Iteration 5

CIS 320-01

Team: Exel Members

Simona Sofronova, Rachel Carr, James Daniel, Mason Tatafu, Nick Frondorf

# Table of Contents

1. Class Diagram

1.1 Class Diagram Narrative

This class diagram outlines the main components of our system, serving as a detailed blueprint of how processes occur and how the website handles events. Each box, or class, represents an essential part of the system. Each class represents a certain facet of the system, like Donors, Volunteers, Events, Forms, or something like the payment system. Within each class are attributes, which define its characteristics (e.g., a donor's name or a volunteer's availability), and methods, which describe actions it can perform (e.g., updating a volunteer's availability or processing a donation). The lines between these boxes are relationships, showing how these classes interact. For example, A Donor (which is a class) fills out a Donation Form (which is also a class).

Verb-noun analysis is a common practice to help show how different classes work together in an entire system. Nouns from the system description, such as Donor, Event, and Website, were identified as entities or classes. Verbs, such as "fill out," "assist," or "process," were used to define the relationships and actions associated with these classes. Cardinality (e.g.,
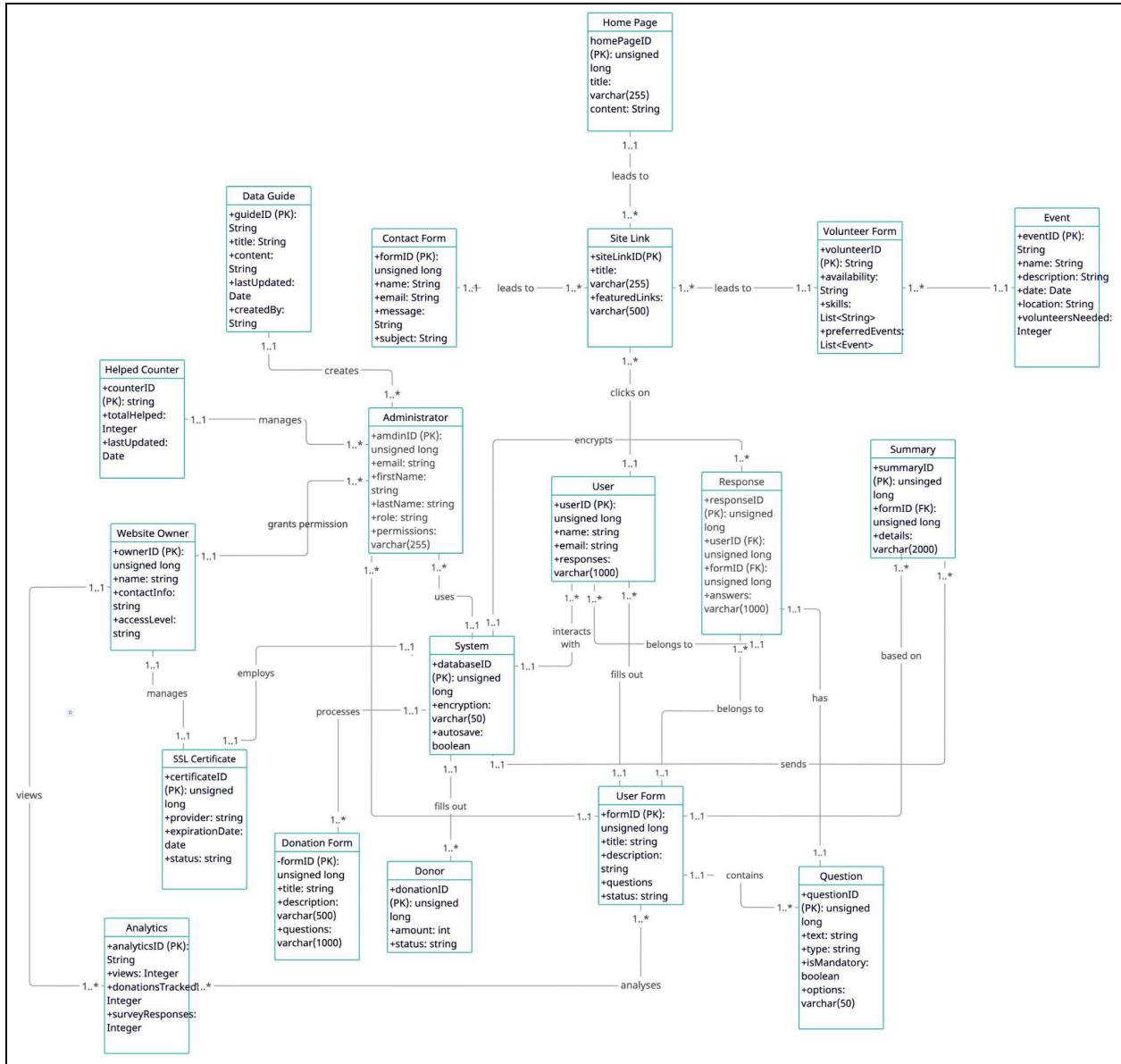
1..1 or 1..*) was added to indicate how many instances of one class relate to another. For example, A single Donor can fill out many Donation Forms is a One-to-Many (1..*) relationship, since one user can fill out many forms. This ensures a logical structure for the system, helping administrators or website developers understand key processes throughout the system.

1.2 Class Diagram

Event
+ eventID: String
+ name: String
+ description: String
+ date: Date
+ location: String
+ volunteersNeeded: Integer
+ assignVolunteer(volunteerID: String): Boolean
+ scheduleEvent(): void
+ organize(): void
+ sendNotifications(): void

1..1 — attends — *1..**

Analytics
+ analyticsID: String
+ views: Integer
+ donationsTracked: Integer
+ surveyResponses: Integer
+ generateSummary(): String
+ analyzeData(): String

Contact Form
+ name: String
+ email: String
+ message: String
+ subject: String
+ sendMessage(): Boolean
+ confirmReceipt(): void

Volunteer
+ availability: String
+ skills: List<String>
+ assignedEvents: List<Event>
+ hoursContributed: Integer
+ signUpForEvent(eventID: String): Boolean
+ logHours(eventID: String, hours: Integer): void
+ viewAssignedEvents(): List<Event>
+ updateAvailability(): void

Volunteer Form
+ volunteerID: String
+ availability: String
+ skills: List<String>
+ preferredEvents: List<Event>
+ registerVolunteer(): Boolean
+ updateVolunteerDetails(): Boolean

analyzes — receives — submits

Base Form
+ formID: String
+ title: String
+ createdBy: Administrator
+ submissionDate: Date
+ submitForm(): Boolean
+ validateForm(): Boolean
+ getFormDetails(): void
+ addQuestion(): void
+ collectResponse(): void
+ encryptResponses(): void

SSL
+ sslID: String
+ certificateType: String
+ validFrom: Date
+ validTo: Date
+ issuingAuthority: String
+ isValid: Boolean
+ validateCertificate(): Boolean
+ renewCertificate(): Boolean
+ getCertificateDetails(): String

Website Owner
+ adminID: String
+ websiteID: String
+ permissions: List<String>
+ updateWebsite(): void
+ manageUsers(): void
+ manageContent(): void
+ analyzeTraffic(): void

