Charitable SDF

Table of Contents

4.0 Software Development Plan

4.1 Plan Introduction

This Software Development Plan provides the details of the planned development for the Charitable CSCI which is a web application that allows users to easily access resources to get more involved in their communities.

Charitable will allow users to maintain a Charitable profile, learn about the nation's trending charitable organizations, research people and organizations, connect with other Charitable users, and share their own Charitable experiences. The intention behind this is to encourage users to get involved in their communities by inviting users to participate in a positive social media platform that revolves around charities and good causes.

Development for Charitable involves building a welcoming, user friendly front-end web application, building a database to hold information regarding users, organizations, and activities, and developing a server which is used to communicate between the front-end web app and back-end database.

Deliverables Schedule		
Number	Deliverable	Due Date
D#01	Project Proposal	Week 02
D#02	Front-End Design	Week 03
D#03	Back-End Database	Week 07
D#04	Back-End Server	Week 08
D#05	Front-End Functioning Web App	Week 09
D#06	Integration	Week 10
D#07	Deployment	Week 11

4.1.1 Project Deliverables

4.2 Project Resources

4.2.1 Hardware Resources

4.2.2 Software Resources

4.3 Project Organization

4.4 Project Schedule

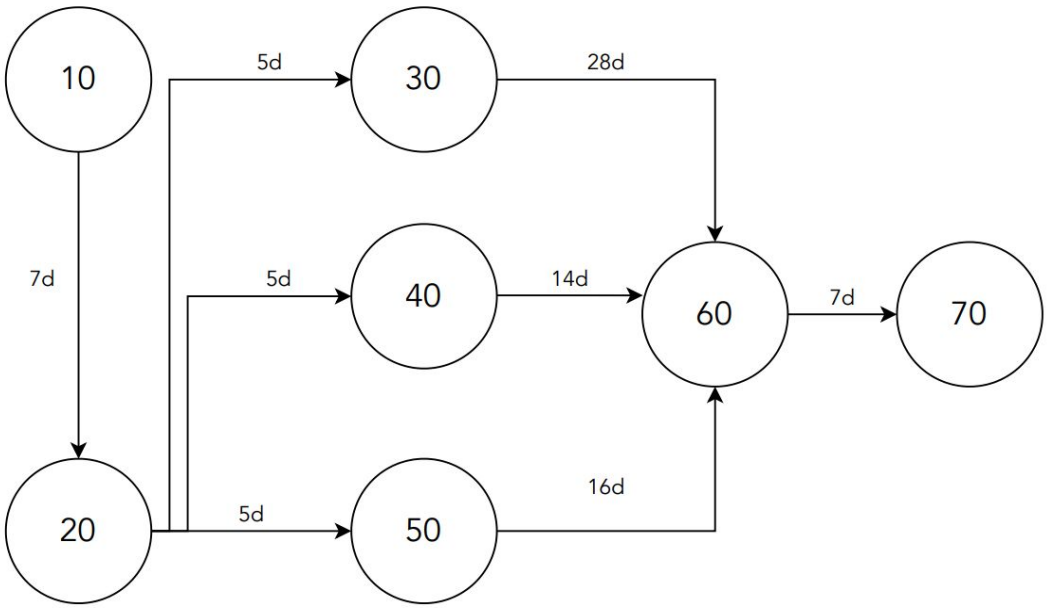
This document intends to outline the development schedule for the web application, Charitable, and its optimistic timeline for development. On the next two pages, you will find a table that lists the milestones, activities, and tasks necessary to complete development. Each milestone, activity, and task has an associated number. These numbers are used to represent its corresponding milestone, activity, or task in the PERT charts that follow the table. Overall, this application is aimed to be completed in 47 days, or roughly 7 weeks.

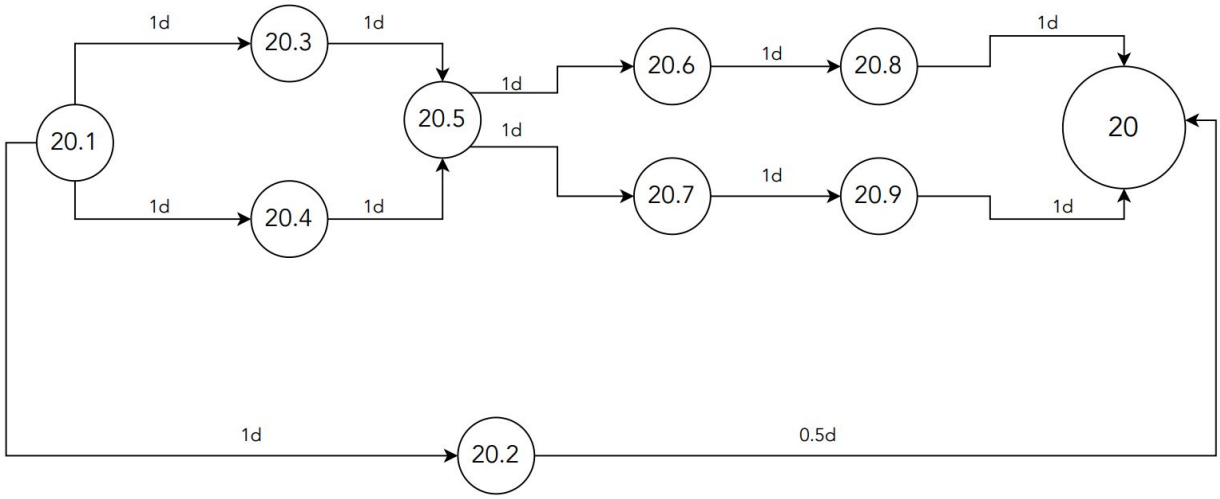
4.4.1 Task Table

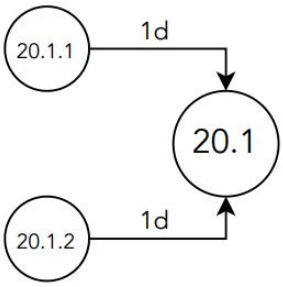
PERT Table					
Milestones		Activities		Tasks	
Project Proposal	10				
Create Art/	20				
		Pick global styles	20.1	Pick site font	20.1.1
				Pick site colors	20.1.2
		Create Logo	20.2	Draw Logo	20.2.1
				Create Favicon.ico	20.2.2
		Create Login Page Prototype	20.3	Create Adobe XD Mockup	20.3.1
				Connect to other mockups	20.3.2
		Create Signup Page Prototype	20.4	Create Adobe XD Mockup	20.4.1
				Connect to other mockups	20.4.2
		Create Navigation Bar	20.5	Create Adobe XD Mockup	20.5.1
				Connect to other mockups	20.5.2
		Create Trending Page Prototype	20.6	Create Adobe XD Mockup	20.6.1
				Connect to other mockups	20.6.2
		Create Search Page Prototype	20.7	Create Adobe XD Mockup	20.7.1
				Connect to other mockups	20.7.2
		Create Timeline Page Prototype	20.8	Create Adobe XD Mockup	20.8.1
				Connect to other mockups	20.8.2
		Create User Profile Page Prototype	20.9	Create Adobe XD Mockup	20.9.1
				Connect to other mockups	20.9.2
Build Website	30				
		Build Login Page	30.1	Set up third party Auth0	30.1.1
				Customize Login Page	30.1.2
		Build Signup Page	30.2	Set up third party Auth0	30.2.1
				Customize Signup Page	30.2.2
		Build Navigation Bar	30.3	Create charitable logo compo	30.3.1
				Create search bar component	30.3.2
				Create trending icon compone	30.3.3
				Create home icon component	30.3.4
				Create profile icon componen	30.3.5
				Connect icon buttons to redire	30.3.6
				Modify for dynamic browser si	30.3.7
		Build Trending Page		Create Scroll View	30.4.2
				Create Scroll View	30.4.4
				Create Display box	30.4.3
				Modify for dynamic browser	30.4.5
		Build Search Page	30.5	Create Web Page	30.5.1
				Create Scroll View	30.5.2
				Create Scroll View	30.5.4
				Create Display Box	30.5.3
				Modify for dynamic browser	30.5.5
		Build Timeline Page	30.6	CreateWeb Page	30.6.1
				Create Scroll View	30.6.2
				Create Scroll View	30.6.3
				Create Individual Post	30.6.4
				Create "Create a Post"	30.6.5
				Add options to attach	30.6.6
				Create Join Friend	30.6.7
				Create profile photo	30.6.8

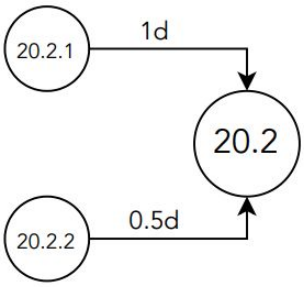
				Modify for dynamic browser	30.6.9
		Build User Profile Page	30.7	Create Web Page	30.7.1
				Create Scroll View	30.7.2
				Create Scroll View	30.7.3
				Create Individual Post	30.7.4
				Create "Create a Post"	30.7.5
				Add options to attach	30.7.6
				Create Join Friend	30.7.7
				Create profile photo	30.7.8
				Create Banner Component	30.7.9
				Add user information	30.7.10
				Create edit profile button	30.7.11
				Modify for dynamic browser	30.7.12
Build Database	40				
		Build Organization Entity	40.1	Obtain organization data	40.1.1
				Create organizations table	40.1.2
				Parse data dump	40.1.3
				Add organizations to table	40.1.4
		Build Activity and Event Entity	40.2	Obtain activity data dump	40.2.1
				Create activity table	40.2.2
				Parse data dump	40.2.3
				Add activites to table	40.2.4
				Allow for user created events	40.2.5
		Build User Entity	40.3	Create user table	40.3.1
				Add users when they sign up	40.3.2
				Allow for user data to be	40.3.3
Build Server	50				
		Make API Calls	50.1	Connect frontend to server api so that it can display the data	50.1.1
				API call to database when user does a search	50.1.2
				API call to database to add user data specific to database	50.1.3
				API call to get user data specific data from the database	50.1.4
				API call to database to add events	50.1.5
				API call to database to get events	50.1.6
		Deploy Server	50.2	Configure API to make calls in non-localhost env	50.2.1
Integrate	60				
		Test	60.1	Run site on server to check	60.1.1
		Debug	60.2	Fix issues and text again until	60.2.1
Deploy Site	70				
		Deploy	70.1	Deploy	70.1.1

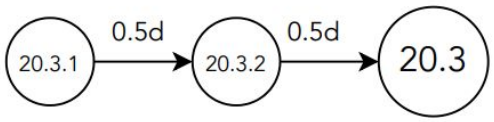
4.4.2 PERT Charts

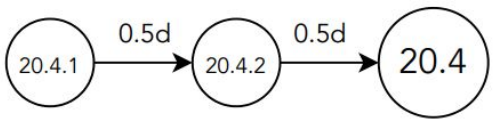


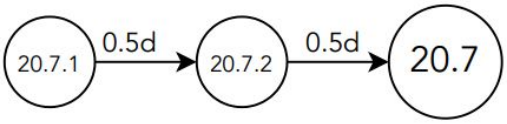

1d
20.3
1d
20.6
1d
20.8
1d
20.1
20.5
1d
20
1d
20.7
1d
20.9
1d
20.4
1d
1d
20.2
0.5d


