIAGRAM DESCRIBING THE SCHEMA OF THE DATABASE**

Complete ERD Diagram



**Figure 130 Completed ERD Diagram**

## UI Workflow Diagrams

Consolidated UI Diagram

Login Page Home Screen

General Volunteers Page

Demographics Financials Page

Page

Child Assignment

Page

Activity Log Page

All Schools Page

Per School Page

Add Page

General

Finance Page

Meal &

Transport Page

PTO Page Fiscal Year Page Grant Year

Page

Users Page Edit Page

Report Builder Page

Annual Checks Reports Page

Volunteer Info Page

**Figure 131 UI Workflow Diagram describing the general flow of how the user will move through the page**



Group UI Diagram

Password Recovery

User Requests New Password

User Confirms New Password

User

Enters App

Login Screen

User Successfully Logs in

Home (Unpopulated

Volunteer Page)

User Selects Desired Tab

Unsuccessful Login Attempt

Current Volunteers Page

Unpopulated Students Page Unpopulated Sites Page

Tracking Page

Main Reporting Page

User Presses View All

Volunteers

User Presses View All

Active Volunteers

User Presses Search by Name or Search by Site

User Presses Search

By Name

User Presses Search

by School or View All

User Filters Results

User Presses View

Grant Report

User Presses View

Fiscal Report

Volunteers(All/Ever) Page

Current Volunteers Page

Populated Volunteer Page

Populated Students Page Populated Sites Page Focused Grant Year Page Focused Fiscal Year Page

User Logs Out

**FIGURE 132 UI WORKFLOW DIAGRAM DESCRIBING THE GENERAL FLOW OF HOW THE USER WILL MOVE THROUGH THE PAGE**

## Battle Tests

* + - 1. UC01: Adding a Volunteer
         1. The user logs in and attempts to add a volunteer, but the volunteer is not added to the system.

Recovery: Prompt the user with an error message and allow for a “try again” button.

* + - 1. UC02: Adding a Student
         1. The user logs in and attempts to add a student, but the student is not added to the system.

Recovery: Prompt the user with an error message and allow for a “try again” button.

* + - 1. UC03: Assigning Volunteer to School
         1. The user logs in and attempts to assign a selected volunteer to a school, but the system fails to assign the volunteer.

Recovery: Prompt the user with an error message and allow for a “try again” button.

* + - 1. UC04: Assigning Volunteer to Student
         1. The user logs in and attempts to assign a selected volunteer to a student, but the system fails to assign the volunteer.

Recovery: Prompt the user with an error message and allow for a “try again” button.

* + - 1. UC05: Logging into the System
         1. User attempts to log in and their credentials are not accepted i.Recovery: Prompt user to double check credentials provided and allow for a “try again” button.
      2. UC09: Generating Volunteer Report
         1. The user selects information to generate a report from and the user hits generate, but the report fails to generate.

Recovery: Prompt the user with an error message, auto-select previously selected information, and allow for a “try again” button.

* + - 1. UC17: Adding to Activity Log
         1. The user selects a volunteer to create an activity entry for and enters desired information into the activity log form, submits the form, but a new entry is not added to the activity log

Recovery: Prompt the user with an error message, pre-fill all previously entered fields, and allow for a “try again” button.

* + - 1. UC18/31: Recording Stipends and PTO
         1. The user selects volunteer to add stipend and PTO information for, enters information onto the form, and submits the form, but stipend and PTO information is not added to the volunteer profile.

Recovery: Prompt the user with an error message, pre-fill all previously entered information, and allow for a “try again” button.

* + - 1. UC19: Changing Timecard
         1. The User selects a volunteer to change timecard information for, updates timecard information, and submits changes, but the timecard information is not updated.

Recovery: Prompt the user with an error message, return to change timecard form, pre-fill with previously entered data, and allow for a “try again” button.