Donor
+ donationHistory: List<Donation>
+ recurringDonations: Boolean
+ donationPreferences: String
+ preferredPaymentMethod: String
+ donate(amount: Decimal): Boolean
+ setRecurringDonation(amount: Decimal, frequency: String): Boolean
+ modifyRecurringDonation(): void
+ viewDonationHistory(): List<Donation>

subscribes to

Subscription
+ subscriptionID: String
+ type: String
+ startDate: Date
+ endDate: Date
+ activate(): void
+ deactivate(): void

views — implements — coordinates — receives updates

Response
+ responseID: String
+ questionID: String
+ userID: String
+ timestamp: DateTime
+ content: String
+ submit(): void
+ update(): void

collects

Social Media Widget
+ widgetID: String
+ platform: String
+ feedURL: String
+ updateFrequency: Integer
+ isActive: Boolean
+ fetchLiveFeed(): List<String>
+ displayFeed(): String
+ deactivateWidget(): Boolean
+ activateWidget(): Boolean

oversees — configures

Donation Form
+ donationID: String
+ donorID: String
+ amount: Decimal
+ date: Date
+ isRecurring: Boolean
+ paymentMethod: String
+ submitDonation(): Boolean
+ validateDonation(): Boolean
+ processDonation(): Boolean
+ modifyRecurring(): void

fills out — used by

Square (Payment System)
+ squareID: String
+ apiKey: String
+ transactionLimit: Decimal
+ supportedCurrencies: List<String>
+ transactionFee: Decimal
+ processPayment(amount: Decimal, donorID: String): Boolean
+ validateTransaction(transactionID: String): Boolean
+ refundTransaction(transactionID: String): Boolean

Question
+ questionID: String
+ text: String
+ responseType: String
+ validateResponse(): Boolean

answers — fill out — asks — superintends

Administrator
+ adminID: String
+ role: String
+ manageSubscription(): void
+ onboardUser(): void
+ generateReport(): void
+ updateWebsite(): void

Help Counter
+ counterID: String
+ totalHelped: Integer
+ lastUpdated: Date
+ incrementCounter(count: Integer): Boolean
+ resetCounter(): Boolean
+ getHelpedCount(): Integer

manages — embeds — creates — modifies

Event Registration Form
+ eventID: String
+ userID: String
+ participationType: String
+ dietaryPreferences: String
+ registerForEvent(): Boolean
+ updateRegistration(): Boolean

Base User
+ userID: String
+ firstName: String
+ lastName: String
+ email: String
+ address: String
+ phone: String
+ role: String
+ createAccount(): void
+ logIn(): void
+ viewHistory(): void
+ submitSupportRequest(): void

completes — assists

Support
+ supportID: String
+ userID: String
+ requestDate: DateTime
+ description: String
+ provideResolution(): String

Guide
+ guideID: String
+ title: String
+ content: String
+ lastUpdated: Date
+ createdBy: String
+ createGuide(title: String, content: String): Boolean
+ updateGuide(content: String): Boolean
+ deleteGuide(): Boolean
+ getGuideDetails(): String

uses

Website
+ websiteID: String
+ domain: String
+ sslConfig: String
+ updateContent(): void
+ managePages(): void

2. Database Design

2.1 Database Diagram

2.2 Database Narrative

This Entity-Relationship-Diagram (ERD) is based on our proposed system. It includes many entities, each with attributes and relationships. Users interact with the website, and those interactions create desired responses from the system. Administrators manage data, permissions, and encryption via the system database. Volunteer and Donation info are tracked and analytics are created based on user activity. The structure allows for data management and analytics tracking with room for more functionality and growth.

2.3 Database Definitions

Data guide

Attribute	Definition	Type	Size	Key
Guide Id	Primary key for entry	int	4 bytes	primary
Title	The title of the entry	string	255 bytes	-
Last updated date	The date updated	date	3 bytes	-
Createdby	Who it was created by	string	255 bytes	-

Contact Form

Attribute	Definition	Type	Size	Key
Name	Name of the contactee	string	255 bytes	-
Email	Email of contactee	string	255 bytes	-
Message	Message being sent	string	255 bytes	-
Subject	Message subject	string	255 bytes	-

Home Page

Attribute	Definition	Type	Size	Key
Home page id	Home page id	int	4 bytes	primary

	num			
Title	Page title	string	255 bytes	-

Site link

Attribute	Definition	Type	Size	Key
SiteLinkId	The site id	int	4 bytes	primary
title	Title for entry	string	255 bytes	-
Featured links	Links to be displayed	string	255 bytes	-

Volunteer Form

Attribute	Definition	Type	Size	Key
VolunteerId	The id of the volunteer	int	4 bytes	primary
Availability	When they are available	date	3 bytes	-
Skills	Their skills	string	255 bytes	-
PreferredEvents	Their preferred events	string	255 bytes	-

Event

Attribute	Definition	Type	Size	Key
EventId	The event id	int	4 bytes	primary
Description	Event Desc	string	255 bytes	-

Date	Event Date	date	3 bytes	-
VolenteersNeeded	Number of volunteers needed	int	4 bytes	-

Helped Counter

Attribute	Definition	Type	Size	Key
CounterId	Id number for the counter	int	4 bytes	primary
TotalHelped	Total Number Of people helped	int	4 bytes	-
LastUpdated	When the counter was last updated	date	3 bytes	-

Administrator

Attribute	Definition	Type	Size	Key
AdminId	Admin key	int	4 bytes	primary
Email	Admin email	string	255 bytes	-
FirstName	Admin first name	string	255 bytes	-
LastName	Admin last name	string	255 bytes	-
Role	Admins role	string	255 bytes	-
Permissions	Admin specific permissions	float	4 bytes	-

User

Attribute	Definition	Type	Size	Key
UserId	Userid num	int	4 bytes	primary
Name	User name	string	255 bytes	-

| Email | User email | string | 255 bytes | - |
| Responses | User response | string | 255 bytes | - |

Response

Attribute	Definition	Type	Size	Key
ResponseId	The response id	int	4 bytes	primary
UserId	The user id	int	4 bytes	Foreign key
FormId	The forms id	int	4 bytes	Foreign key
Answer	The user answer	string	255 bytes	-

Summary

Attribute	Definition	Type	Size	Key
SummaryId	The summary id	int	4 bytes	primary
FormId	The forms Id	int	4 bytes	Foreign
Details	The forms details	string	255 bytes	-

Website Owner

Attribute	Definition	Type	Size	Key
OwnerId	The owners id	int	4 bytes	primary
Name	Owners name	string	255 bytes	-
ContactInfo	Owner Contact Info	string	255 bytes	-
AccessLevel	Owners Access Level	float	4 bytes	-

System

Attribute	Definition	Type	Size	Key

DatabaseId	**The data baseid**	**int**	**4 bytes**	**primary**
Encryption	**The encryption**	**string**	**255 bytes**	**-**
AutoSave	**The ability to save**	**string**	**255 bytes**	**-**

SSL Certificate

Attribute	Definition	Type	Size	Key
CertificateId	**SSL Certificate Id**	**int**	**4 bytes**	**primary**
Provider	**Who provides the certificate**	**string**	**255 bytes**	**-**
ExpirationDate	**When the certificate expires**	**date**	**3 bytes**	**-**
Status	**Its status**	**string**	**255 bytes**	**-**

Donation Form

Attribute	Definition	Type	Size	Key
FormId	**The forms id**	**int**	**4 bytes**	**Foreign**
Title	**The title**	**string**	**255 bytes**	**-**
Description	**The description**	**string**	**255 bytes**	**-**
Questions	**Any questions**	**string**	**255 bytes**	**-**

