

Systems Analysis & Design with
The Center for Service in Action

Alex Huerta
Cristian Ramirez
Katie Blank
Zoe Roberts

11 March 2019

Background Information

The Organization

The Center for Service in Action (CSA) is part of the Cal Poly Dean of Students. Its office is located on campus in Building 52 — Room E-22. The organization employs three full-time staff members—Bradley Kiker, Sr. Coordinator; Megan Rivoire, Coordinator; and Katie Ullrich, SLO AmeriCorps VIP Leader. In addition, 25 student volunteers comprise the Student Community Services (SCS) leadership team, which is recruited annually during Spring Quarter. The CSA aims to connect students with meaningful community service opportunities through its core programs—Alternative Breaks, Service Learning, and Student Community Services. Through these programs, the CSA facilitates service and learning partnerships between Cal Poly and its community partners (i.e., local non-profit agencies). Moreover, the CSA promotes conversations surrounding social justice by hosting monthly Social Justice Series and an annual social justice conference called Change the Status Quo (CSQ).

The People

Founded in 1972, Student Community Services is a student-run organization within the Center for Service in Action. Members of its leadership team are responsible for planning and promoting volunteer events to fellow students. SCS members must attend weekly program and subprogram meetings, participate in weekly office hours, submit event proposals, coordinate outreach to volunteers, and lead volunteers during their approved events. The Coordinator and SLO AmeriCorps VIP Leader are responsible for guiding the student leaders throughout the process and approving their event proposals.

The State of Information Technology in the CSA

Currently, the main information technology used in the CSA is Google Drive. The office has ten PC desktop computers used by the full-time staff and its student leadership team. While the staff requires that some files be stored locally on its computers, the majority of files are stored on Google Drive. However, it's worth mentioning that the CSA is nearly at the capacity of its Drive account, having used 13.3 GB of the available 15 GB. Event approvals are handled using fillable PDF forms, email correspondence between the student leadership team and full-time staff, Google Forms, and Google Sheets. Events are listed on its Google Calendar, which is public to the Cal Poly community. The CSA is scheduled to get new software at the end of the year to help track liability forms and community partner agreements, but they do not have an organized system in place to track their event management process.

Problem Identification

The current status of operations at the CSA remains entirely on Google Drive. As stated in “State of Information Technology in the CSA” (Page 2), the CSA is nearly at the capacity of its Drive account having used 13.3 GB of the available 15 GB. The Google Drive serves as an online storage facility for fillable PDF forms, scanned liability forms, Google Forms, and Google Sheets accessible by both the student leadership team and full-time staff. The complete reliance on the Drive is a challenging platform for the CSA to use to track and store event related paperwork. Google Drive fails to provide CSA with any reference if an event has been approved, denied, or ready for the calendar.

Relying entirely on the Google Drive and email correspondence has resulted in a disjointed process of approving events, tracking liability forms, and verifying community partner agreements. Our system has the ability to simplify the event management process by creating a website that houses all necessary forms associated with an event. Currently, when a student requests an event they download a fillable PDF form, complete it and email it to Katie Ullrich, the SLO AmeriCorps VIP Leader. By emailing the paperwork, rather than uploading the document to the system, Ullrich is the only member of the CSA staff who has access to the event. In addition, emailing the document provides the opportunity for the document to get lost, deleted, or misplaced. Currently, when an event is approved or denied, the update is only tracked via email correspondence. Our system will instead allow for a CSA staff member to review the uploaded PDF document and select its approval status. This will allow other CSA staff members to determine if an event has been approved or denied.

Once an event has been approved, the CSA staff member must determine if the event has all of the associated liability insurance paperwork completed. According to Ullrich, all liability paperwork must be completed by hand rather than electronically. The system currently in place requires the paperwork to be filled out, scanned, and uploaded to the Google Drive. This process provides no clear indication if all necessary paperwork has been completed unless the user combs through the folders within the Drive. While the paperwork must still be filled out by hand and scanned into our system, our system will instead have the ability to indicate to the user if the forms have been completed. When each form is scanned and uploaded, the system will verify the form is complete and archive the document with the associated event and be marked as a completed task. In addition, our system will archive liability forms that typically exist in the system for 5-7 years. Our database will allow liability forms to remain in the system for as long as needed unlike the CSA’s Google Drive that has limited cloud space still available.

After an event has been approved and all associated paperwork has been completed the CSA staff member must determine which of the event times and locations work within the Google

Calendar. The staff member must review the initial event request PDF to determine which days the event was requested for and then update the information to the Google Calendar. Our system will extract the dates from the PDF and update the system's calendar with approved date and time of the event. This process will allow the system to verify an existing event does not conflict with a new event.

Lastly, when an event is going to take place the volunteers sign up via a Google Form. Ullrich indicated that this process of volunteer sign up made it difficult to later track volunteer hours and liability forms. In our system, the volunteer interacts directly with the system to sign up for an event and download the correct form to fill out. Ullrich indicated that volunteers must bring their liability forms to an event rather than uploading it back to the system. After an event, a CSA staff member will have the ability to verify which event each volunteer served at. The system will also be able to store the scanned version of the volunteer's liability form and archive it for later reference after the event has taken place.

Our system will curb the problem of navigating within the Google Drive to find documents and verify that they have been completed. Our system will allow users to more easily find forms, determine if tasks have been performed, update a comprehensive calendar, and streamline the process for volunteer sign ups.

System Request

Center for Service in Action — Event Management Project	
Project Sponsor	Megan Rivoire, Center for Service in Action Coordinator
Business Need	<p>The project has been initiated to streamline event management processes that were previously managed via email, Google Sheets, and Google Forms. Currently,</p> <ul style="list-style-type: none"> • Staff completes event proposals via fillable PDF form and sends to office coordinator via email • Office coordinator approves or denies proposals via email • Staff must manually add approved events to the calendar • Staff uses Google Form/Sheets to track volunteer sign-ups • Volunteers sign liability paperwork in-person at the start of an event, which must be uploaded/tracked after the event
Business Requirements	<p>Using the new system over the traditional one, users will be able to keep better track of all events and the people/organizations involved. The specific functionality of the system should include:</p> <ul style="list-style-type: none"> • Create new events • Approve/deny events • Add documentation to active events • Search for and update events • Automatically schedule events in official calendar • View the calendar • Sign up volunteers for events • Complete liability paperwork prior to event (done on paper) • Download forms • Archive expired event information for up to 7 years and auto delete after.
Business Value	<p>The Center for Service in Action will benefit from this system through improved time management, which will contribute to increased staff productivity and a longer marketing timeline.</p> <ul style="list-style-type: none"> • Drastically reduce waiting time during the Event Proposal process • Increase productivity by allowing more time spent on other tasks • 10% increase in volunteer participation due to improved marketing timeline
Special Issues or Constraints	<p>The new system should be completed, tested, and implemented by the start of the new school year to allow new employees to familiarize themselves with the system.</p>

Analysis

User Requirements

- General staff needs to be able to propose an event
- Office coordinator needs to be able to approve or deny an event
- Users need to be able to view events on the calendar
- Volunteers need to be able to sign up for events
- General staff needs to be able to upload documents when proposing events
- Users need to be able to edit event information
- Users need to be able to download documents

