



SOS CONSULTANTS

ELABORATION SPEC

UNIVERSITY OF LOUISVILLE COLLEGE OF BUSINESS
DR. ROBERT BARKER, PROFESSOR OF SYSTEMS ANALYSIS

PATRICK WARREN, ALEXA BEARDEN, BRAD RIGGS,
ERIC CHEN, AND CAILYN COCHRAN

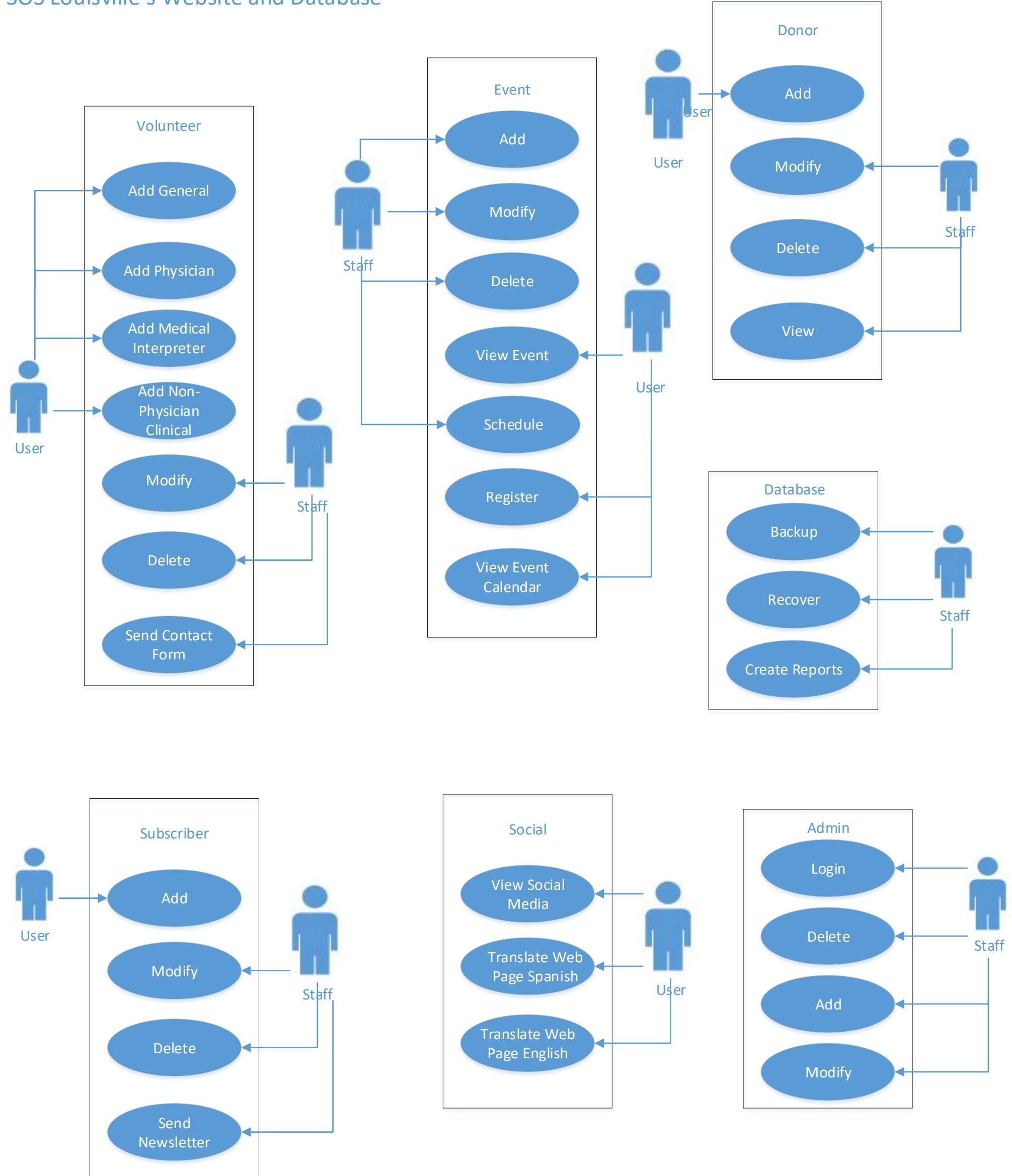
System Requirements

The System Requirements sections gives an hierarchical overview of functional and nonfunctional requirements that this system will rely on. Sections 1 takes in the requirements of the system as a hole. Section 2 deals with the donations and donation portal using PayPal, recurring payments, and one-time payments. Section 3 considers the volunteer form, how the volunteers will fill out their information, and where it will be stored. Section 4 allows for a language translator for the entire website. Language translator will translate to at least Spanish. Section 5 requires the site to have social media links on all web pages. Sections 6 requires that a full calendar be displayed under one webpage.

1.0	Website shall be maintained using WordPress as it's already built out.
1.1	Database shall be maintained using MySQL and will be able to add, modify, and delete volunteers and donors.
1.2	System as a whole shall be maintained using Microsoft Azure.
2.0	Website shall have a donation portal.
2.1	Website shall continue using PayPal.
2.2	Donation portal shall give the user the ability to set up recurring payments or perform a one time donation.
2.3	Donation data shall be gathered using Azure.
3.0	Website shall have a volunteer form.
3.1	Volunteer form shall be changed based on the type of volunteer.
3.2	Volunteer form data shall be maintained by a MySQL database in Microsoft Azure.
3.3	Volunteers shall be added, modified, and deleted.
4.0	Website shall have a language translator.
4.1	Language translator shall be able to interpret to at least Spanish.
5.0	Social media links shall be placed on all web pages.
6.0	An events page shall house a full year calendar.

6.1	The calendar shall store information for organizational events.
6.2	Events can be created, modified, and deleted through Azure.

SOS Louisville's Website and Database



This trace matrix associates the below use cases with my team's system requirements.

FULL USE CASE DESCRIPTIONS

Use Case Specification: ADD GENERAL VOLUNTEER

1. Use-Case Name – Add General Volunteer

1.1 Brief Description

This use case explains how a user will fill out a general volunteer form

2. Flow of Events

2.1 Basic Flow

1. User selects general volunteer form.
2. System loads general volunteer form.
3. User types in first name.
4. User types in last name.
5. User types in city.
6. User types in state.
7. User types in zip code.
8. User types in phone number.
9. User types in date of birth.
10. User types in email.
11. User uploads copy of driver's license.
12. User uploads copy of immunization records.
13. User uploads verification of TB Test.
14. User selects general volunteer type – (medical student, undergraduate student, or general volunteer).
15. User selects interest area.
16. User selects submit general volunteer form.
 - 7.1. System submits completed general volunteer form.
 - 7.2. System asks for all required fields to be filled out.
 - 7.3. System asks for valid email address.
17. System stores all input to database to await approval of SOS Staff.
18. System sends confirmation that general volunteer form has been received.
19. Use case ends.

2.2 Alternative Flows

None

3. Special Requirements

None.

4. Pre-conditions

None.

5. Post-conditions

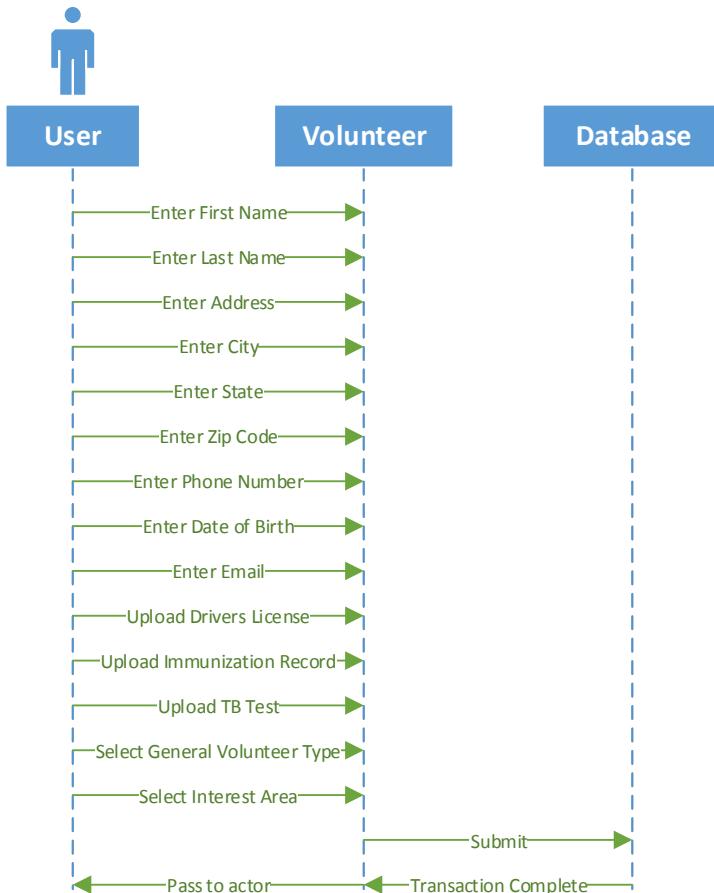
5.1 Approval Pending

System will be pending for approval by staff for general volunteer form.

6. Extension Points

None.

General Volunteer – Sequence Diagram



Use Case Specification: ADD PHYSICIAN VOLUNTEER

1. Use-Case Name – Add Physician Volunteer

1.1 Brief Description

This use case explains how a user will fill out a physician volunteer form

2. Flow of Events

2.1 Basic Flow

1. User selects physician volunteer form.
2. System loads physician volunteer form.
3. User types in first name.
4. User types in last name.
5. User types in city.
6. User types in state.
7. User types in zip code.
8. User types in phone number.
9. User types in date of birth.
10. User types in email.
11. User uploads copy of driver's license.
12. User uploads copy of immunization records.
13. User uploads verification of TB Test.
14. User enters medical license number.
15. User enters practice/medical specialty.
16. User enters current employer/practice.
17. User enters medical centers with privileges to practice.
18. User enters states licensed to practice medicine in.
19. User selects submit physician volunteer form.
 - 7.1. System submits completed physician volunteer form.
 - 7.2. System asks for all required fields to be filled out.
 - 7.3. System asks for valid email address.
20. System stores information to database to await approval of SOS Staff.
21. System sends confirmation that physician volunteer form has been received.
22. Use case ends.

2.2 Alternative Flows

None.

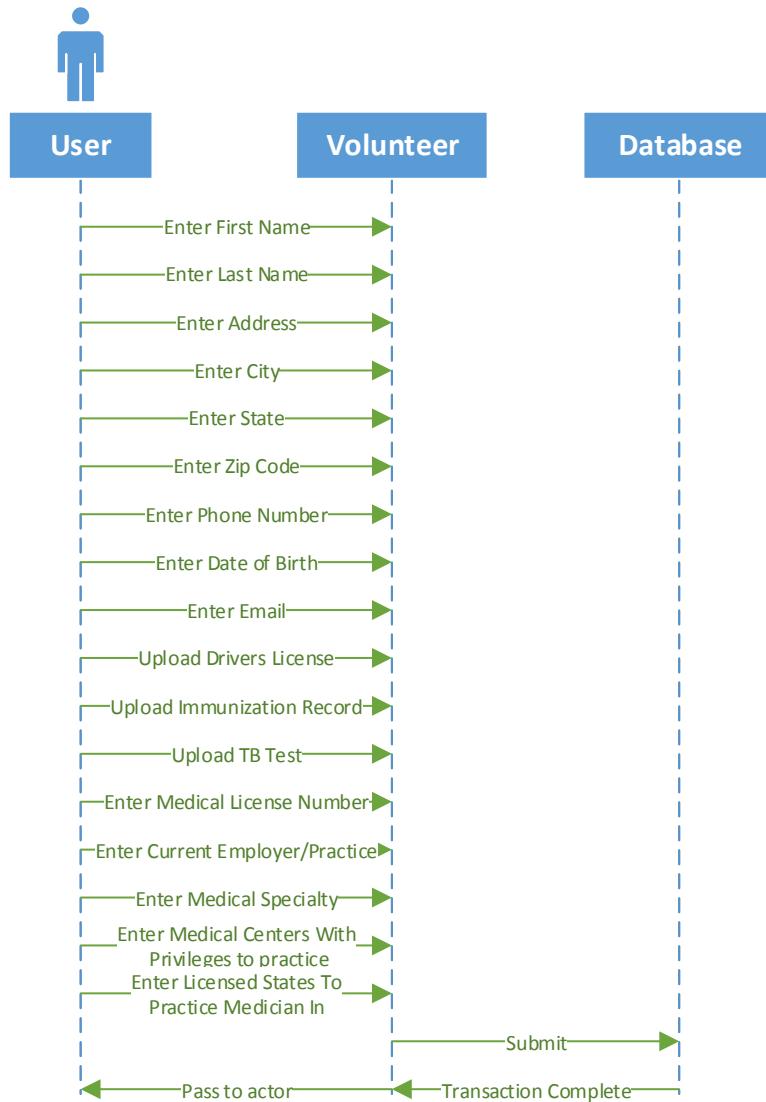
3. Special Requirements

Volunteer must have a medical license number, practice specialty, employer, licensed in one or more states, and privileges to practice in one or more medical centers.

4. Pre-conditions

None.

Physician Volunteer – Sequence Diagram



Use Case Specification: ADD MEDICAL INTERPRETER VOLUNTEER

1. Use-Case Name – Add Medical Interpreter Volunteer

1.1 Brief Description

This use case explains how a user will fill out a medical interpreter volunteer form

2. Flow of Events

2.1 Basic Flow

1. User selects medical interpreter volunteer form.
2. System loads medical interpreter volunteer form.
3. User types in first name.
4. User types in last name.
5. User types in city.
6. User types in state.
7. User types in zip code.
8. User types in phone number.
9. User types in date of birth.
10. User types in email.
11. User uploads copy of driver's license.
12. User uploads copy of immunization records.
13. User uploads verification of TB Test
14. User selects fluent languages
15. User uploads interpreter credentials
16. User selects submit medical interpreter volunteer form.
 - 7.1. System submits completed medical interpreter volunteer form.
 - 7.2. System asks for all required fields to be filled out.
 - 7.3. System asks for valid email address.
17. System stores information to database to await approval of SOS Staff.
18. System sends confirmation that medical interpreter volunteer form has been received.
19. Use case ends.

2.2 Alternative Flows

None.

3. Special Requirements

Volunteer must be fluent in foreign language and have credentials.

4. Pre-conditions

None.

5. Post-conditions

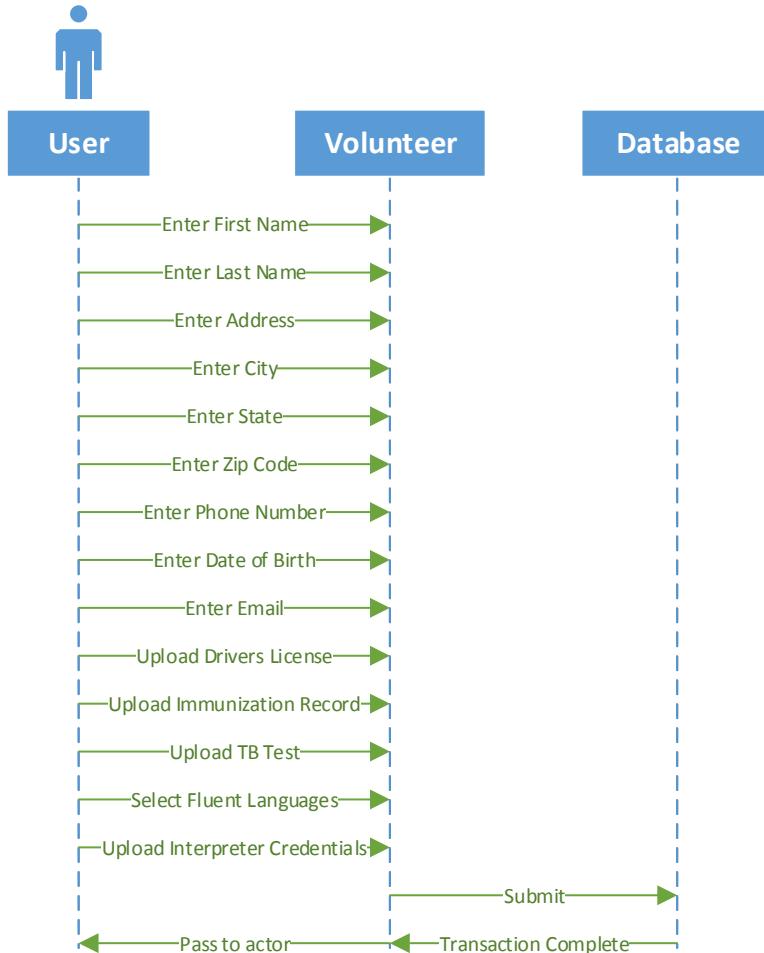
5.1 Approval Pending

System will be pending for approval by staff for medical interpreter volunteer form.

6. Extension Points

None.

Medical Interpreter Volunteer – Sequence Diagram



Use Case Specification: ADD NON-PHYSICIAN CLINICAL VOLUNTEER

1. Use-Case Name – Add Non-Physician Clinical Volunteer

1.1 Brief Description

This use case explains how a user will fill out a non-physician clinical volunteer form

2. Flow of Events

2.1 Basic Flow

1. User selects non-physician clinical volunteer form.
2. System loads non-physician clinical volunteer form.
3. User types in first name.
4. User types in last name.
5. User types in city.
6. User types in state.
7. User types in zip code.
8. User types in phone number.
9. User types in date of birth.
10. User types in email.
11. User uploads copy of driver's license.
12. User uploads copy of immunization records.
13. User uploads verification of TB Test.
14. User selects professional title
15. User states current place of employment
16. User selects submit non-physician clinical volunteer form.
 - 7.1. System submits completed non-physician clinical volunteer form.
 - 7.2. System asks for all required fields to be filled out.
 - 7.3. System asks for valid email address.
17. System stores information to database to await approval of SOS Staff.
18. System sends confirmation that non-physician clinical volunteer form has been received.
19. Use case ends.

2.2 Alternative Flows

None.

3. Special Requirements

Volunteer must have a professional title and current place of employment.

4. Pre-conditions

None.

5. Post-conditions

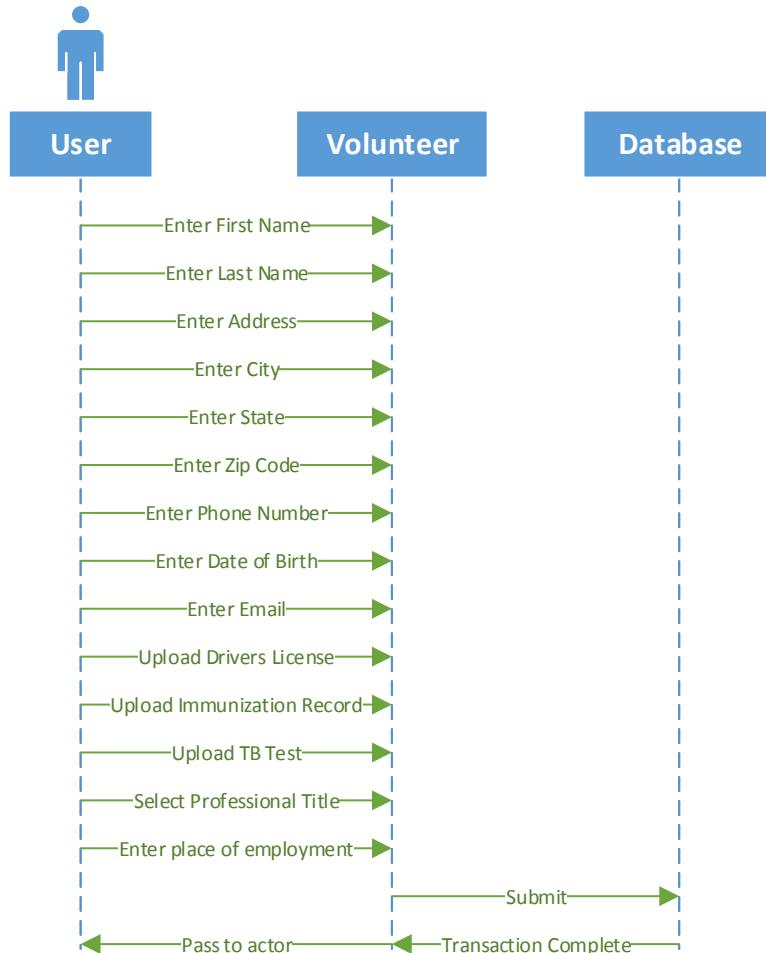
5.1 Approval Pending

System will be pending for approval by staff for non-physician clinical volunteer form.

6. Extension Points

None.

Non-Physician Clinical Volunteer – Sequence Diagram



Use Case Specification: MODIFY VOLUNTEER

1. Use-Case Name – Modify Volunteer

1.1 Brief Description

This use case explains how staff member will modify a volunteer.

2. Flow of Events

2.1 Basic Flow

1. Staff selects Modify Volunteer.
2. System loads page with all of specified volunteer's submitted information and credentials (Contact Info, Credentials, volunteer type – physician, non-physician clinical, medical interpreter, general).
3. Staff makes desired changes to contact info, credentials, and/or volunteer type.
4. Staff selects saves changes.
5. System updates volunteer with specified modifications.
6. Use case ends.

2.2 Alternative Flows

3. Special Requirements

3.1 Modify Privileges

Staff must have delete privilege.

4. Pre-conditions

4.1 Volunteer must exist

A volunteer record must have been created in order for it to be modified.