20.1.1
1d
20.1
20.1.2
1d

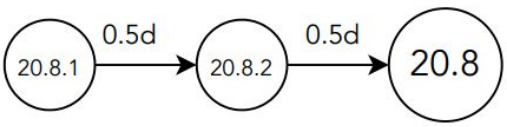

20.2.1
1d
20.2
20.2.2
0.5d

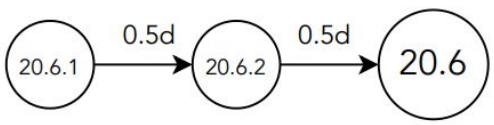

20.3.1
0.5d
20.3.2
0.5d
20.3

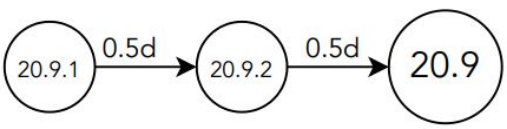

20.4.1
0.5d
20.4.2
0.5d
20.4

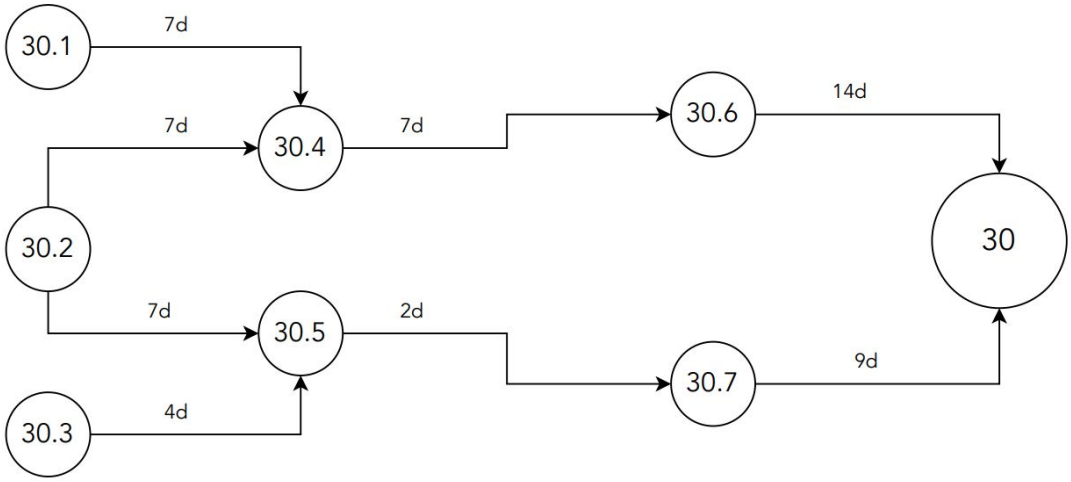

20.7.1
0.5d
20.7.2
0.5d
20.7


20.5.1
0.5d
20.5.2
0.5d
20.5


20.8.1
0.5d
20.8.2
0.5d
20.8

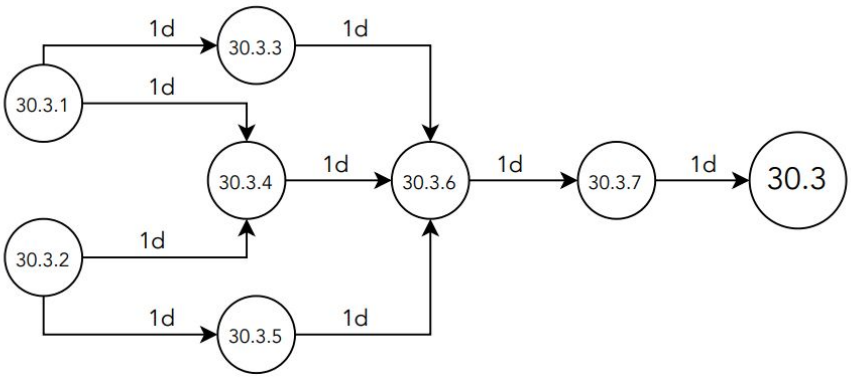

20.6.1
0.5d
20.6.2
0.5d
20.6

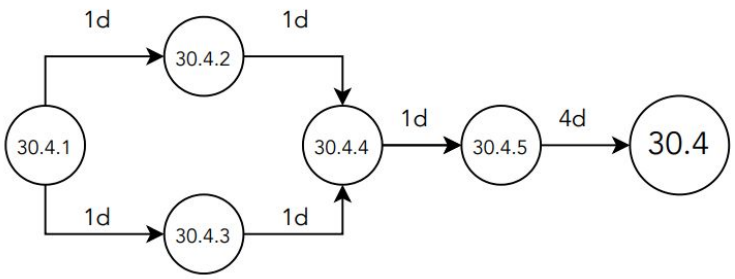

20.9.1
0.5d
20.9.2
0.5d
20.9

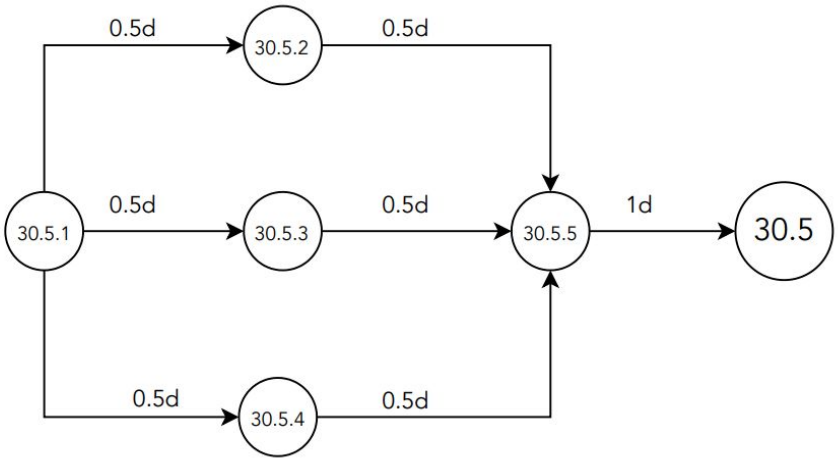

30.1
7d
30.2
7d
30.3
4d
30.4
7d
30.5
2d
30.6
14d
30.7
9d
30

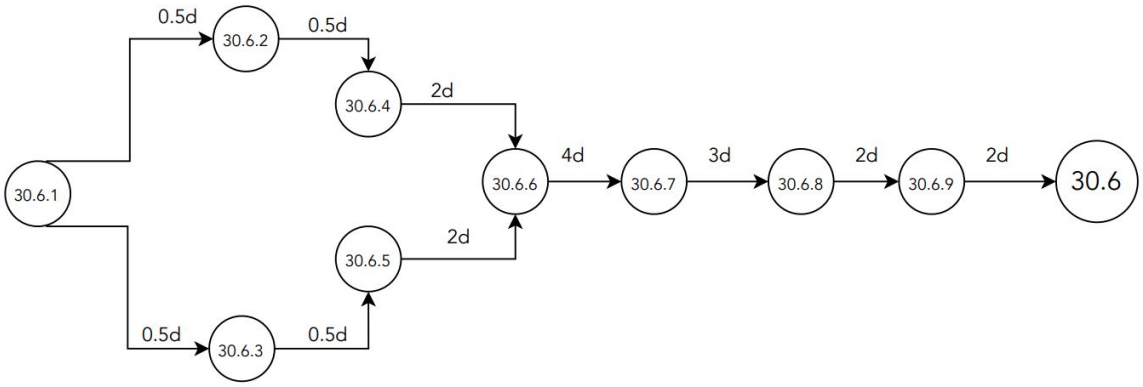

30.1.1
4d
30.1.2
3d
30.1

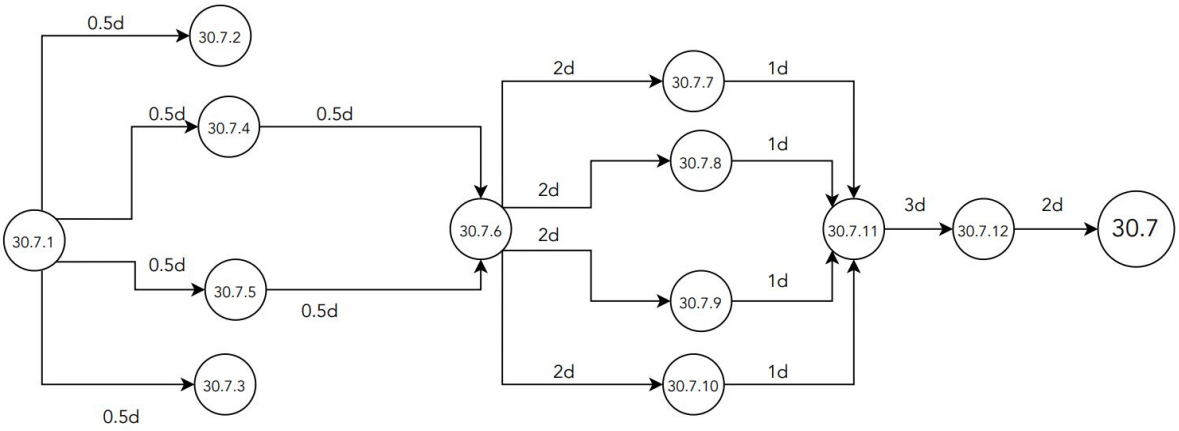

30.2.1
4d
30.2.2
3d
30.2

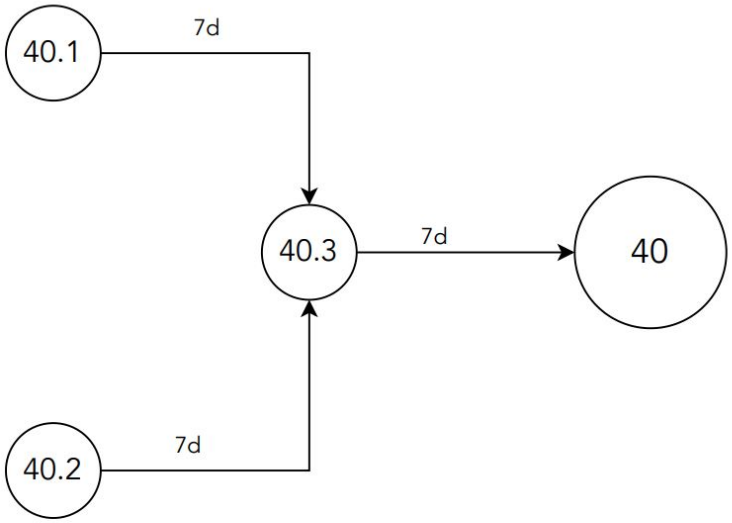

30.3.1
1d
30.3.2
1d
30.3.3
1d
1d
30.3.4
1d
30.3.5
1d
30.3.6
1d
30.3.7
1d
30.3

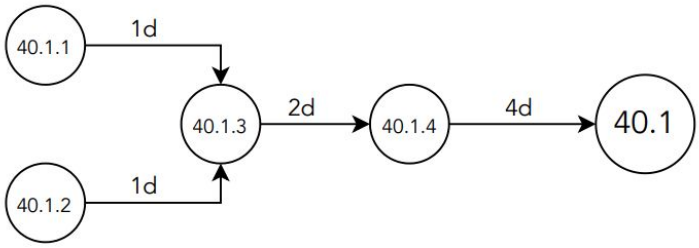

1d
1d
30.4.2
30.4.1
30.4.4
1d
30.4.5
4d
30.4
1d
30.4.3
1d

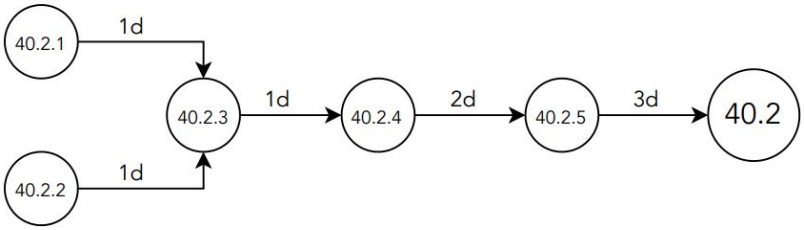

0.5d
30.5.2
0.5d
0.5d
30.5.1
30.5.3
0.5d
30.5.5
1d
30.5
0.5d
30.5.4
0.5d

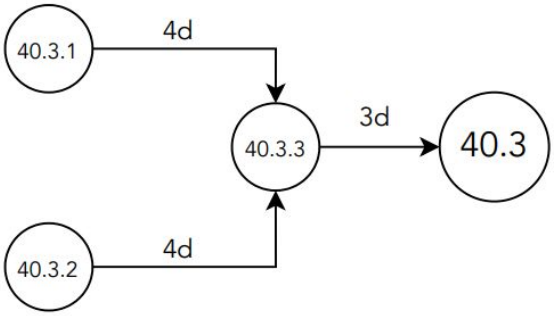

0.5d
30.6.2
0.5d
30.6.4
2d
30.6.1
4d
30.6.6
30.6.7
3d
30.6.8
2d
30.6.9
2d
30.6
2d
30.6.5
0.5d
30.6.3
0.5d

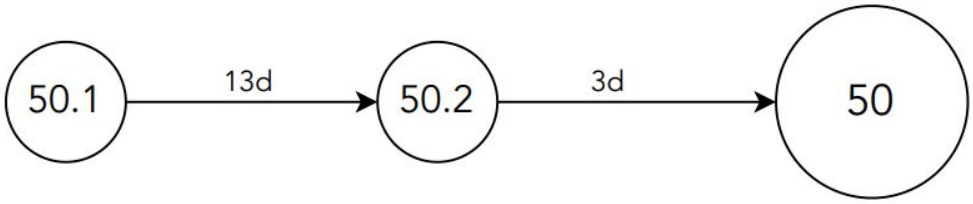

0.5d
30.7.2
2d
30.7.7
1d
0.5d
30.7.4
0.5d
30.7.8
1d
2d
30.7.1
30.7.6
2d
30.7.11
3d
30.7.12
2d
30.7
0.5d
30.7.5
0.5d
30.7.9
1d
30.7.3
2d
30.7.10
1d
0.5d

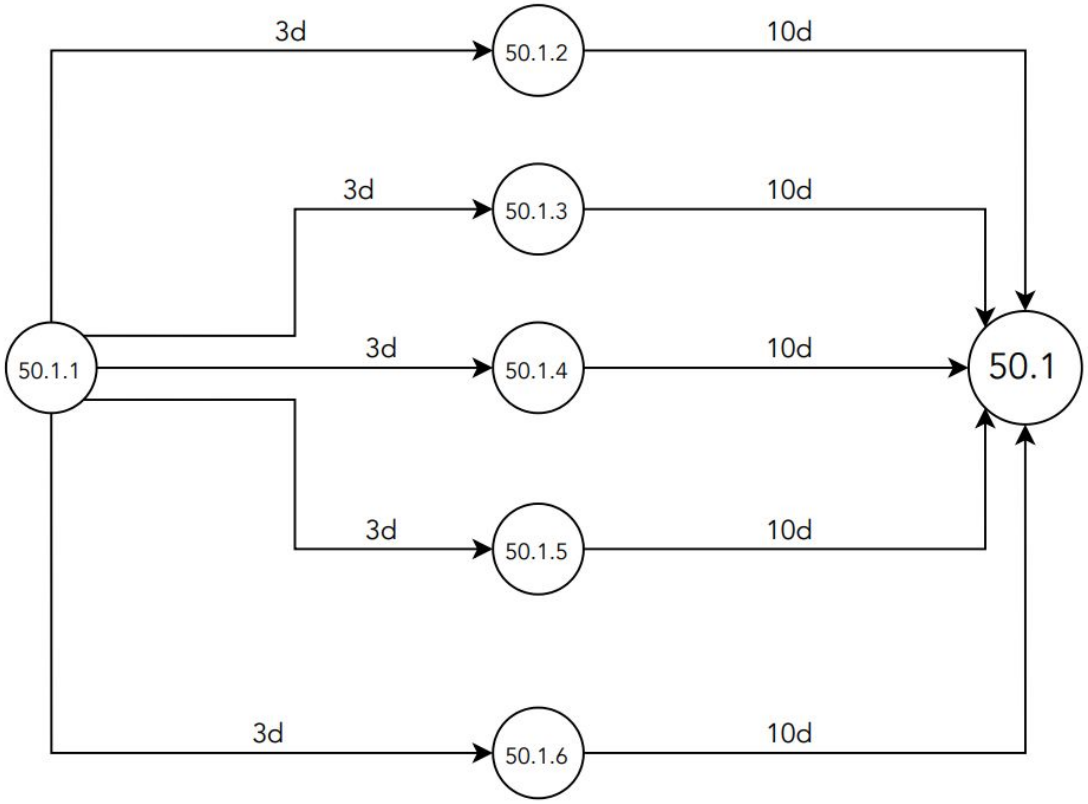

40.1
7d
40.3
7d
40
40.2
7d

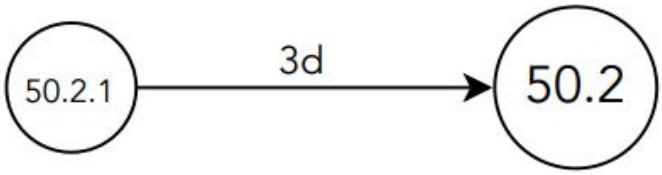