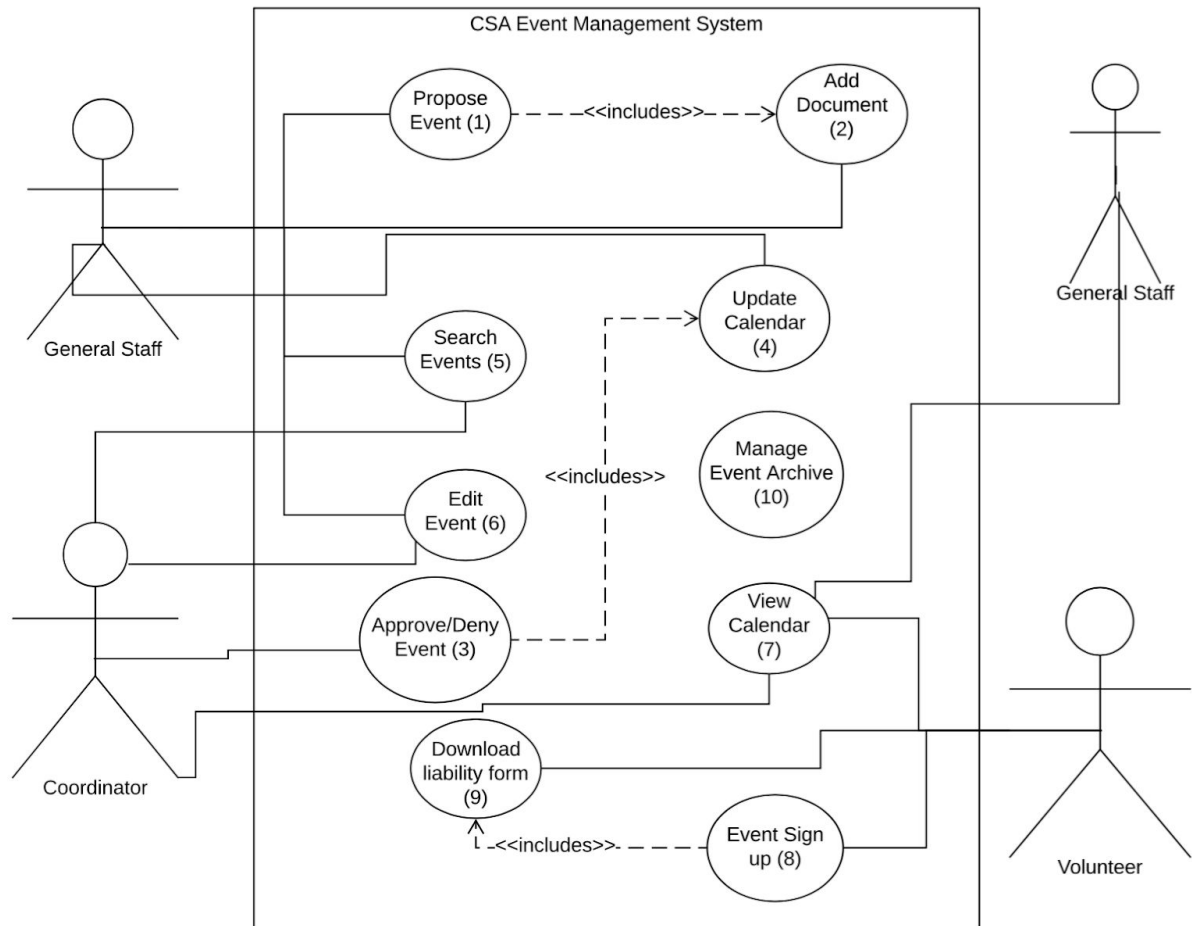
Functional Requirements

- System needs to keep track of event information including title, description, category of volunteer work, event host/creator, date, time, partner agency, transportation, relevant forms
- System needs to automatically add events to the calendar once approved
- System needs to request login credentials from users needing to access secure documents
- System needs to notify office coordinator when an event proposal has been submitted
- System needs to notify general staff when an event has been approved or denied so that they can make edits as necessary
- System must be able to archive event info (liability forms) for up to 7 years before deletion.

Non-functional Requirements

- Systems needs to be secure, especially for the sake of protecting liability forms and agreements
- System needs to be compatible with iOS and PC systems

Use Case Diagram



Use Cases

Title:	Propose Event
Description:	This use case describes the process of general staff proposing new events.
Primary Actor:	General Staff
Trigger:	A staff member wants to create a new volunteer event.
Preconditions:	<ol style="list-style-type: none"> 1. User is logged into the system 2. User has navigated to the main menu
Normal Course:	<ol style="list-style-type: none"> 1. User selects the “Propose Event” option 2. System presents the “Propose Event” dialog 3. User enters the event information <ul style="list-style-type: none"> ○ Title ○ Description ○ Category of volunteer work ○ Event host/creator ○ Date ○ Time ○ Partner agency ○ Transportation 4. Invoke use case “Add Document” 5. User selects the “Save” option 6. System validates event information with user 7. System sends the completed proposal to the office coordinator for approval
Postconditions:	<ol style="list-style-type: none"> 1. Event proposal has been saved in system database 2. Event proposal has been sent to the office administrator
Alternative Courses:	<p>If the user wishes to propose a recurring event [an event like one that has previously happened] (occurs after step 2)</p> <ol style="list-style-type: none"> 3a. User selects the “Propose Recurrent Event” option 4a. System presents prior events 4b. User selects the event they wish to propose again 5a. User enters the updated event date 5b. User selects the “Save” option

	<p>Continue at Step 6</p> <p>User enters invalid information or does not enter required criteria (occurs after step 3):</p> <p>3a. System prompts error message “Invalid criteria” and highlights the section(s) of the form in question.</p> <p>3b. User corrects invalid criteria (enters missing info, enters valid date/time/location)</p> <p>Continue at step 4, repeats until all criteria has been met</p>
Exceptions:	<p>If the user wants to cancel their event proposal (occurs at any time)</p> <ol style="list-style-type: none"> 1. User selects the “Cancel” option 2. System prompts “Are you sure you want to cancel?” 3. User selects “yes” 4. System cancels the event proposal and returns to main menu
Priority:	High

Title:	Add Document
Description:	This use case describes the process of general staff adding documents, such as liability forms and community partner agreements to events.
Primary Actor:	General Staff
Trigger:	A staff member has proposed an event and the appropriate documents must be added for volunteers to complete prior to being able to participate in the event.
Preconditions:	<ol style="list-style-type: none"> 1. User is logged into the system 2. User has navigated to the main menu
Normal Course:	<ol style="list-style-type: none"> 1. User selects “Add Document” option 2. System presents the “Add Document” dialogue, prompts user to search from an event

	<ol style="list-style-type: none"> 3. User selects the event that they wish to add a document(s) to 4. User adds documents through “Add File” option 5. System validates form and confirms upload with user 6. User selects “Save Document(s)” 7. Systems updates database, saves doc’s to event proposal
Postconditions:	<ol style="list-style-type: none"> 1. Documents have been saved to the event 2. Documents are available for volunteers to download, print, and complete
Alternative Courses:	<p>If General Staff needs to remove a document that was previously uploaded (occurs at step 4)</p> <ol style="list-style-type: none"> 4a. User removes a document through “Remove File” option 5a. System confirms “Remove File” option with user (User continues at step 4)
Exceptions:	<p>The user wishes to cancel their document upload (Occurs at step 5)</p> <ol style="list-style-type: none"> 1. System validates document upload and confirms with user with “Are you sure you want to upload these documents?” 2. User selects “Cancel” 3. System terminates use case
Priority:	High

Title:	Approve/Deny Event
Description:	This use case describes the process of approving/denying proposed events.
Primary Actor:	Coordinator
Trigger:	A New Event Proposal has been submitted for Approval.
Preconditions:	<ol style="list-style-type: none"> 1. User is logged into the system 2. User has navigated to the main menu

Normal Course:	<ol style="list-style-type: none"> 1. User selects “Active Event Proposals” option 2. System displays “Active Event Proposals” dialogue 3. User selects the event proposal they wish to approve or deny 4. System displays event proposal information and documents 5. User selects “Approve Event”/“Deny Event” 6. System confirms selection with user 7. User confirms event by selecting “Confirm Event Approval”/“Confirm Event Denial” 8. System updates database 9. System sends an Approved/Denied email notification to the user. 9a. Invoke use case “Update Calendar”, if event approved
Postconditions:	<ol style="list-style-type: none"> 1. New Event Proposal has been Approved/Denied. 2. If approved, the event is added to the calendar.
Alternative Courses:	None, as this use case is expected to be either approved or denied, and account for user error in selecting events. If the Coordinator later decides to cancel the event, they can do so, see Edit Event use case.
Exceptions:	<p>If the user wishes to terminate their progress in approving or denying an event... (Occurs at any time)</p> <ol style="list-style-type: none"> 1. User selects the “Cancel” option 2. System presents “Are you sure you want to cancel your progress?” 3. User selects “Yes” 4. System cancels event approval/denial progress
Priority:	Moderate

Title:	Update Calendar
Description:	This use case describes the process of adding an event to the calendar after being approved.