User Form

Attribute	Definition	Type	Size	Key
FormId	**The form id**	**int**	**4 bytes**	**Primary**
Title	**The title of the form**	**string**	**255 bytes**	**-**
Description	**Form description**	**string**	**255 bytes**	**-**

Questions	Form questions	string	255 bytes	-
Status	Form status	string	255 bytes	-

Analytics

Attribute	Definition	Type	Size	Key
AnalyticsId	Analytics id	int	4 bytes	primary
Views	The different forms of views	string	255 bytes	-
DonationsTracked	Donation Data	string	255 bytes	-
SurveryResponces	Survey Data	string	255 bytes	-

Question

Attribute	Definition	Type	Size	Key
QuestionId	The questions id	int	4 bytes	primary
Text	The displayed text	string	255 bytes	-
Type	Question type	string	255 bytes	-
IsMandatory	Wether the question is mandatory	Boolean		-
Options	What the answer options are	string	255 bytes	-
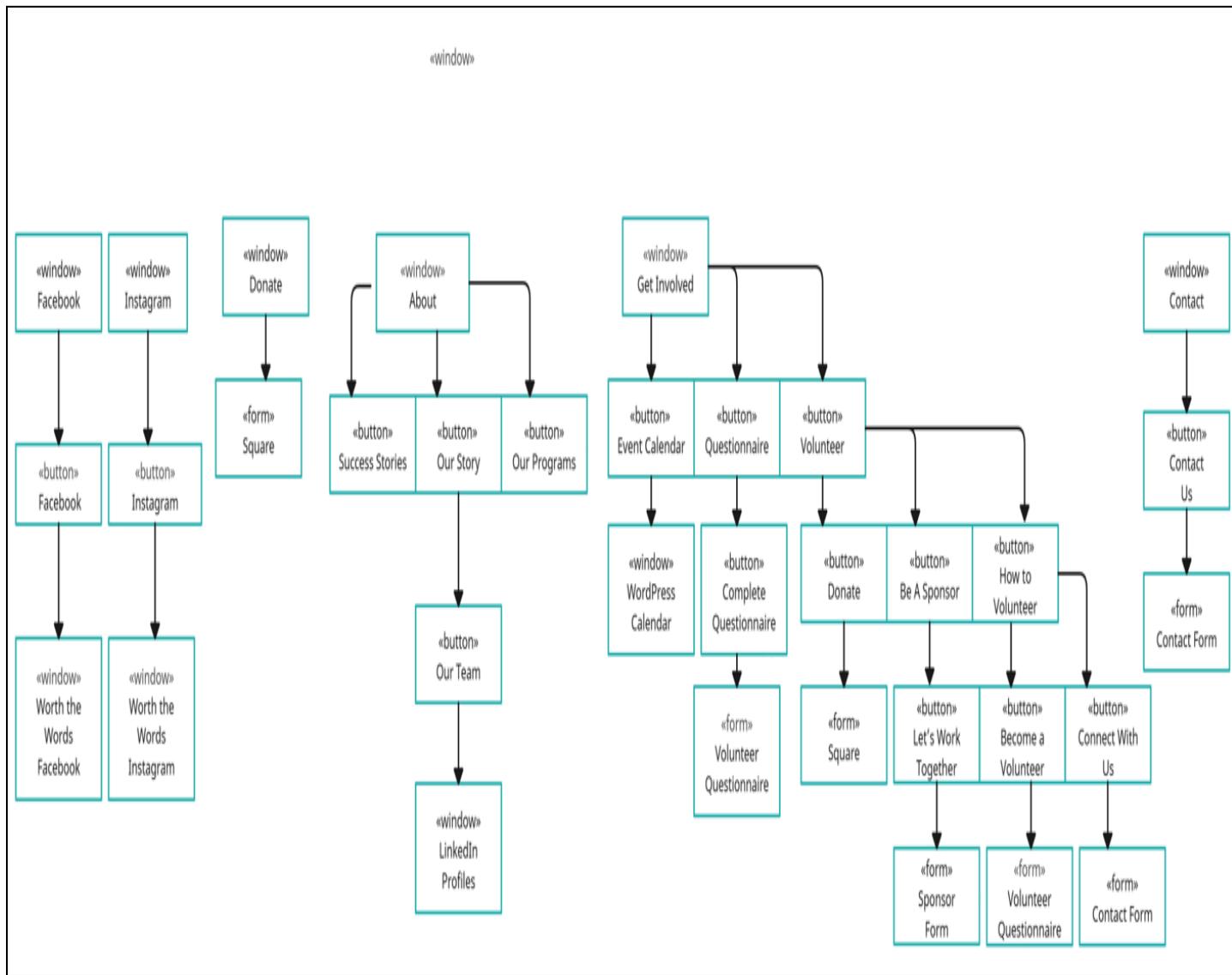
Donor

Attribute	Definition	Type	Size	Key
DonationId	The donations id	int	4 bytes	Primary
Amount	How much	float	4 bytes	–

	donated			
Status	**Donation status**	**string**	**255 bytes**	**-**

3. User Interface Design

3.1 User Interface Navigation Diagram

3.2 User Interface Diagram Narrative

The User Interface Diagram outlines the navigation system for our website's client-based view. It illustrates how users interact with the site, detailing key components and pathways. This diagram highlights user actions and the corresponding responses from the website, ensuring a clear understanding of the user journey and enhancing overall usability.

3.3 Screen Layouts

3.3.A Data Capture Forms

Become A Sponsor Form

Become A Sponsor

racarr07@gmail.com Switch account

✉ Not shared

☁ Draft saved

* Indicates required question

Name *

Rachel Carr

Email *

racarr07@louisville.edu

Company/Organization Name (if applicable):

University of Louisville

Preferred Sponsorship Type (e.g., Event Sponsorship, Program Sponsorship, General Donation) *

Event Sponsorship

How would you like to be recognized? (e.g., Name, Company Logo, Anonymous) *

Name

Why are you Interested in sponsoring us? *

For our students interested in education and music.

This capture shows how the system will react when the user clicks on the "Let's Work Together" link on the Be A Sponsor page of the website. The user is then redirected to a Google Form. The picture shows some of the required fields for the user to fill out to complete a submission. If the form is completed and submitted, the administrator will see the results in Google Forms under the "Responses" section.

Be a Volunteer Form

Be a Volunteer

racarr07@gmail.com Switch account

✉ Not shared

☁ Draft saved

* Indicates required question

Name *

Rachel Carr

Email *

racarr07@louisville.edu

Availability *

weekends

Phone number *

5026402997

Skills and Interests *

helping kids and volunteers sign in at events

Previous Volunteer Experience *

Preferred Volunteer Roles *

sign in desk

Why Are Interested in Volunteering with Us? *

it's a good cause

How You Heard About Us

word of mouth

Do You Consent to a Background Check? *

◉ Yes

◯ No

Comments or Questions

Your answer

Submit   Clear form

This capture shows how the system will react when the user clicks on the "Become A Volunteer" link on the How to Volunteer page of the website. The user is then redirected to a Google Form. The picture shows some of the required fields for the user to fill out to complete a submission. If the form is completed and submitted, the administrator will see the results in Google Forms under the "Responses" section.