* + - 1. UC20: Generating Monthly Site Form
         1. The user selects to generate a monthly site form, but the form does not open.

Recovery: Prompt the user with an error message and allow for a “try again” button.

* + - 1. UC24: Retrieving Timecard Information
         1. The user selects a volunteer for which they wish to retrieve timecard information and selects to retrieve timecard data, but timecard information is not retrieved.

Recovery: Prompt the user with an error message, re-select the previously selected volunteer, and allow for a “try again” button.

* + - 1. UC25: Mass Entering of Volunteers
         1. The user enters a list of volunteers including their information and selects add volunteers, but the volunteers fail to be entered into the system.

Recovery: Prompt the user with an error message, auto-fill all previously entered information, and allow for a “try again” button.

* + - 1. UC28: Entering Meal/Mileage Rates
         1. The user selects rates to be modified, enters a new rate, and selects save, but the rates are not saved to the system.

Recovery: Prompt the user with an error message, pre-fill previously entered rates, and allow for a “try again” button.

* + - 1. UC32: Updating Volunteer Information
         1. User selects volunteer to update information, user updates information, volunteer information fails to update.

Recovery: Prompt the user with an error message, pre-sill all previously entered data, and allow for a “try again” button.

* + - 1. UC33: Adding Schools
         1. The user selects to add a school, fills in the form with school information, and selects to save the school, but the school is not entered into the system.

Recovery: Prompt the user with an error message, pre-fill the form with previously entered information, and allow for a “try again” button.

* + - 1. UC34: Deleting Schools
         1. The user selects the school they wish to delete and selects the delete button, but the school still appears in the system.

Recovery: Prompt the user with an error message, pre-select the school user wishes to delete, and allow for a “try again” button.

* + - 1. UC35: Changing Volunteer’s Status
         1. The user selects volunteer to change the status of, changes status, but the status of the volunteer is not changed.

Recovery: Prompt the user with an error message, pre-select the volunteer user wishes to change the status of, and allow for a “try again” button.

* + - 1. UC36: Add New User
         1. The administrative user selects to add a new user, fills in the form with information regarding the user, and hits submit, but the user is not added to the system.

Recovery: Prompt the user with an error message, pre-fill the form with previously entered information, and allow for a “try again” button.

* + 1. Implementation Alternatives and Recommendations

For the Foster Grandparent Program system, we considered three different application solutions to streamline and better record the needed information to manage and run the volunteer program. The options are the following:

* + - 1. Create a desktop application that will run on Windows 10 and 11. This application will have a custom UI that will allow a user to manage all necessary data relating to the Foster Grandparent Program. This includes creating, updating, reading, and deleting volunteers, schools, and students. This application will also store all necessary financial information and generate reports needed for the function of the program and to submit to governmental entities. The main reason we would adopt this system is that it would eliminate the current excel sheet system with its redundancies and allow for easy management and report creation. A desktop application would allow for greater security as the only outside connection would be to the database. Additionally, it will require users to log in to access the application.
      2. The second option we considered is a web-based application. This would allow for all the benefits of a desktop-based application - such as a custom UI and Windows 10 and 11 requirements - to generate reports for any business need, and allow functionality to a range of devices that may be used within the program. This would allow the application to be used on Windows, iOS, Android, and any other mobile device which would be a benefit to the program if the need for outside-the-office access were to arise. This increase in access would present itself as a greater security risk as it would be able to be accessed from anyone anywhere and would require greater security protocols to stop hackers from accessing sensitive data about finances and volunteer information.
      3. The last option we considered is an existing product named Rosterfy which is a nonprofit volunteer management software used by notable nonprofit organizations such as Habitat for Humanity, The British Heart Foundation, and the Child Cancer Foundation. This software would do everything that is required for the Grandparent Foster Program such as volunteer, school, and student management, easy reporting of needed financials, and would run on any operating system the program requires. The benefit of this approach is that the application could be accessed anywhere via computer or phone with any operating system. This would also eliminate any issue with maintenance