40.1.1
1d
40.1.3
2d
40.1.4
4d
40.1
40.1.2
1d

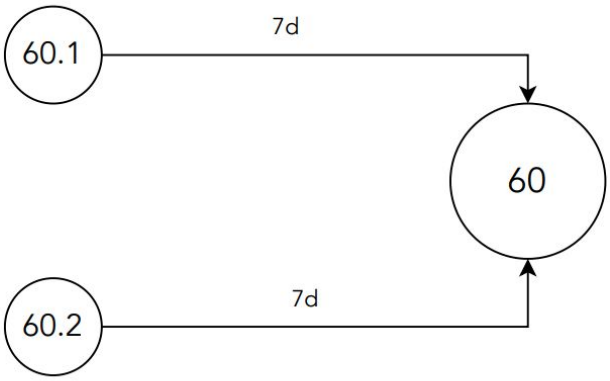

40.2.1
1d
40.2.3
1d
40.2.4
2d
40.2.5
3d
40.2
40.2.2
1d

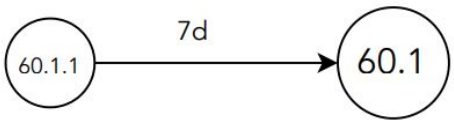

40.3.1
4d
40.3.3
3d
40.3
40.3.2
4d

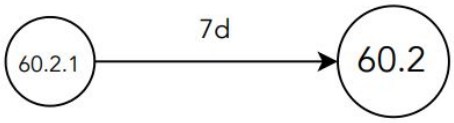

50.1
13d
50.2
3d
50

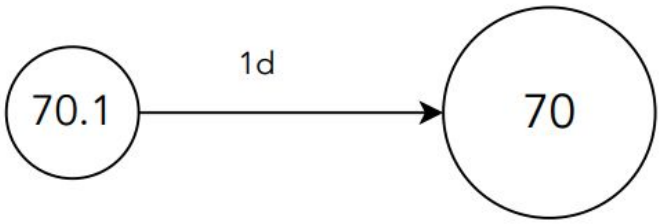

50.1.1
3d
50.1.2
10d
3d
50.1.3
10d
3d
50.1.4
10d
50.1
3d
50.1.5
10d
3d
50.1.6
10d

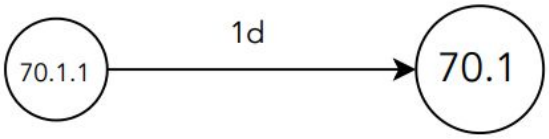

50.2.1
3d
50.2


60.1
7d
60
60.2
7d


60.1.1
7d
60.1


60.2.1
7d
60.2


70.1
1d
70


70.1.1
1d
70.1

5.0 Requirements Specification

5.1 Introduction

This system is built as a web application using React. The system is composed of a web application front-end, a server, and a database back-end that holds datasets for the registered users of the app as well as organizations the app can access. The diagram below illustrates the components of this system. The system will be using an API to perform get and post operations within the application. The system will also use queries to parse the back-end database. This document will

1. List and describe the components of this system.
2. Discuss the functionality requirements of the system and its components.
3. Discuss the performance requirements of the system.
4. Discuss the project environment requirements for the system.

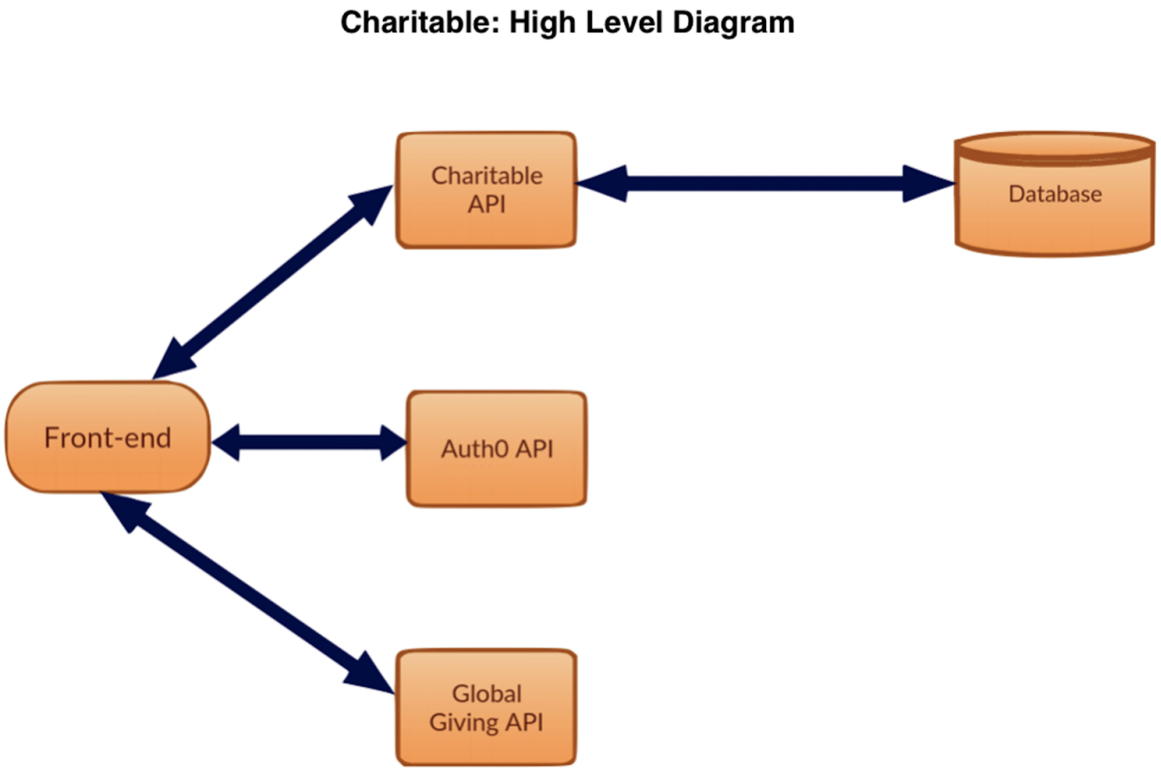


Figure 1.0

5.2 CSCI Component Breakdown

5.2.1 Web Application Front-End CSC

-- Front-end web page interface for user interaction with the system.