5. Post-conditions

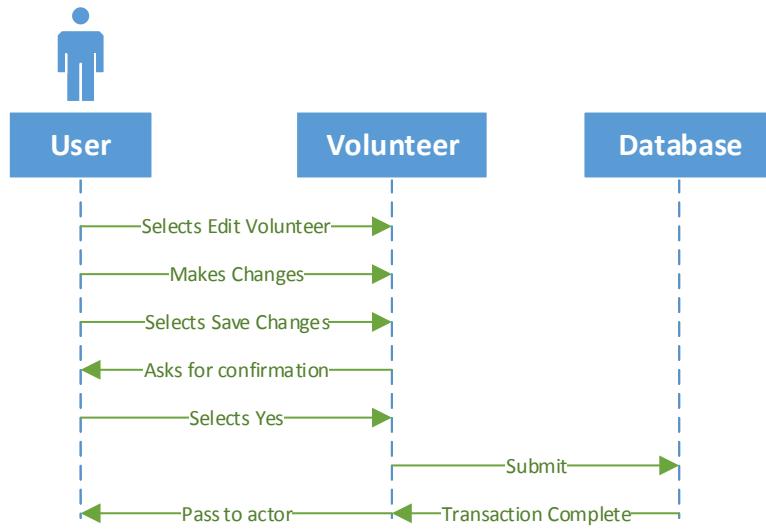
5.1 Volunteer Modified

Volunteer information has been changed/updated.

6. Extension Points

None.

Modify Volunteer – Sequence Diagram



Use Case Specification: DELETE VOLUNTEER

1. Use-Case Name – Delete Volunteer

1.1 Brief Description

This use case explains how staff member will delete a volunteer.

2. Flow of Events

2.1 Basic Flow

1. Staff selects Delete Volunteer
2. System prompts message “Are you sure you want to delete volunteer? All volunteer information will be permanently deleted.”
3. User selects Yes or No
 - 3.1. User selects Yes and system permanently deletes volunteer and all associated information from the database.
 - 3.2. User selects No and the system cancels the request to delete volunteer
4. UseCase Ends

2.2 Alternative Flows

3. Special Requirements

3.1 Delete Privileges

Staff must have delete privilege.

4. Pre-conditions

4.1 Volunteer must exist

A volunteer record must have been created in order for it to be deleted.

5. Post-conditions

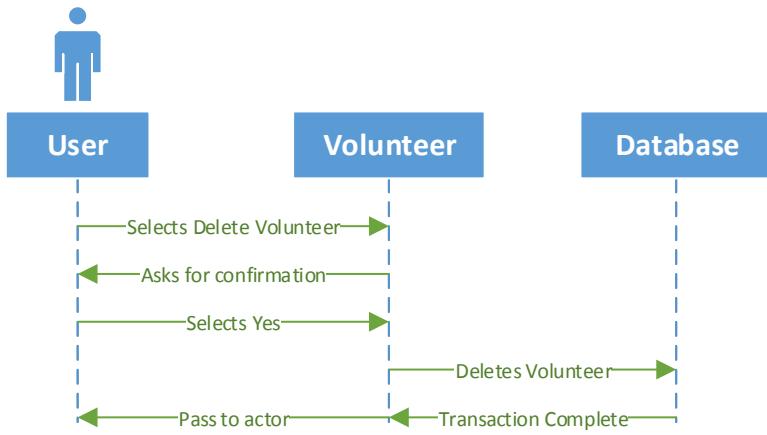
5.1 Volunteer Deleted

Volunteer has been removed from database

6. Extension Points

None.

Delete Volunteer – Sequence Diagram



Use Case Specification: SEND CONTACT FORM

1. Use-Case Name – Send Contact Form

1.1 Brief Description

This use case explains how a user will fill out a contact form.

2. Flow of Events

2.1 Basic Flow

1. User selects contact form on website.
2. System loads contact form.
3. User types in first name.
4. User types in last name.
5. User types in city.
6. User types in state.
7. User types in zip code.
8. User types in phone number.
9. User types in email.
10. User types in comment.
11. User submits contact form.
12. System sends contact form to staff email.
13. Use case ends.

2.2 Alternative Flows

3. Special Requirements

None.

4. Pre-conditions

None.

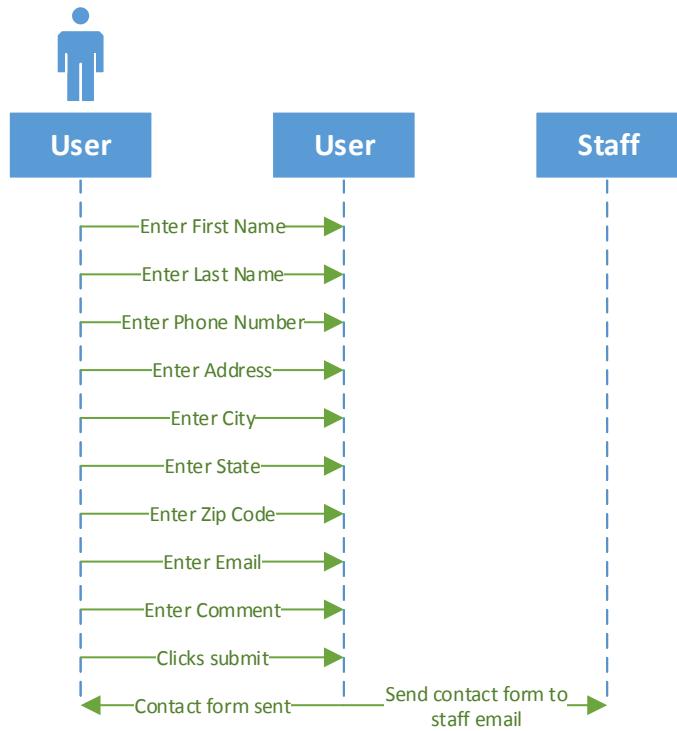
5. Post-conditions

None

6. Extension Points

None.

Send Contact Form – Sequence Diagram



Use Case Specification: Add Donor

1. Use-Case Name: Add Donor

1.1 Brief Description

This use case will allow the actor or user to donate to Surgery on Sunday (SOS) Louisville. Once the user has accessed the SOS Louisville webpage they will click the “DONATE” link. This link will then take the user to a secured PayPal site. Once in PayPal the user will enter in the amount they wish to donate. Next, the user will enter their information. Lastly, once the user has finished entering in their information they will click “Donate Now” to process their donation.

2. Flow of Events

2.1 Basic Flow

1. User clicks the “DONATE” link on homepage
2. User enters in a USD amount
3. User selects a method to donate
4. User enters in the required information based on the method they selected
5. User has the option to add special instructions to the seller
6. User clicks the “Donate Now” button

2.2 Alternative Flows

2.2.1 Donate with PayPal

1. User clicks the “DONATE” link on homepage
2. User enters in a USD amount
3. User clicks the “Donate with PayPal” button
4. User enters their email for PayPal
5. User enters their password for PayPal
6. User clicks the “Log In” button
7. User has the option to enter special instructions to the seller
8. User clicks the “Donate Now” button

2.2.2 Onetime PayPal Payment

1. User clicks the “DONATE” link on homepage
2. User enters in a USD amount
3. User does not check the “Make this a monthly donation” box
4. User clicks the “Donate with PayPal” button
5. User enters their email for PayPal
6. User enters their password for PayPal
7. User clicks the “Log In” button
8. User has the option to add special instructions to the seller
9. User clicks the “Donate Now” button

2.2.3 Monthly PayPal Payment

1. User clicks the “DONATE” link on homepage
2. User enters in a USD amount
3. User checks the “Make this a monthly donation” box
4. User clicks the “Donate with PayPal” button
5. User enters their email for PayPal
6. User enters their password for PayPal
7. User clicks the “Log In” button
8. User has the option to enter special instructions to the seller
9. User clicks the “Donate Now” button

2.2.4 Donate with Debit or Credit Card

1. User clicks the “DONATE” link on homepage
2. User enters in a USD amount
3. User does not check the “Make this a monthly donation” box
3. User clicks the “Donate with Debit or Credit Card” button
4. User selects their Country
5. User enters their card Number
6. User enters their card’s Expiration Date
7. User enters their card’s Security Code
8. User enters their First Name
9. User enters their Last Name
10. User enters their Street Address
11. User has the option to add a second line to their Street Address
12. User enters their City
13. User enters their State
14. User enters their ZIP code
15. User selects which type of phone number they are going to provide
16. User enters in their Phone Number
17. User enters their Email
18. User has the option to check a “Save this information for next time.” box
19. User has the option to enter special instructions to the seller
20. User clicks the “Donate Now” button

3. Special Requirements

3.1 USD Amount

The entered USD amount must be greater than or equal to \$1.00.

3.2 Valid Debit or Credit Card

The entered Debit or Credit Card must not be expired.

3.3 Valid Email

The entered email must be an active account.

4. Pre-conditions

None

5. Post-conditions

5.1 Added Donor

The donor’s information has been automatically added into the Donor Table in the database.

5.2 Successful Donation

The donation has been made to SOS Louisville.

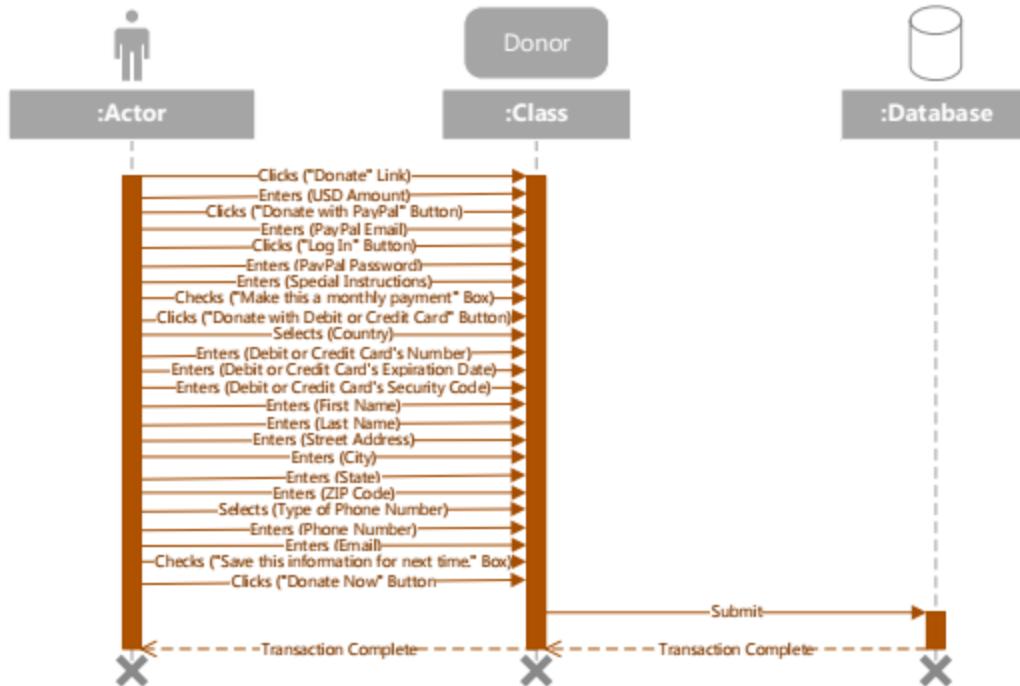
5.3 Confirmation

After a donation has been made a confirmation email will be sent to the user.

6. Extension Points

None

Add Donor – Sequence Diagram



Use Case Specification: Modify Donor

1. Use-Case Name: Modify Donor

1.1 Brief Description

This use case will allow the actor or user to modify a previous donor of Surgery on Sunday (SOS) Louisville. Once the user has accessed the Azure webpage they will click the “Login” link. This link will then prompt the user to enter in their account information. Once the user has successfully logged in they will have access to the database storing the donor information. Next, the user will find the donor they wish to modify. Lastly, once the user finds the donor they will be able to modify the selected donor’s information.

2. Flow of Events

2.1 Basic Flow

1. User will click login once they have access the Azure webpage
2. User enters user name for Azure webpage
3. User enters password for Azure webpage
4. User click the “Login” Button
5. User enters the Email of the Donor they wish to modify
6. User click the “Generate Report” button
7. User selects the Donor they wish to Modify from the report
8. User clicks the “Edit” button
9. User can modify the Donor’s First Name
10. User can modify the Donor’s Last Name
11. User can modify the Donor’s Email
12. User click the “Save” button
13. User selects “Logout”

2.2 Alternative Flows

2.2.1 Search with First Name

1. User will click login once they have access the Azure webpage
2. User enters user name for Azure webpage
3. User enters password for Azure webpage
4. User click the “Login” Button
5. User enters the First Name of the Donor they wish to modify
6. User click the “Generate Report” button
7. User selects the Donor they wish to Modify from the report
8. User clicks the “Edit” button
9. User can modify the Donor’s First Name
10. User can modify the Donor’s Last Name
11. User can modify the Donor’s Email
12. User click the “Save” button
13. User selects “Logout”

2.2.2 Search with Last Name

1. User will click login once they have access the Azure webpage
2. User enters user name for Azure webpage
3. User enters password for Azure webpage
4. User click the “Login” Button
5. User enters the Last Name of the Donor they wish to modify
6. User click the “Generate Report” button

7. User selects the Donor they wish to Modify from the report
8. User clicks the “Edit” button
9. User can modify the Donor’s First Name
10. User can modify the Donor’s Last Name
11. User can modify the Donor’s Email
12. User click the “Save” button
13. User selects “Logout”

3. Special Requirements

3.1 Simultaneous Login

User cannot login if already logged in.

4. Pre-conditions

4.1 Existing Donor

The donor being modified must have donated in the past.

5. Post-conditions

5.1 Modified Donor

The donor’s information has been modified.

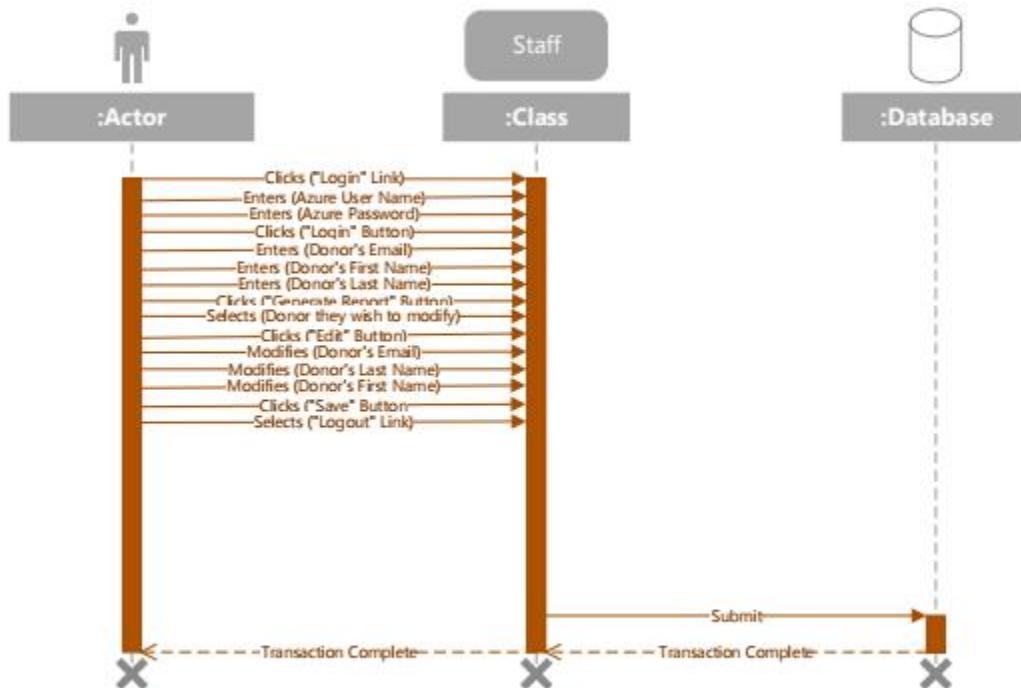
5.2 Confirmation

After the donor’s information has been modified a confirmation email will be sent to the user.

6 Extension Points

None

Modify Donor – Sequence Diagram



Use Case Specification: Delete Donor

1. Use-Case Name: Delete Donor

1.1 Brief Description

This use case will allow the actor or user to delete a previous donor of Surgery on Sunday (SOS) Louisville. Once the user has accessed the Azure webpage they will click the “Login” link. This link will then prompt the user to enter in their account information. Once the user has successfully logged in they will have access to the database storing the donor information. Next, the user will find the donor they wish to delete. Lastly, once the user finds the donor they will be able to delete the donor.

2. Flow of Events

2.1 Basic Flow

1. User will click login once they have access the Azure webpage
2. User enters user name for Azure webpage
3. User enters password for Azure webpage
4. User click the “Login” Button
5. User enters the Email of the Donor they wish to delete
6. User click the “Generate Report” button
7. User selects the Donor they wish to delete from the report
8. User clicks the “Delete” button
9. User click the “Save” button
10. User selects “Logout”

2.2 Alternative Flows

2.2.1 First Name

1. User will click login once they have access the Azure webpage
2. User enters user name for Azure webpage
3. User enters password for Azure webpage
4. User click the “Login” Button
5. User enters the Last Name of the Donor they wish to delete
6. User click the “Generate Report” button
7. User selects the Donor they wish to delete from the report
8. User clicks the “Delete” button
9. User click the “Save” button
10. User selects “Logout”

2.2.2 Last Name

1. User will click login once they have access the Azure webpage
2. User enters user name for Azure webpage
3. User enters password for Azure webpage
4. User click the “Login” Button
5. User enters the Last Name of the Donor they wish to delete
6. User click the “Generate Report” button
7. User selects the Donor they wish to delete from the report
8. User clicks the “Delete” button
9. User click the “Save” button
10. User selects “Logout”

3. Special Requirements

3.1 Simultaneous Login

User cannot login if already logged in.

4. Pre-conditions

4.1 Existing Donor

The donor being deleted must have donated in the past.

5. Post-conditions

5.1 Deleted Donor

The donor has been deleted.

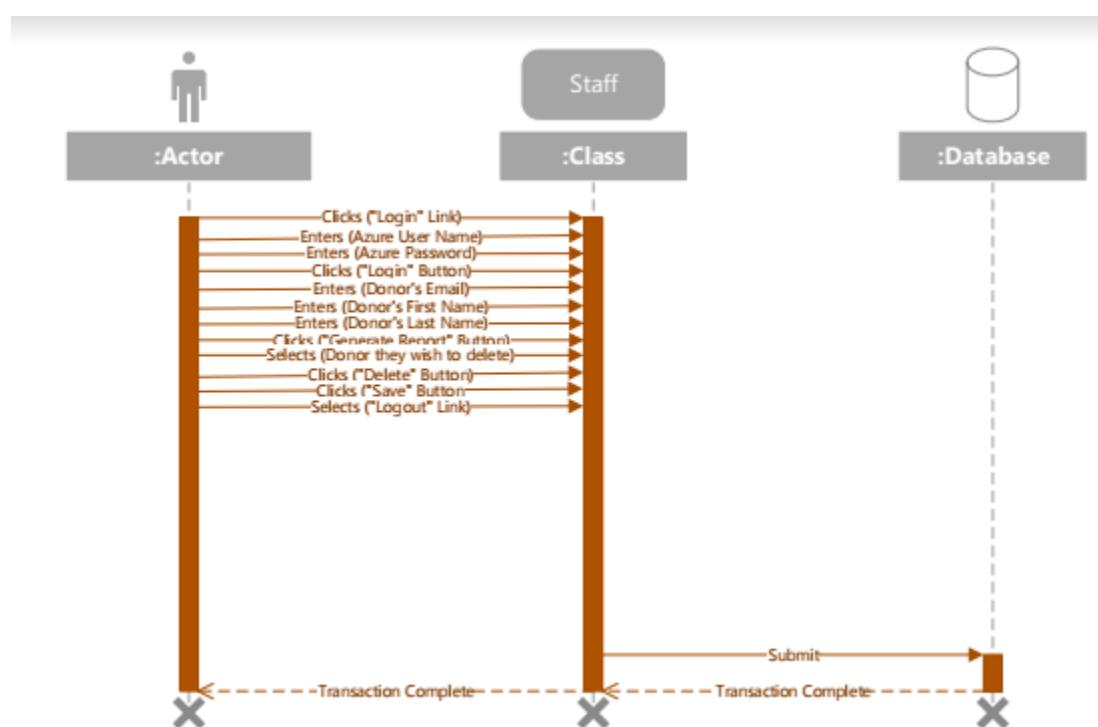
5.2 Confirmation

After the donor has been deleted a confirmation email will be sent to the user.

6. Extension Points

None

Delete Donor – Sequence Diagram



Use Case Specification: View Donation

1. Use-Case Name: View Donation

1.1 Brief Description

This use case will allow the actor or user to view a previous donation to Surgery on Sunday (SOS) Louisville. Once the user has accessed the Azure webpage they will click the “Login” link. This link will then prompt the user to enter in their account information. Once the user has successfully logged in they will have access to the database storing the donor information. Next, the user will find the donation they wish to view. Lastly, the user will view the donation.