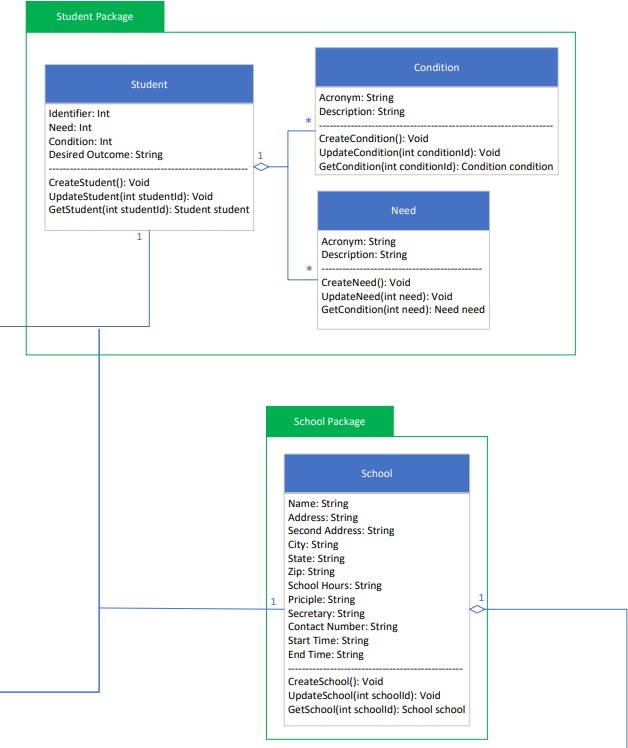
as Rosterfy would fix any issues that arise and would be able to provide support to the users in the Foster Grandparent Program. The downside to this approach would be that the program would have to pay a monthly or yearly fee to access the program which could be a burden on an already minimal budget. Also, the users would be locked into the constraints of Rosterfy and would not be able to have custom functions if needed.

After weighing all the options and discussing the pros and cons of each of the above options, our group decided to go with Option 1 and build a custom desktop application for the Foster Grandparent Program. We agreed to this option based on four criteria, cost, security, reliability, and ease of maintenance. Option 1 is the cheapest option to develop as the cost of development will be less than building a web-based application as the expertise required is less for a Windows application. Also, the cost of hosting this application will be nominal as the database will be on the existing Saginaw County servers and the application will be hosted locally placed on the machines the Saginaw Commission of Aging uses. This is contrary to the cost that it would be to maintain the entire application on a web server such as Amazon Web Service if we chose option 2 and the cost of a monthly subscription if we chose option 3. Option 1 is also the most secure as it is not vulnerable to web-based attacks which would affect a web application and Rosterfy if they were the victim of a DDoS attack. The only vulnerability would be if the Saginaw County servers got hacked or if a user allowed a virus to be downloaded on their local machine. Furthermore, Option 1 is the most reliable of the three options as the only interruption to functionality would be if the Saginaw County servers went down. Option 1 avoids any connectivity issues that could arise with a web-based application and avoids the reliance on having the entire operation dependent on a third party as in option 3. Lastly, option 1 allows the greatest ease of maintenance as it will avoid any hosting maintenance, updates to web tools that could break the application, and having to wait on an outside party to fix issues if they arise. In our opinion we believe Option 1 to be the best as it will save the county money, the application will be secure from most modern hacking attempts, the application will always be available to its users, and will require minimal maintenance to use for many years to come.