Contact Form

* Indicates required question

Name *

Rachel Carr

Email *

racarr07@louisville.edu

This capture shows how the system will react when the user clicks on the "Connect With Us" link on the How to Volunteer page of the website. The user is then redirected to a Google Form. The picture shows some of the required fields for the user to fill out to complete a submission. If the form is completed and submitted, the administrator will see the results in Google Forms under the "Responses" section.

Get Involved Questionnaire Form
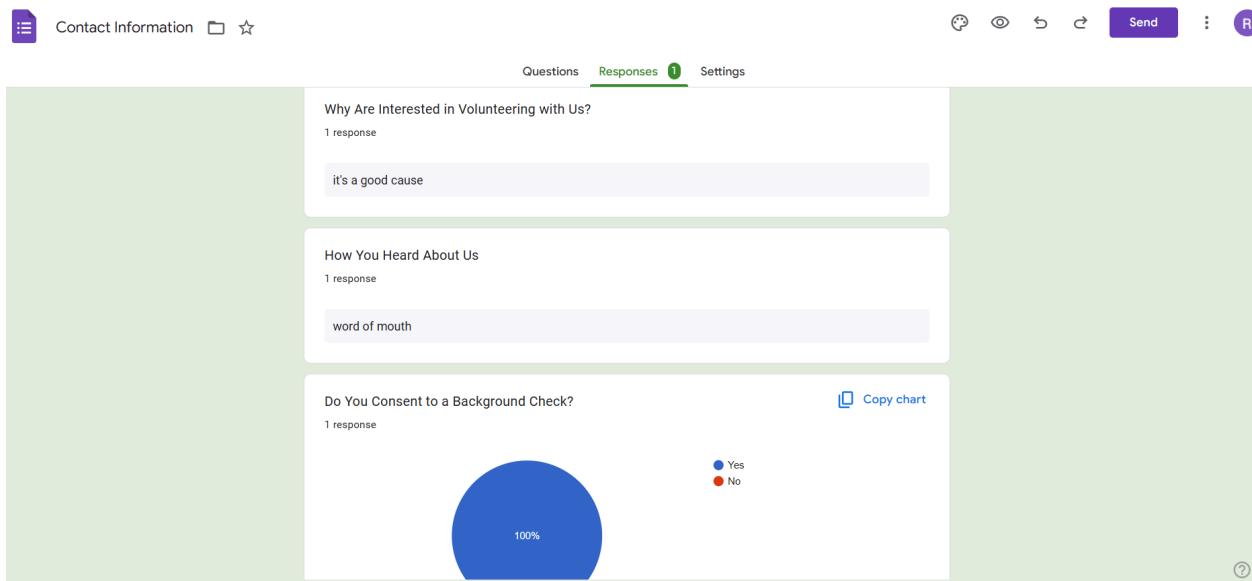
This capture shows how the system will react when the user clicks on the "Complete Questionnaire" link on the Questionnaire page of the website. The user is then redirected to a Survey Monkey Form. The picture shows the first question of the survey as well as the survey progress at the bottom. If the form is completed and submitted, the administrator will see the results on the Survey Monkey homepage on the user dashboard.

3.3B Data Presentation Forms

Company/Organization Name (if applicable):

1 response

University of Louisville

Preferred Sponsorship Type (e.g., Event Sponsorship, Program Sponsorship, General Donation)

1 response

Event Sponsorship

How would you like to be recognized? (e.g., Name, Company Logo, Anonymous)

1 response

Name

Why are you Interested in sponsoring us?

1 response

Questions   Responses 1   Settings

1 response

⊞ Link to Sheets   ⋮

Accepting responses ⬤

Summary   Question   Individual

Name

1 response

Rachel Carr

Email

1 response

racarr07@louisville.edu

The following entails the administrator's view of submitted forms. The administrator can view the user responses as well data visualizations of yes/no questions. The administrator can

choose to view the results by summary, question, or individual. The administrator can export the results to a Google Sheets document.

The following images entail the administrators view on Survey Monkey. The administrator receives an email to their personal email connected to their Survey Monkey account when a user has submitted a survey. The administrator then can go to their dashboard on the homepage of Survey Monkey to view the results. After clicking analyze results, they can choose to view the results as summaries, as data visualizations, or by individual responses. The administrator can click the "Present Results" function to create a summary data visualization of the survey results or with a paid subscription can export the results to their preferred document format.