Primary Actor:	The System
Trigger:	<ol style="list-style-type: none"> 1. An event has just been approved by a coordinator. 2. Clock time reaches 12:00am
Preconditions:	<ol style="list-style-type: none"> 1. The system is online 2. The event was approved by a coordinator. ("Approve/Deny Event" use case)
Normal Course:	<ol style="list-style-type: none"> 1. System queries for all new event approval(s). 2. System reviews event approval(s), analyzing form(s) 3. System updates calendar in database, blocking off dates/times for the approved event(s)
Postconditions:	<ol style="list-style-type: none"> 1. Calendar is up to date and currently reflects all approved events.
Alternative Courses:	<p>Staff member needs to manually update the calendar (occurs before step 1):</p> <ol style="list-style-type: none"> 1a. Staff member logs into the system and navigates to main menu 1b. Staff member selects "Update Calendar" 1c. System prompts list of all new event approvals 1d. Staff member selects event(s) to add to the calendar. 1e. System adds event information to the calendar.
Exceptions:	<p>If System finds no new event approval during daily check (occurs after step 1):</p> <ol style="list-style-type: none"> 1. System query reveals no new event approvals 2. System ends use case
Priority:	High

Title:	Search Events
Description:	This use case describes the process of users searching for events and event information.
Primary Actor:	General Staff, Coordinator, Volunteer

Trigger:	User requires more information about the certain event(s).
Preconditions:	<ol style="list-style-type: none"> 1. User is logged into the system 2. User has navigated to main menu 3. Event has been approved, and updated in database and calendar
Normal Course:	<ol style="list-style-type: none"> 1. User selects "Search Events" 2. System prompts a search criteria look-up tool: <ul style="list-style-type: none"> Optional Filters <ol style="list-style-type: none"> a. Title b. Description c. Category of volunteer work d. Event host/creator e. Date/Time f. Partner agency 3. User enters what ever event identifiers they have into the look-up tool and selects "search" 4. System retrieves and prompts user with event information.
Postconditions:	<ol style="list-style-type: none"> 1. System has prompted user with desired event information.
Alternative Courses:	<p>If user enters criteria that has no match in the database (occurs after step 3)</p> <ol style="list-style-type: none"> 3a. System prompts user with error message "could not find results with search criteria, verify there are no spelling errors" 3b. User enters corrections and hits "search". 3c. Repeat until correct criteria is given <p>Continues at Step 4.</p>
Exceptions:	<p>If user selects to cancel (occurs any any step)</p> <ol style="list-style-type: none"> 1. System prompts message: "are you sure?" 2. User selects "yes" 3. System navigates back to Main menu.
Priority:	Moderate

Title:	Edit Event
Description:	This use case describes the process of general staff or coordinators editing an event.
Primary Actor:	General Staff, Coordinator
Trigger:	Event details have changed from the submission of the initial event proposal.
Preconditions:	<ol style="list-style-type: none"> 1. User is logged into the system 2. User has navigated to the main menu 3. At least one event proposal has been submitted to the system
Normal Course:	<ol style="list-style-type: none"> 1. User selects the "Edit Event" option 2. System presents the user with all available event proposals and approved events that have not occurred. 3. User selects the event or proposal that they want to edit 4. System prompts user to edit event details/proposal forms 5. User makes necessary changes (changes date, changes approval status, etc) and selects "save" 6. System updates database.
Postconditions:	<ol style="list-style-type: none"> 1. User's changes to the event/proposal have been saved to the database.
Alternative Courses:	<p>If the coordinator chooses to cancel an event that has already been approved (occurs after step 4)</p> <ol style="list-style-type: none"> 4a. User selects the option to "cancel event" 4b. System prompts user for coordinator cal poly account credentials. 4c. User enters email and password. 4d. System prompts a confirmation message "Are you sure?" 4e. User selects "yes" <p>Continue at Step 6</p>

Exceptions:	<p>If user selects to exit the “edit event” process (occurs at any step)</p> <ol style="list-style-type: none"> 1. User selects “x” 2. System prompts message: “are you sure?” 3. User selects “yes” 4. System navigates back to Main menu.
Priority:	Moderate

Title:	View Calendar
Description:	This use case describes the process of users viewing the calendar to see upcoming events.
Primary Actor:	General Staff, Coordinator, Volunteers
Trigger:	Users would like to know what events are scheduled this month.
Preconditions:	<ol style="list-style-type: none"> 1. User is logged into the system 2. User has navigated to the main menu
Normal Course:	<ol style="list-style-type: none"> 1. User scrolls down web page to view calendar. 2. System displays up-to-date calendar on home page 3. User hovers cursor over a certain date 4. System prompts a window with an expanded view of the day’s itinerary
Postconditions:	<ol style="list-style-type: none"> 1. User has obtained quick information regarding events by date.
Alternative Courses:	<p>If user wants more information regarding events scheduled for other months (occurs after step 2)</p> <p>2a. User selects prev/next arrow (next to the month title) above the calendar.</p> <p>2b. System loads prev/next month on calendar. Repeat step until gets to desired month.</p> <p>Continue at Step 3</p> <p>If calendar is not up to date (after step 1)</p>

	1a. System displays calendar on home screen 1b. System prompts a caption "Calendar may not reflect most current schedule" Continue at Step 3.
Exceptions:	If user no longer needs to view calendar (occurs at any step) 1. Navigates to different part of the webpage
Priority:	High

Title:	Event Sign Up
Description:	This use case describes the process of volunteers signing up for an event.
Primary Actor:	Volunteer
Trigger:	A volunteer wants to sign up for an event.
Preconditions:	1. User is logged in to the system 2. User has navigated to the main menu 3. Event(s) the volunteer selects have been approved by a coordinator.
Normal Course:	1. User selects "Volunteer" option 2. System presents approved events with "Sign Up" option next to each 3. User selects "Sign Up" option on the desired event 4. System confirms with user "Are you sure you want to sign up for (insert event title)" 5. User selects "Yes" option 6. System prompts "Event Sign Up" form: <ul style="list-style-type: none"> ○ Name ○ Phone Number ○ Year (Freshman, Sophomore, etc) ○ Email ○ Do you need a ride to the event?

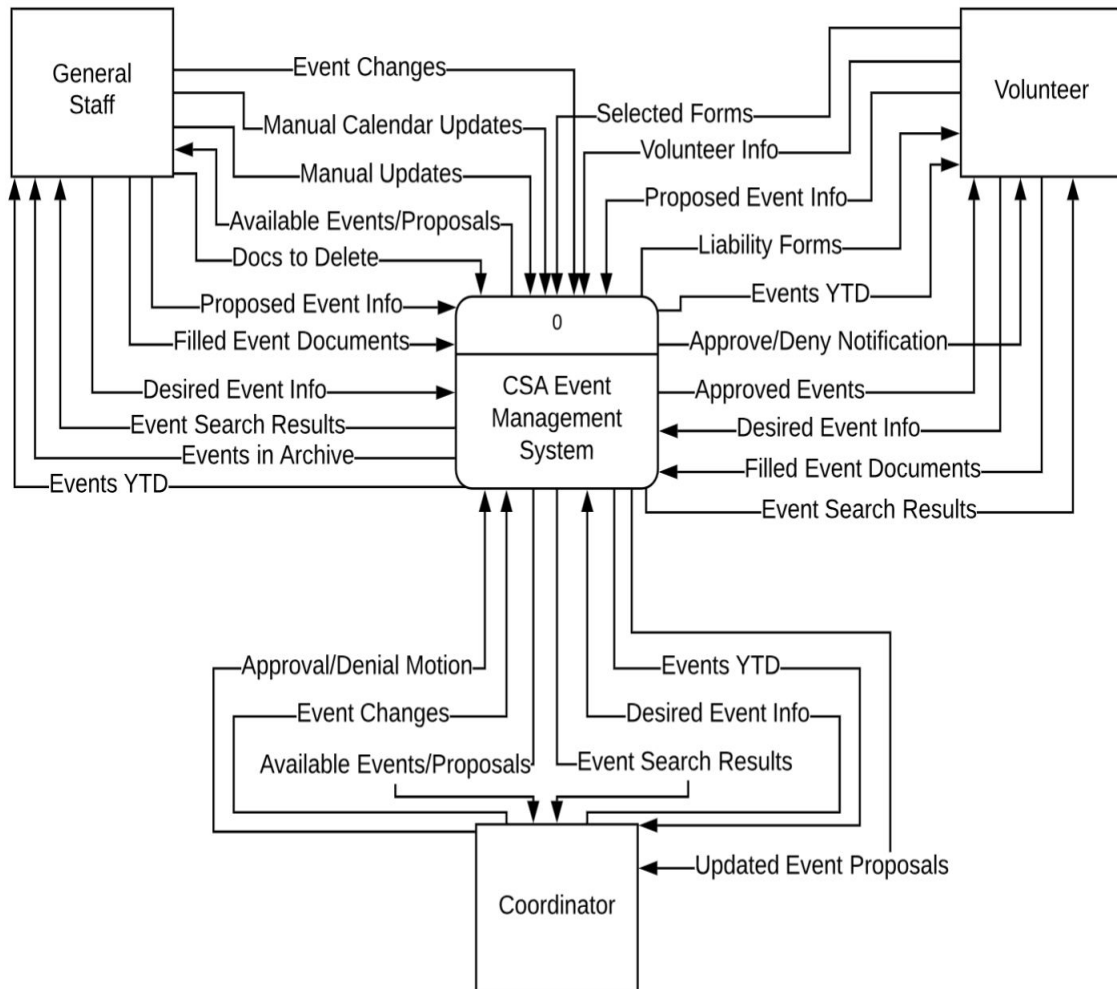
	<ul style="list-style-type: none"> i. System displays volunteers signed up as drivers <ul style="list-style-type: none"> ○ Can you drive people to/from the event? (Y/N) ○ If "Y" <ul style="list-style-type: none"> i. Number of seats available <ol style="list-style-type: none"> 7. User selects "Confirm Information" option 8. System verifies form 9. System saves volunteer(s) to the event in the database 10. User is taken to the event page 11. Invoke use case "Download Liability Form"
Postconditions:	<ol style="list-style-type: none"> 1. Volunteer is signed up for the desired event 2. If volunteer is able to drive, they are listed as "Driver" in volunteer sign up for the event
Alternative Courses:	<p>User enters invalid information or does not enter required criteria (occurs after step 7):</p> <ol style="list-style-type: none"> 7a. System prompts error message "Invalid criteria" and highlights the section(s) of the form in question. 7b. User corrects invalid criteria (enters missing info, enters valid phone and email) <p>Continue at step 8, repeats until all criteria has been met</p> <p>Event is full and no more volunteers can sign up (occurs after step 3)</p> <ol style="list-style-type: none"> 4a. System displays "Event is full" dialogue 5a. User is returned to the approved events list (continue at step 3 of normal course)
Exceptions:	<p>If the user wishes to cancel (occurs at any time)</p> <ol style="list-style-type: none"> 1. User selects the "Cancel" option 2. System presents "Are you sure you want to cancel your progress?" 3. User selects "Yes" 4. System cancels sign up and returns to main menu
Priority:	Moderate