## Graphical user interface, text, application Description automatically generatedRecommended Implementation Architecture Diagram

Financial Package (PTO Stipend)

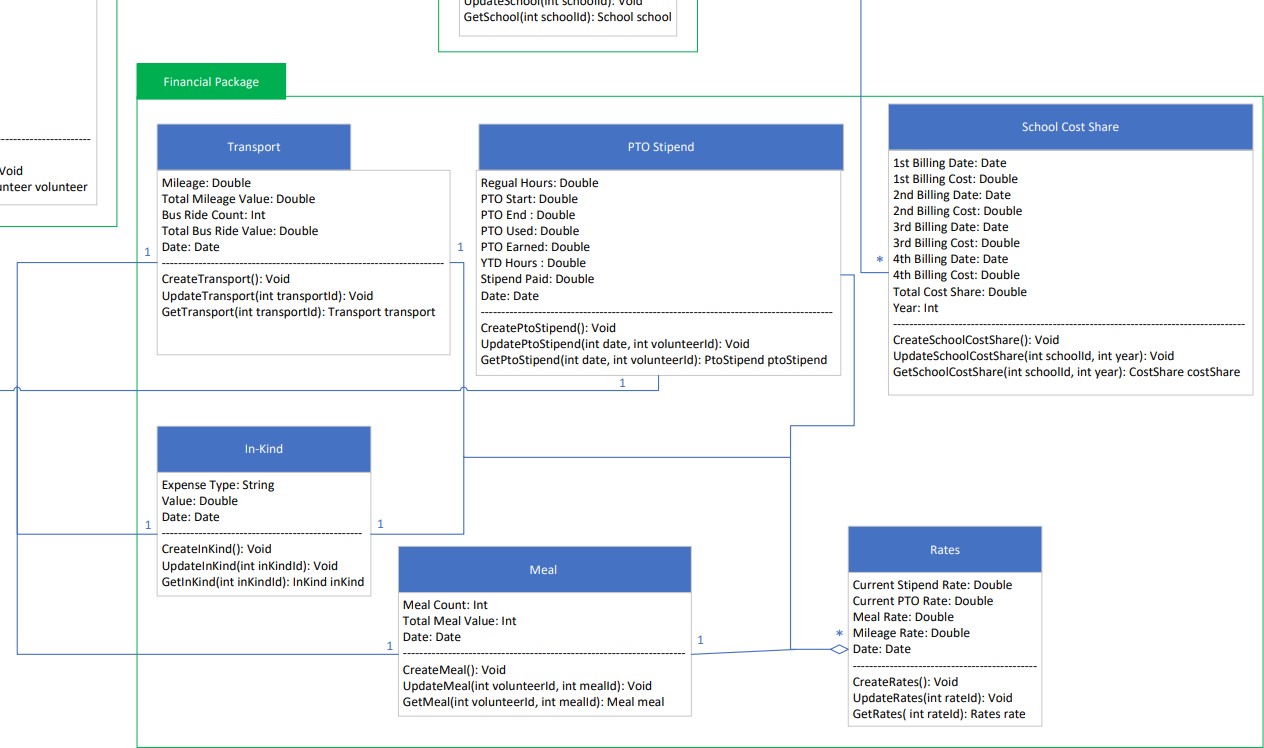
**Figure 133 Architecture diagram for the volunteer package**



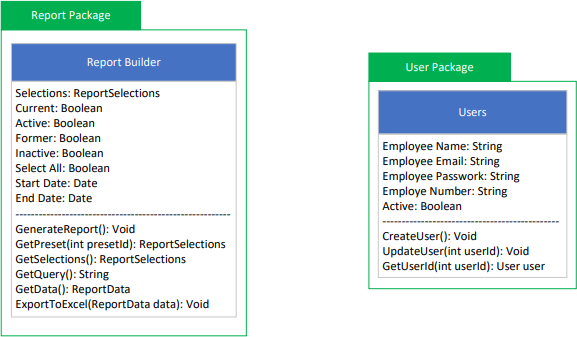
Financial Package (School Cost Share)