2. Flow of Events

2.1 Basic Flow

1. User will click login once they have access the Azure webpage
2. User enters user name for Azure webpage
3. User enters password for Azure webpage
4. User click the “Login” Button
5. User enters the Email of the Donor who donated the donation they wish to view
6. User click the “Generate Report” button
7. User selects the Donor who donated the donation they wish to view
8. User views the donation
9. User selects “Logout”

2.2 Alternative Flows

2.2.1 First Name

1. User will click login once they have access the Azure webpage
2. User enters user name for Azure webpage
3. User enters password for Azure webpage
4. User click the “Login” Button
5. User enters the First Name of the Donor who donated the donation they wish to view
6. User click the “Generate Report” button
7. User selects the Donor who donated the donation they wish to view
8. User views the donation
9. User selects “Logout”

2.2.2 Email

1. User will click login once they have access the Azure webpage
2. User enters user name for Azure webpage
3. User enters password for Azure webpage
4. User click the “Login” Button
5. User enters the Last Name of the Donor who donated the donation they wish to view
6. User click the “Generate Report” button
7. User selects the Donor who donated the donation they wish to view
8. User views the donation
9. User selects “Logout”

3. Special Requirements

3.1 Simultaneous Login

User cannot login if already logged in.

4. Pre-conditions

4.1 Existing Donor

The donation being viewed must have been donated in the past.

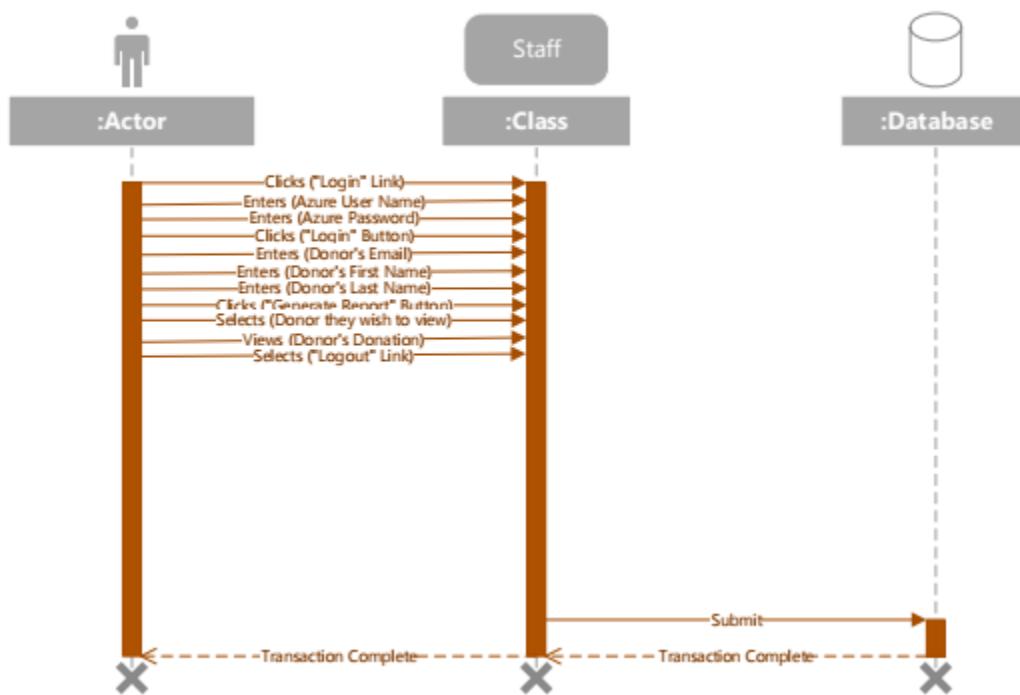
5. Post-conditions

None

6. Extension Points

None

View Donation – Sequence Diagram



Use Case Specification: Add Event

1. Use-Case Name: Add Event

1.1 Brief Description

This use case describes how a staff would add an event on the website.

2. Flow of Events

2.1 Basic Flow

1. Staff goes to www.soslouisville.org
2. Staff clicks on “Event” tab
3. Staff is on the event list and calendar page
4. Staff clicks on “Add event” button
5. Staff enters event time
6. Staff enters event date
7. Staff enters event location
8. Staff enters event name
9. Staff enters event descriptions
10. Staff clicks “Submit” to calendar and event list when finished
11. The system prints the event onto the calendar and event list
12. The system stores the information into database
13. Use case ends

2.2 Alternative Flows

1. Staff goes to www.soslouisville.org
2. Staff clicks “Event” tab
3. Staff is on the event list and calendar page
4. Staff clicks on “Add event” button
5. Staff enters event time
6. Staff enters event date
7. Staff enters event location
8. Staff enters event name
9. Staff enters event descriptions
10. Staff clicks “Cancel”
11. The system does not print the event onto the calendar and event list
12. The system does not store the information into database
13. Use case ends

3. Special Requirements

Event must be in the future.

4. Pre-conditions

The staff needs to have the permission to add event.

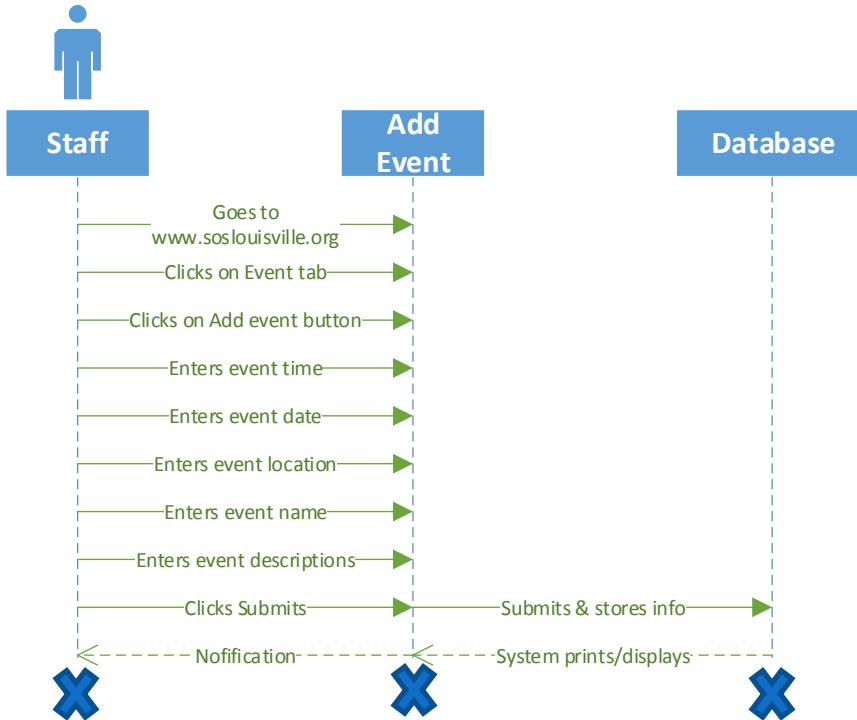
5. Post-conditions

The system posts the event onto the calendar and event list.
Issue notification that an event has been added.

6. Extension Points

None

Add Event – Sequence Diagram



Use Case Specification: Modify Event

1. Use-Case Name: Modify Event

1.1 Brief Description

This use case describes how a staff would modify an event on the website.

2. Flow of Events

2.1 Basic Flow

1. Staff goes to www.soslouisville.org
2. Staff clicks "Event" tab
3. Staff is on the event list and calendar page
4. Staff clicks on "Edit event" button
5. Staff edits event time
6. Staff edits event date
7. Staff edits event location
8. Staff edits event name
9. Staff edits event descriptions
10. Staff clicks "Submit" to calendar and event list when finished
11. The system prints the updated event onto the calendar and event list
12. The system stores the information into database
13. Use cases ends

2.2 Alternative Flows

1. Staff goes to www.soslouisville.org
2. Staff clicks "Event" tab
3. Staff is on the event list and calendar page
4. Staff clicks on "Edit event" button
5. Staff edits event time
6. Staff edits event date
7. Staff edits event location
8. Staff edits event name
9. Staff edits event descriptions
10. Staff clicks "Cancel"
11. The system does not print the updated event onto the calendar and event list
12. The system does not store the information into database
13. Use case ends

3. Special Requirements

Must have an existing event.

4. Pre-conditions

The staff needs to have the permission to edit event.

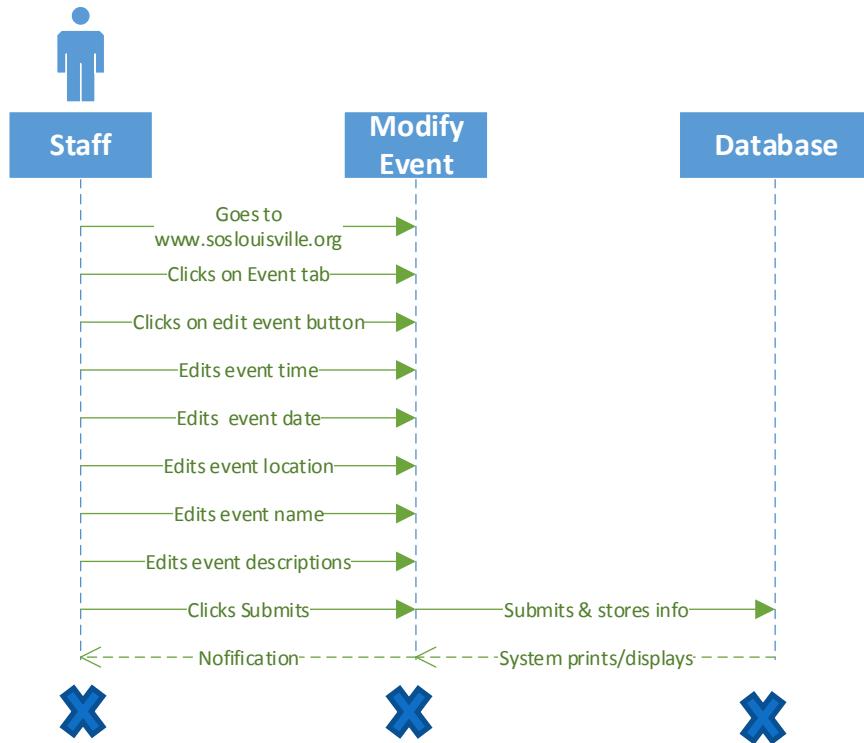
5. Post-conditions

The system posts the updated even onto the calendar and event list.
Issue notification that an event has been edited.

6. Extension Points

None

Modify Event – Sequence Diagram



Use Case Specification: Delete Event

1. Use-Case Name: Delete Event

1.1 Brief Description

This use case describes how a staff would delete an event on the website.

2. Flow of Events

2.1 Basic Flow

1. Staff goes to www.soslouisville.org
2. Staff clicks “Event” tab
3. Staff is on the event list and calendar page
4. Staff clicks on “Delete event” button
5. Staff clicks “Submit” to confirm
6. The system deletes the event off the calendar and event list
7. The system removes the information into database
8. Use cases ends

2.2 Alternative Flows

1. Staff goes to www.soslouisville.org
2. Staff clicks “Event” tab
3. Staff is on the event list and calendar page
4. Staff clicks on “Delete event” button
5. Staff clicks “Cancel” instead of “Submit”
6. The system does not delete the event
7. The system does not remove the information into database
8. Use case ends

3. Special Requirements

Must have an existing event.

4. Pre-conditions

The staff needs to have the permission to delete event.

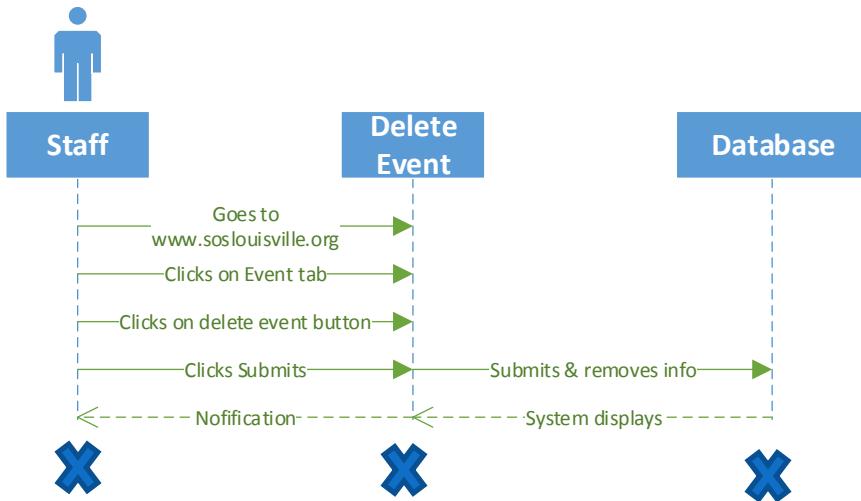
5. Post-conditions

The system removes the event off the calendar and event list.
Issue notification that an event has been deleted.

6. Extension Points

None

Delete Event – Sequence Diagram



Use Case Specification: Register for Event

1. Use-Case Name: Register for Event

1.1 Brief Description

This use case describes how a user would register for event.

2. Flow of Events

2.1 Basic Flow

1. User goes to www.soslouisville.org
2. User clicks on “Event” tab
3. User is on the event list and calendar page
4. User clicks “Register for event”
5. User select the event
6. User fills out a form
7. User enters name
8. User enters type of volunteer
9. User enters email
10. User enters phone number
11. User clicks “Submit” to confirm
12. The system displays a confirmation pop up that an event has been registered
13. The system stores the information into database
14. Use cases ends

2.2 Alternative Flows

1. User goes to www.soslouisville.org
2. User clicks on “Event” tab
3. User is on the event list and calendar page
4. User clicks “Register for event”
5. User select the event
6. User fills out a form
7. User enters name
8. User enters type of volunteer
9. User enters email
10. User enters phone number
11. User clicks “Cancel”
12. The system does not display a confirmation pop up that an event has been registered
13. The system does not store the information into database
14. Use cases ends

3. Special Requirements

Must have an existing event.

4. Pre-conditions

The user needs to be on the event page.

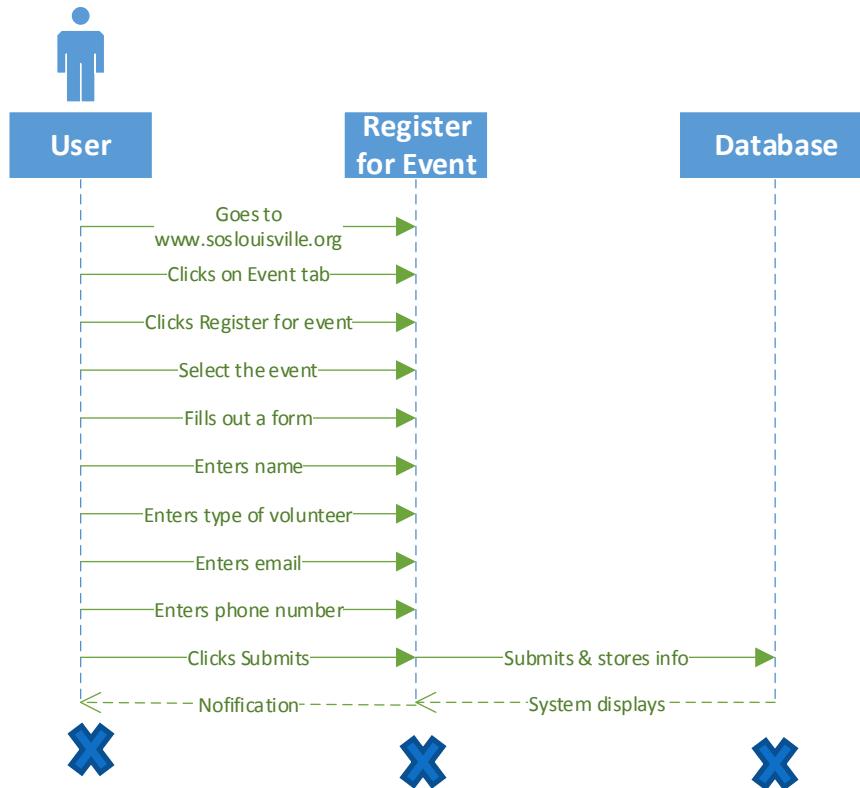
5. Post-conditions

The system displays user have registered for event.
Issue notification for the volunteers.

6. Extension Points

None

Register for Event – Sequence Diagram



Use Case Specification: View Event

1. Use-Case Name: View Event

1.1 Brief Description

This use case describes how a staff and user would view an event on the website.

2. Flow of Events

2.1 Basic Flow

1. Staff/user goes to www.soslouisville.org
2. Staff/user clicks on “Event” tab
3. Staff/user is on the event list and calendar page
4. Staff/user clicks “Event list” for more details
5. The system displays the event list in detail
6. Use cases ends

2.2 Alternative Flows

1. Staff/user goes to www.soslouisville.org
2. Staff/user clicks on “Event” tab
3. Staff/user is on the event list and calendar page
4. Staff/user does not click on the event list
5. The system does not display the event list in detail
6. Use case ends

3. Special Requirements

Must have an existing event.

4. Pre-conditions

The staff/user needs to be on the event page.

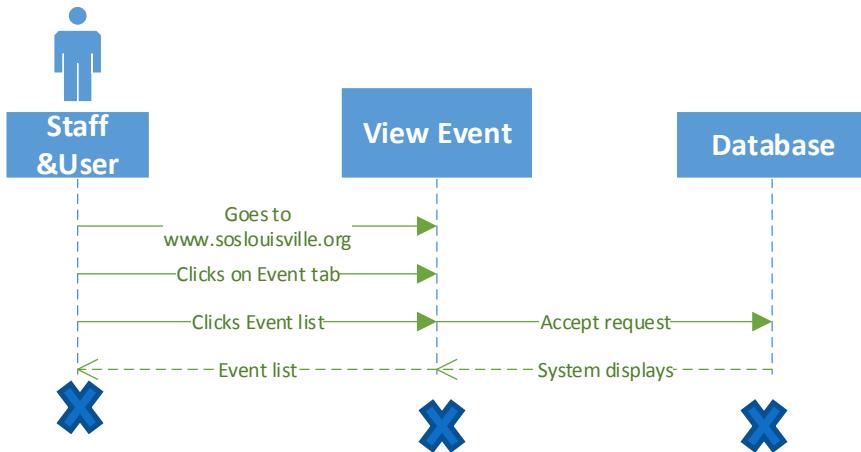
5. Post-conditions

The system displays the event list in detail.
Setup notification for an event.

6. Extension Points

None

View Event – Sequence Diagram



Use Case Specification: View Event Calendar

1. Use-Case Name: View Event Calendar

1.1 Brief Description

This use case describes how a staff and user would view an event calendar on the website.

2. Flow of Events

2.1 Basic Flow

1. Staff/user goes to www.soslouisville.org
2. Staff/user clicks on “Event” tab
3. Staff/user is on the event list and calendar page
4. Staff/user clicks “Event calendar” for more details
5. The system displays the event calendar in detail
6. Use cases ends

2.2 Alternative Flows

1. Staff/user goes to www.soslouisville.org
2. Staff/user clicks on “Event” tab
3. Staff/user is on the event list and calendar page
4. Staff/user does not click on the event calendar
5. The system does not display the event calendar in detail
6. Use case ends

3. Special Requirements

Must have an existing event.

4. Pre-conditions

The staff/user needs to be on the event page.

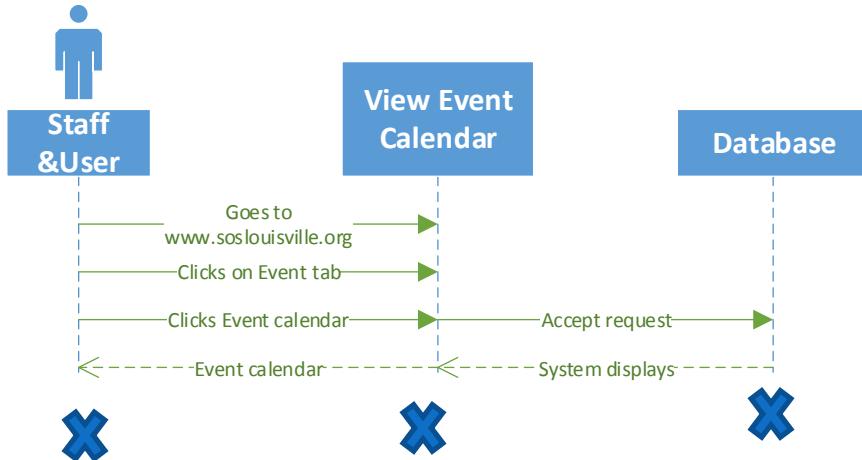
5. Post-conditions

The system displays the event calendar in detail.
Setup notification for an event.

6. Extension Points

None

View Event Calendar – Sequence Diagram



Use Case Specification: Schedule Volunteer for Event

1. Use-Case Name: Schedule Volunteer for Event

1.1 Brief Description

This use case describes how a staff would schedule volunteer for event.

2. Flow of Events

2.1 Basic Flow

1. Staff goes to www.soslouisville.org
2. Staff clicks on “Event” tab
3. Staff is on the event list and calendar page
4. Staff clicks “Schedule event”
5. Staff enters volunteer’s email
6. Staff clicks “Submit” to confirm
7. The system displays the number of volunteers for an event
8. The system stores the information into database
9. Use cases ends

2.2 Alternative Flows

1. Staff goes to www.soslouisville.org
2. Staff clicks on “Event” tab
3. Staff is on the event list and calendar page
4. Staff clicks “Schedule event”
5. Staff enters volunteer’s email
6. Staff clicks “Cancel”
7. The system does not display the number of volunteers for an event
8. The system does not store the information into database
9. Use case ends

3. Special Requirements

Must have an existing event.

4. Pre-conditions

The staff needs to be on the event page and have permission to schedule.