4. Gantt Chart
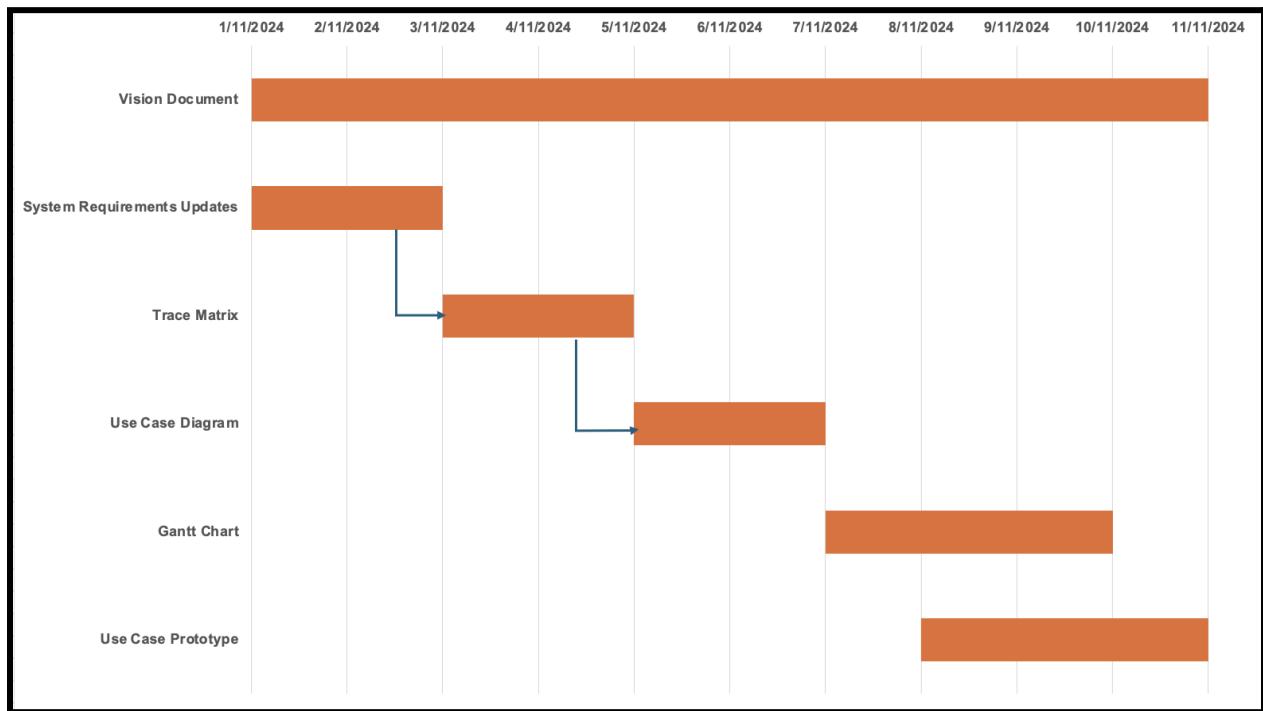
4.1 Gantt Chart Diagram

Task ID	Task Name	Start Date	Duration (Days)	End Date	Task Responsibility	Dependencies
ID0200	Vision Document	15/10/2024	2	16/10/2024	James	None
ID0201	Feasibility Analysis	16/10/2024	2	17/10/2024	James	ID0200
ID0202	System Requirements	16/10/2024	3	18/10/2024	Nick	ID0200
ID0203	List of Use Cases	18/10/2024	2	19/10/2024	Nick	ID0202
ID0204	Initial Architecture Considerations	18/10/2024	2	19/10/2024	Rachel	None
ID0205	Risk Analysis	19/10/2024	2	20/10/2024	Simona	ID0203
ID0206	Trace Matrix	19/10/2024	2	20/10/2024	Simona	ID0203, ID0204
ID0207	Gantt Chart	15/10/2024	6	20/10/2024	Mason	None
ID0208	Inception Phase Prototype	19/10/2024	2	20/10/2024	Rachel	None
ID0300	Vision Document	1/11/2024	10	11/10/2024	Rachel	None
ID0301	System Requirements Updates	1/11/2024	2	11/2/2024	Nick	None
ID0302	Trace Matrix	3/11/2024	2	11/4/2024	Simona	ID0301
ID0303	Use Case Diagram	5/11/2024	2	11/6/2024	Simona	ID0302
ID0304	Gantt Chart	7/11/2024	3	11/9/2024	Mason	None
ID0305	Use Case Prototype	8/11/2024	3	11/10/2024	Rachel	None
ID0401	Class Diagram	15/11/2024	4	18/11/2024	Simona	None
ID0402	Database Diagram	18/11/2024	4	21/11/2024	Nick, James, Rachel	ID0401
ID0403	User Interface Design	19/11/2024	4	22/11/2024	Rachel, James	ID0402, ID0401
ID0404	Gantt Chart	20/11/2024	5	24/11/2024	Mason	None
ID0405	User Interface Prototypes	22/11/2024	3	24/11/2024	Rachel, Nick	ID0402, ID403
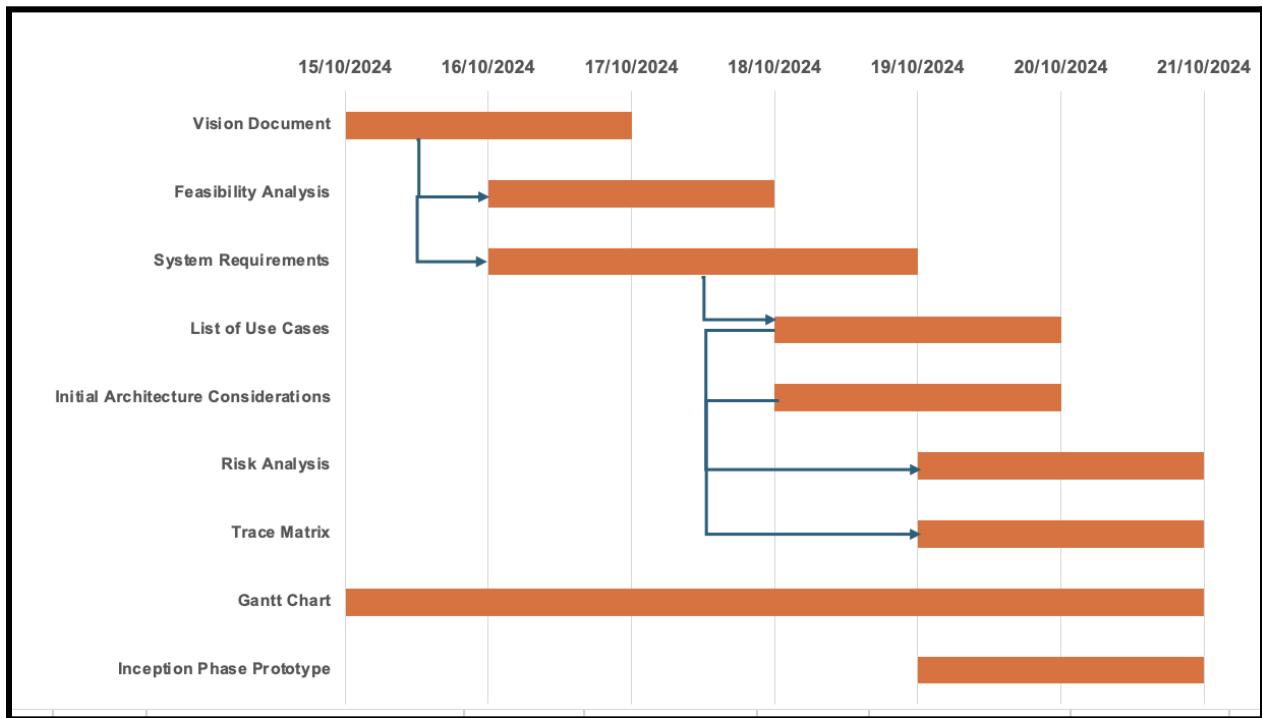
Link to excel sheet.