**Figure 134 Architecture Diagram for the student and school package**

**Figure 135 Architecture Diagram for financial package**

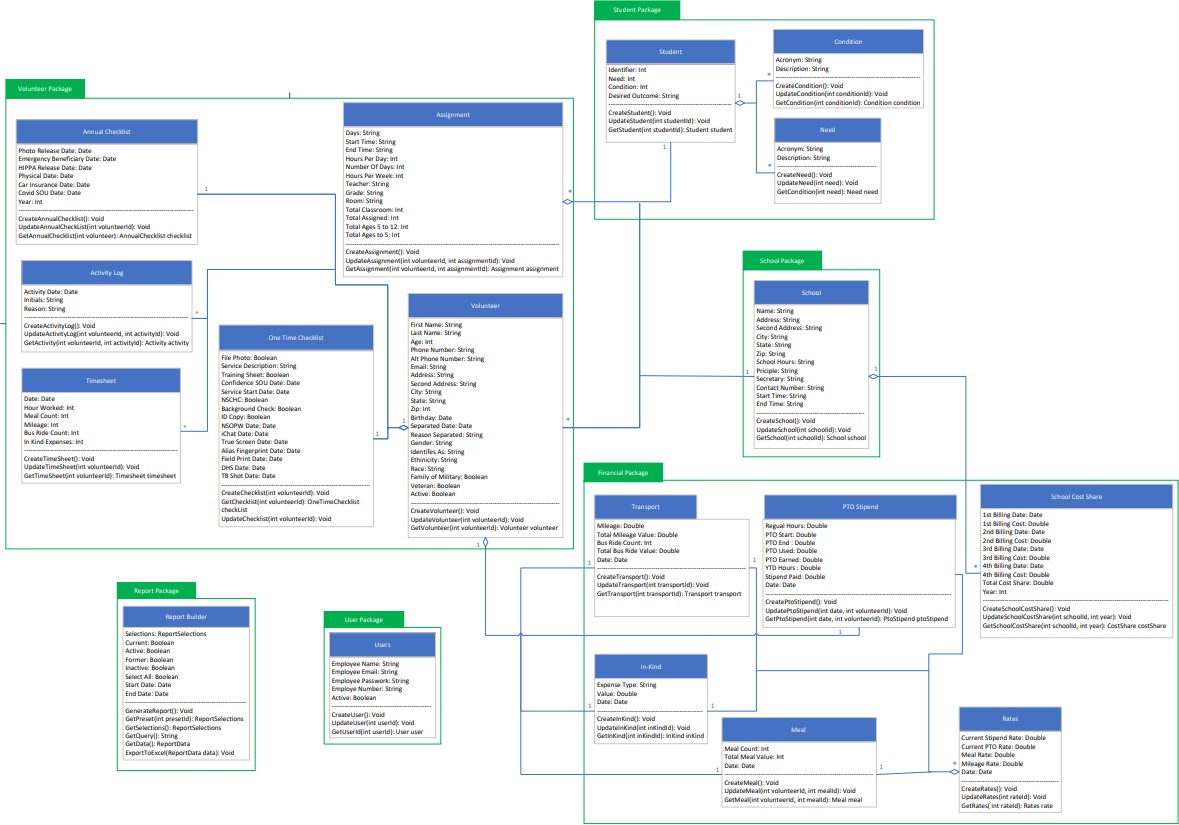


School Package (School)



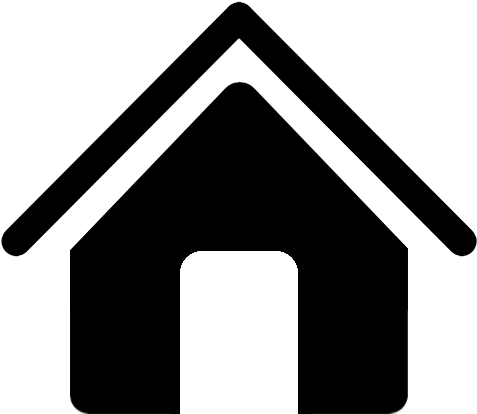
**Figure 136 Architecture diagram for Report and User Package**