5. Post-conditions

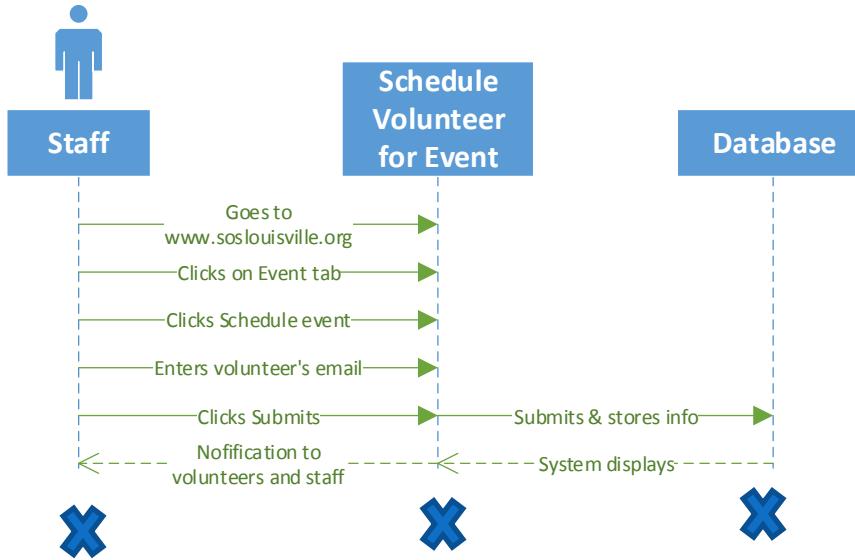
The system displays the number of volunteers for an event.

Issue notification for the volunteers and they will have to fill out information

6. Extension Points

None

Schedule Volunteer for Event – Sequence Diagram



Use Case Specification: View Social Media

1. Use-Case Name: View Social Media

1.1 Brief Description

This use case will describe how a visitor would view Surgery on Sunday Louisville's social media platforms.

2. Flow of Events

2.1 Basic Flow

1. User goes to www.soslouisville.org
2. User clicks on one of three social media icon links: "Facebook", "Twitter", or "Google Plus"
3. User is redirected to the selected social media page
4. User enters log in credentials for selected social media platform
5. User is directed to SOS Louisville's social media page
6. User views SOS Louisville's social media content
7. Use case ends

2.2 Alternative Flows

2.2.1 View Facebook Page

1. User goes to www.soslouisville.org
2. User clicks on "Facebook" icon on homepage
3. User is redirected to Facebook.
4. User enters email
5. User enters password
6. User selects "Enter"
7. User is directed to SOS Louisville's Facebook page
8. User views SOS Louisville's content
9. Use case ends

2.2.2 View Twitter Page

1. User goes to www.soslouisville.org
2. User clicks on "Twitter" icon on homepage
3. User is redirected to Twitter.
4. User enters username
5. User enters password
6. User selects "Enter"
7. User is directed to SOS Louisville's Twitter page
8. User views SOS Louisville's content
9. Use case ends

2.2.3 View Google Plus Page

1. User goes to www.soslouisville.org
2. User clicks on "Google Plus" icon on homepage
3. User is redirected to Google Plus.
4. User enters email
5. User enters password
6. User selects "Enter"
7. User is directed to SOS Louisville's page
8. User views SOS Louisville's content
9. Use case ends

3. Special Requirements

None

4. Pre-conditions

4.1 Facebook Account

The user must have a Facebook account to post or share SOS Louisville's content.

4.2 Twitter Account

The user must have a Twitter account to post or share SOS Louisville's content.

4.3 Google Plus Account

The user must have a Google Plus account to post or share SOS Louisville's content.

5. Post-conditions

5.1 View Social Media Pages

User is directed to SOS Louisville's social media pages.

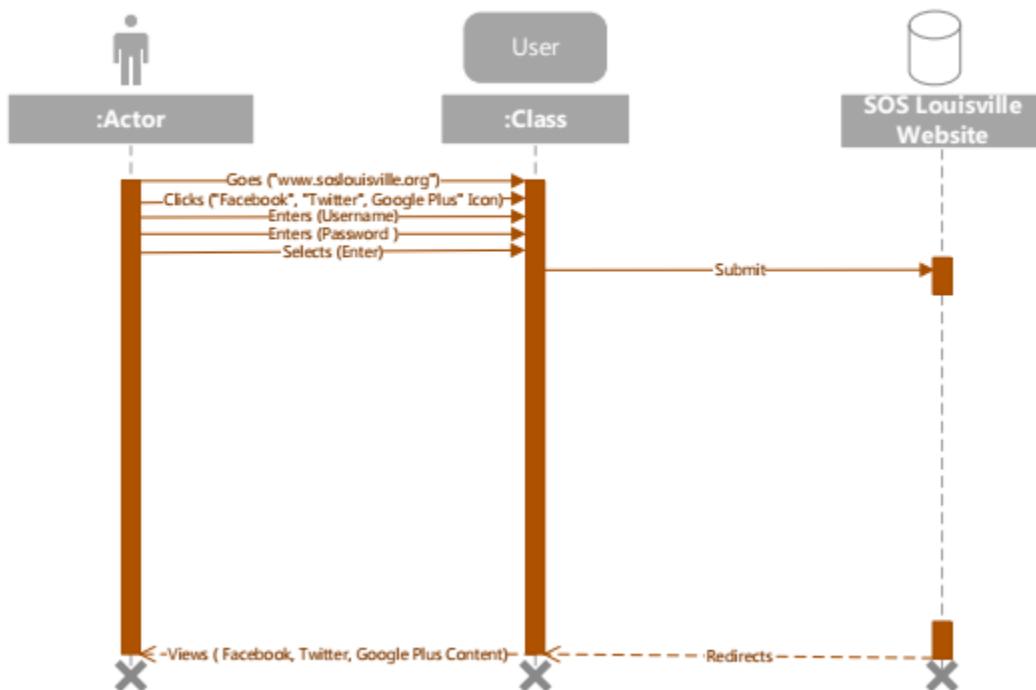
5.2 Notification

SOS Louisville will receive a notification when a user likes their social media page

6 Extension Points

None

View Social Media – Sequence Diagram



Use Case Specification: Translate Web Page to English

1. Use-Case Name: Translate Web Page to English

1.1 Brief Description

This use case will describe how a user will translate Surgery on Sunday's page from Spanish to English.

2. Flow of Events

2.1 Basic Flow

1. User goes to www.soslouisville.org
2. User locates language dropdown bar on the homepage
3. User clicks "Select Language" button
4. User selects "English" option
5. User views page in English
6. Use case ends

3. Special Requirements

3.1 User is an English speaker.

4. Pre-conditions

4.1 View website in Spanish

Website language is in Spanish

5. Post-conditions

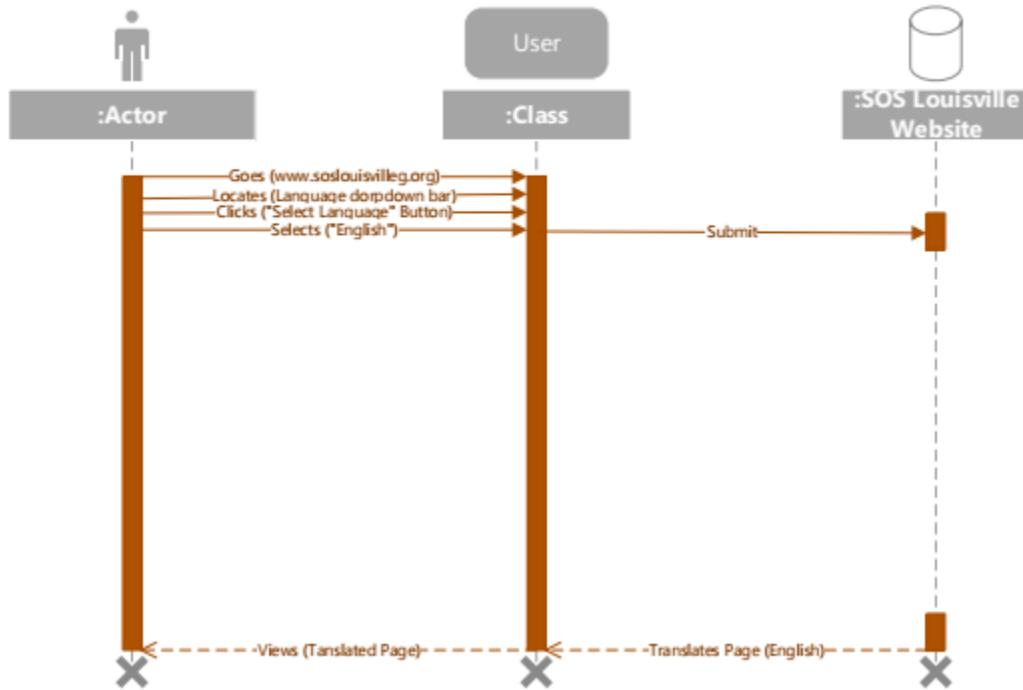
5.1 View website in English

Website is translated to English

6. Extension Points

None

Translate Web Page to English – Sequence Diagram



Use Case Specification: Translate Web Page to Spanish

1. Use-Case Name: Translate Web Page to Spanish

1.1 Brief Description

This use case will describe how a user will translate Surgery on Sunday's page from English to Spanish.

2. Flow of Events

2.1 Basic Flow

1. User goes to www.soslouisville.org
2. User locates language dropdown bar on the homepage
3. User clicks "Select Language" button
4. User selects "Spanish" option
5. User views page in Spanish.
6. Use case ends

3. Special Requirements

3.1 User is a Spanish speaker.

4. Pre-conditions

4.1 View website in English

Website language is in English

5. Post-conditions

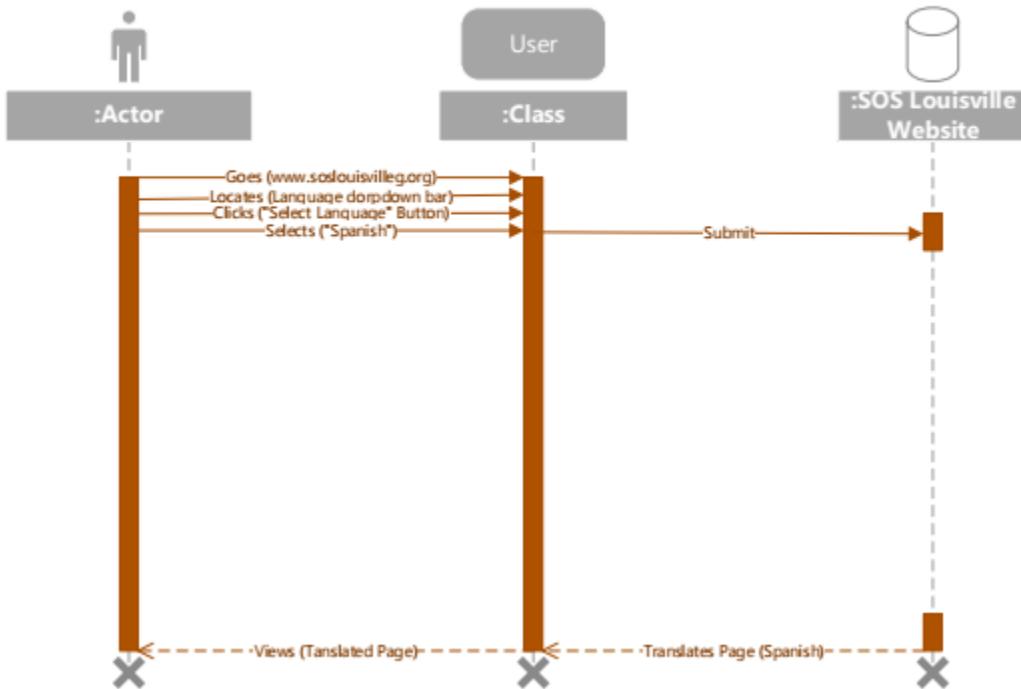
5.1 View website in Spanish

Website is translated to Spanish

6. Extension Points

None

Translate Web Page to Spanish – Sequence Diagram



Use Case Specification: Subscribe to Email Newsletter

1. Use-Case Name: Subscribe to Email Newsletter

1.1 Brief Description

This use case will describe how a user will subscribe to SOS Louisville's newsletter.

2. Flow of Events

2.1 Basic Flow

1. User goes to www.soslouisville.org
2. User locates "Newsletter" form on homepage
3. User enters email into form
4. User enters name into form
5. User views option to "Subscribe" or "Unsubscribe"
6. User selects "Subscribe"
7. User receives subscription confirmation email.
8. Use case ends

2.2 Alternative Flows

2.2.1 Unsubscribe to Newsletter

1. User goes to www.soslouisville.org
2. User locates "Newsletter" form on homepage
3. User enters email into form
4. User enters name into form
5. User views option to "Subscribe" or "Unsubscribe"
6. User selects "Unsubscribe"
7. User receives un-subscription confirmation email.
8. Use case ends

3. Special Requirements

3.1 Valid Email

The email entered must be an active account.

3.2 Subscribe or Unsubscribe

The user must select "Subscribe" or "Unsubscribe"

3.3 No information left blank

No information in the form can be left blank

4. Pre-conditions

4.1 Not Subscribed

The user does not have a subscription to the newsletter.

4.2 Subscribed

The user is subscribed to the email newsletter.

5. Post-conditions

5.1 User is Subscribed

User receives subscription confirmation email from SOS Louisville

5.2 Newsletter

User will receive week/monthly newsletter from SOS Louisville

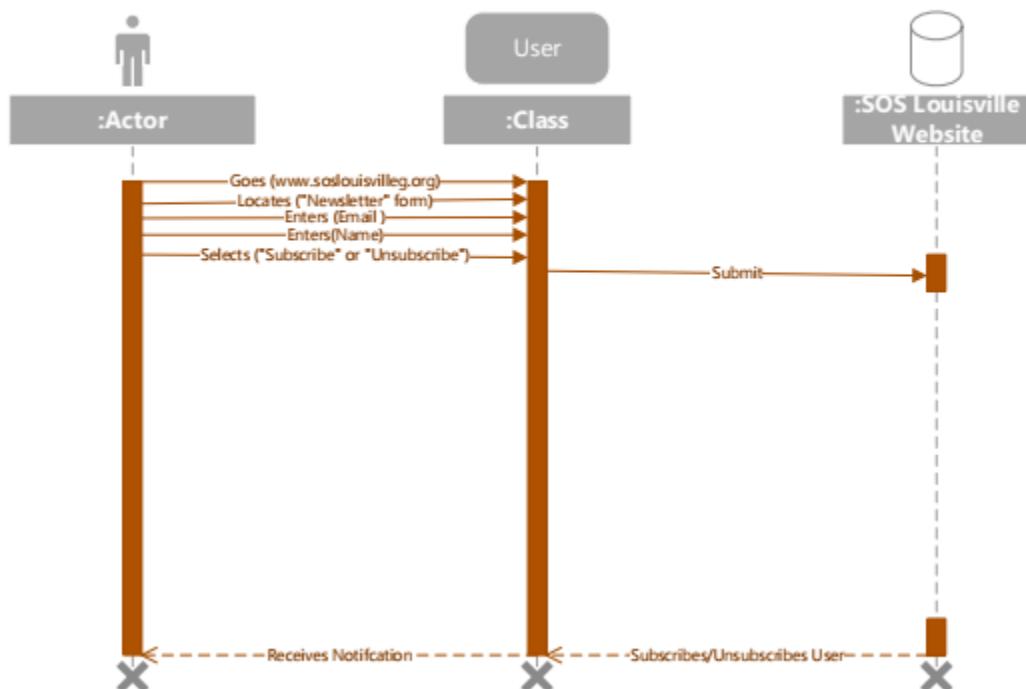
5.3 User is Unsubscribed

User receives un-subscription confirmation email from SOS Louisville

6. Extension Points

None

Subscribe to Email Newsletter – Sequence Diagram



Use Case Specification: Modify Subscriber

1. Use-Case Name: Modify Subscriber

1.1 Brief Description

This use case will describe how a staff member will modify a pre-existing subscriber in SOS Louisville's database.

2. Flow of Events

2.1 Basic Flow

1. Staff enters subscriber database
2. Staff locates subscriber
3. Staff modifies subscriber's email
4. Staff selects "save"
5. End of use case

2.2 Alternative Flows

2.2.1 Modify Subscriber's name

1. Staff enters subscriber database
2. Staff locates subscriber
3. Staff modifies subscriber's name
4. Staff selects "save"
5. End of use case

3. Special Requirements

3.1 Valid Email

The email entered must be an active account.

3.2 Subscriber's Name

Subscriber's name must be correctly associated with the email

4. Pre-conditions

4.1 Subscribed

The user is subscribed to the email newsletter.

5. Post-conditions

5.1 Modified Subscriber's Email

The new email address will receive the weekly/monthly subscription

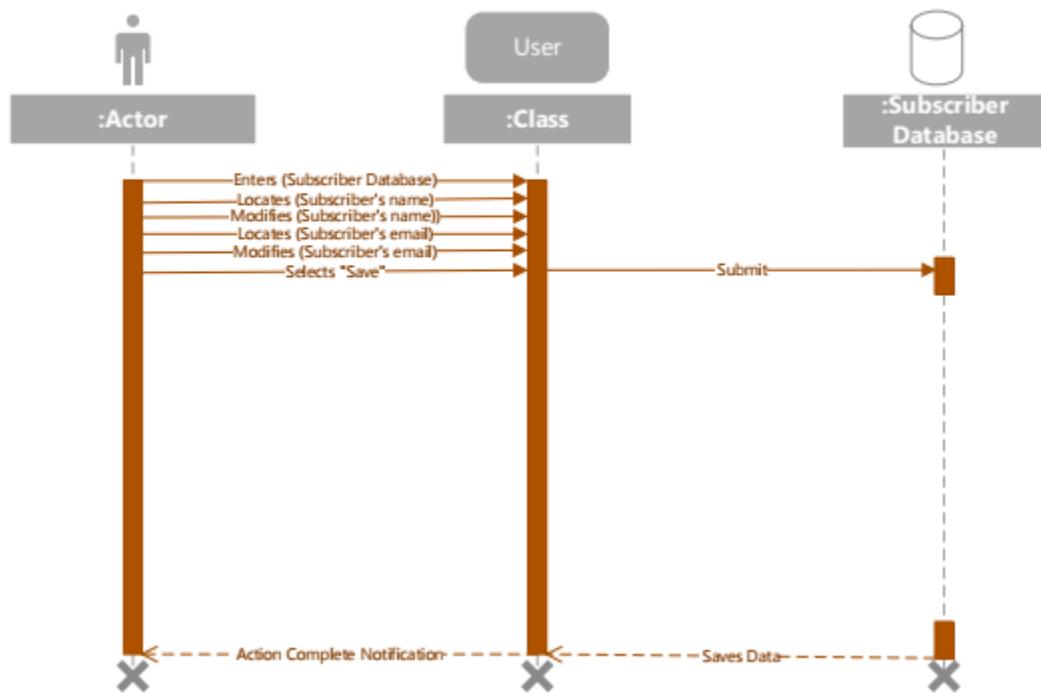
5.2 Modified Subscriber's Name

The weekly/monthly newsletter will be addressed to the modified subscriber name

6. Extension Points

None

Modify Subscriber – Sequence Diagram



Use Case Specification: Delete Subscriber

1. Use-Case Name: Delete Subscriber

1.1 Brief Description

This use case will describe how a staff member will delete a pre-existing subscriber in SOS Louisville's database.

2. Flow of Events

2.1 Basic Flow

1. Staff enters subscriber database
2. Staff locates subscriber
3. Staff unchecks subscriber box next to name
4. Staff selects "save"
5. End of use case

3. Special Requirements

3.1 Subscribed

The name and email must of an active subscriber.

4. Pre-conditions

4.1 Subscribed

The user is subscribed to the email newsletter.

5. Post-conditions

5.1 Deleted Subscriber

Subscriber name and email will no longer be available in the database.

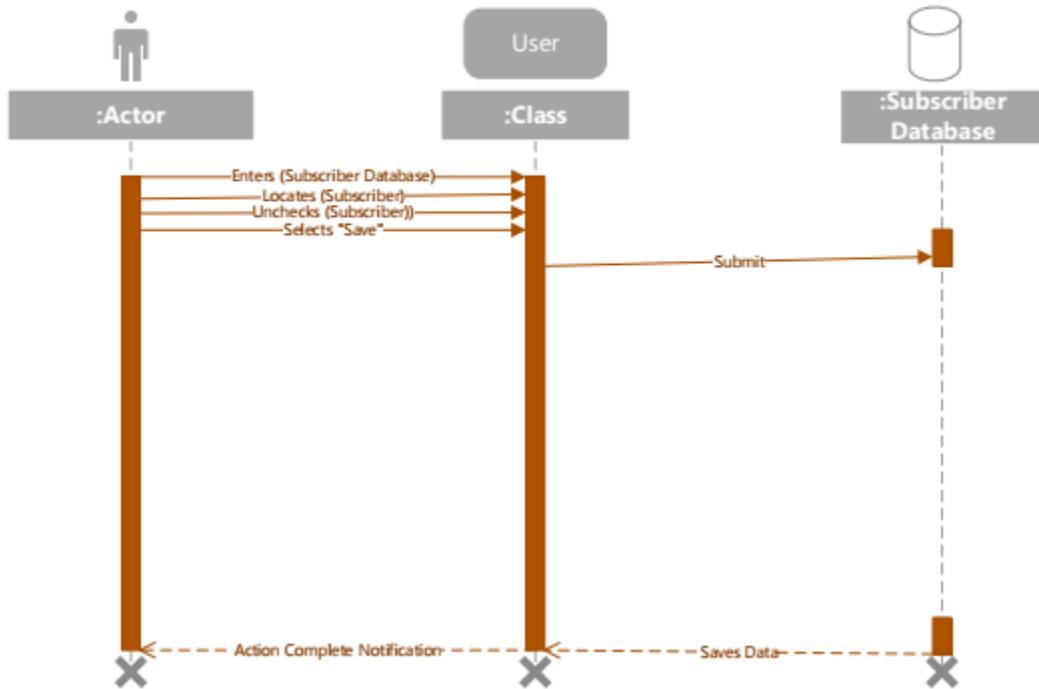
5.2 Unsubscribed to Newsletter

Subscriber will not receive weekly/monthly newsletter

6. Extension Points

None

Delete Subscriber – Sequence Diagram



Use Case Specification: Send Newsletter

1. Use-Case Name: Send Newsletter

1.1 Brief Description

This use case will describe how a staff member will send a weekly/monthly newsletter to subscribers.