Title:	Download Liability Form
Description:	This use case describes how volunteers complete a liability release form for an event they signed up for
Primary Actor:	Volunteer
Trigger:	A volunteer has signed up for an event on the CSA Event Management System
Preconditions:	<ol style="list-style-type: none"> 1. User is logged into the system 2. User is signed up for an event
Normal Course:	<ol style="list-style-type: none"> 1. User selects "Download Liability Forms" option 2. System prompts user to click the files they wish to download 3. System prompts user to enter how many copies of each they would prefer. 4. User selects "download selected forms" option 5. System notifies user that "Forms Successfully Downloaded"
Postconditions:	<ol style="list-style-type: none"> 1. User has downloaded the liability forms required to participate in an event
Alternative Courses:	<p>Download error (bad user internet connection) (occurs after step 4)</p> <p>4a. System prompts "unable to download forms, check your connection and try again", repeat as needed</p> <p>Continue at Step 5</p>
Exceptions:	<p>System unable to provide documents (bad system connection) (occurs after step 1)</p> <p>1a. System prompts message "Unable to display at this time"</p> <p>1b. System ends use case</p>
Priority:	Moderate

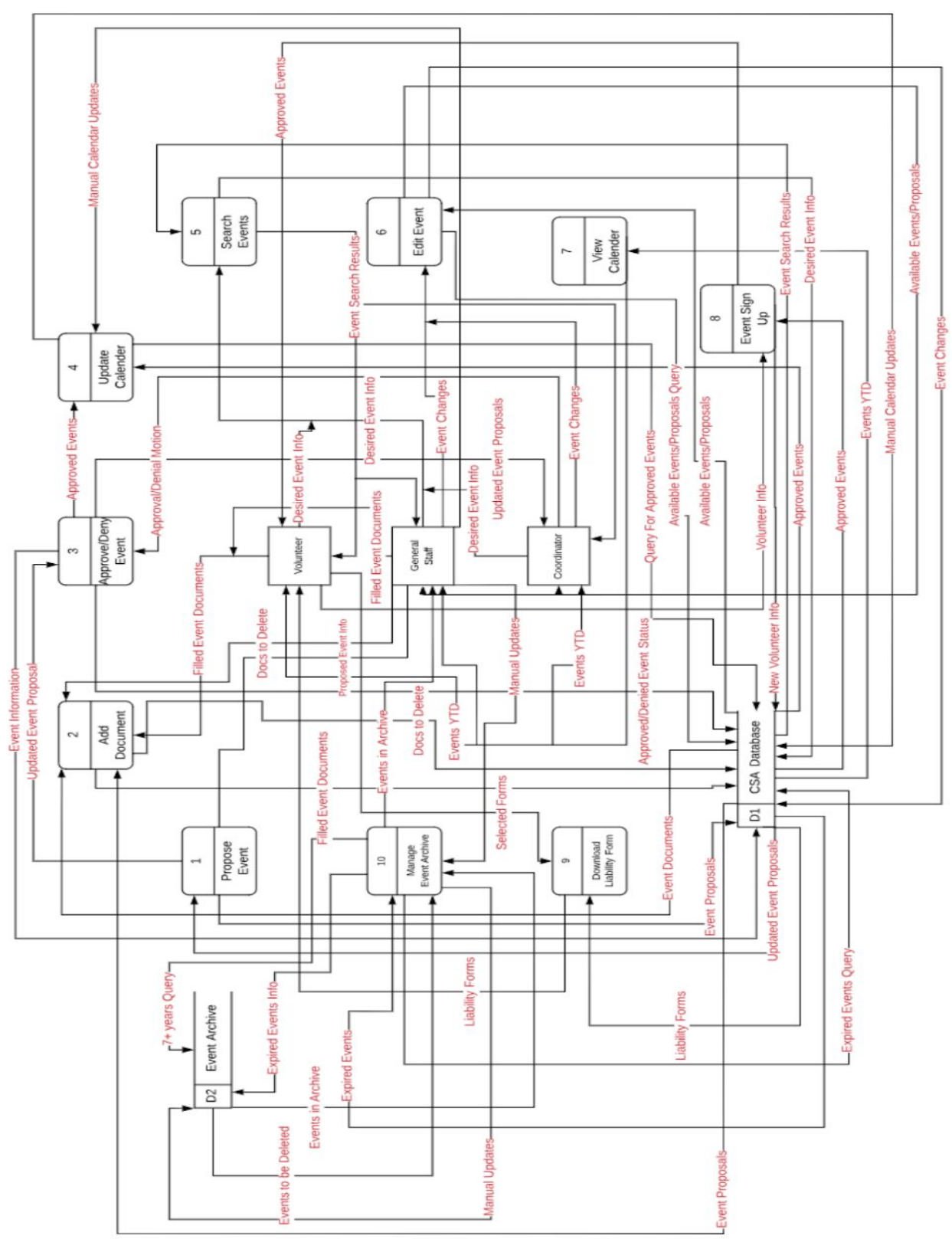
Title:	Manage Event Archive
Description:	This use case describes the process of archiving events after they occur and removing ones that are older than 7 years
Primary Actor:	The System
Trigger:	<ol style="list-style-type: none"> 1. An event date has passed and needs to be archived for documentation purposes 2. Clock time reaches 12:00 am
Preconditions:	<ol style="list-style-type: none"> 1. The system is online 2. The event was approved by a coordinator 3. The event date has passed
Normal Course:	<ol style="list-style-type: none"> 1. Daily, system queries all events with dates that have passed 2. System sends all event and volunteer information to the "Event Archive" database <ol style="list-style-type: none"> a. Title b. Event host/creator c. Date d. Partner agency e. Volunteers <ol style="list-style-type: none"> i. Name ii. Phone Number iii. Year (Freshman, Sophomore, etc) 3. System updates "Event Archive" database with expired event information 4. System queries "Event Archive" database for any events older than 7 years and deletes them from the database.
Postconditions:	<ol style="list-style-type: none"> 1. The event is now archived in the "Event Archive" data store for a duration of 7 years 2. Event archive is up to date, only has records from events within 7 years ago

Alternative Courses:	<p>Staff member needs to manually update event archive (occurs before step 1):</p> <ol style="list-style-type: none"> 1a. Staff member logs into the system and navigates to main menu 1b. Staff member selects "Event Archive" 1c. System prompts list of all current events in archive at the time 1d. Staff member selects to: add event(s) from database / remove events from the "Event Archive" 1e. System verifies selection with user specifying "actions can not be undone" 1f. User selects "confirm" <p>Continue at Step 3</p>
Exceptions:	<p>If staff member wants to cancel manually updating the event archive (occurs at any point after step 1b in alt. course)</p> <ol style="list-style-type: none"> 1. User selects the "Cancel" option 2. System prompts "Are you sure you want to cancel your progress?" 3. User selects "Yes" 4. System returns to main menu
Priority:	Low