Complete Architecture Diagram

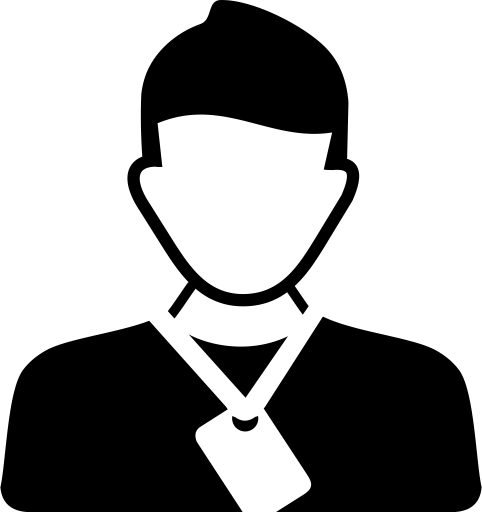


**Figure 137 Completed Architecture diagram**

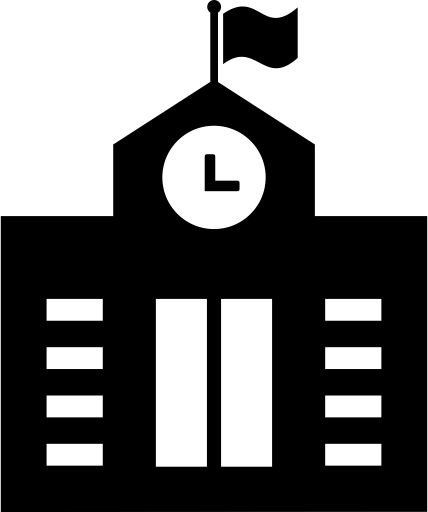
## Image Citations



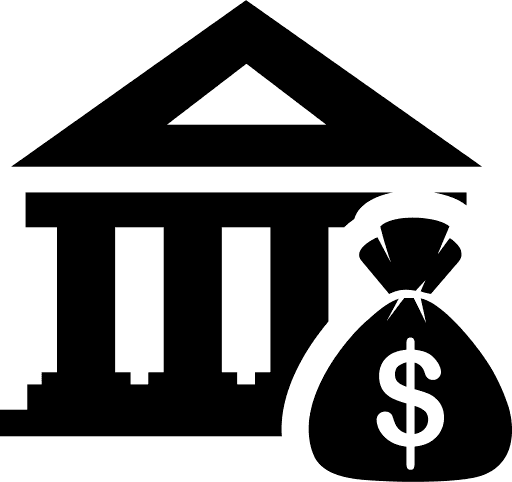
**Figure 138 Home page icon, cited from https://www.clipartmax.com/download/m2K9A0A0b1d3N4Z5\_free-icons-png-new-home-icon/**



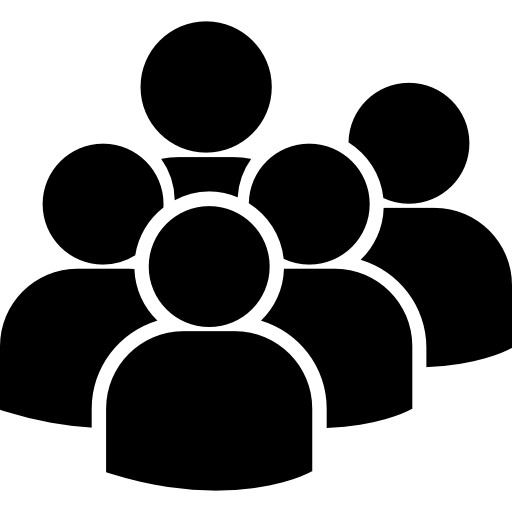
**Figure 139 volunteer icon, cited from https://uxwing.com/volunteer-icon/**