2. Flow of Events

2.1 Basic Flow

1. Staff creates content for weekly/monthly newsletter.
2. Staff enters subscriber database
3. Staff extrapolates subscribers' emails from database
4. Staff navigates to organization's email page
5. Staff selects compose email
6. Staff enters subscriber's emails into "to" textbox
7. Staff enters newsletter subject into "subject" textbox
8. Staff enters newsletter content into "body" textbox
9. Staff reviews information entered
10. Staff clicks "send" button
11. Use case ends

3. Special Requirements

3.1 Valid Email

The email entered must be an active account.

3.2 Subscribed User

The email entered must be of a subscribed user

4. Pre-conditions

4.1 Subscribed

The user is subscribed to the email newsletter.

5. Post-conditions

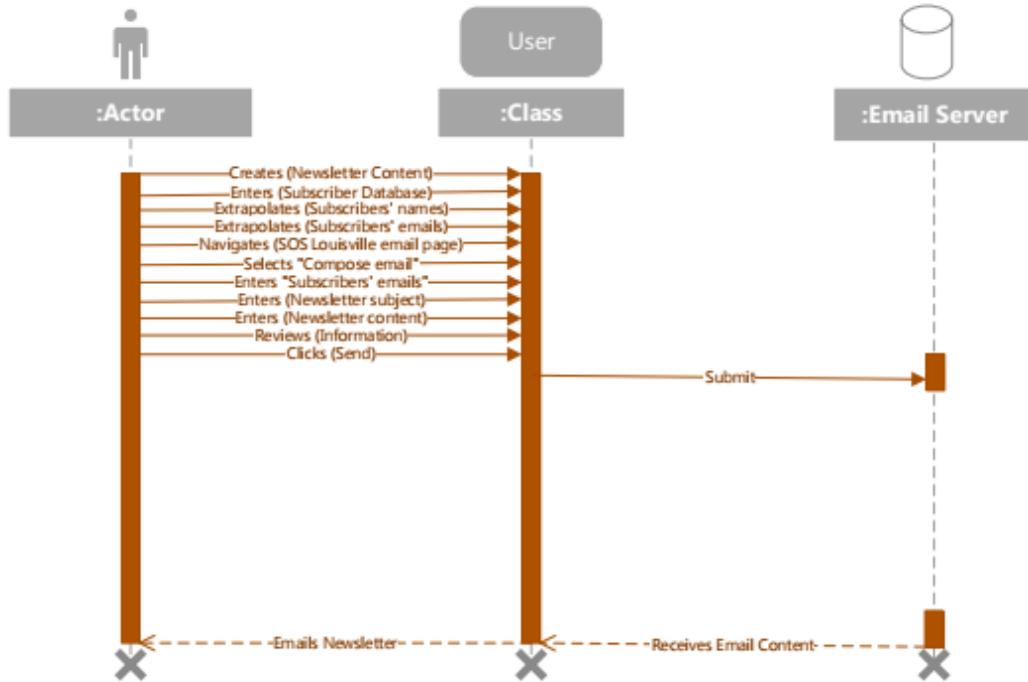
5.1 Newsletter

User receives weekly/monthly newsletter

6. Extension Points

None

Send Newsletter – Sequence Diagram



Use Case Specification: Create Admin Account

1. Create Admin Account

1.1 Brief Description

This use case will allow the actor to successfully create an administrator account for Microsoft Azure and WordPress. Initial creation of administrator accounts will be done by an SOS Administrator. Once initial admin account is created, the admin will then login into the Azure portal, select a subscription you want the admin to access, and add them as an owner. Once complete, you will then be able to add them as a co-administrator to the Azure account.

2. Flow of Events

2.1 Basic Flow

1. Admin logs into Azure Portal
2. Admin logs into the Subscription portal
3. Admin clicks Add Control (IAM) button
4. Admin selects Add
5. Admin selects Role
6. Admin selects Owner
7. Admin types email address of new owner.
8. Admin selects Save

2.2 Alternative Flows

2.2.1 Co-Administrator Created from Owner

1. Admin logs into Azure Portal
2. Admin logs into the Subscription portal
3. Admin clicks Add Control (IAM) button
4. Admin selects Add
5. Admin selects Role
6. Admin selects Owner
7. Admin types email address of new owner.
8. Admin selects Save
9. Admin right-clicks the new Owner
10. Admin selects Add as co-administrator

3. Special Requirements

3.1 Active Microsoft Azure Subscription

SOS Louisville administrators and staff will purchase a subscription for Microsoft Azure.

3.2 Valid Email Address

A valid email address is required to be an administrator. This email address will typically be on with domain of SOS Louisville.

4. Pre-conditions

4.1 Purchased a Microsoft Azure Subscription

SOS Louisville has an active account with Microsoft Azure.

5. Post-conditions

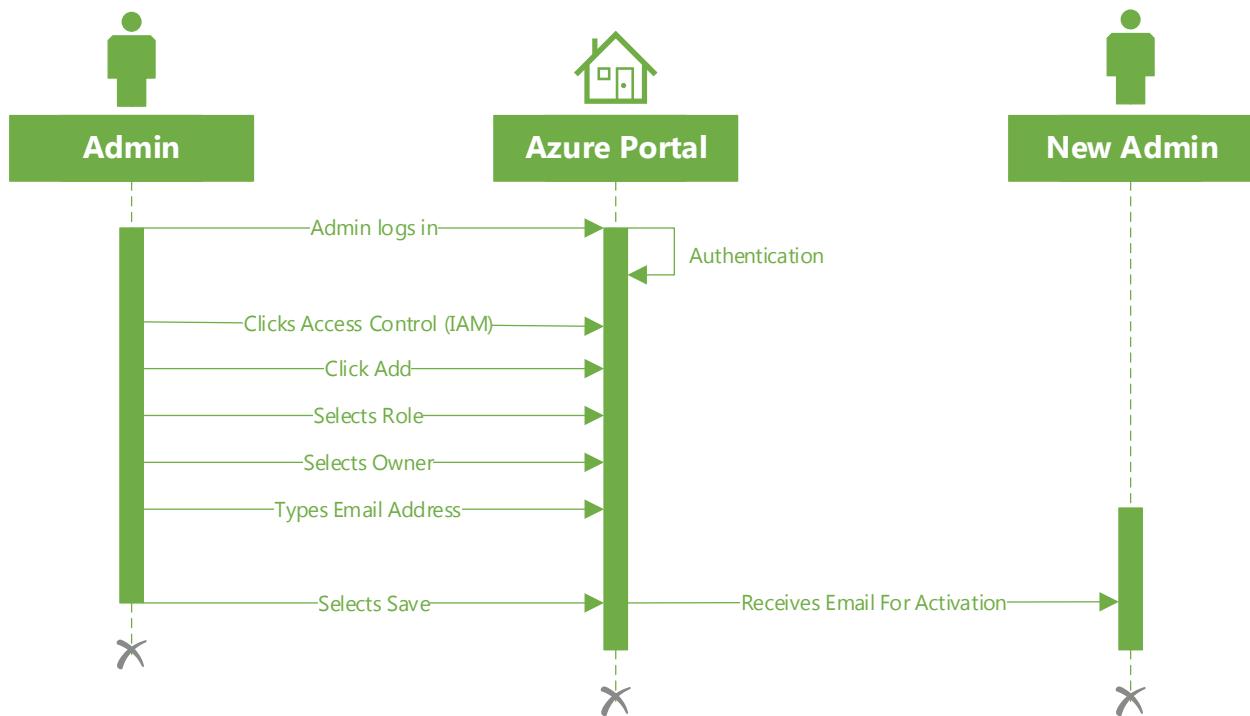
5.1 Administrator Created

Administrator has been created and thus can manipulate the entire system.

6. Extension Points

None

Create Admin Account – Sequence Diagram



Use Case Specification: Modify Admin Account

1. Modify Admin Account

1.1 Brief Description

The administrator will be able to modify an existing admin account. They will have to login to either a top tier admin account above all others, or login to the admin account being modified.

2. Flow of Events

2.1 Basic Flow

1. Admin logs into their account
2. Admin selects the subscription the admin to be modified is under
3. Admin selects Access Control (IAM)
4. Admin may modify access levels of the account.

2.2 Alternative Flows

2.2.1 Modify Admin from Root Account

1. Admin logs into their account
2. Admin selects the subscription the admin to be modified is under
3. Admin selects Access Control (IAM)
4. Admin may modify access levels of the account

2.2.2 Modify Admin Account from User Perspective

1. Admin logs into their account
2. Admin goes to Azure Account Center
3. Admin selects Edit
4. Admin selects Profile
5. Admin select Edit Details
6. Admin may change Name
7. Admin may change Email Address
8. Admin may change Password
9. Admin may change Home Address
10. Admin will click Save Changes

3. Special Requirements

3.1 Admin Account Previously Created

An active admin account to be modified must already be created

4. Pre-conditions

4.1 Admin Account Previously Created

An active admin account to be modified must already be created

5. Post-conditions

5.1 Admin Account Access Level Modified

The access level of the admin account has been changed

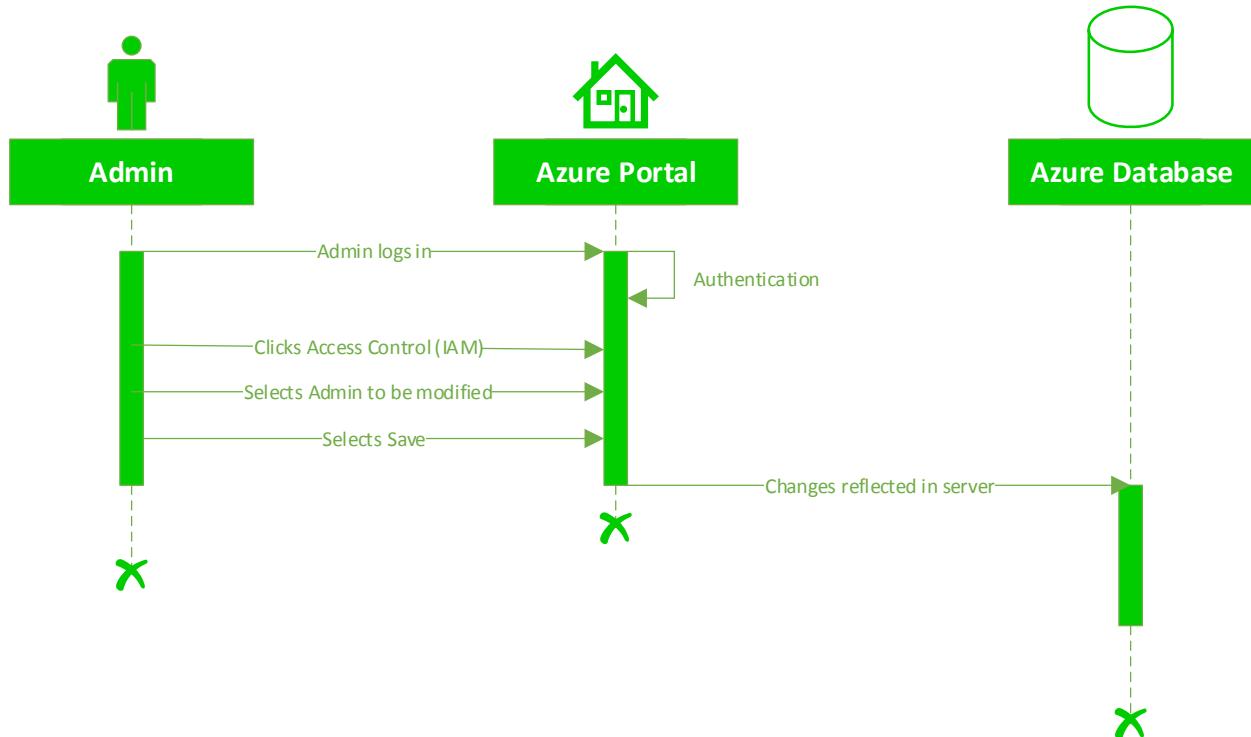
5.2 Admin Account Information Modified

The information for the admin account has changed (i.e., Name, email address, home address, password, etc.)

6. Extension Points

None

Modify Admin Account – Sequence Diagram



Use Case Specification: Delete Admin Account

1. Delete Admin Account

1.1 Brief Description

Administrators will be able to log in to Microsoft Azure and delete an admin account that was previously created.

2. Flow of Events

2.1 Basic Flow

1. Admin logs into Azure Portal
2. Admin logs into the Subscription portal
3. Admin clicks Add Control (IAM) button
4. Admin right-click the user
5. Admin selects Remove Co-Administrator
6. Admin unchecks Owner box

3. Special Requirements

3.1 Valid Administrators

Subscription must have more than one administrator to delete an account.

4. Pre-conditions

None

5. Post-conditions

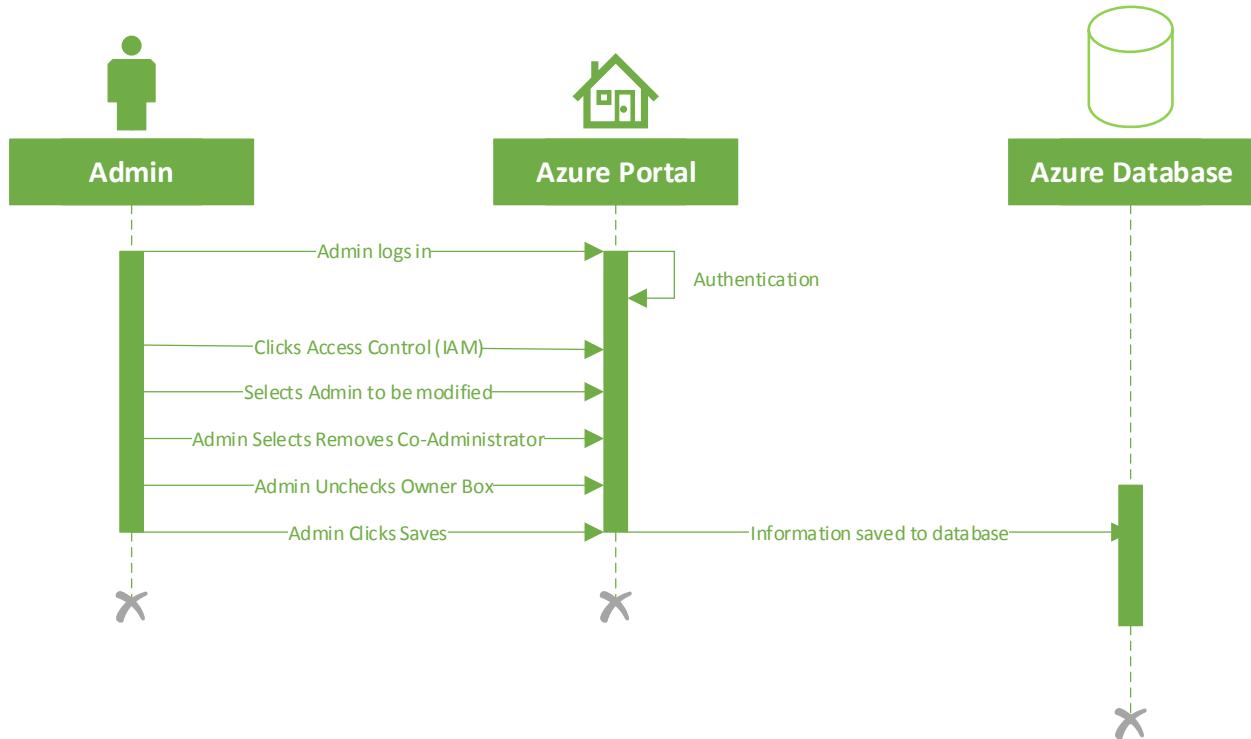
5.1 Admin Account deleted

Once the administrator has unchecked the Owner box, the user will no longer be an admin.

6. Extension Points

None

Delete Admin Account – Sequence Diagram



Use Case Specification: Login Admin Account

1. Login Admin Account

1.1 Brief Description

Administrators are taken through the process of logging into Microsoft Azure.

2. Flow of Events

2.1 Basic Flow

- 1 Admin goes to <https://portal.azure.com>
- 2 Admin will type in the username
- 3 Admin will type in the password
- 4 Admin presses Enter

4.1 Alternative Flows

4.1.1 Admin Logs in For First Time

- 1 Admin receives an email regarding their new account
- 2 Admin will be given their username
- 3 Admin will be given their temporary password
- 4 Admin goes to <https://portal.azure.com>
- 5 Admin will type in the username
- 6 Admin will type in the temporary password
- 7 Admin presses Enter
- 8 Admin is instructed to change their temporary password
- 9 Admin can access the full dashboard

3. Special Requirements

None

4. Pre-conditions

4.1 Valid Email

New admin must have a valid email addresses

5. Post-conditions

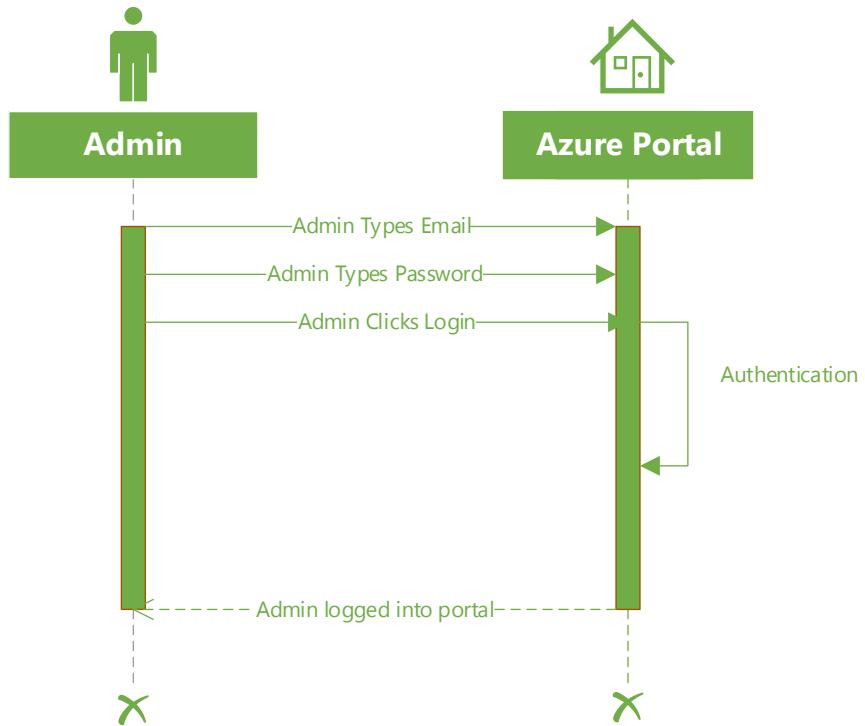
5.1 Admin Is Logged In

Admin can log in to the organizational dashboard

6. Extension Points

None

Login Admin Account – Sequence Diagram



Use Case Specification: Backup Database

1. Backup Database

1.1 Brief Description

This use case describes the steps needed to back up the relation database in Microsoft Azure. There are two ways this can easily be done. One, we back up the database using Azure's geo-redundant functions. This is already set up once the database is created. Second, we can export all the data to a local PC.

2. Flow of Events

2.1 Basic Flow

1. Admin logs into Microsoft Azure
2. Admin clicks on their database
3. Admin clicks Export
4. Admin clicks Storage
5. Admin enters username
6. Admin enters password
7. Admin clicks Create to export database

2.2 Alternative Flows

2.2.1 Automatic Data Retention Via Azure

1. Admin logs into Microsoft Azure
2. Admin clicks Recovery Services Vaults
3. Admin creates a new Recovery Services Vault
4. Admin clicks on their database
5. Admin clicks on their server name
6. Admin clicks Long-term backup retention
7. Admin clicks on their database
8. Admin clicks Configure
9. Admin selects their Recovery Services Vault
10. Admin selects their retention policy.
11. Admin selects Save Changes.

3. Special Requirements

None

4. Pre-conditions

4.1 Server Present

Server is up and running with Microsoft Azure

4.2 Database is Present

Database is up and running with Microsoft Azure and data is populated to it.