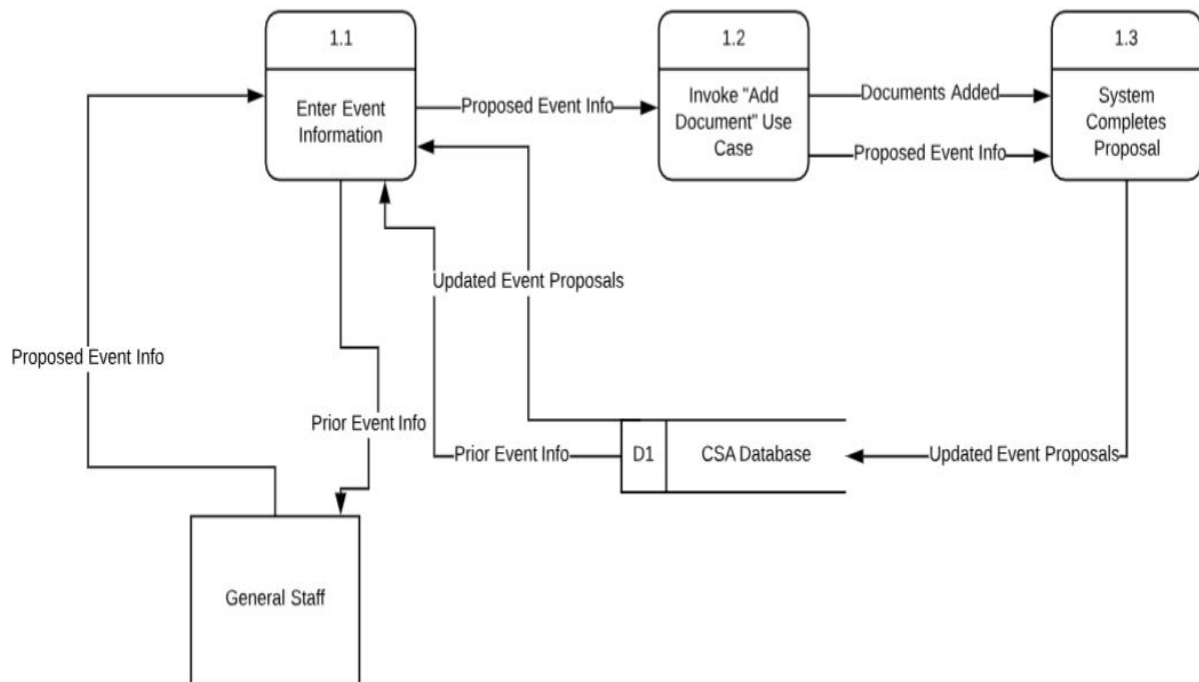
Context Diagram



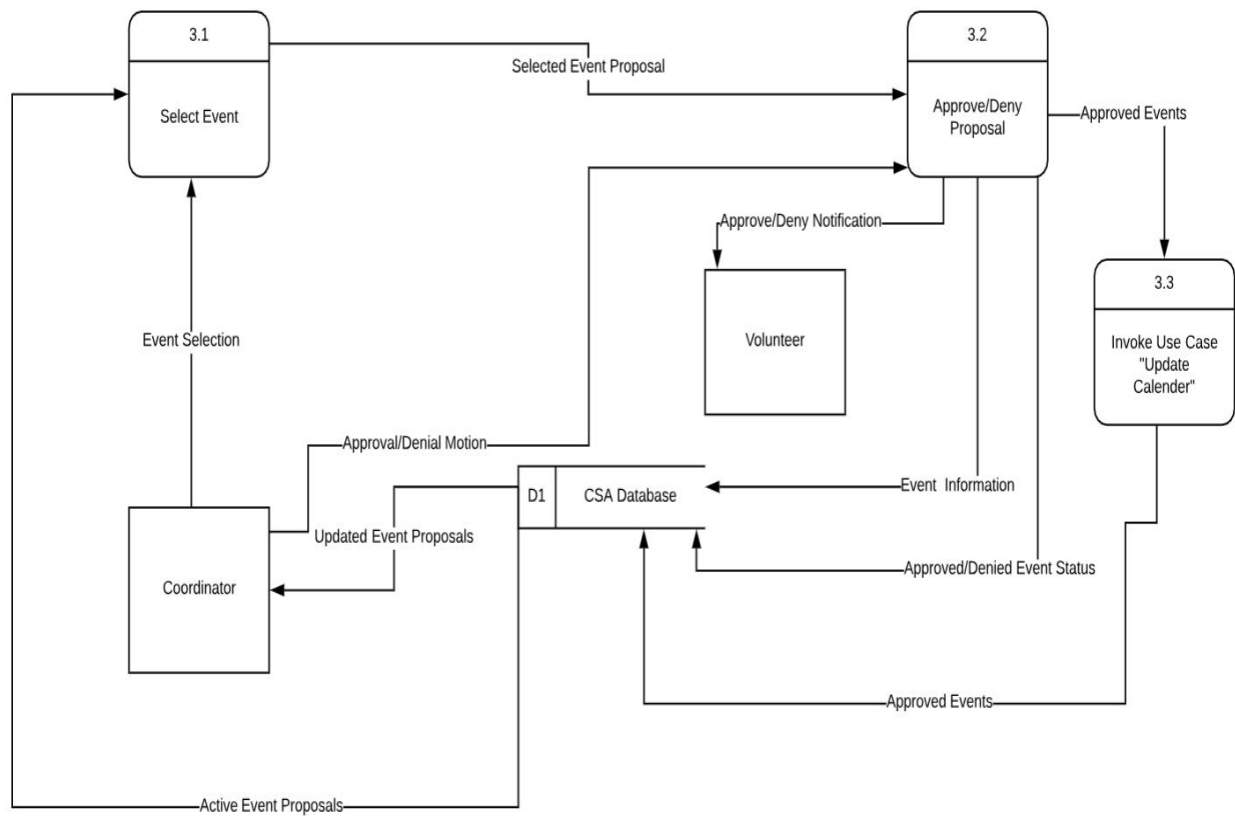
Level 0 Diagram



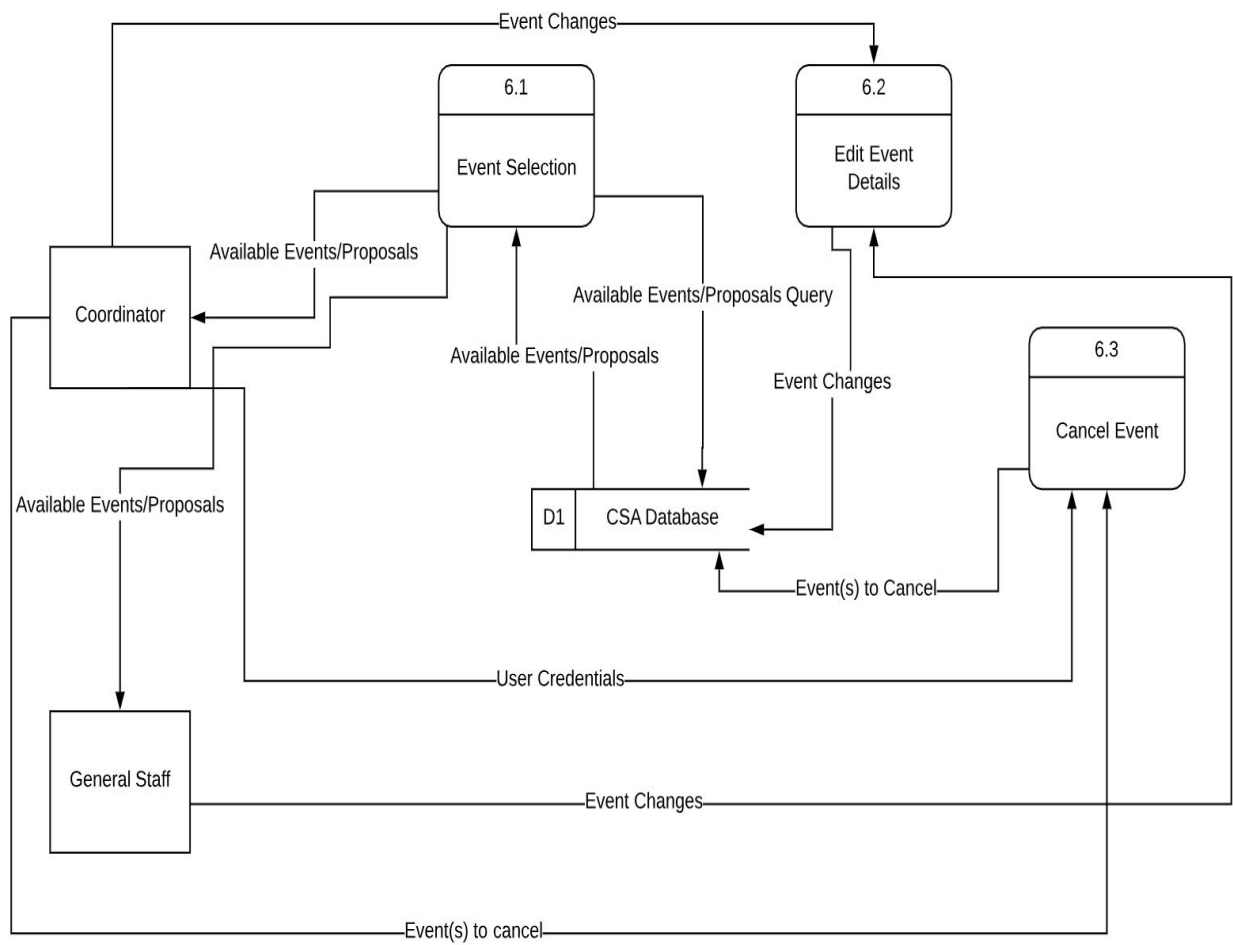
Level 1 Diagram - Propose Event



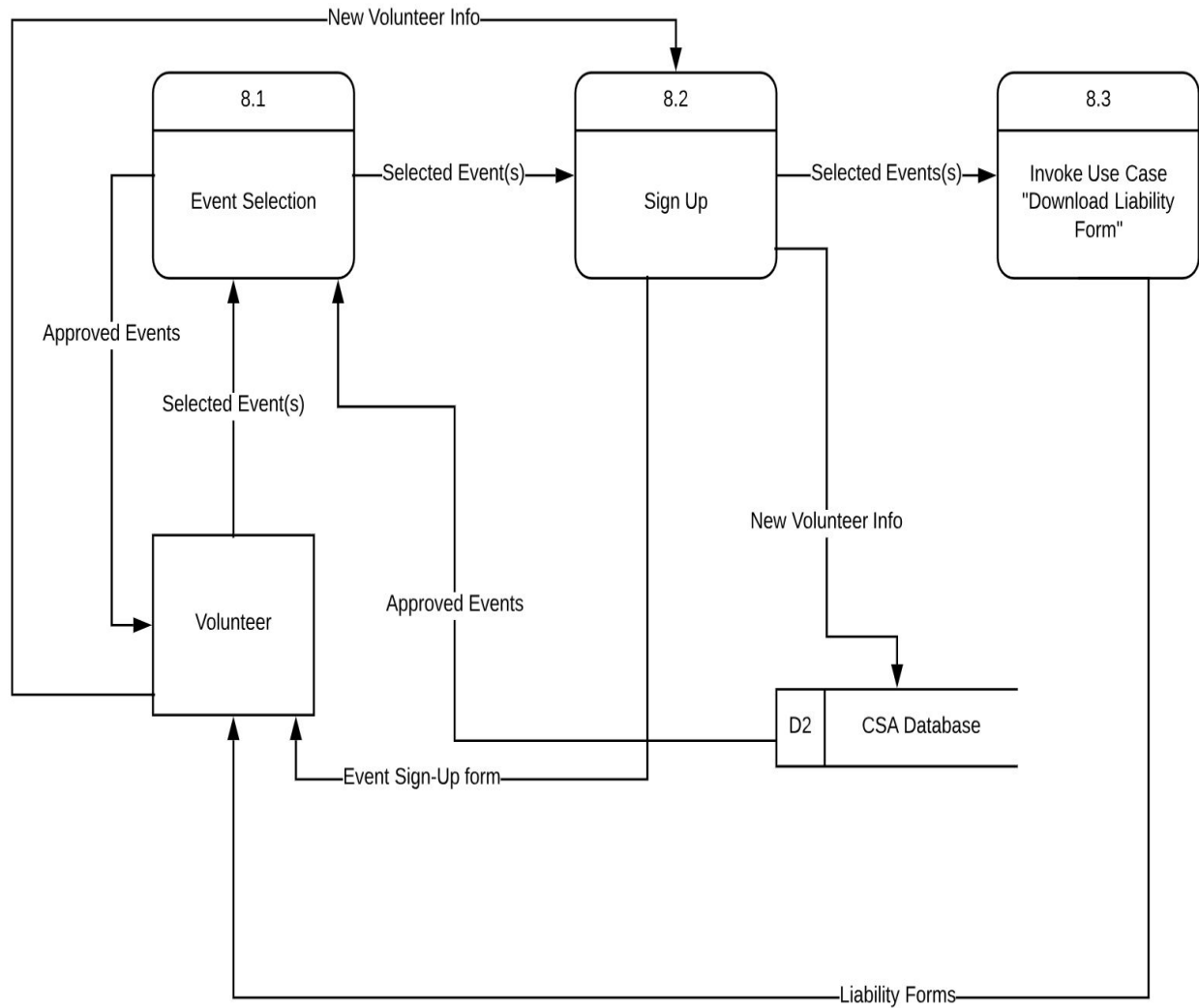
Level 1 Diagram - Approve/Deny Event



Level 1 Diagram - Edit Event



Level 1 Diagram - Approve/Deny Event



Discussion of Proposed Solution

Given the needs of the Center for Service in Action (CSA) and their current lack of a structured information technology system, we propose the creation of an event management system to streamline the processes associated with planning and maintaining events. This system will offer specific functionality depending on whether the user is a staff member of the CSA or a student volunteer. The proposed system is a website, which will allow staff members and volunteers alike to easily access the system on a variety of devices. This is critical because some staff members might choose to work off of their personal computer or laptop rather than the desktop computers in the office. Additionally, the website will be most easily accessed by volunteers as well, especially because some volunteers may be infrequent users of the site. As such, it would be overburdensome to require these types of users to download an application that they would not use on a regular basis.

The architecture of the proposed system is a thin client, cloud-based computing system. Currently, the CSA stores its documents on Google Drive. During interviews with the CSA, the staff explained that would not be necessary for the system or any of its data to be stored locally on the office's computers. Thus, a cloud-based system is the ideal system to meet the needs of the CSA staff and the other external users of their system.

When a new user visits the site, they will be asked to designate themselves as either a staff member or volunteer. For the sake of security, staff members will be required to provide their login credentials in order to proceed. From there, the user will be presented with a unique view of the system based on their designation. Staff members will be presented with the event proposal database with the capabilities of creating new events, editing events, and approving/denying events. Meanwhile, student volunteers will be presented with the CSA calendar, which shows approved events that they can sign up for. Student volunteers will also have the option to download liability forms prior to the event so that they can print and sign the documents before an event.