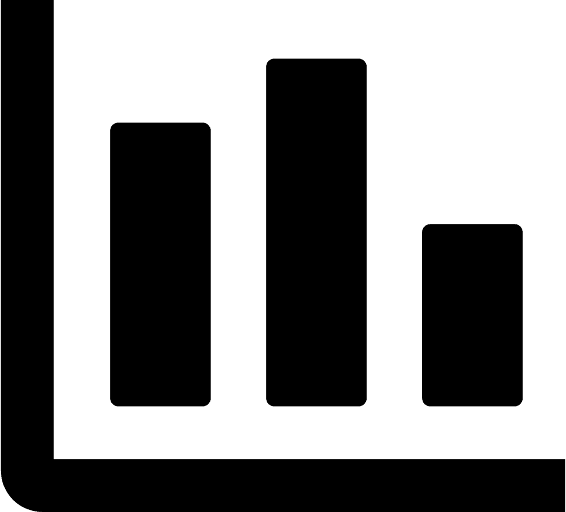
**Figure 140 school page icon, cited from https://uxwing.com/high-school-icon/**



**Figure 141 Finance page icon, cited from https://uxwing.com/bank-finance-loan-icon/**



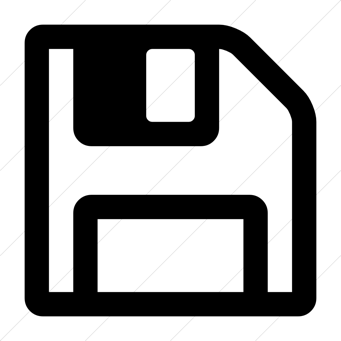
**Figure 142 users page Icon, cited from https://www.flaticon.com/free-icon/users-group\_32441**



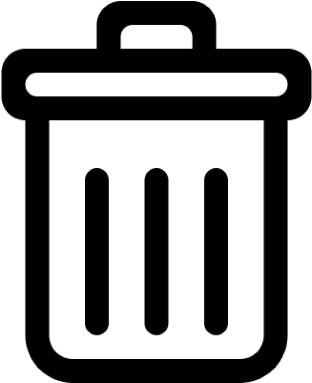
**Figure 143 report page icon, cited from https://uxwing.com/bar-chart-icon/**



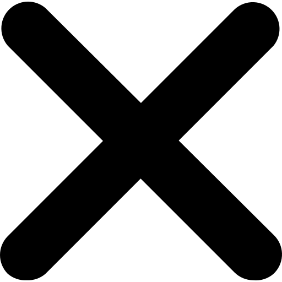
**Figure 144 Export to Microsoft Excel Icon, cited from https://uxwing.com/file-download-import-icon/**



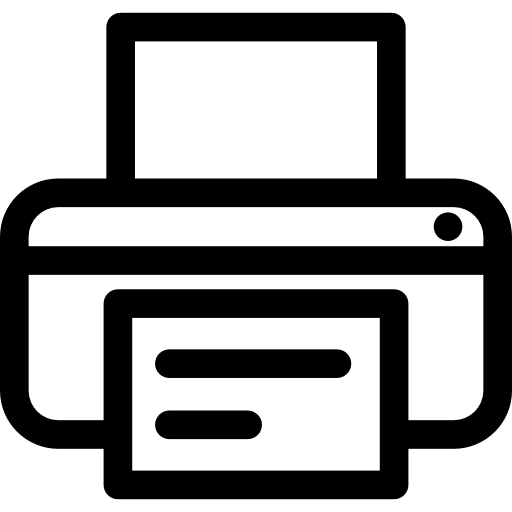
**Figure 145 save icon, cited from https://www.pngfind.com/mpng/iRhwxTo\_black-save-icon-36517-free-icons-and-png/**



**Figure 146 delete icon, cited from https://toppng.com/show\_download/190698/recycling-bin-vector-delete-icon-png-black/large**