5. Post-conditions

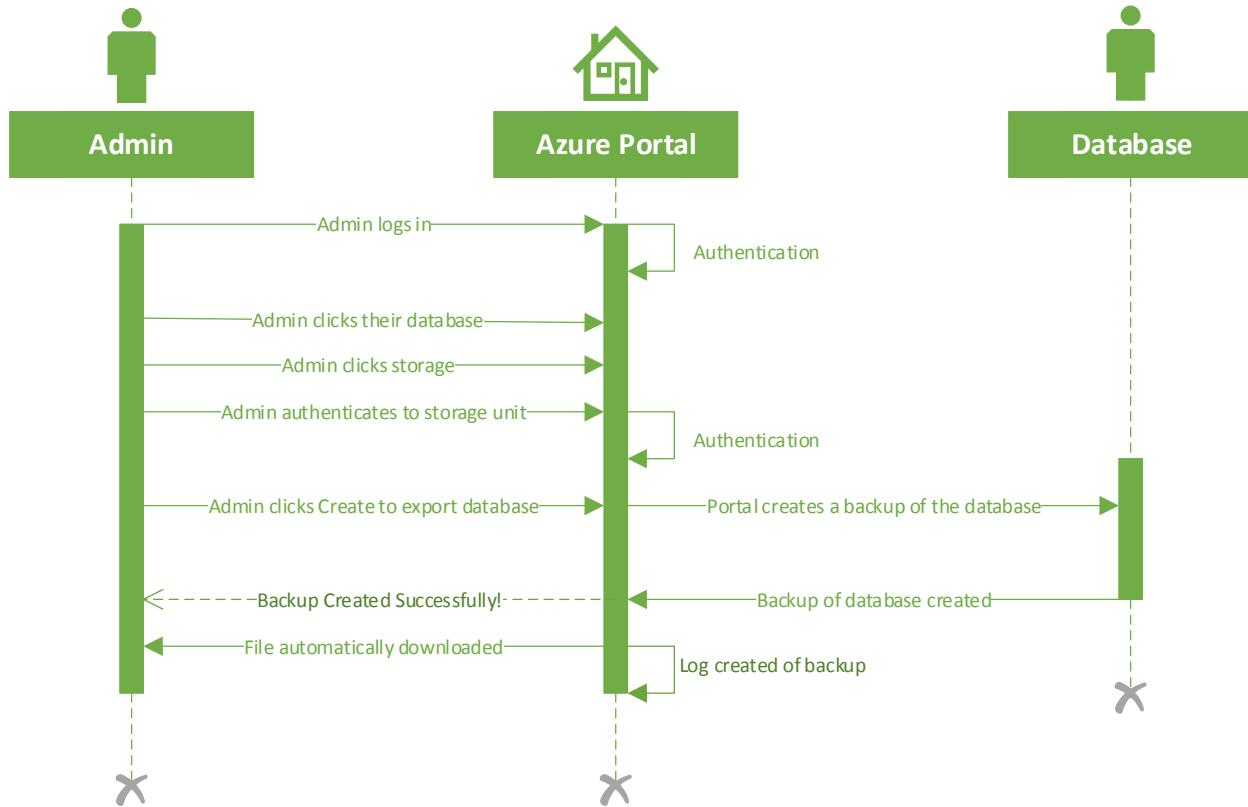
5.1 Data is Backed Up

Data is successfully backed up for their retention period.

6. Extension Points

None

Backup Database – Sequence Diagram



Use Case Specification: Recover Database

1. Recover Database

1.1 Brief Description

This use case defines how the administrators will be able to recover their data in the event there is data loss. If the administrators regularly export their database, they can import their database back in. If they have set up long-term data retention, they will be able to restore their data from the last backup in Microsoft Azure.

2. Flow of Events

2.1 Basic Flow

1. Admin logs into Azure Portal
2. Admin selects their SQL Server
3. Admin clicks Import Database
4. Admin selects their storage
5. Admin enters their username
6. Admin enters their password
7. Admin selects the latest backup from file
8. Admin clicks OK

2.2 Alternative Flows

2.2.1 Data Backup via Azure

1. Admin logs into Azure portal
2. Admin clicks on their database
3. Admin clicks on their server name
4. Admin clicks Long-term backup retention
5. Admin clicks on their database
6. Admin clicks on Overview
7. Admin clicks Restore on the top bar
8. Admin selects the backup they want to restore from
9. Admin clicks OK.

3. Special Requirements

3.1 Manual Backups Created

The admins have already backed up their database manually

3.2 Backups Created via Azures Retention

Backups have been previously set up in Long-term data retention

4. Pre-conditions

4.1 Database Experiences Emergency Loss

Database and/or server has quit working and there has been a loss of data

5. Post-conditions

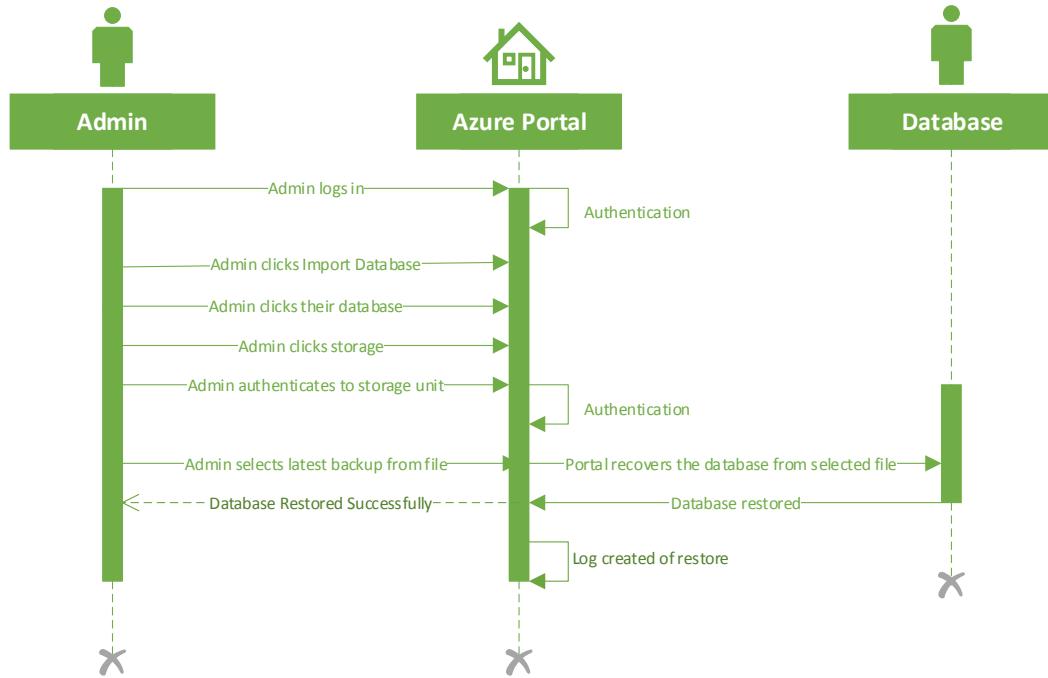
5.1 Database Restored

Database is restored from a manual backup or the retention policy

6. Extension Points

None

Recover Database – Sequence Diagram



Use Case Specification: Create Reports

1. Create Reports

1.1 Brief Description

This use case explains how the administrators of SOS Louisville can successfully and efficiently build reports based off the needs of an event or to build the distribution list for current subscribers. They will be able to select a Volunteer's qualifications, staff needed, and other entities.

2. Flow of Events

2.1 Basic Flow

1. Admin logs into Microsoft Azure
2. Admin clicks SQL Database on the left side.
3. Admin clicks Object-Relational Mapping (ORM) extension
4. Admin will be able to report contents of volunteer
5. Admin will be able to report contents of staff
6. Admin will be able to report contents of donor
7. Admin will be able to report contents of donations
8. Admin will be able to report the newsletter list
9. Admin will click Save Changes

2.2 Alternative Flows

2.2.1 Report Volunteers

1. Admin logs into Microsoft Azure
2. Admin clicks SQL Database on the left side
3. Admin clicks Object-Relational Mapping (ORM) extension
4. Admin can report the name
5. Admin can report the home address
6. Admin can report the email address
7. Admin can report the phone number
8. Admin can report the qualifications
9. Admin can report their specialty
10. Admin can report their availability
11. Admin will click Save Changes

2.2.2 Report Staff

1. Admin logs into Microsoft Azure
2. Admin clicks SQL Database on the left side
3. Admin clicks Object-Relational Mapping (ORM) extension
4. Admin can report the name
5. Admin can report the home address
6. Admin can report the email address
7. Admin can report the phone number
8. Admin can report the specialty
9. Admin can report the role
10. Admin will click Save Changes

2.2.3 Report Donor

1. Admin logs into Microsoft Azure
2. Admin clicks SQL Database on the left side
3. Admin clicks Object-Relational Mapping (ORM) extension

4. Admin can report the name
5. Admin can report the email address
6. Admin will click Save Changes

2.2.4 Report Donation

1. Admin logs into Microsoft Azure
2. Admin clicks SQL Database on the left side
3. Admin clicks Object-Relational Mapping (ORM) extension
4. Admin can report the donation amount
5. Admin will click Save Changes

2.2.5 Report Subscriber

1. Admin logs into Microsoft Azure
2. Admin clicks SQL Database on the left side
3. Admin clicks Object-Relational Mapping (ORM) extension
4. Admin can report the subscriber's name
5. Admin can report the subscriber's email address
6. Admin will click Save Changes

3. Special Requirements

None

4. Pre-conditions

4.1 Object-Relational Mapping Set Up

SOS Consultants must provide you with an ORM that is functional so that reporting from the database is easy and efficient.

5. Post-conditions

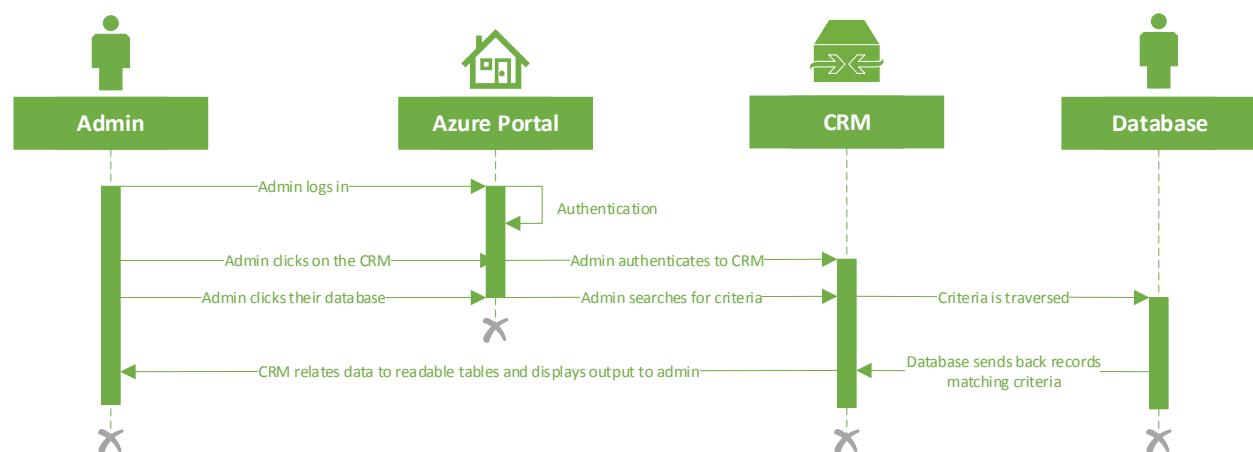
5.1 Reports Created

Reports are created based on the admin specifications.

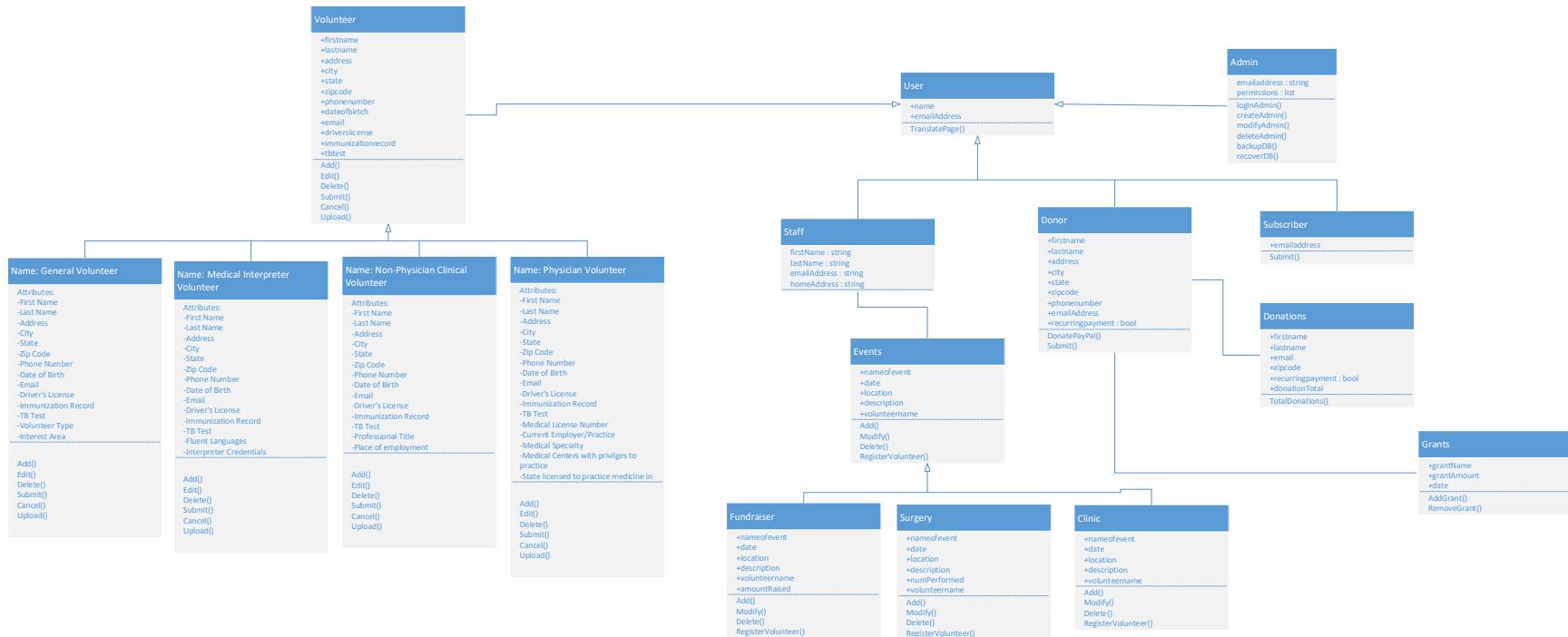
6. Extension Points

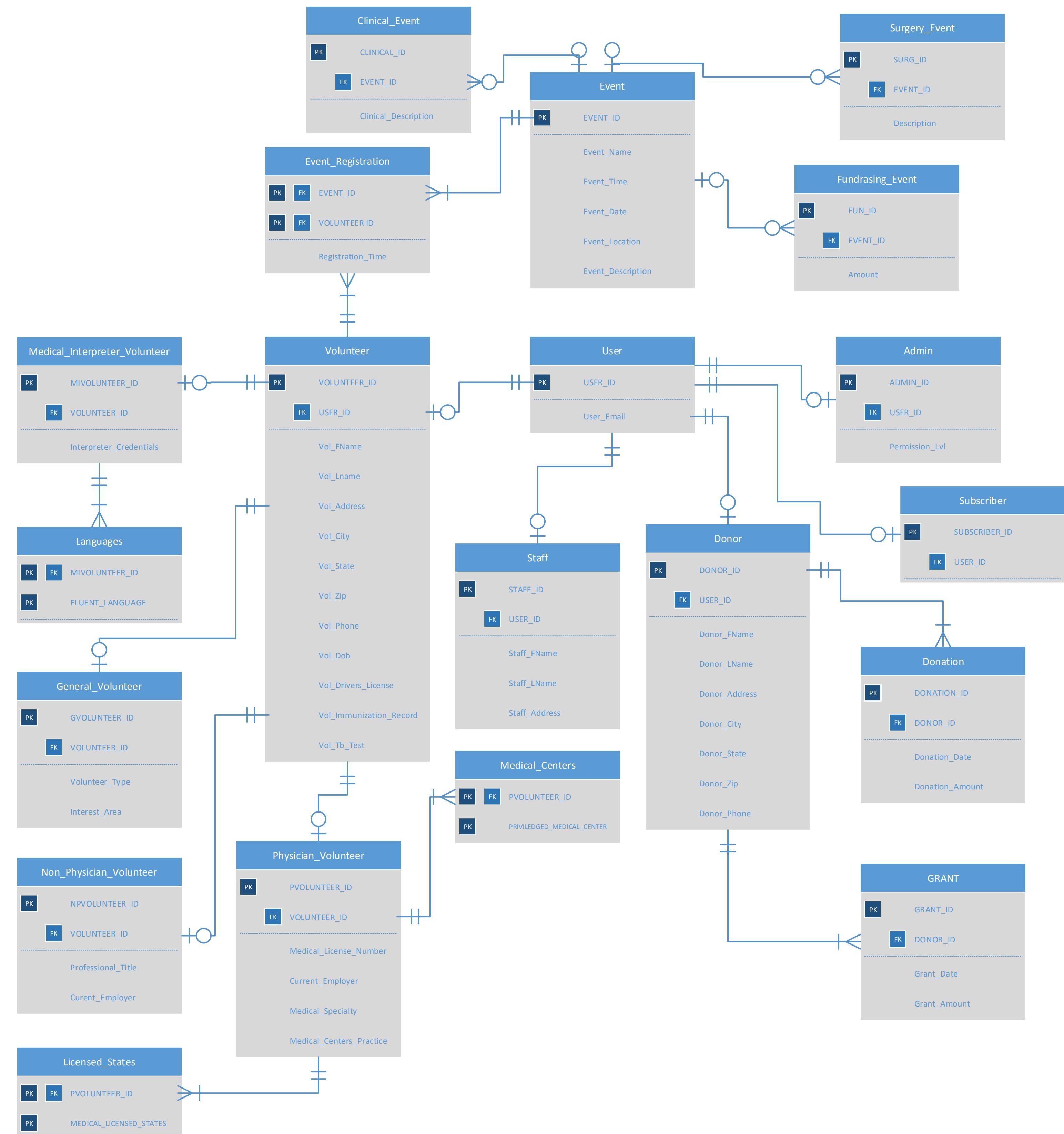
None

Create Reports – Sequence Diagram



Within the class diagram, the user is the base class. A user can be 1 of 4 types of volunteers; General, Medical Interpreter, Non-Physician Clinical, Physician. A user can subscribe for a newsletter. A user can be an admin. A can be a staff member which can register for 1 of 3 types of events; Fundraiser, Surgery, Clinical. A user can be a donor which can make donations or provide grants.





This data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a data definition language, which can then be used to create a database.

The diagram shown here contains the tables, relationships, and dependencies that will populate SOS Louisville's database.