5.2.1.1 Login Page CSU

-- Web page for logging into system.

5.2.1.2 Signup Page CSU

-- Web page for registering in system.

5.2.1.3 Timeline Page CSU

-- Web page to display the user and user's friend activity.

5.2.1.4 Trending Page CSU

-- Web page to display trending organizations in the nation.

5.2.1.5 User Profile Page CSU

-- Web page for editing and maintaining user information.

5.2.1.6 Search Results Page CSU

-- Web page to display user search results.

5.2.1.7 Navigation Bar CSU

-- UI element used to navigate between web pages in application.

5.2.2 Server CSC

-- Backend server to allow the front-end web page to interact with the database.

5.2.3 Database CSC

-- Backend database to hold datasets for the system.

5.2.3.1 User Dataset

-- Data table to hold registered user information.

5.2.3.2 Organization Dataset

-- Data table to hold registered organization information.

5.2.3.3 Activities and Events Dataset

-- Data table to hold registered activities and events.

5.3 Functional Requirements

5.3.1 Web Application Front-End CSC

5.3.1.1 The web application front-end shall be displayed in a web browser.

5.3.1.2 The web application front-end shall react to mouse clicks on displayed buttons.

5.3.1.3 The web application front-end shall be composed of 6 web pages.

5.3.1.4 The web application front-end shall have one page for logging into the system.
This page will be known as the Login Page.

5.3.1.4.1 The Login Page shall display the Charitable Logo.
This logo will require no action.

5.3.1.4.2 The Login Page shall provide an option for Google login.

5.3.1.4.3 The Login Page shall provide 2 text entry fields.

5.3.1.4.4 The Login Page shall provide one text entry field for an email address entry.
This text entry field will be known as the Login Email Field.

5.3.1.4.5 The Login Page shall provide a second text entry field for a password entry.
This text entry field will be known as the Login Password Field.
This Login Password Field will hide the text entered by replacing each character typed with a dot.

5.3.1.4.6 The Login Page shall provide a link to reset a password.

This link will be known as the Password Reset Link.

5.3.1.4.7 The Password Reset Link shall redirect to a third party account service called Auth0 upon a mouse click.

5.3.1.4.8 The Login Page shall display a button to complete the login.
This button will be known as the Login Button.

5.3.1.4.9 The Login Button shall redirect the web page to the user account tied to the login credentials upon mouse click.

5.3.1.4.10 The Login Page shall provide an option to a signup page.
This link will be known as the Signup Link.

5.3.1.4.10 The Signup Link shall redirect the web page to a third party account service called Auth0 upon mouse click.

5.3.1.5 The web application front-end shall have one page for registering in the system.
This page will be known as the Signup Page.

5.3.1.5.1 The Signup Page shall display the Charitable Logo.
This logo will require no action.

5.3.1.4.2 The Signup Page shall provide options for Google login.

5.3.1.5.3 The Signup Page shall provide 2 text entry fields.

5.3.1.5.4 The Signup Page shall provide one text entry field for an email address entry.
This text entry field will be known as the Signup Email Field.

5.3.1.5.5 The Signup Page shall provide one text entry field for a password entry.
This text entry field will be known as the Signup Password Field.
This Signup Password Field will hide the text entered by replacing each character typed with a dot.
This Signup Password Field will provide a button to reveal the text entered.

5.3.1.5.6 The Signup Page shall state that by signing up you agree to the terms and policies.

5.3.1.5.7 The Signup Page shall display a button to complete the account registration.
This button will be known as the Signup Button.

5.3.1.5.8 The Signup Button shall redirect the webpage to the Login Page upon mouse click.

5.3.1.5.9 The Signup Page shall provide a link to the Login Page.
This link will be known as the Login Link.

5.3.1.5.10 The Login Link shall redirect the webpage to the Login Page upon mouse click.

5.3.1.6 The web application front-end shall have one page for displaying the user and user's friend activity.
This page will be known as the Timeline Page.

5.3.1.6.1 The Timeline Page shall display a scroll view.

5.3.1.6.2 The scroll view shall display rectangular text boxes stacked vertically.
These rectangular text boxes will be known as Posts.

5.3.1.6.3 The scroll view's first element above the Posts shall display a rectangular box for creating a new post to the timeline.
This rectangular box will be known as the Create New Post Box.

5.3.1.6.4 The Create New Post Box shall provide a text entry field.
This text entry field will be known as the Caption Text Box.

5.3.1.6.5 The Create New Post Box shall provide a button to attach a file.
This button will be known as the Attach File Button.

5.3.1.6.6 The Attach File Button shall open a popup upon mouse click.
This popup will be known as the Attach File Popup.

5.3.1.6.7 The Attach File Button shall access the files of the computer system the application is being used on.

5.3.1.6.8 The Attach File Popup shall display the files on the computer system the application is being used on.

5.3.1.6.9 The Attach File Popup shall display a button to complete the file attachment.
This button will be known as the Open File Button.

5.3.1.6.10 The Open File Button shall attach the chosen file from the popup to the text entered in the Caption Text Box upon mouse click.

5.3.1.6.11 The Attach File Popup shall display a button to close the file popup.
This button will be known as the Cancel File Attachment Button.

5.3.1.6.12 The Cancel File Attachment Button shall close the file popup upon mouse click.

5.3.1.6.13 The Create New Post Box shall provide a button to attach an image.
This button will be known as the Attach Image Button.

5.3.1.6.14 The Attach Image Button shall open a popup upon mouse click.
This popup will be known as the Attach Image Popup.

5.3.1.6.15 The Attach Image Button shall access the files of the computer system the application is being used on.

5.3.1.6.16 The Attach Image Popup shall display the files on the computer system the application is being used on.

5.3.1.6.17 The Attach Image Popup shall display a button to complete the image attachment.
This button will be known as the Open Image Button.

5.3.1.6.18 The Open Image Button shall attach the chosen image from the popup to the text entered in the Caption Text Box upon mouse click.

5.3.1.6.19 The Attach Image Popup shall display a button to close the file popup.
This button will be known as the Cancel Image Attachment Button.

5.3.1.6.20 The Cancel Image Attachment Button shall close the file popup upon mouse click.

5.3.1.6.21 A Post shall display the profile picture of the user creating the Post.

5.3.1.6.22 A Post shall display the name of the user creating the Post.

5.3.1.6.23 A Post shall display the date the Post was created.

5.3.1.6.24 A Post shall display the location the user creating the Post specifies.

5.3.1.6.25 A Post shall display any text message entered in the Create New Post Box.

5.3.1.6.26 A Post shall display any attachments of images attached to the Post.

5.3.1.6.27 The scroll view shall display any posts created by the user or the user's connections on the application in the order of date created.

5.3.1.7 The web application front-end shall have one page for displaying trending organizations.
This page will be known as the Trending Page.

5.3.1.7.1 The Trending Page shall display a scroll view.
This scroll view will be displayed on the left half of the page.

5.3.1.7.2 The Trending Page shall display a text box.

This text box will be displayed on the right half of the page.

5.3.1.7.3 The scroll view shall display rectangular text boxes stacked vertically.
These rectangular boxes will be known as Trending Listings.

5.3.1.7.4 Each Trending Listing shall display the name of a trending organization in the nation.

5.3.1.7.5 Each Trending Listing shall display the logo of a trending organization in the nation.

5.3.1.7.6 Each Trending Listing shall display the mission statement of a trending organization in the nation.

5.3.1.7.7 The Trending Listing shall respond to a mouse click.
The mouse click on a Trending Listing in the scroll view will create a new selection in the scroll view.

5.3.1.7.8 A selected Trending Listing from the scroll view shall be outlined in the color blue.

5.3.1.7.9 The text box shall display information about the organization that is selected from the scroll view.

5.3.1.7.10 The default selection in the scroll view shall be the first Trending Listing.

5.3.1.7.11 The text box shall display a button to save the organization.
This button will be known as the Save Trending Organization Button.

The Save Trending Organization Button will be the shape of a heart.

5.3.1.7.12 The Save Trending Organization Button shall display the outline of a heart when the organization is not saved.

5.3.1.7.13 The Save Trending Organization Button shall display a filled heart when the organization is saved.

5.3.1.7.14 The Save Trending Organization Button shall respond to a mouse click.

The mouse click on the Save Trending Organization Button will change the outline of the heart to a filled heart if the organization had not been saved.

The mouse click on the Save Trending Organization Button will change the filled heart to an outline of a heart if the organization had been saved.

5.3.1.8 The web application front-end shall have one page for displaying a user profile.

This page will be known as the User Profile Page.

5.3.1.8.1 The User Profile Page shall display a profile photo.

This profile photo will be displayed in the shape of a circle.

This profile photo will be displayed at the top of the left side of the web page.

5.3.1.8.2 The User Profile Page shall display a banner photo.

This banner photo will be displayed in the shape of a rectangle.

This banner photo will begin where the profile photo is placed.

This banner photo will extend across the page.

5.3.1.8.3 The User Profile Page shall display the user first name.
This first name will be displayed on the banner photo.
This first name will be displayed to the right of the profile photo.

5.3.1.8.4 The User Profile Page shall display the user last name.
This last name will be displayed to the right of the first name on the banner photo.

5.3.1.8.5 The User Profile Page shall display a user biography.
This biography will be displayed under the name on the banner photo.

5.3.1.8.6 The User Profile Page shall display the user's location if specified.
This location will be displayed to the right of the biography

5.3.1.8.7 The User Profile Page shall display 2 scroll views.

5.3.1.8.8 The User Profile Page shall display one scroll view on the majority of the left side of the web page.
This scroll view will be known as the Timeline Scroll View.