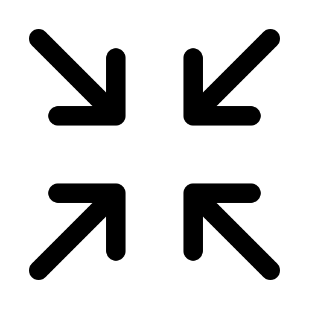
**Figure 147 cancel icon, cited from https://www.flaticon.com/free-icon/close\_2976286?term=x&page=1&position=4&origin=search&related\_id=2976286**



**Figure 148 print icon, cited from https://www.pngwing.com/en/free-png-pvjto/download**



**Figure 149 search Icon, cited from https://www.flaticon.com/free-icon/magnifier\_64673?term=magnify&page=1&position=1&origin=search&related\_id=64673**



**Figure 150 minimize icon, cited from https://www.flaticon.com/free-icon/minimize\_2989876?term=shrink&page=1&position=1&origin=search&related\_id=2989876**

## 

## O.Final Database Scheme

Graphical user interface, application, Word

Description automatically generated

**Figure 151 Part one of the database scheme.**

Graphical user interface, diagram

Description automatically generated**Figure 152 Part two of the database scheme.**

Diagram

Description automatically generated**Figure 153 Part three of the database scheme.**

Diagram

Description automatically generated

**Figure 154 Part four of the database scheme.**

## P.Final Implementation

The system was implemented as a Windows Presentation Foundation(WPF) application using C#, SQL, and LINQ with a SQL Server database and a .NET framework. Github was used as a repository and as source control tool.

## Index

Activity log 10, 15, 29, 55, 117, 136

Annual Check 129

API 3, 13, 35, 36

Budget 19, 79, 80, 81, 99, 100

C# 3, 12

Child assignment 114, 115

Database 1, 4, 10, 33

Desktop application 7, 12, 34, 137, 138

Director/Coordinator 3, 12

Excel 4, 7, 8, 9, 11, 12, 13, 35, 67, 70, 71, 72, 73, 74, 75

Finance 121

Financial 4, 11, 22, 55, 63, 64, 122, 123, 124, 125, 137

Fiscal Year 4, 64, 97, 124, 127

Foster Grandparent Program 0, 9, 10, 11, 33, 137, 138

GEM 3, 4, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29,

30, 31, 32, 33, 34, 35, 36, 38, 39, 40, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 69, 70, 71, 72, 73,

74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 133

Grant Year 63, 98, 125, 126

Login 22, 26, 105

Meal & Transport 122

PTO 5, 19, 20, 34, 55, 77, 123, 136

Report 15, 16, 20, 21, 22, 27, 30, 32, 44, 55, 136, 137

Requirements 0, 1, 2, 10, 13, 14, 15, 16, 19, 22, 27, 29, 30, 32, 33, 85, 86

RESTful APIs 5, 9

Rosterfy 138

Saginaw County Commission on Aging 5, 10, 11

School 5, 8, 11, 13, 16, 17, 18, 19, 33, 34, 55, 62, 69, 119, 120, 136, 138

Site 1, 7, 10, 16, 28, 69, 81, 94, 95, 103, 137

SQL 6, 7, 8, 10

Stipend 6, 19, 34, 55, 76

UI 2, 6, 87, 105, 135, 137

UML 56, 57, 58, 59

Use Case 2, 56, 57, 58, 85

User . 1, 6, 7, 11, 12, 35, 61, 62, 63, 64, 65, 66, 67, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80,

81, 82, 83, 84, 131, 132, 136, 137

Volunteer 4, 5, 7, 10, 11, 14, 15, 18, 19, 20, 21, 27, 28, 29, 30, 31, 32, 33, 34, 49, 55, 60, 65, 66,

69, 74, 76, 77, 78, 79, 80, 81, 82, 92, 93, 108, 109, 136, 137, 138

Windows 7, 34

Word 5, 7, 8, 9, 10, 12, 13, 36