Donor
User who gives organization money

Attribute Name	Attribute Description	Type	Size
DONOR_ID (PK)	Unique donor identification	Integer	10
USER_ID (FK)	Unique user identification	Integer	10
Donor_FName	Donor's first name	String	25
Donor_LName	Donor's last name	String	25
Donor_Address	Donor's address	String	50
Donor_City	Donor's city	String	25
Donor_State	Donor's State	String	15
Donor_Zip	Donor's zip code	String	5
Donor_Phone	Donor's phone number	String	11

Donation
Money given to the organization by users

Attribute Name	Attribute Description	Type	Size
DONATION_ID (PK)	Unique donation identification	Integer	10
DONOR_ID (FK)	Unique donor Identification	Integer	10
Donation_Date	Date donation was given	String	25
Donation_Amount	Dollar amount given	Money	10

User

Anyone who may use the system

Attribute Name	Attribute Description	Type	Size
USER_ID (PK)	Unique user identification	Integer	10
User_Email	Email address of user	String	254

Staff

Staff member that uses the system

Attribute Name	Attribute Description	Type	Size
STAFF_ID (PK)	Unique staff identification	Integer	10
USER_ID (FK)	Unique user identification	Integer	10
Staff_FName	First name of staff member	String	25
Staff_LName	Last name of staff member	String	25
Staff_Address	Address of staff member	String	50

Subscriber
One who subscribes for monthly newsletters

Attribute Name	Attribute Description	Type	Size
SUBSCRIBER_ID (PK)	Unique subscriber identification	Integer	10
USER_ID (PK)	Unique user identification	Integer	10

Admin
Administrator who can change the system

Attribute Name	Attribute Description	Type	Size
ADMIN_ID (PK)	Unique admin identification	Integer	10
USER_ID (FK)	Unique user identification	Integer	10
Permission_Lvl	Permission level of the admin	String	25

Grant

Large amount of money given to the organization

Attribute Name	Attribute Description	Type	Size
GRANT_ID (PK)	Unique identification number	Integer	10
DONOR_ID (FK)	Unique identification number	Integer	10
Grant_Date	Date donation was given	String	25
Grant_Amount	Dollar amount given	Money	10

Volunteer

Individual who volunteers time and skills to the organization

Attribute Name	Attribute Description	Type	Size
VOLUNTEER_ID (PK)	Unique volunteer identification	Integer	10
USER_ID (FK)	Unique user identification	Integer	10
Vol_Fname	First name of volunteer	String	25
Vol_Lname	Last name of volunteer	String	25
Vol_Address	Address of volunteer	String	50
Vol_City	City of volunteer	String	25
Vol_State	State of volunteer	String	25
Vol_Zip	Zip code of volunteer	String	5
Vol_Phone	Phone number of volunteer	String	11
Vol_Dob	Date of birth of volunteer	String	8
Vol_Drivers_License	Driver's license of volunteer	String	15
Vol_Immunization_Record	Immunization record of volunteer	String	500
Vol_Tb_Test	TB Test	BLOB	100 MB

Event

An event which SOS Louisville hosts

Attribute Name	Attribute Description	Type	Size
EVENT_ID (PK)	Unique identification number	Integer	10
Event_Name	Name of the event	String	25
Event_Description	Description of the event	String	500
Event_Date	Date of the event	String	25
Event_Time	Start time of the event	String	10
Event_Location	Location of the event	String	50

Surgery_Event
An event which SOS Louisville hosts

Attribute Name	Attribute Description	Type	Size
SURG_ID (PK)	Unique identification number	Integer	10
EVENT_ID (FK)	Unique identification number	Integer	10
Description	Description of the event	String	500

Fundraising_Event
An event which SOS Louisville hosts

Attribute Name	Attribute Description	Type	Size
FUN_ID (PK)	Unique identification number	Integer	10
EVENT_ID (FK)	Unique identification number	Integer	10
Amount	Amount of money fundraised	String	10

Clinical_Event
An event which SOS Louisville hosts

Attribute Name	Attribute Description	Type	Size
CLINICAL_ID (PK)	Unique identification number	Integer	10
EVENT_ID (FK)	Unique identification number	Integer	10
Clinical_Description	Description of the event	String	500

Event_Registration
The scheduling of volunteers for events

Attribute Name	Attribute Description	Type	Size
EVENT_ID (PK) (FK)	Unique identification number	Integer	10
VOLUNTEER_ID (PK) (FK)	Unique identification number	Integer	10
Registration_Time	Date volunteer registered for event	String	25

Non_Physician_Volunteer
Individual who volunteers as a non-physician

Attribute Name	Attribute Description	Type	Size
NPVOLUNTEER_ID (PK)	Unique identification number for a Non Physician Volunteer	Integer	10
VOLUNTEER_ID (FK)	Unique volunteer identification	Integer	10
Professional_Title	Non-Physician's professional title	String	50
Current_Employer	Non-Physician's current place of employment	String	50

General_Volunteer
Individual who is a general volunteer

Attribute Name	Attribute Description	Type	Size
GVOLUNTEER_ID (PK)	Unique identification number for a General Volunteer	Integer	10
VOLUNTEER_ID (FK)	Unique volunteer identification	Integer	10
Volunteer_Type	Type of general volunteer	String	200
Interest_Area	General volunteer's area of interest	String	200

Medical_Interpreter_Volunteer

Individual who volunteers as a medical interpreter

Attribute Name	Attribute Description	Type	Size
MIVOLUNTEER_ID (PK)	Unique identification number for a Medical Interpreter Volunteer	Integer	10
VOLUNTEER_ID (FK)	Unique volunteer identification	Integer	10
Interpreter_Credentials	Credentials for medical interpreter volunteer	String	200

Languages

The languages in which the medical interpreter is fluent

Attribute Name	Attribute Description	Type	Size
MIVOLUNTEER_ID (PK) (FK)	Unique identification number for medical interpreter volunteer	Integer	10
FLUENT_LANGUAGE (PK)	Specified language of fluency	String	25

Physician_Volunteer
Individual who volunteers as a physician

Attribute Name	Attribute Description	Type	Size
PVOLUNTEER_ID (PK)	Unique identification number for a Physician Volunteer	Integer	10
VOLUNTEER_ID (FK)	Unique volunteer identification	Integer	10
Medical_License_Number	Physician's medical license number	Integer	20
Current_Employer	Physician's current employer	String	50
Medical_Specialty	Physician's medical specialty	String	200
Medical_Centers_Practice	Physician's current medical center of practice	String	50

Medical_Centers

Physician volunteers and medical centers they can volunteer in.

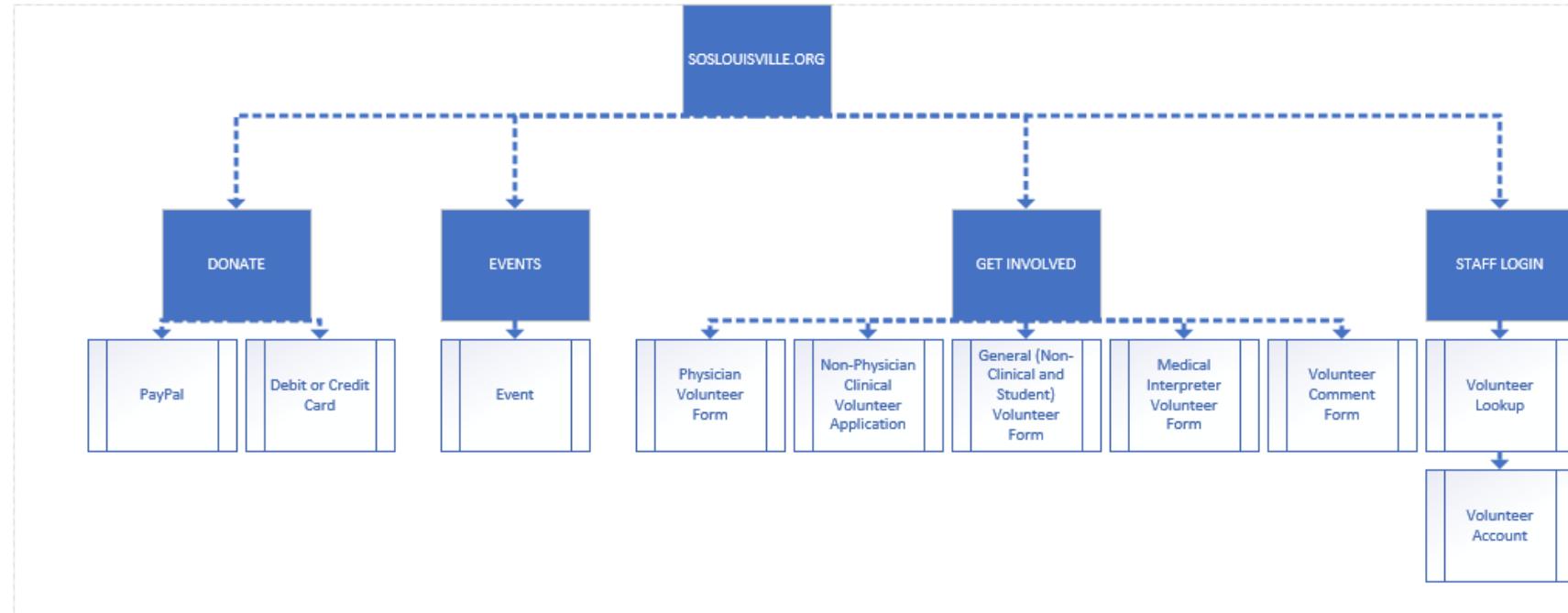
Attribute Name	Attribute Description	Type	Size
PVOLUNTEER_ID (PK)(FK)	Unique identification number for a Physician Volunteer	Integer	10
PRIVILEGED_MEDICAL_CENTER (PK)	Medical center that volunteer can practice in	String	50

Licensed_States

States in which physician volunteer holds a medical license

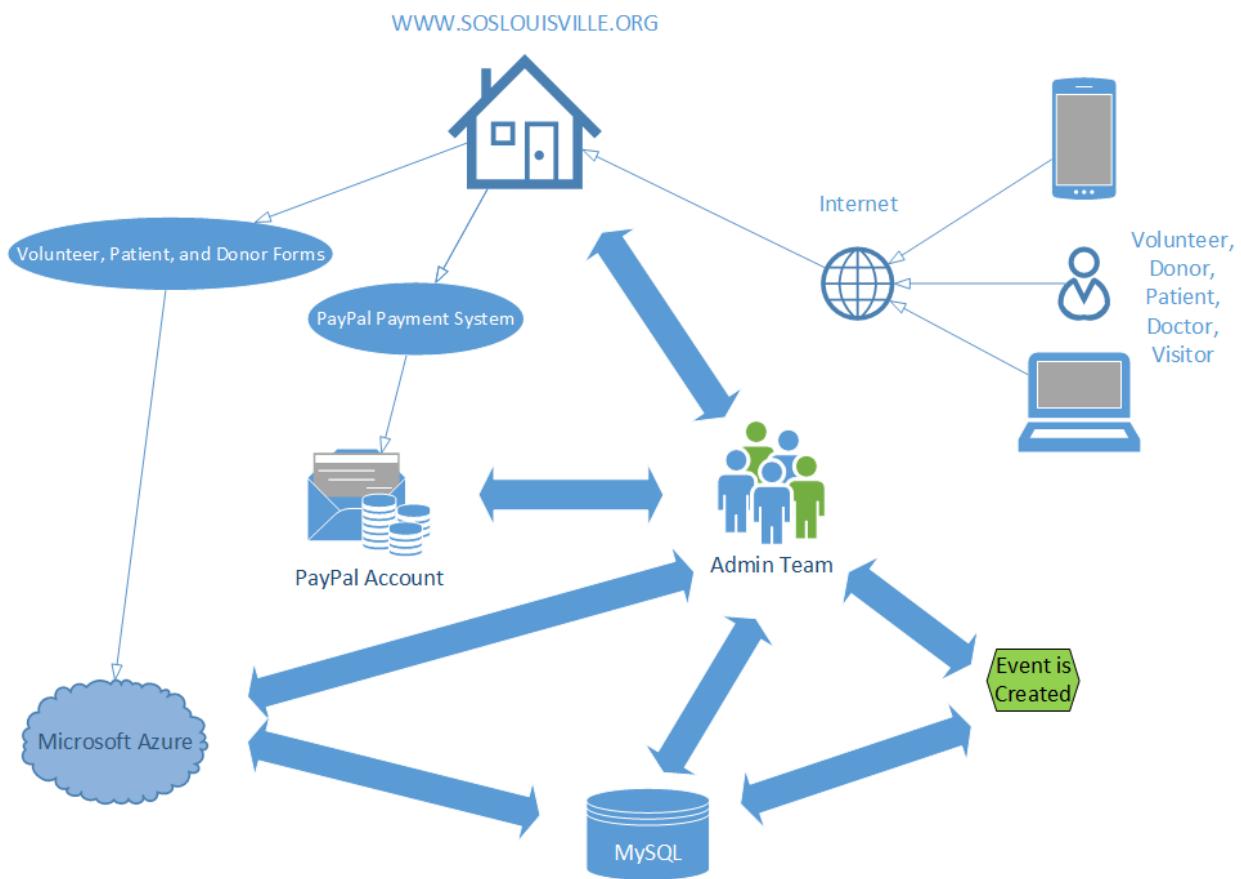
Attribute Name	Attribute Description	Type	Size
PVOLUNTEER_ID (PK) (FK)	Unique identification number for a physician volunteer	Integer	10
MEDICAL_LICENSED_STATES (PK)	States in which physician holds a medical license	String	200

This Window Navigation Diagram depicts the screens needed to realize each use case.



Physical Architecture Design

With this in place, we can begin to build on the architecture already in place to streamline processes and cut down on unnecessary tasks. Volunteers, donors, patients, and visitors can login through the internet and access www.soslouisville.org. From there, they can fill volunteer forms, donor forms, and donate money to the organization's cause. Admins have free reign of the website, the PayPal account, the GSuite account, MySQL, and are in charge of creating events.



Security Procedures

Backup

- **Date:** Third Friday of every month
- **Time:** 10 PM Eastern Standard Time
- **Person:** Technology Coordinator (Jordan Licata)

Recovery

- **Date:** First business day of December
- **Time:** 10 PM Eastern Standard Time
- **Person:** Technology Coordinator (Jordan Licata)

Iteration 2 Gantt Chart

I2												
ID	Task Name	Duration (Days)	Resource	Predecessor	25-Sep	26-Sep	27-Sep	28-Sep	29-Sep	30-Sep	1-Oct	2-Oct
A	Gantt Chart	2	Cailyn Cochran	--								
B	Use Cases	3	Eric Chen	--								
C	Vision Document	3	Brad Riggs	--								
D	Initial Architecture Considerations	3	Patrick Warren	--								
E	Risk Analysis - Use Case Risks	3	Eric Chen	A, B, C, D								
F	Risk Analysis - Other Risk Areas	3	Patrick Warren	A, B, C, D								
G	System Requirements	3	Alexa Bearden	A, B, C, D								
H	Inception Phase- Prototypes	3	Cailyn Cochran /Patrick Warren	A,B,C,D								

The first gantt chart, I2, lists and describes the tasks for the completion of iteration 2. The first four tasks include the gantt chart, use cases, vision document, and initial architecture considerations. These tasks did not have any predecessors, and had duration of at most 3 days. The following tasks were risk analysis for the use cases and other risk areas, system requirements, and inception phase (prototypes). These had predecessors of the first four tasks, so they must be completed first.

Iteration 3 Gantt Chart

I3													
ID	Task Name	Duration (Days)	Resource	Predecessor	16-Oct	17-Oct	18-Oct	19-Oct	20-Oct	21-Oct	22-Oct	23-Oct	24-Oct
A	Use case descriptions for high risk use cases	3	Brad Riggs/Alexa Bearden	--									
B	Updated Gantt chart	3	Cailyn Cochran	--									
C	Use case diagram	3	Eric Chen	A,B									
D	Use case HTML prototype (high risk use cases)	3	Patrick Warren	C									

The second Gantt chart ,I3, lists and outlines the 4 tasks for iteration 3. The tasks for this iteration included use case descriptions for high risk cases, updated gantt chart, use case diagram, and use case HTML prototype. For the first two tasks, there were no dependencies and had 3 days of duration. The use case diagram will begin after the first two tasks, and lastly the use case HTML can begin after the third task is completed.

Iteration 4 Gantt Chart

I4												
ID	Task Name	Duration (Days)	Resource	Predecessor	1-Nov	2-Nov	3-Nov	4-Nov	5-Nov	6-Nov	7-Nov	8-Nov
A	Gantt Chart	2	Cailyn Cochran	--								
B	Use Cases Descriptions - Low Risk	3	Cailyn / Eric / Alexa / Brad / Patrick	--								
C	Use Case Diagram	3	Cailyn / Eric / Alexa / Brad / Patrick	A,B								
D	Sequence Diagram - High and Low Risk	3	Cailyn / Eric / Alexa / Brad / Patrick	A,B,C								
E	Class Diagram	3	Cailyn / Eric / Alexa / Brad / Patrick	A, B, C								
F	CRC Cards, Textual Analysis, and/or Prototype Analysis	3	Cailyn / Eric / Alexa / Brad / Patrick	A, B, C								
G	System Requirements	3	Alexa Bearden	A, B, C, D								
H	Use Case Prototypes - Low and High Risk	3	Cailyn / Eric / Alexa / Brad / Patrick	A,B,C,D								

The third Gantt chart, I4, lists and outlines the 8 tasks for iteration 4. The tasks for this iteration include an updated Gantt chart, use case descriptions for low and high risk use cases, use case diagrams for each use case, sequence diagrams for low and high risk use cases, class diagram, CRC cards, textual analysis, and prototype analysis. The total duration of time to complete these tasks is a week. For the first two tasks, there were no dependencies and had total duration of 2 days. Use case descriptions were completed by each group member. The next task use case diagrams were created for each use case. The next 5 tasks were dependent upon the first three tasks, and had a total duration of 4 days to complete the remaining tasks.

Iteration 5 Gantt Chart

I5												
ID	Task Name	Duration (Days)	Resource	Predecessor	20-Nov	21-Nov	22-Nov	23-Nov	24-Nov	25-Nov	26-Nov	27-Nov
A	Gantt Chart	2	Cailyn Cochran	--								
B	User Interface Prototype	3	Cailyn / Eric / Alexa / Brad / Patrick	--								
C	Class Diagram	3	Brad / Patrick	A,B								
D	Database Design - ERD	3	Cailyn / Eric / Alexa / Brad / Patrick	A, B, C								
E	Data Dictionary	3	Cailyn / Eric / Alexa / Brad / Patrick	A, B, C, D								
F	User Interface Navigation / Site Map	3	Alexa Bearden	A, B, C, D								
G	Screen Layouts	3	Alexa Bearden	A, B, C, D								

The fourth Gantt chart I5, lists and outlines the 7 tasks for iteration 5. The tasks for this iteration included updated Gantt chart, user interface prototypes, class diagram, database design (ERD), data dictionary, user interface navigation site map, and screen Layout. For the first two tasks, there were no dependencies and had 2 days of duration total. The class diagrams can be completed after the user interface prototypes are updated. Next, all group members will collectively work on the Database Design, based off each members assigned use case. The next 3 task to be completed include data dictionary, site map, and screen layouts, which have a duration of two days.

ADD DONOR

Donate to
Surgery on Sunday Louisville

\$ 0.00
USD



Make this a monthly donation [?](#)

Donate with PayPal

or

Donate with a Debit or Credit Card

[Cancel and return to Surgery on Sunday Louisville](#)

ADD DONOR WITH PAYPAL

Donate to
Surgery on Sunday Louisville

\$1.00
USD

Add special instructions to the seller:

Donating from >

 BANK \$1.00

The money will be sent right away.
[Change Payment Method](#)

You'll donate \$1.00

[View PayPal policies](#) and your payment method rights.

Donate Now

[Cancel and return to Surgery on Sunday Louisville](#)

ADD DONOR WITH CREDIT OR DEBIT CARD

Donate to
Surgery on Sunday Louisville

\$1.00
USD

Make this a monthly donation ?

♂ Add special instructions to the seller:

Have a PayPal account? [Log In](#)

Donate with a debit or credit card

United States (dropdown)

Card number

Expires (dropdown) Security code (dropdown)

First name Last name

Billing address

Street address

Apt., ste., bldg. (optional)

City

State (dropdown) ZIP code (dropdown)

Contact Information

Mobile (dropdown) Phone number

Email

Save this information for next time.

View PayPal policies and your payment method rights.

Donate Now

Cancel and return to Surgery on Sunday Louisville

MODIFY DONOR and DELETE DONOR

Search Options

First Name:

Last Name:

or

Email:

Search Results

Table	First Name	Last Name	Email
Donor	John	Smith	jsmith@gmail.com
Volunteer	John	Doe	jdoe@yahoo.com

Select from Results

Donor	John	Smith	jsmith@gmail.com
-------	------	-------	------------------

Table: Donor

First Name:

Last Name:

Email:

Modify or Delete

Method:

Amount:

Special
Instructions:

Add General Volunteer

www.soslouisville.org/general-volunteer-form

General Volunteer Form

Participant Info

First Name*	Drivers License*	<input type="button" value="Choose File"/> No File Chosen Please upload picture of state issued Drivers License or ID.
Last Name*	Immunization Records*	<input type="button" value="Choose File"/> No File Chosen Please upload a copy of current immunization records.
Address Line 1*	Verification of TB Test*	<input type="button" value="Choose File"/> No File Chosen Please upload verification of receiving a TB Test within the past 12 months
Address Line 2	Volunteer Type:	<input checked="" type="checkbox"/> Undergraduate Student <input checked="" type="checkbox"/> General Volunteer <input checked="" type="checkbox"/> Medical Student <input checked="" type="checkbox"/> other: <input type="text"/>
City*	Interest Areas:	<input checked="" type="checkbox"/> Registration <input checked="" type="checkbox"/> Fundraising <input checked="" type="checkbox"/> Cleaning <input checked="" type="checkbox"/> Catering <input checked="" type="checkbox"/> Shadowing: Interpreter <input checked="" type="checkbox"/> Shadowing: Surgery/Clinical
State*		
Zip Code*		
Phone*		
Date of Birth*		
Email*		

SOS Louisville Logo

Volunteer With Us!

Add Non-Physician Clinical Volunteer

www.soslouisville.org/non-physician-clinical-volunteer-form

Non-Physician Clinical Volunteer Form

Participant Info

First Name*	Drivers License*	<input type="button" value="Choose File"/> No File Chosen Please upload picture of state issued Drivers License or ID.
Last Name*	Immunization Records*	<input type="button" value="Choose File"/> No File Chosen Please upload a copy of current immunization records.
Address Line 1*	Verification of TB Test*	<input type="button" value="Choose File"/> No File Chosen Please upload verification of receiving a TB Test within the past 12 months
Address Line 2	Professional Title*	<input type="checkbox"/> CRNA <input type="checkbox"/> APRN <input type="checkbox"/> PA <input type="checkbox"/> CMA <input type="checkbox"/> Scrub Tech <input type="checkbox"/> other: _____
City*	Current Employer/Practice/Hospital*	
State*		
Zip Code*		
Phone*		
Date of Birth*		
Email*		

SOS Louisville Logo

Volunteer With Us!

Add Physician Volunteer

www.soslouisville.org/physician-volunteer-form

Physician Volunteer Form

Participant Info

First Name*	<input type="text"/>	Drivers License*	<input type="file"/> No File Chosen Please upload picture of state issued Drivers License or ID.
Last Name*	<input type="text"/>	Immunization Records*	<input type="file"/> No File Chosen Please upload a copy of current immunization records.
Address Line 1*	<input type="text"/>	Verification of TB Test*	<input type="file"/> No File Chosen Please upload verification of receiving a TB Test within the past 12 months
Address Line 2	<input type="text"/>	Medical License Number*	<input type="text"/>
City*	<input type="text"/>	Practice Specialty*	<input type="text"/>
State*	<input type="text"/>	Current Employer/Practice*	<input type="text"/>
Zip Code*	<input type="text"/>	Privileges*	<input type="text"/>
Phone*	<input type="text"/>	List hospitals that you're privileged to practice	<input type="text"/>
Date of Birth*	<input type="text"/>	Licensed States*	<input type="text"/>
Email*	<input type="text"/>	List states licensed to practice medicine in	<input type="text"/>

SOS Louisville Logo

Volunteer With Us!