Colour Key
I2
I3
I5
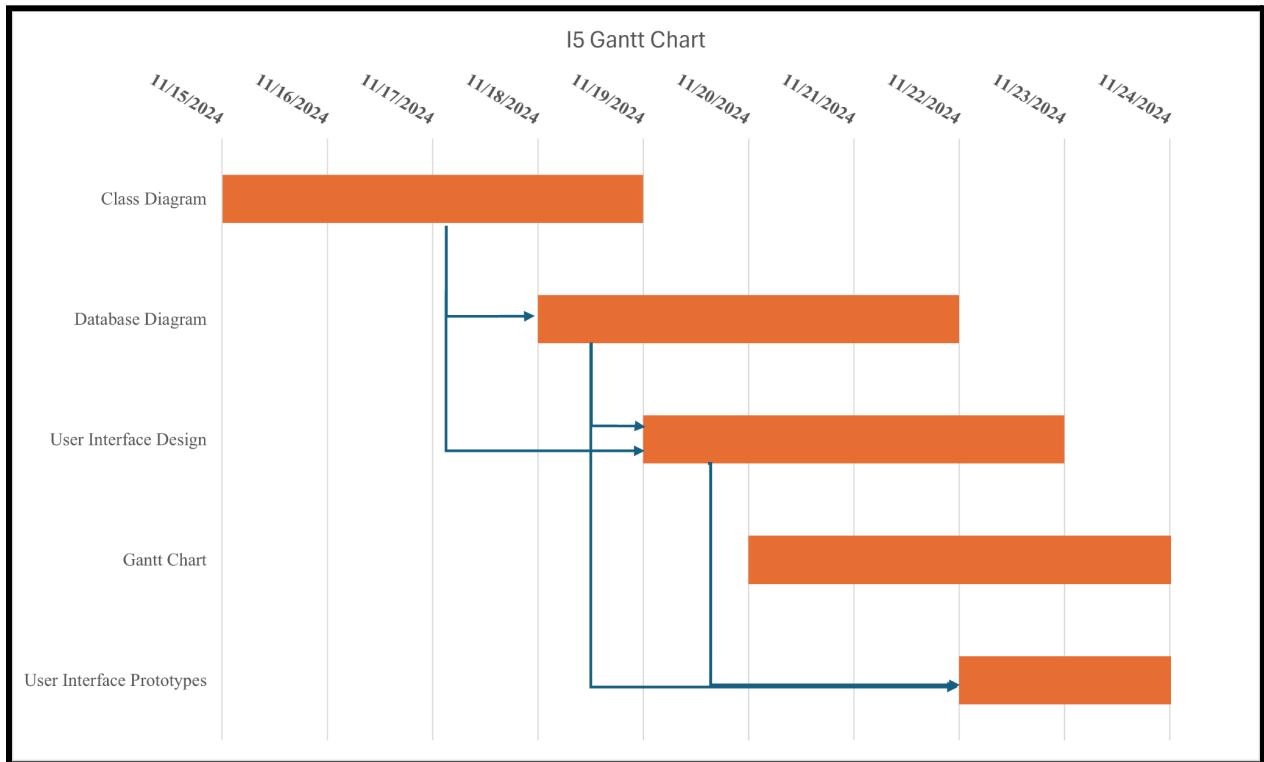
I2 Gantt Chart

I3 Gantt Chart

I5 Gantt Chart
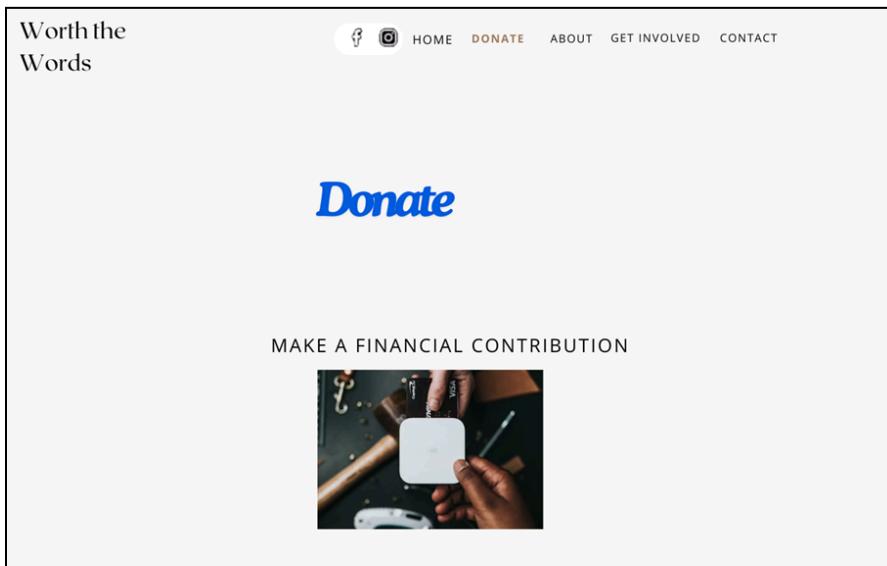
I5 Gantt Chart

4.2 Gantt Chart Narrative
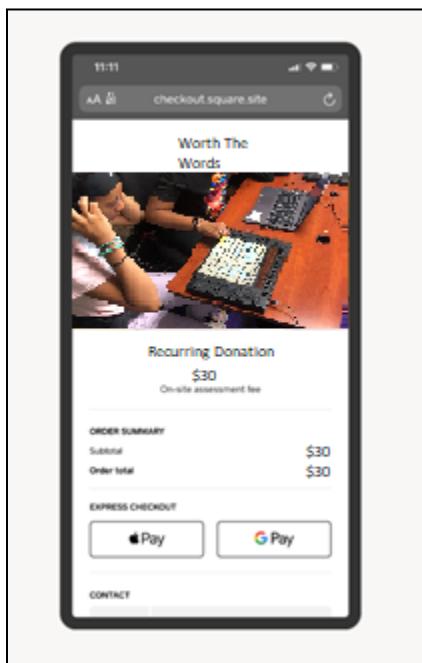
Each team member had a share of the workload. I5 was split into specific tasks. The main tasks were class diagram, database diagram, user interface diagram, gantt chart and user interface prototypes. Some tasks were allocated to only one group member, while most tasks were worked on by multiple members.

5. User Interface Prototypes

Use Case 1: Set Up Recurring Monthly Donations
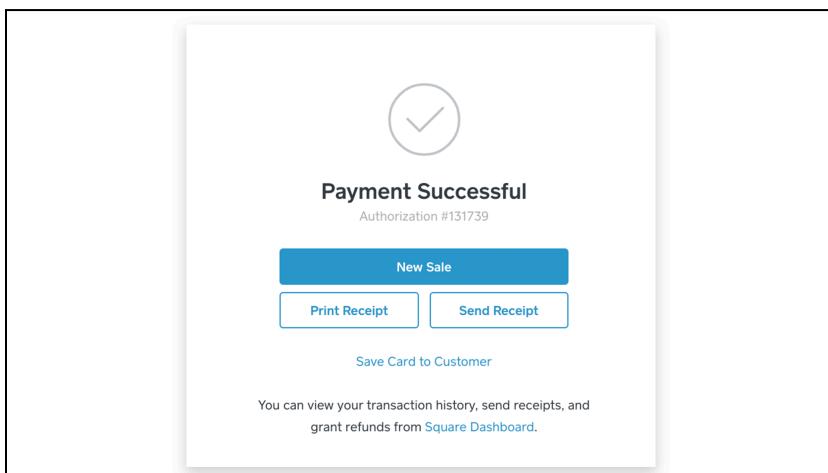
Use Case 2: Secure Payment Processing
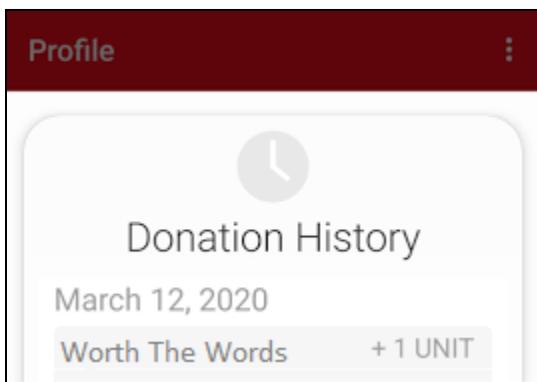
Use Case 3: Manage Donation Subscriptions

Use Case 4: Send Payment Confirmations

Use Case 5: Manage Donation History

Use Case 6: Display Upcoming Events

Upcoming Events

Search by Category: [Enter category] Search by Date:
[dd/mm/yyyy 📅] [Search] [Reset]

New Year's Celebration
Date: 2023-01-01
Category: Holiday

Celebrate the new year with us!
[View Details]

Valentine's Day Dance
Date: 2023-02-14
Category: Party

Join us for a Valentine's Day dance.
[View Details]

Spring Festival
Date: 2023-03-21
Category: Festival

Welcome the spring season with a festival!
[View Details]

Use Case 7: View Event Details

Event Details

Spring Festival

Date: 2023-03-21

Category: Festival

Join us as we welcome the spring season with music, food, and fun activities for all ages! Enjoy live performances, local vendors, and more.

Use Case 8: Create New Event Entry

Manage Events

Create New Event

Title: [   ]
Date: [dd/mm/yyyy   📅]
Category: [   ]

Description: [   ]
[Create Event]

_____

Existing Events

New Year's Celebration (Date: 2023-01-01, Category: Holiday)
Description: Celebrate the new year with us!
[Edit]

Valentine's Day Dance (Date: 2023-02-14, Category: Party)
Description: Join us for a Valentine's Day dance.
[Edit]

Spring Festival (Date: 2023-03-21, Category: Festival)
Description: Welcome the spring season with a festival!
[Edit]

Use Case 9: Modify Existing Event

Modify Event Details

Select Event to Modify:
[New Year's Celebration ⌄]

Modify Selected Event

Title:
[Event Title]

Date:
[dd/mm/yyyy   📅]

Category:
[Holiday ⌄]

Description:
[Event Description]

[Save Changes]

Existing Events

New Year's Celebration

Date: 2023-01-01

Category: Holiday

Description: Celebrate the new year with us!