The proposed system will have the capacity to store event data and documents for up to seven years after the event has occurred. This is the general rule of thumb used for storing data in non-profit work, and was a requirement discussed by the CSA. In order to increase office efficiency, the proposed system will notify staff members when event have been proposed so that the office coordinator can quickly approve or deny the proposal. Furthermore, the proposed system would automatically add events to the public calendar in order to optimize sign up time.

Design - Prototype Mockups

GENERAL LOG IN


The mockup is a web browser window with the address bar showing `https://serviceinaction.calpoly.edu/SCS`. The main content area has a light beige background. On the left, a grey box contains the text "Welcome back to" in a script font, followed by the Center for Service in Action logo (a stylized 'G' with a 'P' inside) and the text "CENTER FOR SERVICE IN ACTION". To the right, a group photo of students is shown above the text "HOW WOULD YOU LIKE TO PROCEED?". Below this text are two green buttons: "STAFF" and "VOLUNTEER".

Below the "STAFF" button, a "STAFF LOGIN" modal is displayed. It has a title bar with a close button (X). Inside, there are two input fields: "username" and "password", both with green labels. Below the fields is a green "SUBMIT" button.

Below the "VOLUNTEER" button, a yellow note box contains the text: "System will present volunteer: homepage to user. No login is required to proceed."

STAFF VIEW: HOMEPAGE

← → ↻ 🏠 <https://serviceinaction.calpoly.edu/SCS>

 CENTER FOR SERVICE IN ACTION [Katie Ullrich](#) | [Log Out](#)

HOME PROPOSE EVENT EVENT ARCHIVE

🔍

March 2019

S	M	T	W	Th	F	S
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

NOTIFICATIONS

- Event proposed on 2/21/19 [OPEN](#)
- Event proposed on 2/23/19 [OPEN](#)
- Next event on 3/8/19 [OPEN](#)

INVALID EVENT ✕

The event you entered does not exist. Please use other search criterion.

ST. PATRICKS DAY

Volunteer at a St. Patricks Day-themed event with cookie decorating, games and all things St. Patrick's Day.