Add Medical Interpreter Volunteer

www.soslouisville.org/medical-interpreter-volunteer-form

Medical Interpreter Volunteer Form

Participant Info

First Name*	Drivers License*	<input type="button" value="Choose File"/> No File Chosen Please upload picture of state issued Drivers License or ID.
Last Name*	Immunization Records*	<input type="button" value="Choose File"/> No File Chosen Please upload a copy of current immunization records.
Address Line 1*	Verification of TB Test*	<input type="button" value="Choose File"/> No File Chosen Please upload verification of receiving a TB Test within the past 12 months
Address Line 2	Languages Fluent:	<input type="checkbox"/> Spanish <input type="checkbox"/> Arabic <input type="checkbox"/> Chinese <input type="checkbox"/> ASL <input type="checkbox"/> other
City*	Interpreter Credentials:	<input type="button" value="Choose File"/> No File Chosen Please upload any applicable interpreter certifications
State*		
Zip Code*		
Phone*		
Date of Birth*		
Email*		

SOS Louisville Logo

Volunteer With Us!

Modify Volunteer

www.soslouisville.org/edit-volunteer

Edit Volunteer

Participant Info

First Name*	Brad	Drivers License*	<input type="button" value="Edit"/> license.jpg
Last Name*	Riggs	Immunization Records*	<input type="button" value="Edit"/> IR.pdf
Address Line 1*	1234 Street Rd	Verification of TB Test*	<input type="button" value="Edit"/> VOTBT.pdf
City*	Louisville	Physician	
State*	KY	Medical License Number:	<input type="button" value="Edit"/>
Zip Code*	40219	Practice Specialty:	<input type="button" value="Edit"/>
Phone*	123-456-7890	Privileges:	<input type="button" value="Edit"/>
Date of Birth*	04/14/1989	Licensed States:	<input type="button" value="Edit"/>
Email*	example@louisville.edu	Current Employer/Practice:	<input type="button" value="Edit"/>

Medical Interpreter	
Interpreter Credentials:	<input type="button" value="Edit"/>
Languages Fluent:	<input type="button" value="Edit"/>
Non-Physician Clinical	
Professional Title:	<input type="button" value="Edit"/>
Current Employer/Practice/Hospital:	<input type="button" value="Edit"/>
General Volunteer	
General Volunteer Type:	<input type="button" value="Edit"/>
Interest Area:	<input type="button" value="Edit"/>

SOS Louisville Logo

Delete Volunteer

The screenshot shows a web browser window for the URL www.soslouisville.org/volunteer-search. The page title is "VOLUNTEER SEARCH". On the left, there are buttons for "New Search" and "Search". On the right, there is a logo for "SOS Louisville" with a stylized building icon. The main content area displays a table titled "Search Results" with one row of data. The columns are: Volunteer #, Name, Street #, Street Name, City, Zip Code, Volunteer Type, and Delete. The data in the first row is: 10001, Riggs, Brad, 1234, Street Rd, Louisville, 40219, General, and a red circular delete button.

Volunteer #	Name	Street #	Street Name	City	Zip Code	Volunteer Type	Delete
10001	Riggs, Brad	1234	Street Rd	Louisville	40219	General	

Displaying results 1 of 1

[Open Selected Account](#) [Close](#)

Send Contact Form

Contact Form

www.soslouisville.org/contactform

SOSLouisville LOGO

First Name

Last Name

City

State

Zip Code

Phone Number

Email

Comment

Submit

Prototype - Login

Left picture shows where the admin will enter their email address. Right picture shows where the admin will enter their password.

The image contains two side-by-side screenshots of the Microsoft Azure sign-in interface. Both screenshots feature the Microsoft logo at the top left and a grey header bar with the text "Sign in to Microsoft Azure".
Left Screenshot (Email Input): The input field for the email address, containing "ptwarr01@cardmail.louisville.edu", is highlighted with a yellow rectangular box. Below the input field are two buttons: "Back" (grey) and "Next" (blue). At the bottom, there are links for "No account? Create one!" and "Can't access your account?".
Right Screenshot (Password Input): The input field for the password, containing ".....", is highlighted with a yellow rectangular box. Below the input field are two buttons: "Back" (grey) and "Sign in" (blue). At the bottom, there is a link for "Forgot my password".

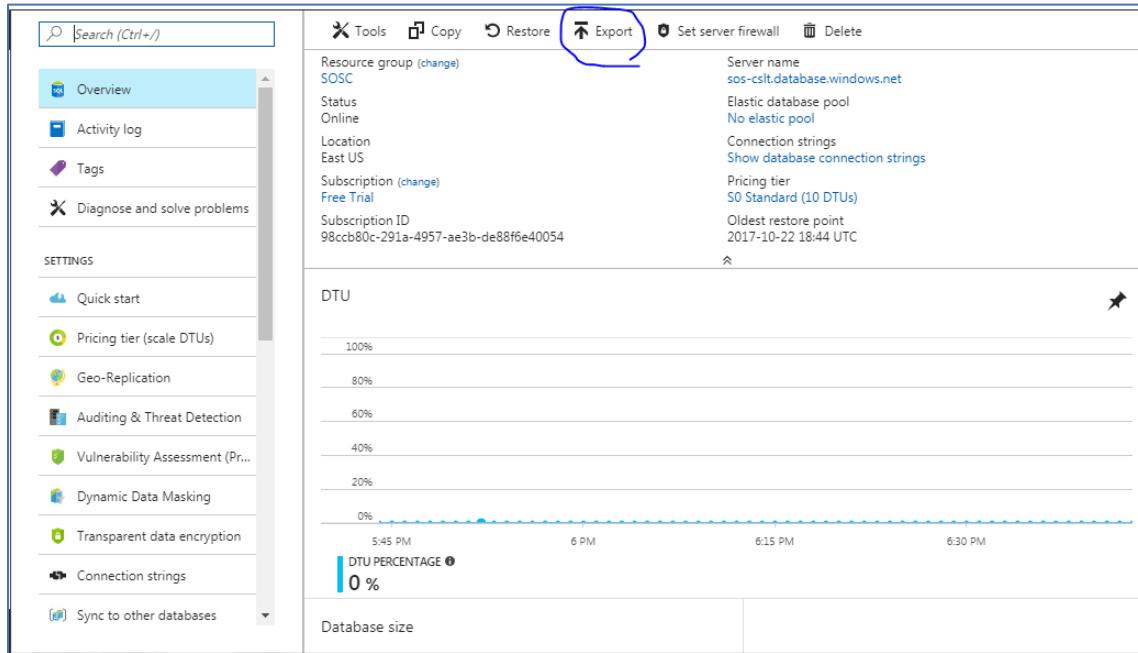
Prototype - Delete

The image shows the Microsoft Azure Access control (IAM) blade. The left sidebar includes options like Overview, Activity log, Access control (IAM), Tags, and Diagnose and solve problems. The main area has tabs for Add, Remove (circled in blue), Roles, Refresh, and Help.
The central part of the screen displays a list of users with the following details:

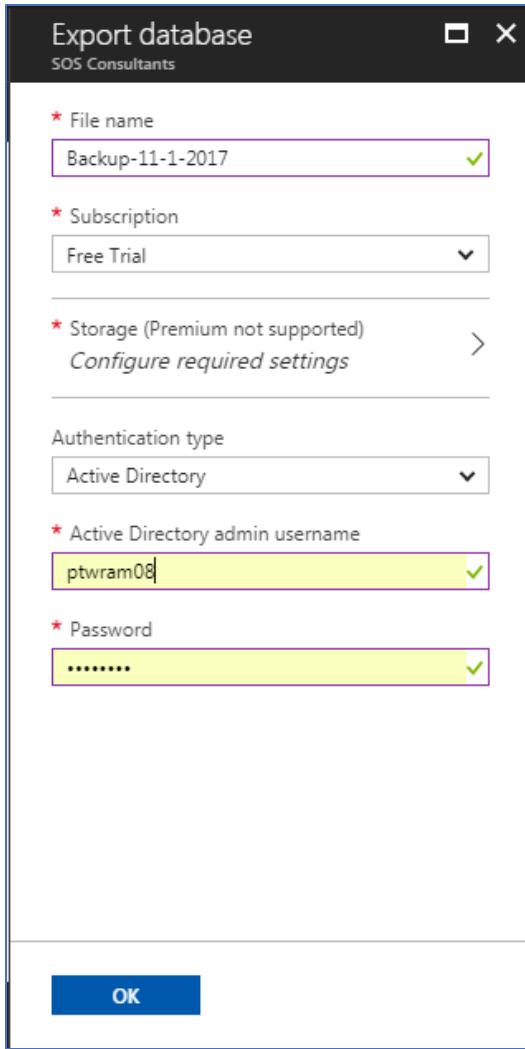
NAME	TYPE	ROLE	SCOPE
Bearden,Alexa Rose arbear02@cardmail....	User	Contributor	This resource
Chen,Eric e0chen04@cardmai...	User	Owner	This resource
Cochran,Cailyn Soo ... cscoch03@cardmail...	User	Owner	This resource
Riggs,Bradley W bwrigg01@cardmai...	User	Owner	This resource
Warren,Patrick Tho... ptwarr01@cardmail....	User	Owner	Subscription (Inherited)

Prototypes - Backup

First picture shows that the admin will need to click Export first. The second photo gives a description of what needs to be filled for the backup to succeed. Please note, you will need storage available in order for the backup to succeed.



The screenshot shows the Azure portal's 'Overview' blade for a database named 'sos-cs1t.database.windows.net'. The 'Export' button in the top navigation bar is highlighted with a blue circle. The main content area displays various database settings and metrics. A chart titled 'DTU' shows usage over time, with the percentage at 0% and the last update at 6:15 PM. Below the chart, there is a section for 'Database size'.



Prototypes - Recover

First picture shows that the admin will need to select Restore. The second asks which backup you would like to restore.

Searched for SOSC

Resource group (change)
SOSC

Status
Online

Location
East US

Subscription (change)
Free Trial

Subscription ID
98ccb80c-291a-4957-ae3b-de88f6e40054

Server name
sos-cslt.database.windows.net

Elastic database pool
No elastic pool

Connection strings
[Show database connection strings](#)

Pricing tier
S0 Standard (10 DTUs)

Oldest restore point
2017-10-22 18:44 UTC

DTU

100%
80%
60%
40%

Restore

SOS Consultants

* Database name
SOS Consultants_2017-11-08T02-30Z

Point-in-time Long-term

Oldest restore point
2017-10-22 18:44 UTC

Restore point (UTC)
2017-11-08 2:30:00 AM

* Target server
sos-cslt East US

Elastic database pool
None

* Pricing tier
Standard S0: 10 DTU, 250 GB

Pin to dashboard

OK

View Social Media

1

www.soslouisville.org

About Us Get Help Get Involved Refer a Patient Events DONATE Staff Login English

Last Name

Subscribe

PLEASE FOLLOW & LIKE US

[f](#) [t](#)

3

https://www.facebook.com/SurgeryonSundayLouisville/

Log In

Surgery on Sunday Louisville
@SurgeryonSundayLouisville

Home About Events Photos Community Reviews Posts

[Like](#) [Create Fundraiser](#) [Donate](#) [Message](#)

Photos

Nonprofit Organization in Louisville, Kentucky
5.0 ★★★★
945 people like this

Community See All

945 people like this

See more of Surgery on Sunday Louisville on Facebook

Log In OR Create New Account

2

www.soslouisville.org

About Us Get Help Get Involved Refer a Patient Events DONATE Staff Login English

Last Name

Subscribe

PLEASE FOLLOW & LIKE US

[f](#) [t](#)

4

https://www.facebook.com/SurgeryonSundayLouisville/

[f](#) [t](#) [Callys](#) Home

Surgery on Sunday Louisville
@SurgeryonSundayLouisville

Home About Events Photos Community Reviews Posts

[Like](#) [Follow](#) [Create Fundraiser](#) [...](#)

Status Photo/Video

Write something on this Page...

Photos

Nonprofit Organization in Louisville, Kentucky
5.0 ★★★★
945 people like this

Community See All

Invite your friends to like this Page
945 people like this

944 people follow this

Tonya Polk Cochran and 3 other friends like this or have checked in

About See All

(502) 242-2089

Subscribe to Newsletter

1

A screenshot of a website with a navigation bar at the top. The navigation bar includes links for 'About Us', 'Get Help', 'Get Involved', 'Refer a Patient', 'Events', 'DONATE', 'Staff Login', and a language selection dropdown set to 'English'. Below the navigation bar, the text 'Louisville community.' is displayed. The main content area contains the heading 'Subscribe to our monthly mailing list!' followed by a form with three input fields: 'Email Address *', 'First Name', and 'Last Name', each with a corresponding text input box. A large black 'Subscribe' button is positioned below the input fields.

2

Subscribe to our monthly mailing list!

Email Address *



cailyncochran@gmail.com

First Name



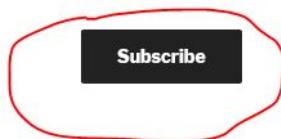
Cailyn

Last Name



Cochran

Subscribe

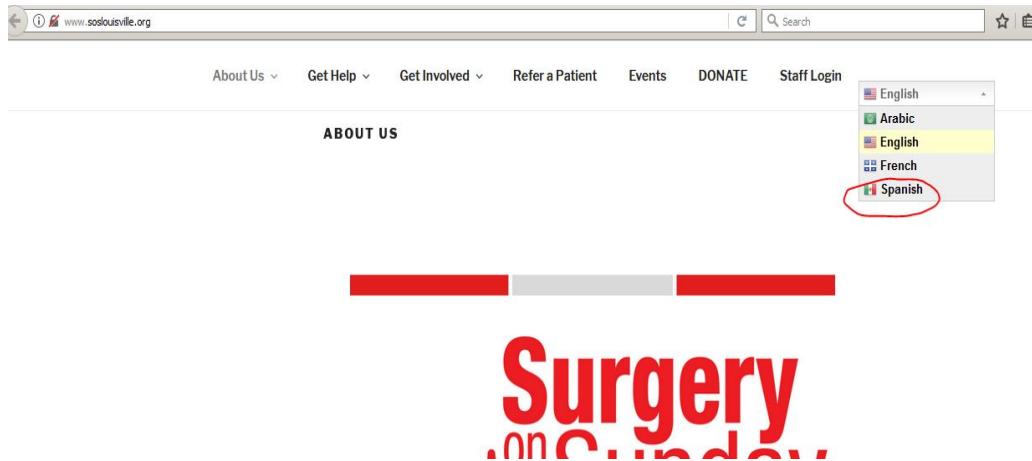


View Website Spanish

1



2



3



Estamos cirugía el domingo Louisville, Inc.

View Website Spanish

1

A screenshot of a web browser showing the homepage of www.sosLouisville.org. The page features a navigation bar with links for About Us, Get Help, Get Involved, Refer a Patient, Events, DONATE, and Staff Login. A dropdown menu for language selection is open, showing options: English (selected), Arabic, French, and Spanish. The 'English' option is highlighted with a red oval. Below the navigation is a section titled 'ABOUT US' with a red and grey horizontal bar above it. The main title 'Surgery on Sunday' is displayed in large red letters, with 'on Sunday' in a smaller font.

2

A screenshot of the same website as in step 1, but with a different language selection. The dropdown menu shows 'English' selected again, indicated by a yellow background and a red oval. The rest of the page content, including the navigation bar and the 'Surgery on Sunday' logo, remains the same.

3

A screenshot of the website after the language has been changed. The dropdown menu now shows 'Spanish' selected, indicated by a yellow background and a red oval. The rest of the page content, including the navigation bar and the 'Surgery on Sunday' logo, remains the same.

We are Surgery on Sunday Louisville, Inc.

Modify Subscriber

1

www.soslouisville.org/subscriber-search

New Search Search **Subscriber Database**



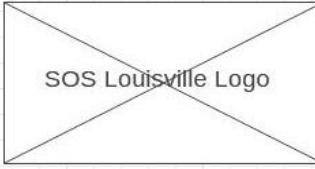
Name	Email	Subscribed
Cailyn Cochran	Cailyn.Cochran@louisville.edu	<input checked="" type="radio"/>
Brad Riggs	Brad.Riggs@louisville.edu	<input checked="" type="radio"/>
Alexa Bearden	Alexa.Bearden@louisville.edu	<input checked="" type="radio"/>
Patrick Warren	Patrick.Warren@louisville.edu	<input checked="" type="radio"/>
Eric Chen	Eric.Chen@louisville.edu	<input checked="" type="radio"/>
		<input type="radio"/>
		<input type="radio"/>
		<input type="radio"/>
		<input type="radio"/>

Save

2

www.soslouisville.org/subscriber-search

New Search Search **Subscriber Database**



Name	Email	Subscribed
Cailyn Cochran	Cailyn.Cochran@louisville.edu	<input checked="" type="radio"/>
Brad Riggs	Brad.Riggs@louisville.edu	<input checked="" type="radio"/>
Alexa Bearden	Alexa.Bearden@louisville.edu	<input checked="" type="radio"/>
Patrick Warren	Patrick.Warren@louisville.edu	<input checked="" type="radio"/>
Eric Chen	Eric.Chen@louisville.edu	<input checked="" type="radio"/>
		<input type="radio"/>
		<input type="radio"/>
		<input type="radio"/>
		<input type="radio"/>

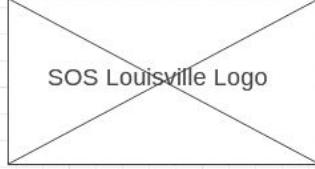
Save

Delete Subscriber

1

www.soslouisville.org/subscriber-search

New Search Search **Subscriber Database**



Name	Email	Subscribed
Cailyn Cochran	Cailyn.Cochran@louisville.edu	<input checked="" type="radio"/>
Brad Riggs	Brad.Riggs@louisville.edu	<input checked="" type="radio"/>
Alexa Bearden	Alexa.Bearden@louisville.edu	<input checked="" type="radio"/>
Patrick Warren	Patrick.Warren@louisville.edu	<input checked="" type="radio"/>
Eric Chen	Eric.Chen@louisville.edu	<input checked="" type="radio"/>
		<input type="radio"/>
		<input type="radio"/>
		<input type="radio"/>
		<input type="radio"/>

Save

2

www.soslouisville.org/subscriber-search

New Search Search **Subscriber Database**



Name	Email	Subscribed
Cailyn Cochran	Cailyn.Cochran@louisville.edu	<input checked="" type="radio"/>
Brad Riggs	Brad.Riggs@louisville.edu	<input type="radio"/>
Alexa Bearden	Alexa.Bearden@louisville.edu	<input checked="" type="radio"/>
Patrick Warren	Patrick.Warren@louisville.edu	<input checked="" type="radio"/>
Eric Chen	Eric.Chen@louisville.edu	<input checked="" type="radio"/>
		<input type="radio"/>
		<input type="radio"/>
		<input type="radio"/>
		<input type="radio"/>

Save

Send Newsletter

1



2

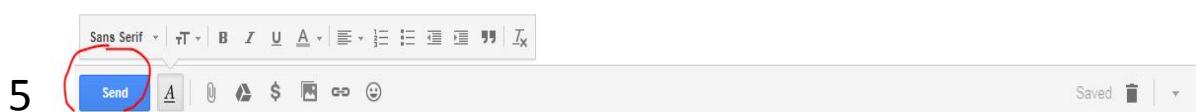
→ *Subscribers*

3

→ *Monthly Newsletter*

4

→ Newsletter Content|



About Us ▾ Get Help ▾ Get Involved ▾ Refer a Patient Events DONATE Staff Login English

Partner Clinics

< > today November 2017 month week day

Sun	Mon	Tue	Wed	Thu	Fri	Sat
29	30	31	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	1	2
3	4	5	6	7	8	9

Add Event

View event calendar, Add event

The prototype above shows how the calendar would look and the functions on the calendar.

Clinic

Date/Time
Date(s) - 11/09/2017
4:45 pm - 8:00 pm

[Register](#) [Schedule](#)

Categories

By appointment only.



Please follow and like us:
[Like](#) [Share](#) [Tweet](#) [Share](#)

[Edit](#) [Delete](#)

Modify, Delete, Register, Schedule event

This prototype shows the event in detail with the functions edit, delete, register, and schedule.

Add New Calendar

My Events

Add Media

Visual Text

B I ~~A~~ ~~E~~ ~~H~~ ~~L~~ ~~T~~ ~~C~~ ~~S~~ ~~P~~ ~~U~~ ~~M~~ ~~A~~ ~~O~~ ~~N~~ ~~F~~ ~~R~~ ~~G~~ ~~?~~

Paragraph ~~U~~ ~~M~~ ~~A~~ ~~O~~ ~~N~~ ~~F~~ ~~R~~ ~~G~~ ~~?~~

Add event

Above is the add event function.

Participant Info

First Name*	<input type="text"/>
Last Name*	<input type="text"/>
Address Line 1*	<input type="text"/>
Address Line 2	<input type="text"/>
City*	<input type="text"/>
State*	<input type="text"/>
Zip Code*	<input type="text"/>
Phone*	<input type="text"/> Your primary contact number
Date of Birth*	<input type="text"/>
Email*	<input type="text"/>
<input type="button" value="Previous"/> No file selected <input type="button" value="Next"/>	

Register for event

Above is the register event in detail.



Email



Enter

Schedule volunteer

This prototype displays the schedule volunteer for an event.