5.3.1.8.9 The Timeline Scroll View shall be displayed under the profile photo and banner photo.

5.3.1.8.10 The Timeline Scroll View shall display rectangular text boxes stacked vertically.
These rectangular text boxes will be known as Posts.

5.3.1.8.11 The Timeline Scroll View's first element above the Posts shall always be a rectangular box for creating a new post to the timeline.
This rectangular box will be known as the Create New Post Box.

5.3.1.8.12 The Create New Post Box shall provide a text entry field.
This text entry field will be known as the Caption Text Box.

5.3.1.8.13 The Create New Post Box shall provide a button to attach a file.
This button will be known as the Attach File Button.

5.3.1.8.14 The Attach File Button shall open a popup upon mouse click.
This popup will be known as the Attach File Popup.

5.3.1.8.15 The Attach File Button shall access the files of the computer system the application is being used on.

5.3.1.8.16 The Attach File Popup shall display the files on the computer system the application is being used on.

5.3.1.8.17 The Attach File Popup shall display a button to complete the file attachment.
This button will be known as the Open File Button.

5.3.1.8.18 The Open File Button shall attach the chosen file from the popup to the text entered in the Caption Text Box upon mouse click.

5.3.1.8.19 The Attach File Popup shall display a button to close the file popup.
This button will be known as the Cancel File Attachment Button.

5.3.1.8.20 The Cancel File Attachment Button shall close the file popup upon mouse click.

5.3.1.8.21 The Create New Post Box shall provide a button to attach an image.
This button will be known as the Attach Image Button.

5.3.1.8.22 The Attach Image Button shall open a popup upon mouse click.
This popup will be known as the Attach Image Popup.

5.3.1.8.23 The Attach Image Button shall access the files of the computer system the application is being used on.

5.3.1.8.24 The Attach Image Popup shall display the files on the computer system the application is being used on.

5.3.1.8.25 The Attach Image Popup shall display a button to complete the image attachment.
This button will be known as the Open Image Button.

5.3.1.8.26 The Open Image Button shall attach the chosen image from the popup to the text entered in the Caption Text Box upon mouse click.

5.3.1.8.27 The Attach Image Popup shall display a button to close the file popup.
This button will be known as the Cancel Image Attachment Button.

5.3.1.8.28 The Cancel Image Attachment Button shall close the file popup upon mouse click.

5.3.1.8.29 A Post shall display the name of the user creating the Post.

5.3.1.8.30 A Post shall display the date the Post was created.

5.3.1.8.31 A Post shall display the location the user creating the Post specifies.

5.3.1.8.32 A Post shall display any text message entered in the Create New Post Box.

5.3.1.8.33 A Post shall display any attachments of images attached to the Post.

5.3.1.8.34 The Timeline Scroll View shall display any posts created by the user in the order of date created.

5.3.1.8.35 The User Profile Page shall display a second scroll view on the right side of the web page.
This scroll view will be known as the Saved Scroll View.

5.3.1.8.36 The Saved Scroll View shall be displayed under the profile photo and banner photo.

5.3.1.8.37 The Saved Scroll View shall display rectangular text boxes vertically.
These rectangular text boxes will be known as Saved Listings.

5.3.1.8.38 Each Saved Listing shall display the name of a trending organization in the nation.

5.3.1.8.39 Each Saved Listing shall display the logo of a trending organization in the nation.

5.3.1.8.40 Each Saved Listing shall display the mission statement of a trending organization in the nation.

5.3.1.8.41 The Saved Listing shall respond to a mouse click.
The mouse click on a Saved Listing in the scroll view will create a new selection in the scroll view.

5.3.1.8.42 A selected Saved Listing from the scroll view shall be outlined in the color blue.

5.3.1.8.43 Selecting a Saved Listing from the scroll view shall open a popup.
This popup will be known as the Saved Listing Information Popup.

5.3.1.8.44 The Saved Listing Popup shall display information about the selected Saved Listing.

5.3.1.8.45 The Saved Listing Popup shall display a scroll view if the information does not fit on the popup.

5.3.1.8.46 The Saved Listing Popup shall provide a button in the shape of an 'x' at the top right of the popup.
This button will be known as the Close Saved Button.

5.3.1.8.47 The Close Saved Button shall close the Saved Listing Popup upon mouse click.

5.3.1.9 The web application front-end shall have one page for displaying search results.

This page will be known as the Search Results Page.

5.3.1.9.1 The Search Results Page shall display a scroll view.

This scroll view will be displayed on the left half of the page.

5.3.1.9.2 The Search Results Page shall display a text box.

This text box will be displayed on the right half of the page.

5.3.1.9.3 The scroll view shall display rectangular text boxes vertically.

These rectangular boxes will be known as Search Result Listings.

5.3.1.9.4 Each Search Result Listing shall display the name of an organization or person that matches the search query.

5.3.1.9.5 Each Search Result Listing shall display the logo or profile photo of the organization or person.

5.3.1.9.6 Each Search Result Listing shall display the mission statement or biography of the organization or person.

5.3.1.9.7 The Search Result Listing shall respond to a mouse click.

The mouse click on a Search Result Listing in the scroll view will create a new selection in the scroll view.

5.3.1.9.8 A selected Search Result Listing from the scroll view shall be outlined in the color blue.

5.3.1.9.9 The text box shall display information about the organization that is selected from the scroll view.

5.3.1.9.10 The default selection in the scroll view shall be the first Search Result Listing.

5.3.1.9.11 The text box shall display a button to save the organization.
This button will be known as the Save Searched Organization Button.
The Save Searched Organization Button will be the shape of a heart.

5.3.1.9.12 The Save Searched Organization Button shall display the outline of a heart when the organization is not saved.

5.3.1.9.13 The Save Searched Organization Button shall display a filled heart when the organization is saved.

5.3.1.9.14 The Save Searched Organization Button shall respond to a mouse click.
The mouse click on the Save Searched Organization Button will change the outline of the heart to a filled heart if the organization had not been saved.
The mouse click on the Save Searched Organization Button will change the filled heart to an outline of a heart if the organization had been saved.

5.3.1.10 The web application front-end shall display a bar to navigate between web pages in the application.
This element will be known as the Navigation Bar.

5.3.1.10.1 The Navigation Bar shall display the Charitable Logo at the left.

This logo will require no action.

5.3.1.10.2 The Navigation Bar shall provide a text entry field to enter a search query.

This text entry field will be known as the Search Bar.
The Search Bar will be displayed to the left of the Charitable Logo.

5.3.1.10.3 The Search Bar shall be required to respond to the Key press of the keyboard ‘Enter’ key.

This response will redirect the web page to the Search Results Page.
The Search Results page will be populated with the search results of the search query.

5.3.1.10.4 The Navigation Bar shall display a button to navigate to the Trending Page.

This button will be known as the Trending Button.
The Trending Button will be in the shape of an upward pointing trendline.
The Trending Button will be displayed to the left of the Search Bar.

5.3.1.10.5 The Navigation Bar shall display a button to navigate to the Timeline Page.

This button will be known as the Home Button.
The Home Button will be in the shape of a house.
The Home Button will be displayed to the left of the Trending Button.

5.3.1.10.6 The Navigation Bar shall display a button to navigate to the User Profile Page.

This button will be known as the Profile Button.
The Profile Button will be in the shape of the outline of a person’s head and shoulders.
The Profile Button will be displayed to the left of the Home Button.

5.3.1.10.7 The Navigation Bar shall appear on all web pages in the application with the exception of the Login and Signup Pages.

5.3.2 Server CSC

5.3.2.1 The server shall be built using Next.js and Flask.

5.3.2.2 The server shall communicate actions on the web page to the database back-end.

5.3.3 Database CSC

5.3.3.1 The database shall be composed of 3 data entities.

5.3.3.2 The database shall host a user entity.

5.3.3.2.1 The user entity shall store a primary key id attribute. This attribute will be a unique identifier for each user.

5.3.3.2.2 The user entity shall store a user first name attribute.

5.3.3.2.3 The user entity shall store a user last name attribute.

5.3.3.2.4 The user entity shall store a user password attribute.

5.3.3.3 The database shall host an organization entity.

5.3.3.3.1 The organization entity shall store a primary key id attribute.

5.3.3.3.2 The organization entity shall store an organization name attribute.

5.3.3.3.3 The organization entity shall store a mission statement attribute.

5.3.3.3.4 The organization entity shall store a website url attribute.

5.3.3.4 The database shall host an activity and event entity.

5.3.3.3.1 The activity and event entity shall store an organization id attribute.
5.3.3.3.2 The organization entity shall store a user id attribute.
5.3.3.3.3 The organization entity shall store a description attribute.
5.3.3.3.4 The organization entity shall store a website url attribute.

5.4 Performance Requirements

5.4.1 This product shall be based on the web.

-- This product must be used on a web browser.

5.4.2 This product shall take its initial load time based on internet connection strength.

-- This product must be used on a web browser.

5.4.3 This product's web application front-end shall be intuitive.

-- Users shall feel comfortable navigating the web app front-end, for it shall be reflective of other social media platform's functionalities.

5.4.4 The first signup shall be processed in 5 seconds.

-- When a user clicks on the Signup Button after entering their credentials into the text entry fields, it should take no more than 5 seconds for the system to register the new user and redirect the web page to the login page.

5.4.5 The first login shall be processed in 5 seconds.

-- When a user clicks on the Login Button after entering their credentials into the text entry fields, it should take no more than 5 seconds for the system to access the users account and log them into the system.

5.4.6 The first search request results shall be returned in 15 seconds.

--When a user enters a search query into the Search Bar, it should take no more than 15 seconds for the system to send the query to the database and return the first matching search result.

5.4.9 A post shall be completed in 10 seconds.

--When a user creates a new post, it should take no more than 10 seconds for the system to send the post to the server and have the post appear on the user timeline.

5.4.10 A friend request shall be completed in 10 seconds.

--When a user requests to connect with another user, it should take no more than 5 seconds for the system to process the request and notify the other user.

5.4.12 Trending page shall load trending results in 20 seconds.

-- When a user navigates to the trending web page, the results for trending organizations should take no more than 10 seconds.

5.4.13 Trending organizations shall update every hour.

-- The Global Giving API should update trending organizations every hour.

5.4.14 Profile picture updates shall be processed in 10 seconds.

-- Updates made to a user's profile photo should take no more than 10 seconds.

5.4.15 Profile banner updates shall be processed in 10 seconds.

-- Updates made to a user's banner should take no more than 10 seconds.

5.4.16 Profile biography updates shall be processed in 10 seconds.

-- Updates made to a user's biography should take no more than 10 seconds.

5.4.17 Loading Timeline results shall be processed in 10 seconds.

-- When visiting your own timeline, the general timeline should load in no more than 10 seconds.

5.4.18 Timeline updates shall be processed in 15 seconds.

-- Updates made to a user’s timeline, new posts from user's, new posts from other users, should take no more than 10 seconds.

5.4.19 Loading a user's profile photo shall be processed in 10 seconds.

-- When visiting a user's profile, the profile photo should load in no more than 10 seconds.

5.4.20 Loading a user's banner shall be processed in 10 seconds.

-- When visiting a user's profile, the banner should load in no more than 10 seconds.

