

# CS 407 Group 27 Design Document

Project Name: GlassGift

Peter Xeros (pxeros@purdue.edu)

Anjali Malik (malik11@purdue.edu)

Jordan Pitlor (jpitlor@purdue.edu)

Noah Rinehart (nrinehar@purdue.edu)

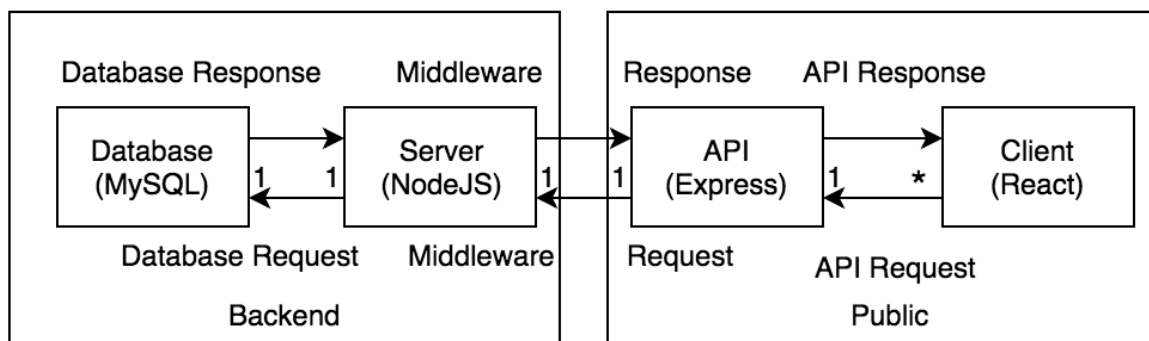
|   |          |
|---|----------|
| <b>Purpose</b>  | <b>2</b> |
| <b>Design Outline</b>   | <b>2</b> |
| <b>Design Issues</b>  | <b>3</b> |
| Functional Issues   | 3        |
| 1) How will users receive notifications?                          | 3        |
| 2) What Payment API are we using?                                 | 3        |
| 3) How should account creation be handled?                        | 3        |
| 4) Will NGOs be able to make donations?                           | 3        |
| 5) Should we have donation types?                                 | 4        |
| 6) Should NGO's have more than one category.                      | 4        |
| 7) Should we let donors be anonymous?                             | 4        |
| Non-Functional Issues   | 4        |
| 8) What database will we use?                                     | 4        |
| 9) What frontend framework will we use?                           | 5        |
| 10) What transfer protocol will we use between client and server? | 5        |
| 11) What backend will we use to power our service?                | 5        |
| 12) What will be the transfer format between client and server?   | 6        |
| 13) What hosting service will we use to host our web application? | 6        |
| <b>Design Details</b>   | <b>7</b> |
| Sequence Diagrams   | 7        |
| Register as a User  | 7        |
| Search and View an NGO  | 7        |
| Making a Donation   | 8        |
| Class Diagram   | 8        |
| UI Mock-Up  | 8        |

# Purpose

We are designing a Web Application that would allow users to donate to NGO's while keeping track of all donations so that all trades are transparent. These transparent trades can be used to verify an NGO and help with keeping track of monetary gains. We plan on having a thick server design in order to keep the app streamlined, and simply because it is faster with that design. The Client will function similar to GlassDoor, in which a user can search for an NGO to search for and use a designated payment method to make a donation to an NGO, while the server will link up between the Database and the client. The database will have all the data regarding every donation made, and all the information on the users and the NGOs.

## Design Outline

For our application, we will be using a client-server model. This model seems to be the best model to use for a web app. The client, or web app, will make API calls to the backend server, which will make CRUD calls to the database. The client's purpose is to give a UI for the user to interact with, and send any requests (or changes) to the backend to process. The server will process these requests, and send any needed changes to the database, and send back the client anything requested. The database will receive requests and process them, responding to the server a status.



# Design Issues

*\*Boldened answer is the chosen option*

## Functional Issues

### 1) How will users receive notifications?

**Email**

Text

In-app notifications

***Justification:***

We are already collecting their email, so we may as well use it. Texting often costs money, and notifications require 3 code bases (web, Android, and iOS)

### 2) What Payment API are we using?

**Paypal Sandbox**

Ether

Stripe

***Justification:***

Paypal is very widely used, and their sandbox is very developer friendly

### 3) How should account creation be handled?

**Username & password**

OAuth2 (Ex: Facebook, Google)

No login

***Justification:***

We should have authentication to ensure payments aren't fraudulent. If we use OAuth2, we run the risk of the API of our choice changing, which would cause us more work. Implementing our own authentication system shouldn't be too challenging.

#### **4) Will NGOs be able to make donations?**

**No**

Yes

#### ***Justification:***

This wouldn't have to happen enough for it to be worth supporting

#### **5) Should we have donation types?**

**Yes**

No

#### ***Justification:***

There are multiple reasons someone may want to make a donation, and we want to recognize that.

#### **6) Should NGO's have more than one category.**

**Yes**

No

#### ***Justification:***

Some NGOs don't necessarily fit within one category. For example, one category is international. This is a helpful category, but it doesn't completely describe an NGO. The effort to support more than one category isn't very difficult, so we decided the pros were worth it.

#### **7) Should we let donors be anonymous?**

**Yes**

No

#### ***Justification:***

There are no cons to letting donors be anonymous. We can still collect statistics on their donation without displaying their name.

## Non-Functional Issues

### **8) What database will we use?**

**mySQL**

MongoDB

PostgreSQL

#### ***Justification:***

The first decision we had in regards to what type of database we were going to use was picking between SQL and No-SQL. Although both have their distinct pros and cons, ultimately, we based our decision off of familiarity, and eventually chose SQL as our database. After this decision was made, the next one concerned which particular database server we would use, and we also based our decision off of familiarity and chose mySQL, which some team members had experience with. Fortunately, many of the different SQL database servers function very similarly, and any minute differences will probably never be faced.