[EDIT EVENT](#)

GROWING GROUNDS WORKDAY: ACTION REQUIRED

Event Title <input type="text" value="Growing Grounds Workday"/>	Event Host/Creator <input type="text" value="Monique Rea"/>
Event Description <input type="text" value="Join SCS to volunteer at Growing Grounds Farm. Fun and friendship will be involved!"/>	Community Partner <input type="text" value="Transitions Mental Health Association"/>
Category of Volunteer Work <input type="text" value="Health"/>	Transportation Type <input type="text" value="Carpool"/>
Date and Time <input type="text" value="4/5/2019 9AM to 12PM"/>	Certificate of Liability Insurance <input type="text" value="See Document"/>

[APPROVE](#) [DENY](#)

STAFF VIEW: PROPOSE EVENT

← → ↻ ⌂ https://serviceinaction.calpoly.edu/SCS

CENTER FOR SERVICE IN ACTION Katie Ullrich | Log Out

HOME PROPOSE EVENT EVENT ARCHIVE

NEW EVENT PROPOSAL

Event Title Growing Grounds Workday	Event Host/Creator Katie Ullrich
Event Description Join SCS to volunteer at Growing Grounds Farm. Fun and friendship will be involved!	Community Partner Transitions Mental Health Association
Category of Volunteer Work Health	Transportation Type Carpool
Date and Time 4/5/2019 9AM to 12PM	Certificate of Liability Insurance Upload Document

Health ▼
 Youth
 Animal Welfare
 Homelessness
 Elderly
 Environmental
 Food Insecurity
 Disabilities

Carpool ▼
 Bus
 Walking
 Biking

ERROR ✕
 All fields are required. Please complete highlighted fields and resubmit the proposal.

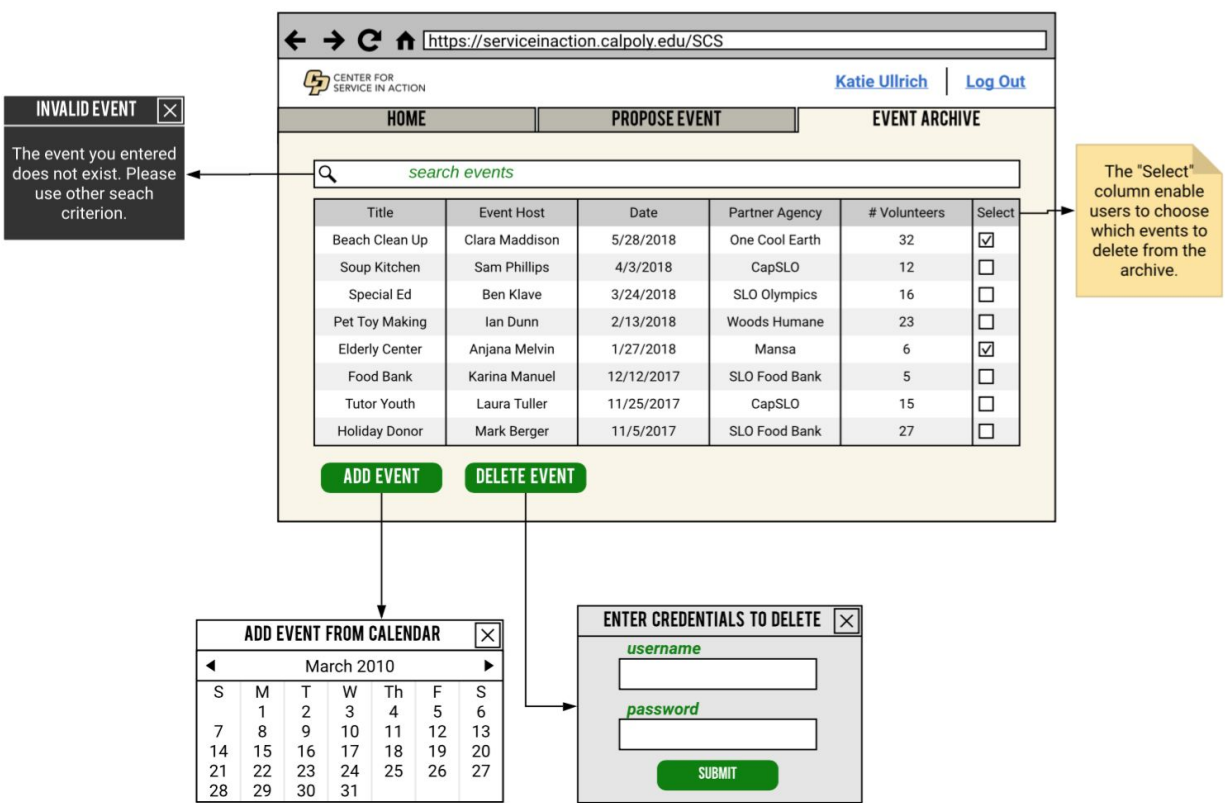
THANK YOU ✕
 Your event proposal has been submitted. The coordinator will be notified of your submission.

CHOOSE EVENT FROM CALENDAR ✕
 March 2010

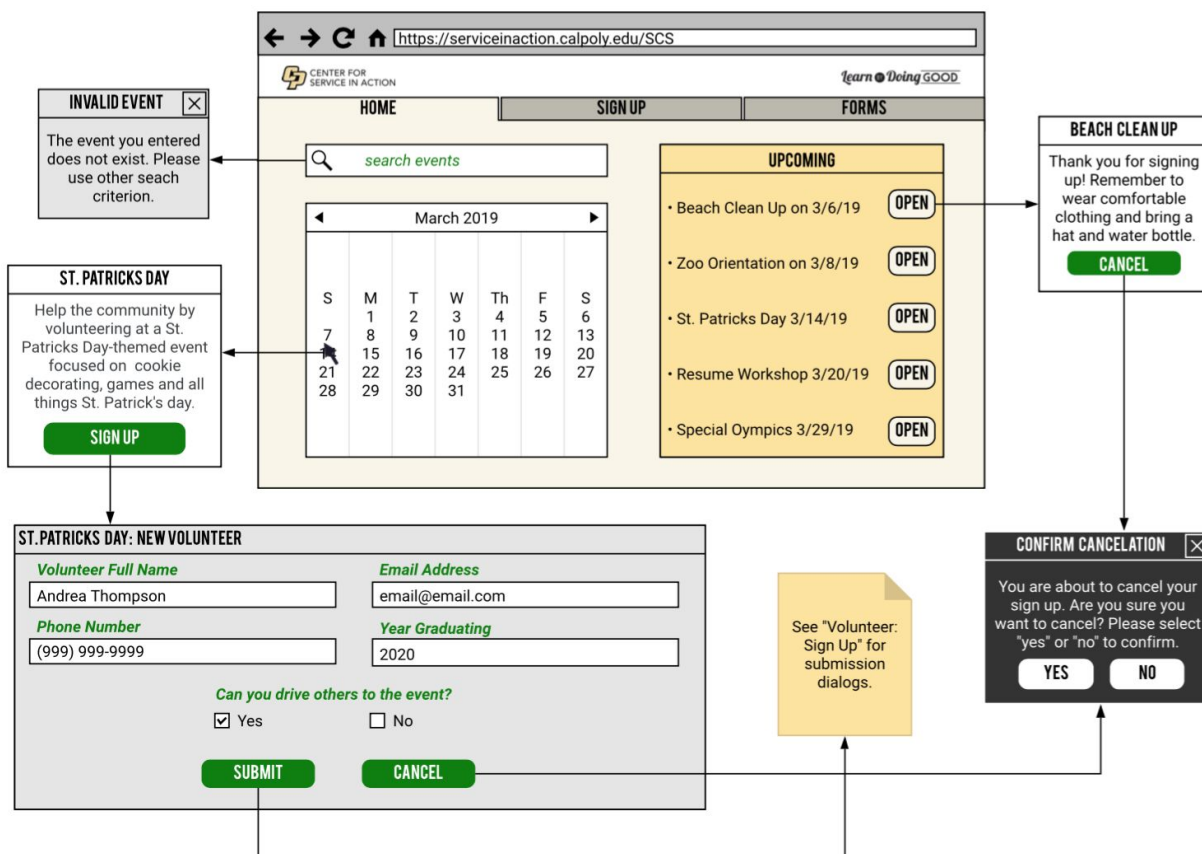
S	M	T	W	Th	F	S
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