5.4.21 Loading a user's biography shall be processed in 10 seconds.

-- When visiting a user's profile, the biography should load in no more than 10 seconds.

5.4.22 Saving an organization shall add the organization to the user’s saved list in 15 seconds.

-- When a user saves an organization to their profile, the organization should be added to their saved list in no more than 15 seconds.

5.4.23 Selecting a different trending or search result shall render the information in the display box in 10 seconds.

-- When a user selects a trending or search result, the information about the selected result should take no more than 10 seconds to render in the display box.

5.5 Project Environment Requirements

5.5.1 Development Hardware Requirements

Following are the hardware requirements for Charitable:

Category	Requirement
Processor	1.8 GHz Intel Core i5
Display	13.3-inch (1440 x 900)

Hard Drive Space	500MB
Memory	8 GB 1600 MHz DDR3

Following are the software requirements for Charitable:

Category	Requirement
Operating System	Mac, Windows, Linux
Compiler	Babel
Graphics	Adobe XD Figma Material UI
Collaboration and Version Control	GitHub VS Code
Front-End Framework	React.js Next.js
Server	Flask
Database	PostgreSQL
Development Framework	Agile SCRUM
SCRUM Manager	Jira
Web Browser	Any web browser that can run JavaScript

5.5.2 Execution Hardware Requirements

Following are the hardware requirements for Charitable:

Category	Requirement
Processor	1.8 GHz Intel Core i5
Memory	8 GB 1600 MHz DDR3

Following are the software requirements for Charitable:

Category	Requirement
Operating System	Mac, Windows, Linux
Web Browser	Any web browser that can run JavaScript

6.0 Software Design Document

6.1 Introduction

This document presents the architecture and detailed design for the software for the web application, Charitable. Charitable is a web application that allows users to easily access resources and get more involved in their communities.

Charitable will allow users to maintain a Charitable profile, learn about the nation's trending charitable organizations, research people and organizations, connect with other Charitable users, and share their own Charitable experiences. The intention behind this is to encourage users to get involved in their communities by inviting users to participate in a positive social media platform that revolves around charities and good causes.

Development for Charitable involves building a welcoming, user friendly front-end web application, building a database to hold information regarding users, posts, organizations, and activities, and developing a server which is used to communicate between the front-end web app and back-end database.

6.1.1 System Objectives Section

The objective of this application is to provide an easy-to-use social media platform that encourages user involvement in communities. Users can search for organizations, activities, and other users by using the search bar at the top of every page. Users can save organizations to their saved list by clicking the heart button at the top-right corner of every organization listing. Users can maintain a personal profile in which they can upload and change a profile picture, a banner photo, and a biography statement. Users can find trending organizations on the trending page. Users can use the navigation bar at the top of every page to navigate between the different pages in the system.

6.1.2 Hardware, Software, and Human Interfaces Section

6.1.2.1 Hardware Interfaces

6.1.2.1.1 Mouse

The mouse is used to click on buttons, text fields, and dialogs.

6.1.2.1.2 Keyboard

The keyboard is used to type in search queries.

6.1.2.1.3 Screen

The screen is used to see the website.

6.1.2.2 Software Interfaces Section

6.1.2.2.1 Https

Used to communicate through the browser.

6.1.2.2.2 TCP/IP

Used to communicate to our backend in MongoDB, as well as Auth0

6.1.2.3 Human Interface Section

6.1.2.3.1 Monitor

The monitor displays the GUI to the user.

6.1.2.3.2 Mouse

The mouse is used to click on buttons and dialogs, as well as to navigate scrollviews.

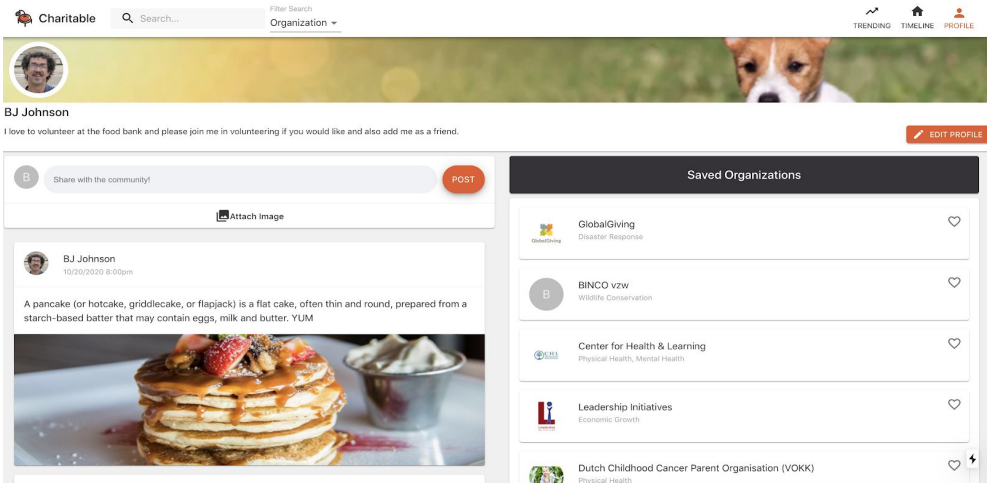
6.1.2.3.3 Keyboard

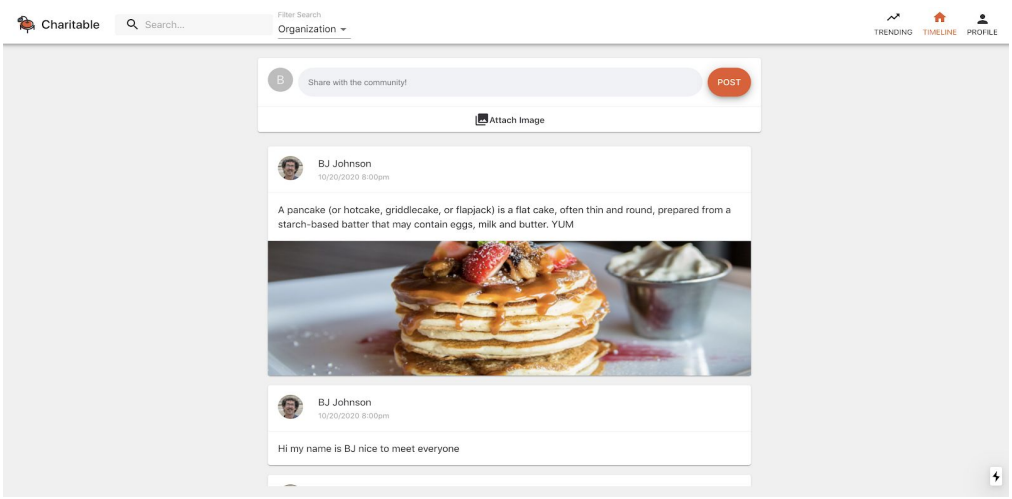
The keyboard is used to type in text fields.

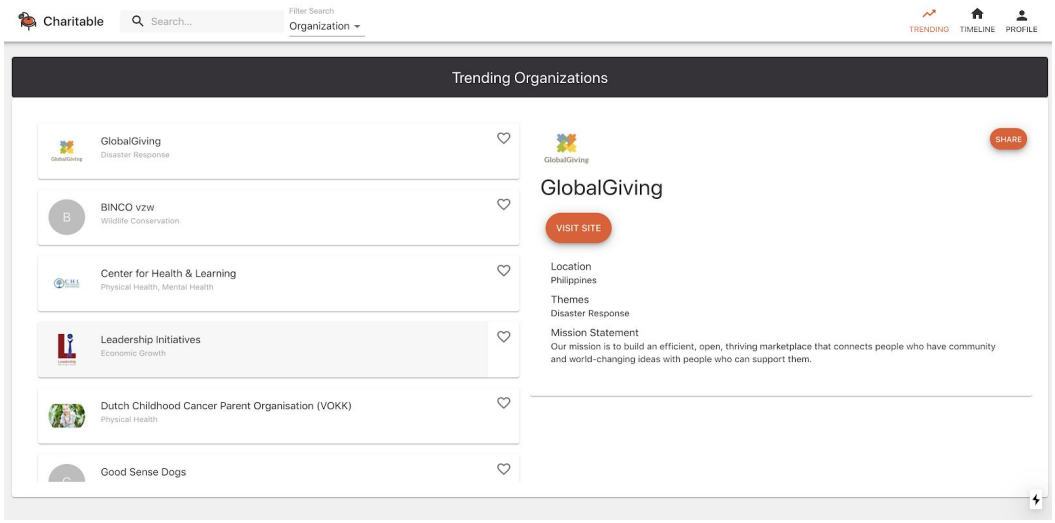
6.1.2.3.4 GUI

The GUI will contain a search field for users to search for organizations, users, or activities. The user can filter between the 3 options by using the dropdown directly to the right of the search bar. There are 4 pages the user can visit: Trending, Profile, Timeline, and Search. In the trending page the user will be able to see the top 10 trending organizations and can click on a given organization to see more info. They can also share or save organizations from the larger description. In the profile page users can edit their profile, view their saved organizations and click on them for more information and un-save them, and create a post. On the timeline page, users can see posts from those who they are following. On the search page, the user can search for organizations, users, or activities. After searching for organizations, the user can click on an organization to see more detail, save it, or share it. After searching for activities, the user can click on an activity to see more detail, save it, or share it. After searching for other users, the user can click on a user to see their profile, where they can see

the users profile information, past posts, and saved organizations. In addition, the user can choose to follow the user they are viewing.







6.2 Architectural Design

6.2.1 Major Software Components

The major software components in our application are a front-end web application, a server, and a backend database. The front-end web-application has a Registration Page, Login Page, Search Page, Trending Page, Profile Page, Timeline Page and Navigation Bar. The server that the application uses is the MongoDB cloud-based API. The backend database component is a document-centric database built using MongoDB which holds the information of our users, charitable organizations, charitable activities, and posts made by users. Users interact with the frontend through the web browser, which sends requests to the server and retrieves information from the MongoDB database. The data is then sent to the frontend and is displayed back to the user through the browser.