Use Case 10: Search For Events by Date or Category

Use Case 11: Administrator Creates Customized Questions

Use Case 12: Administrator Manages the Form Design

Use Case 13: Administrator Configures Question Format
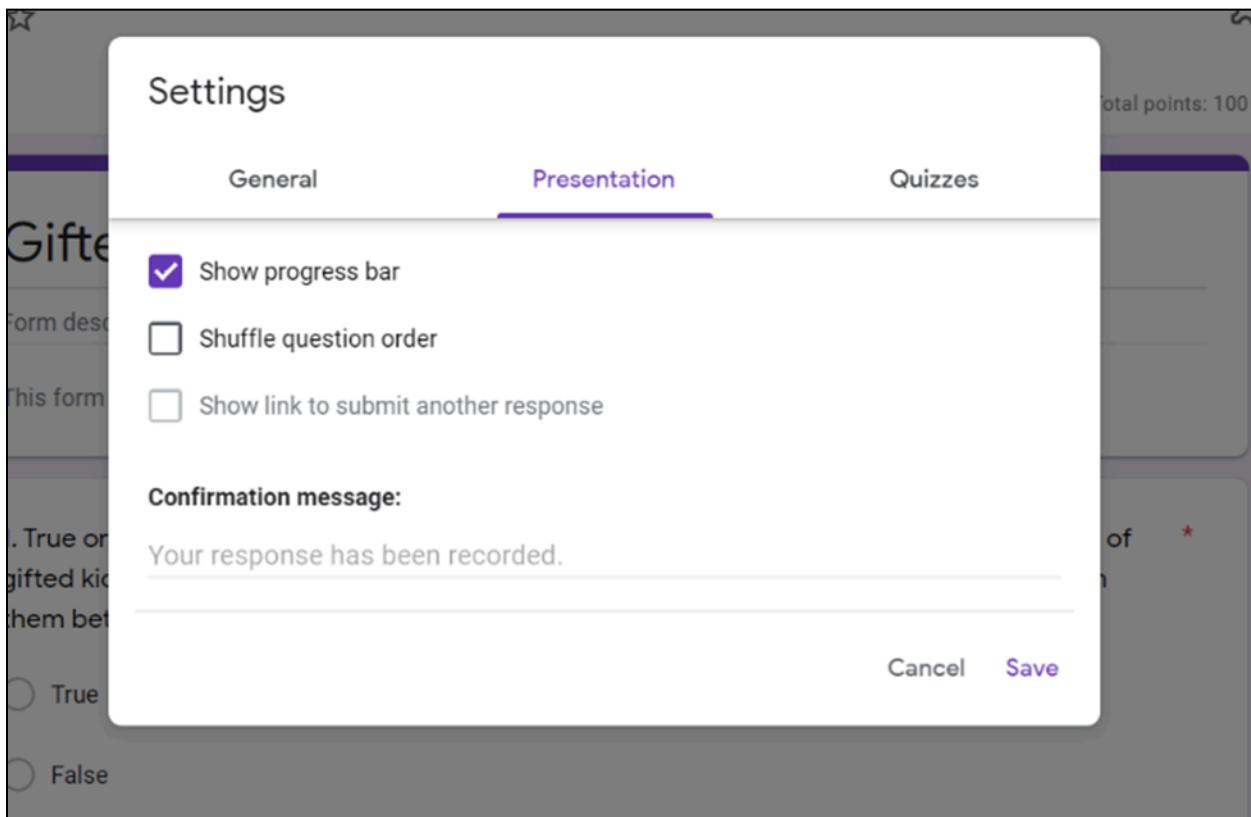
Use Case 14: Administrator Modifies Question Order

Use Case 15: Administrator Sets the Question as Mandatory or Optional
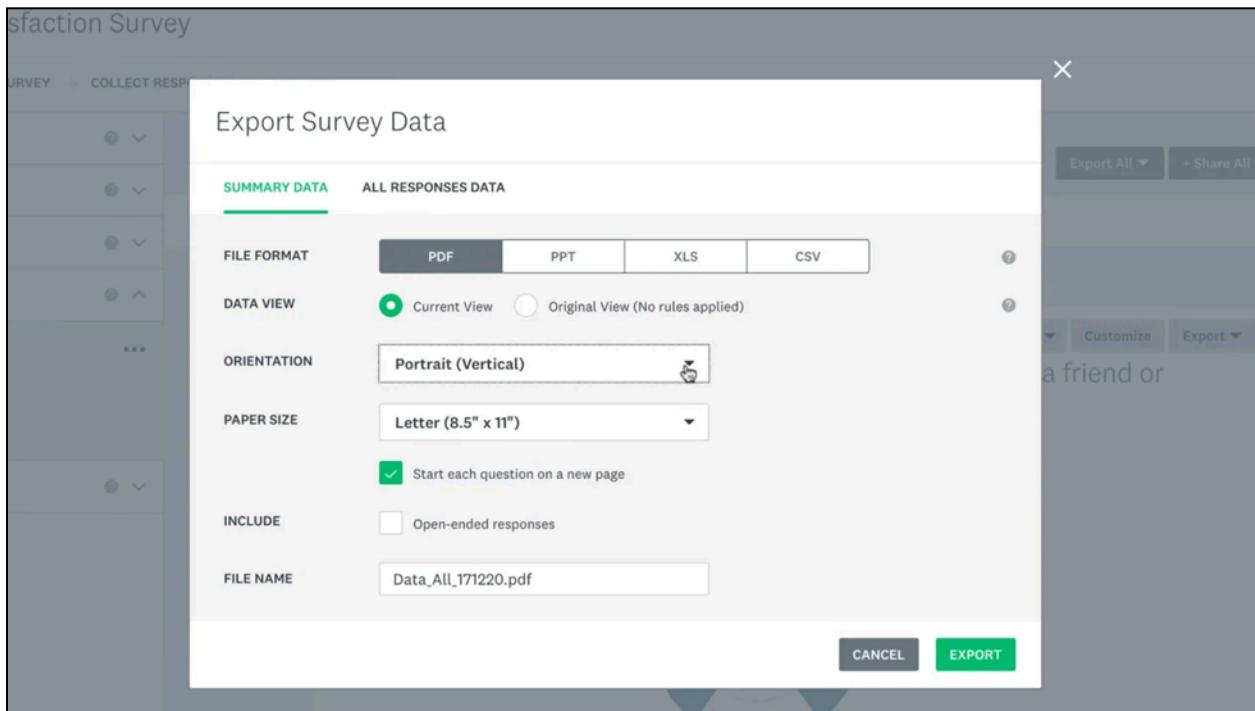
Use Case 16: User Uses the Autosave Feature to Return to Questionnaire Later to Complete

Use Case 17: Administrator Previews Form Before Submission

Use Case 18: Administrator Reviews User Responses

Use Case 19: System Encrypts User Response

Availability *

Long answer text

Phone number *

Short answer text

Skills and Interests *

Long answer text

Previous Volunteer Experience *

Long answer text

Use Case 20: System Validates User Input

Use Cases 21-23: Volunteer Form

Use Cases 24-26: Contact Form

Contact us!