ENTER CREDENTIALS TO CANCEL ✕
 username
 password

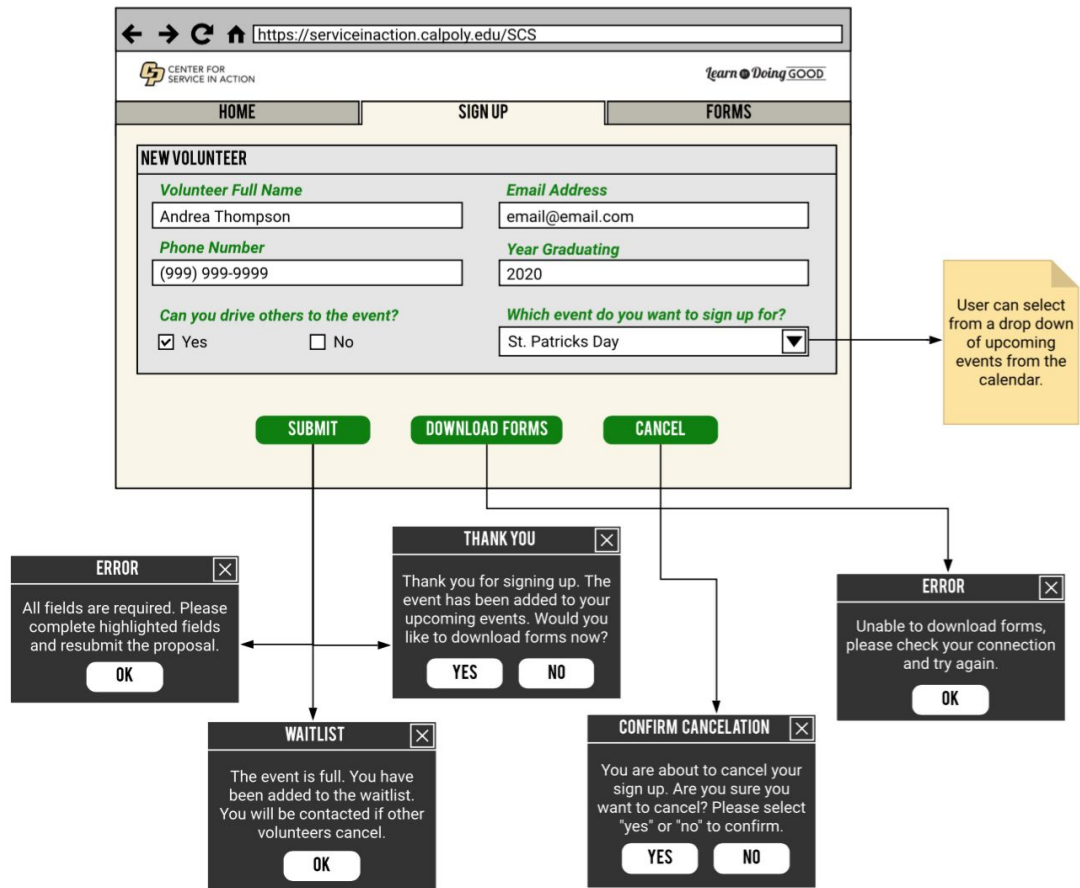
STAFF VIEW: EVENT ARCHIVE



VOLUNTEER VIEW: HOMEPAGE



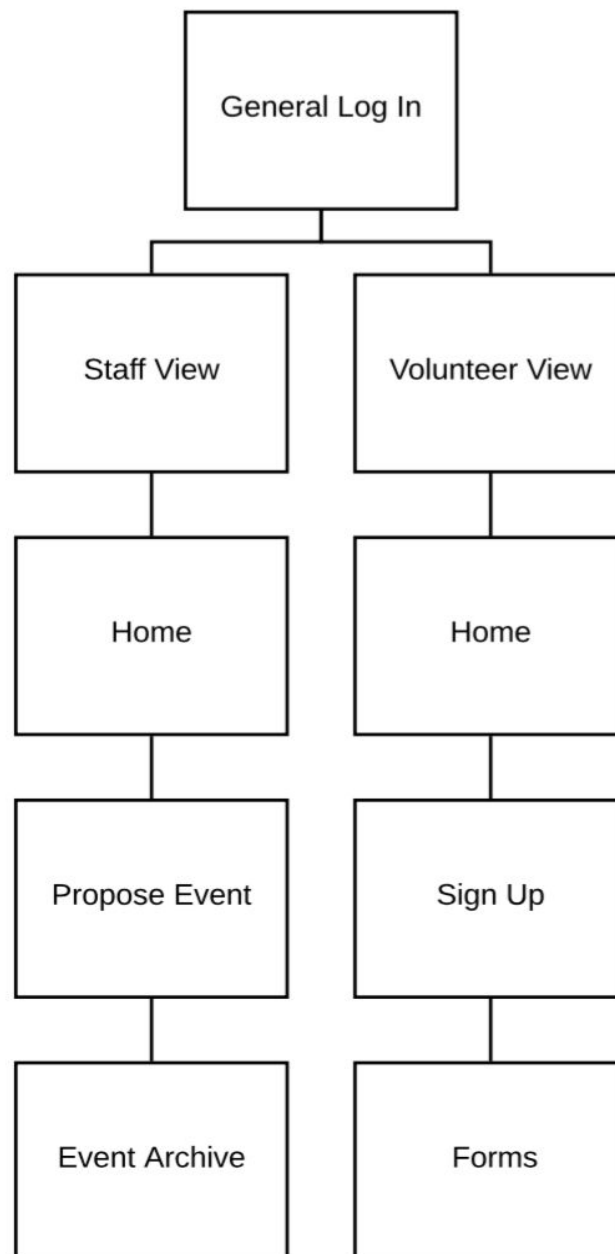
VOLUNTEER VIEW: SIGN UP



VOLUNTEER VIEW: FORMS

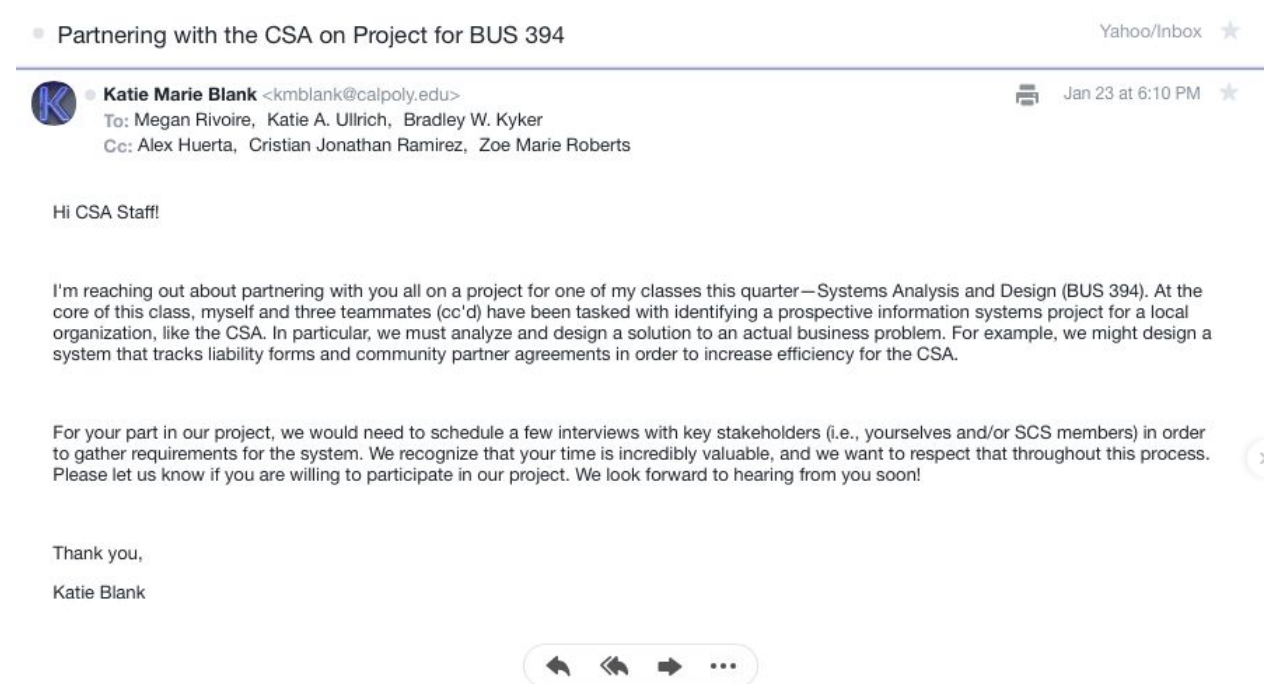
[illegible]

Interface Structure Diagram: CSA System



Appendix

On 1/23/19, our group sent the following message to the CSA staff regarding our partnership on the project. Shortly after, our group member Katie confirmed their approval for the partnership in person.



In order to fully understand and document the processes that the Center for Service in Action followed on a day to day basis, we scheduled an interview session with Katie Ollrich, SLO Americorp VIP representative, on February 19th, 2019. The group members in attendance during the interview were Katie Blank, Zoe Roberts, Alex Huerta, and Cristian Ramirez. The interview lasted approximately 30 minutes and consisted of mostly Q&A.