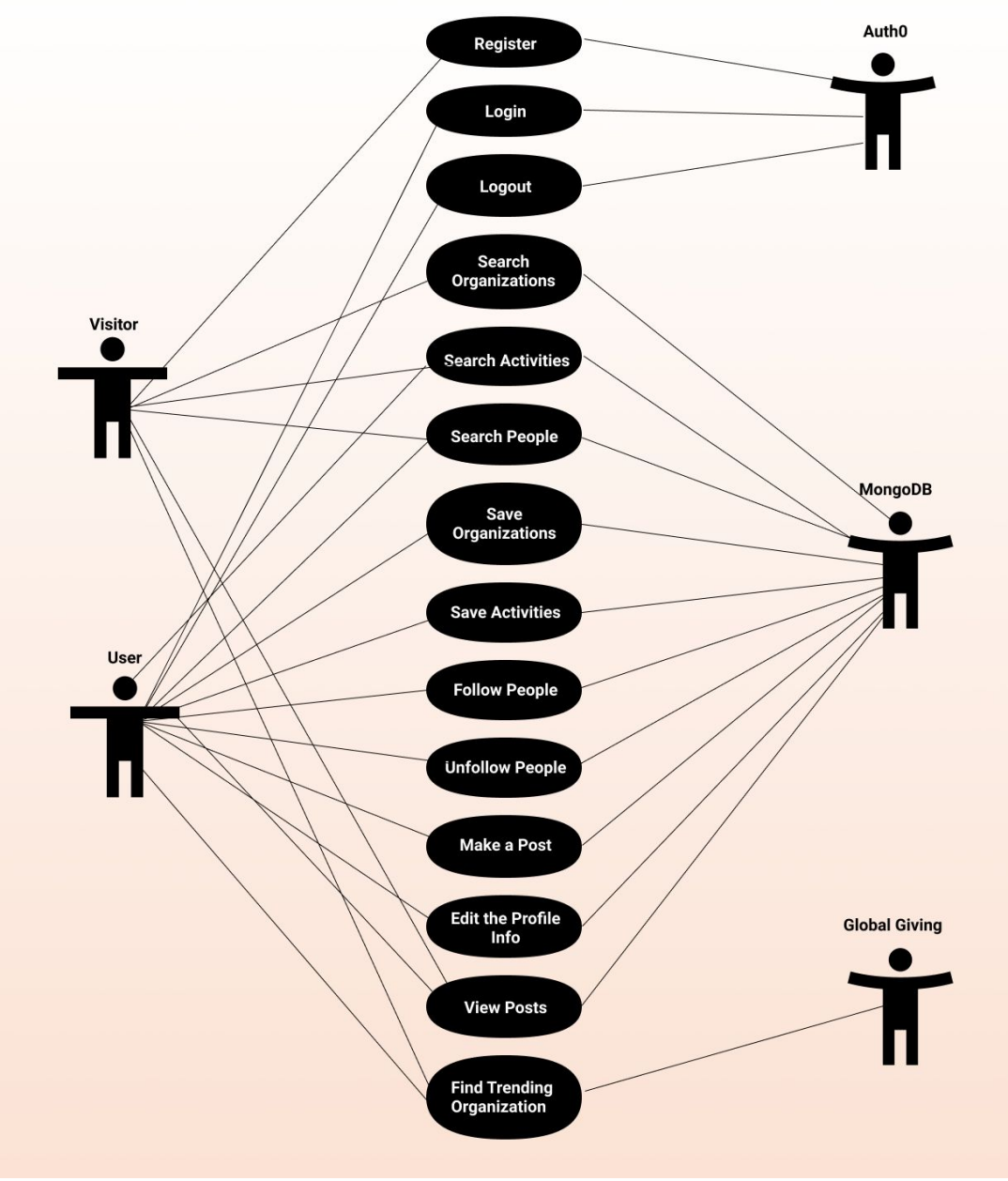
6.2.2 Major Software Interactions

The front-end web application uses HTTPS. Clients communicate with the database server through a regular TCP/IP socket.

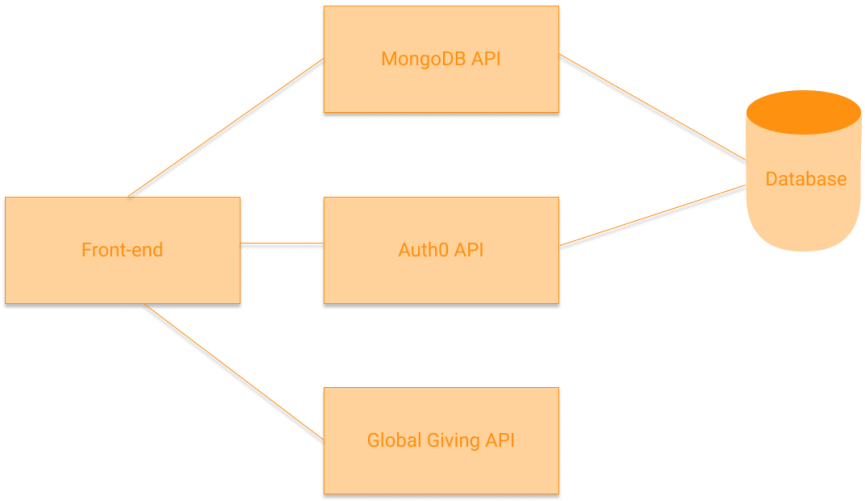
The front-end communicates with the backend database through MongoDB and MongoDB's cloud-based API. The MongoDB Wire Protocol is a simple socket-based, request-response style protocol. The API sends queries from the front-end to the database. Queries sent to the database return JSON Objects. The JSON Object from the back-end database will be sent to the front-end through API calls made with the MongoDB API.

6.2.3 Architectural Design Diagrams

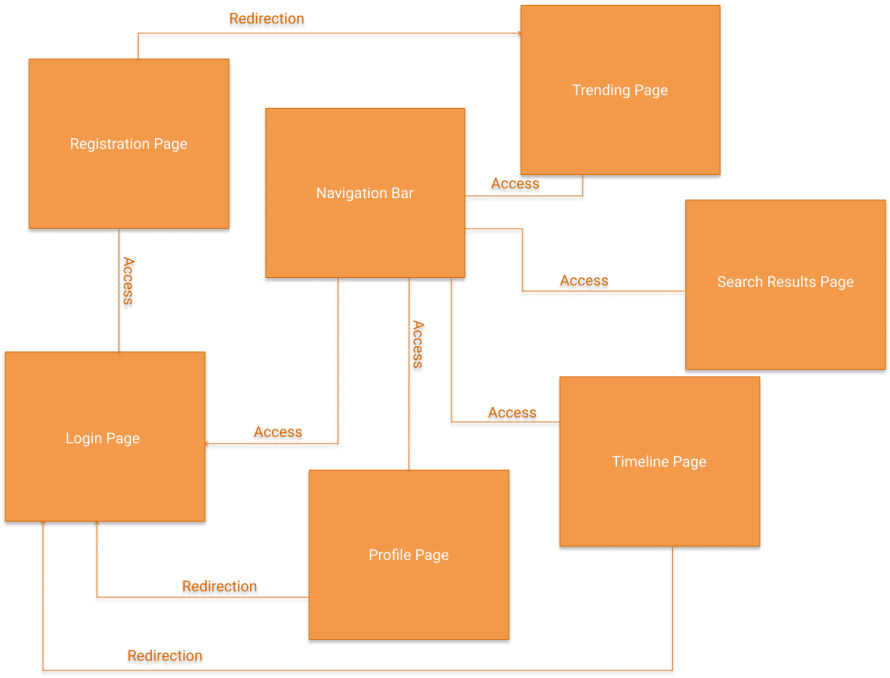
Use-Case Diagram



Top-level Diagram



Component Diagram



6.3 CSC and CSU Descriptions Section

6.3.1 Detailed Class Descriptions Section

6.3.1.1 Web Application Front-End CSC

-- Front-end web page interface for user interaction with the system.

6.3.1.1.1 Login Page CSU Class

-- Web page for logging into system.

- Email/Username Textbox
  - Field for user to type their email/username
- Password Text Box Field
  - Field for user to type their password
- Login Button Field
  - Button for user to login
- Reset Password Button
  - Button for user to reset password
- Login Method
  - The method that runs with user hits login button
- ResetPassword Method
  - The method that runs when user hits reset password button

6.3.1.1.2 Signup Page CSU Class

-- Web page for registering in system.

- Email Textbox
  - Field for user to type their email to use for their account.
- Username Textbox
  - Field for user to type their username to use for their account.
- Password Textbox
  - Field for user to type their password to use for their account. Characters are censored
- Sign Up button

        - When pressed creates an account for the user.
    - Signup Method

6.3.1.1.3 Timeline Page CSU Class

-- Web page to display the user and user's friend activity.

- Scroll View Field
    - This scrollview field displays the post
- Create Post Box Field
    - This is creates a post with text and possibly images
- Image Button
    - This button attaches an image
- Post Button
    - This button submits a post
- Post Method
    - This does an API call to the database that creates a post

6.3.1.1.4 Trending Page CSU Class

-- Web page to display trending organizations in the nation.

- Scroll View Field
    - List of Organization list items
- Organization List Item
    - Button that opens the Organization modal or Organization Description box
- Organization Modal
    - Modal that opens with Organization Description box if the page is smaller width
- Organization Description Box
    - Specific Organization information is listed here
- Share Button
    - Within the Organization Description Box, opens a Share Modal
- Share Modal
    - Modal that contains a Create Post Component that will create a post
- Save Button

        - On the Organization List Item, clicking will save the organization to a user
    - Visit Site Button
        - Button on Description box that links to an Organization Page
    - Fetch Organizations Method
        - API call method that gets the Trending Organization
    - Render Scroll View Method
        - Maps the data into Organization List Items
    - Open Modal Method
        - Opens modal if the page is width small

6.3.1.1.5 User Profile Page CSU Class

-- Web page for editing and maintaining user information.

- Profile Banner
    - Displays the name, profile picture, and Bio
- Create Post box
    - Creates new posts for the user
- Post text field
    - Text input for new post in create post box
- Attach image button
    - Button to attach an image to a new post, in create post box
- Post button
    - Button in create post box, to push the post to the backend
- Posts Scrollview
    - Scrollview that displays all the previous posts of the user
- Saved Organizations scrollview
    - Scrollview That displays all the users saved organizations. Functionally equivalent to the trending organizations scrollview, with the only difference

being the organizations that are displayed
- Edit Profile Button
    - Opens the Edit Profile Modal
- Edit Profile Modal
    - Contains fields for Bio, Name, and Avatar, allowing the user to alter these fields
- Edit Profile Submit
- Follow Button
    - Button that allows users to follow another user
- Unfollow Button
    - Button that allows users to unfollow another user
- Follow Method
    - Method that adds user ID to the given users following array
- Unfollow Method
    - Method that removes a user ID from the given users array
- Edit Profile Method
    - Method that runs when a user submits their profile edits
- Save Organizations Method
    - Method to save organization by adding it to an array of saved organizations for the user

6.3.1.1.6 Search Results Page CSU Class

-- Web page to display user search results.

- Scroll View Field
    - Displays results of search query
- Organization and Activity Description Box
    - Specific Organization and Activity information is listed here
- Organization and Activity Description Modal

        - Modal that opens with Organization and Activity Description box if the page is smaller width

6.3.1.1.7 Navigation Bar CSU Class

-- UI element used to navigate between web pages in application.

- Charitable Icon
    - Icon showing Charitable's logo.
- Search Bar
    - Text field for users to type in their search.
- Search Filter
    - Dropdown for users to choose a filter for their search, such as "organization," "activity," or "user."
- Trending Button
    - When pressed, redirects user to Trending Page.
- Timeline Button
    - When pressed, redirects users to timeline page if they have an account.
- Profile Button
    - When pressed, redirects users to profile page if they have an account.
- Search Organization Method
    - Method to send search by organization query to database
- Search Activity Method
    - Method to send search by activity query to database
- Search User Method
    - Method to send search by user query to database

6.3.1.2 Server CSC Class

-- Backend server to allow the front-end web page to interact with the database.