Any questions, comments, concerns? Let us know!

jamesdog2003@gmail.com Switch account

✉ Not shared

Any questions for us?
we'll try our best to respond as soon as possible!

Your answer

How was your experience with us?

Your answer


Looking to partner with us?

○ Yes

○ No

○ Maybe, send me more info!

Submit   Clear form

Use Case 27: Donation Banner Redirect

Use Case 29: Redirect Header to Home

Use Case 30: Socials Link Button

Use Case 34: Add Security Plug-Ins

Use Case 35: Create Encryption for Personal Information

Encrypt Personal Information

Select Data Type to Encrypt

Email ⌄

Encryption Method

AES (Advanced Encryption Standard) ⌄

Key Length (Bits)

Enter key length (e.g., 128, 256)

Encrypt Data

Use Case 36: Add Analytics Plug-In

Add Analytics
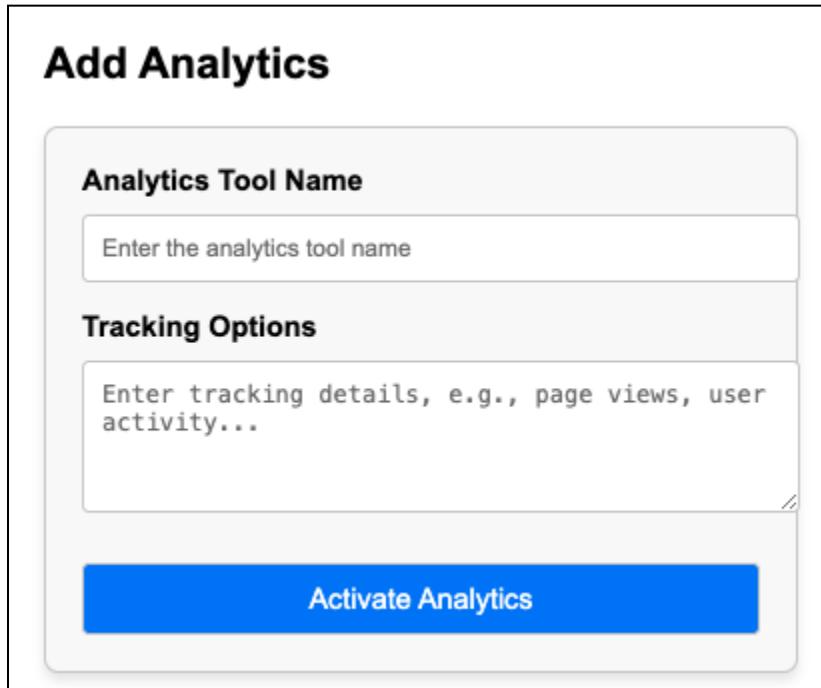
Analytics Tool Name

Enter the analytics tool name

Tracking Options

Enter tracking details, e.g., page views, user activity...

Activate Analytics

Use Case 37: Add Square Analytics

Enable Sales Analytics

Square Account Name

Enter your Square account name

Analytics Type

Specify the analytics to track, e.g., sales
trends, revenue growth

Activate Sales Analytics

Use Case 38: Manage Owner and Admin Privileges

Allow stats reports to be viewed by

- Administrator
- Editor
- Author
- Contributor
- Subscriber
- Copy Editors
- Reporters
- Revisor
- SEO Manager
- SEO Editor

Use Case 39: Viewing for Analytics Formatting

**Use Case 40: Exporting of Questionnaire Form**
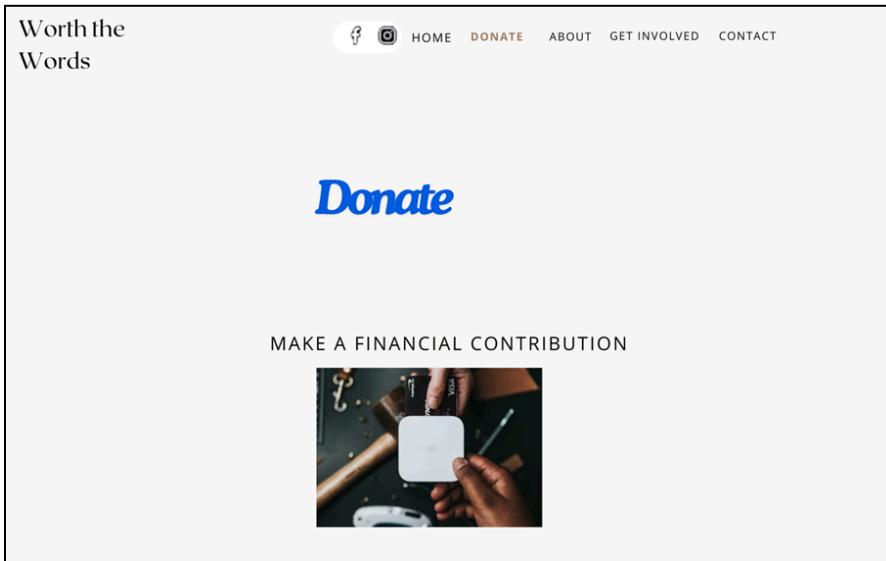
Use Case 41: View Helped Youth Counter


Youth Helped
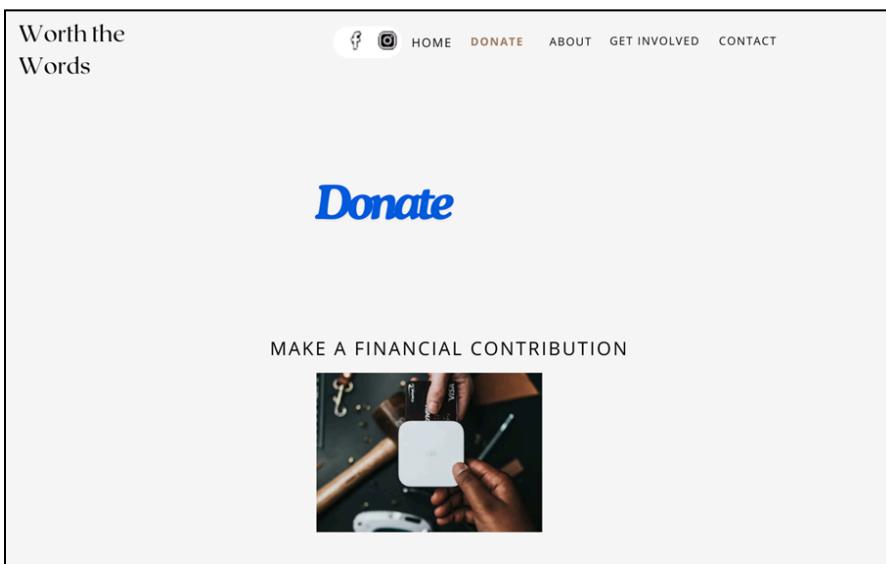
43

Use Case 42: Formatting Content for All Screens

Fits the same across screens for phones and laptops

Use Case 43: Create Heading and Footer Uniformity

The footer and header is the same across pages

Use Case 44: Viewing the Data Guide

Data Guide

Lots of data

Put data in format a, or format b for this case

Use Case 45: Viewing the Editing Guide

Editing Guide

Lots of editing

How to change text

How to change images

Use Case 46: Viewing the Analytics Guide

Troubleshooting Guide

How to fix an issue with crashing

How to fix an issue with images not displaying