- MongoDB API

        - Queries our database and sends response in JSON format
    - Auth0 API
        - Creates users and allows us to query user information from the Auth0 database.
    - GlobalGiving API
        - Provides the trending organizations and updates hourly

6.3.1.3 Database CSC

-- Backend database to hold datasets for the system.

- Users send API calls to the backend through interaction with the frontend. The backend takes the call and returns a JSON object back to the frontend which is then displayed back to the user.

6.3.1.3.1 User Dataset CSU Class

-- Data table to hold registered user information.

- _Id
    - Unique ID provided by MongoDB to differentiate users
- Email
    - String containing the user's email
- Password
    - String containing the user's (hashed) password
- Username
    - String containing the user's username
- Nickname
    - String containing the user's nickname (identical to the user's username)
- Name
    - String containing the User's name which is displayed on their profile. Default's to "Please add your name"
- Profile picture
    - String containing url to the User's profile picture
- Banner picture

        - String containing url to the user's banner picture
    - Bio
        - String containing the user's bio. Defaults to "Please add your own bio"
    - Posts
        - Array containing the user's posts
    - Saved Organizations
        - Array containing the user's saved organizations
    - Saved Activities
        - Array containing the user's saved activities
    - Following
        - Array containing the ids of the users that the user is following

6.3.1.3.2 Organization Dataset CSU Class

-- Data table to hold registered organization information.

- _Id
    - Unique ID provided by MongoDB to differentiate organizations
- Name
    - String containing the organization's name
- Mission
    - String containing the organization's mission
- Logo URL
    - String containing a link to an image of the organization's logo
- Global Giving Id
    - String containing Id given to organization by GlobalGiving
- Countries
    - Array of countries the organization resides in
- Themes

        - Array of themes relating to the organization

6.3.1.3.3 Activities Dataset CSU Class

-- Data table to hold registered activities and events.

- _Id
    - Unique ID provided by MongoDB to differentiate organizations
- Global Giving Id
    - String containing Id given to organization by GlobalGiving
- Purpose
    - String containing the activity's purpose
- Description
    - String containing the activity's description
- Contact name
    - String containing the name of the person you can contact for more information about the activity
- Contact title
    - String containing the title of the person you can contact for more information about the activity
- Contact URL
    - String containing the URL of the person you can contact for more information about the activity
- Country
    - String containing the country that the activity is in
- Donation options
    - Array of Strings of possible donation options
- Project link
    - String that is a URL to a link sharing more information about the activity
- Region

        - String that indicates the region of the activity
    - Status
        - String that indicates whether the activity is active or retired
    - Summary
        - String that contains the summary of the activity
    - Global Giving organization Id
        - String that contains the ID to the organization given by GlobalGiving
    - Theme
        - String that indicates the theme of the activity
    - Impact
        - String that indicates the impact of the activity

6.3.1.3.4 Posts Dataset CSU Class

-- Data table to hold user's posts.

- _Id
    - Unique ID provided by MongoDB to differentiate posts
- User Id of poster
    - _Id of user who made the post
- Image URL
    - String containing url to a attached image
- Organization Id
    - Id of a attached organization
- Activity Id
    - Id of a attached activity
- Date posted
    - String containing the date the post was made
- Typed content
    - String containing the type content of the post

6.3.2 Detailed Interface Descriptions Section

6.3.2.1 Frontend Web Application

- Registration Page
  - Interfaces user data to the database on submit through TCP/IP
- Login Page
  - Interfaces with auth0 to get user session on submit using HTTPS
- Search Page
  - Interfaces with database through TCP/IP when users save organizations/activities or share posts about said organizations or activities
  - Receives search query through URL parameters
- Trending Page
  - Interfaces with Global Giving API through HTTPS
- Profile Page
  - Interfaces with database through TCP/IP when users edit their profile, follow another person, unfollow another person, unsave or save organizations/activities, create a post
- Timeline Page
  - Interfaces with database through TCP/IP when create a post, an when fetching posts from users they are following to display on the timeline
- Navigation Bar
  - Interfaces with Auth0 when clicking the login button to redirect to Auth0 login screen
  - Search box alters the URL

6.3.2.2 Server

- MongoDB cloud-based API
  - Interfaces with database through TCP/IP

6.3.2.3 MongoDB backend

- Users
  - Users can display their own info or search for other uses via the frontend. The frontend sends an API call to the database, which returns a JSON object containing all of the User's fields.

- Organizations
    - Organizations can be searched through the Search Page. Users can search organizations by entering a field that gets sent to the backend. The backend will then return a JSON organization object based on whether the field best matches the organization name, country, or theme.
- Activities
    - Activities are displayed on the User's Profile page or Posts page. For one's profile page, the backend returns all of the posts in the user's own Posts array. For the Posts page, the backend returns all of the posts from their page as well as the ones from everyone they are following. These will be displayed in chronological order.

6.3.3 Detailed Data Structure Descriptions Section

6.3.3.1 Web Application Front-End CSC

6.3.3.1.1 Login Page CSU Class

- Higher order function that returns a callback when login is successful

6.3.3.1.2 Signup Page CSU Class

- Higher order function that returns a callback when sign up is successful

6.3.3.1.3 Timeline Page CSU Class

- Array used to hold scroll view post elements

6.3.3.1.4 Trending Page CSU Class

- Array used to hold scroll view trending organizations elements

6.3.3.1.5 User Profile Page CSU Class

- Array used to hold scroll view post elements
- Array used to hold scroll view saved organizations elements
- Array used to hold saved activities elements

6.3.3.1.6 Search Results Page CSU Class

- Array used to hold scroll view search results elements

- Array of JSON Objects to hold search results
- Array used to hold search result list items

6.3.3.1.7 Navigation Bar CSU Class

- Pushes elements to router

6.3.3.2 Server CSC Class

- MongoDB API
  - Filters arrays and JSON objects

6.3.3.3 Database CSC

6.3.3.3.1 User Dataset CSU Class

- Document-centric database used to store user data
- Array of JSON Objects to hold user information
  - Array used to hold list of user's friends
  - Array used to hold user's saved organizations
  - Array used to hold user's saved activities

6.3.3.3.2 Organization Dataset CSU Class

- Document-centric database used to store organization data
- Array of JSON Objects to hold organization information
  - Array used to hold countries that organization is present in
  - Array used to hold themes that organization follows

6.3.3.3.3 Activities Dataset CSU Class

- Document-centric database used to store activity data
- Array of JSON Objects to hold activity information

- Array used to hold possible donation options

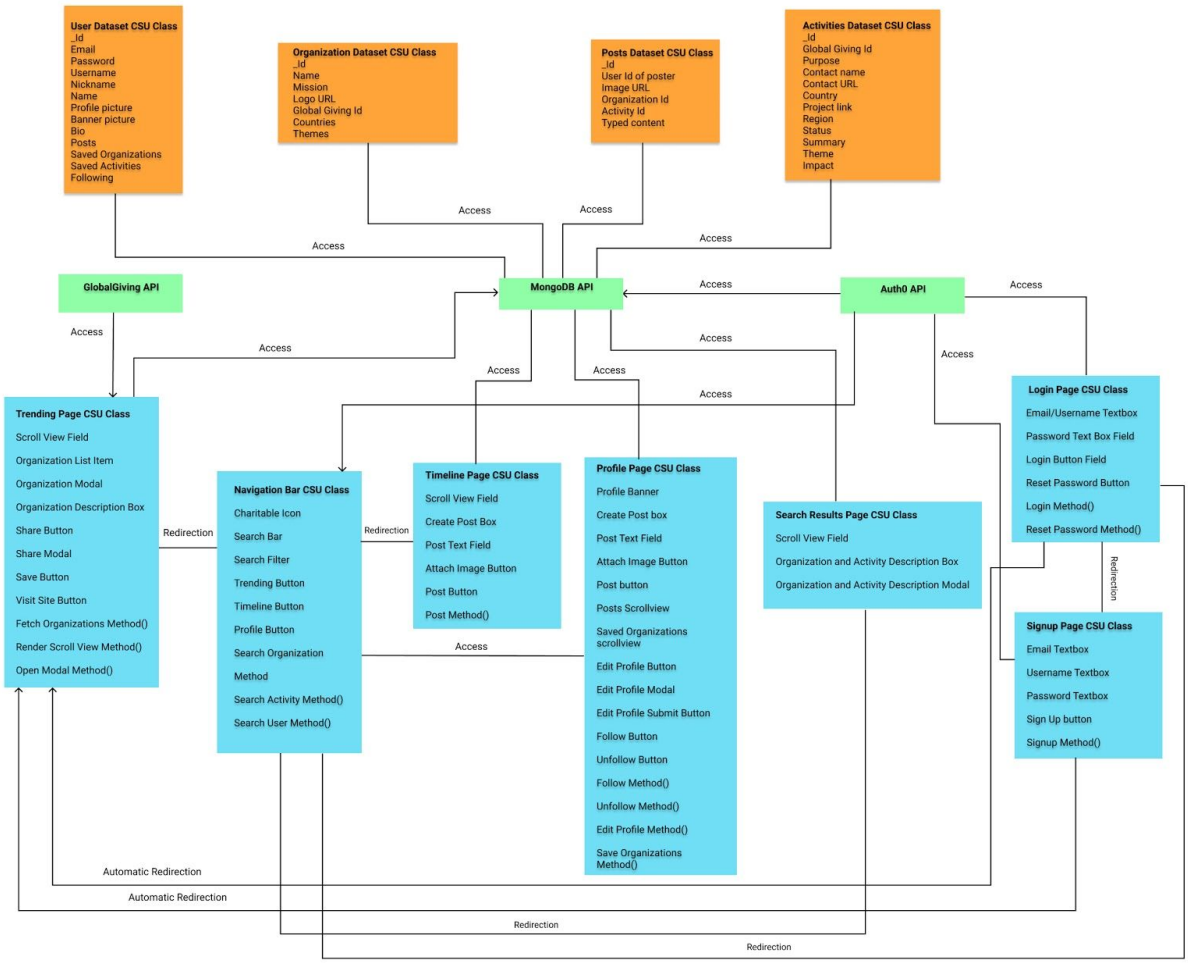
6.3.3.3.4 Posts Dataset CSU Class

- Document-centric database used to store post data
- Array of JSON Objects to hold information from user's posts

6.3.4 Detailed Design Diagrams Section

Legend

- Orange - Database CSUs
- Green - Server CSUs
- Blue - Frontend CSUs



6.4 Database Design and Description Section

6.4.1 Database Design ER Diagram Section

6.4.2 Database Access Section

Our database will be accessed using the MongoDB cloud-based API. We will run queries to get search results from the database to the front-end. We will also be

using the API calls to update user information, including new friend connections, new saved organizations, and updated bio statements.

6.4.3 Database Security Section

We will be using Auth0 to collect user registration information. Auth0 uses its own encryption and email verification process to protect its login data. When users login or sign up, they are redirected to Auth0 under our tenant, so Auth0 makes sure that their data is secure while they input their passwords or create passwords. Since our main concern is privacy of user data, our utilization of the third party Auth0 will provide the security measures we need to protect the important information in our database.

Additionally, our MongoDB Database is private and requires a username and password to access. The passwords are also encrypted in the database, so even people who have access to the database cannot view the user passwords for maximum user privacy.