### **9) What frontend framework will we use?**

Vue

**React**

Angular

#### ***Justification:***

Very similarly to our decision with what database we will use, we chose React as the frontend framework we will use due to experience. Because some members of the team already had used React before, it made sense to pick React as the frontend framework to use.

### **10) What transfer protocol will we use between client and server?**

SOAP

**REST**

RPC

***Justification:***

Although there are plenty of different transfer protocols to use for communication between client and server, REST by far makes the most sense. SOAP has aged out, and is no longer popular, while REST is now the de-facto standard for web applications. RPC, on the other hand, could work in this scenario, but it is not really meant for this. Instead, it is for inter-service communication, something that we will not have to focus on.

**11) What backend will we use to power our service?**

**NodeJS**

PHP

Python

Java

***Justification:***

The decision to use NodeJS was easy, since some of our team had a lot of experience with it, while to others had an interest to learn it due to its explosion in popularity in recent years. Our team feels more confident choosing something we have experience in or the desire to learn as it will help in the development of our project.

**12) What will be the transfer format between client and server?**

**JSON**

XML

***Justification:***

The format that we will use for transferring data between client and server is JSON, because it is the de-facto standard, as well as something that the team has experience on. This couples well with the frontend framework (React) we will be using as well as our backend (NodeJS) since both are based on JavaScript, and the transferred data would not have to be explicitly serialized and deserialized.

**13) What hosting service will we use to host our web application?**

**Heroku**

Github pages

AWS

Google Cloud

Azure

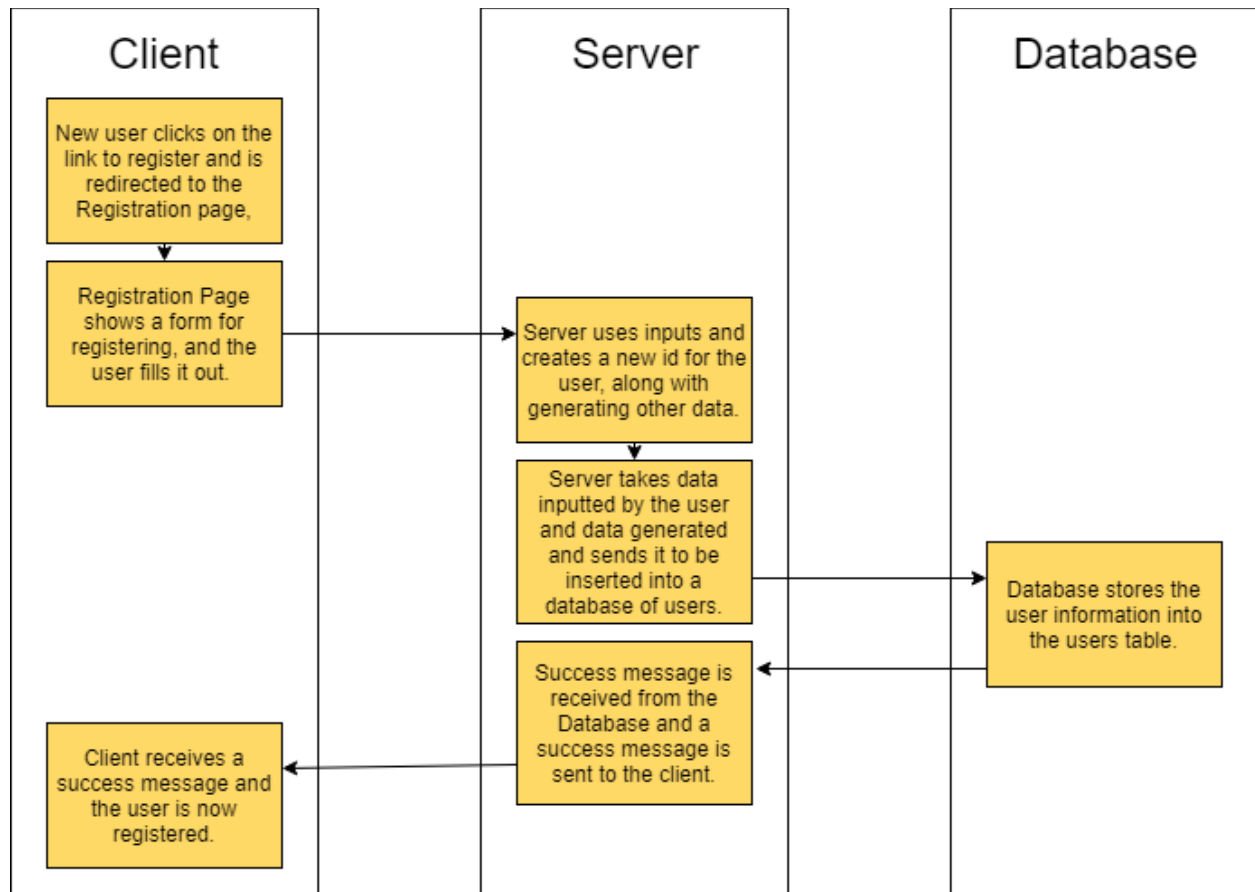
***Justification:***

Choosing a hosting service depended on how easy we wanted to deploy our project. Although the big cloud providers AWS, Google Cloud, and Azure have “free trials” to use their service, it may be troublesome to setup and use. Github pages also does not do dynamic hosting, which is something that we need since we’re using a backend server and a database. Heroku looked to be our ideal candidate, since of its ease of use (just pushing to an external repo to deploy), as well as its free price.

# Design Details

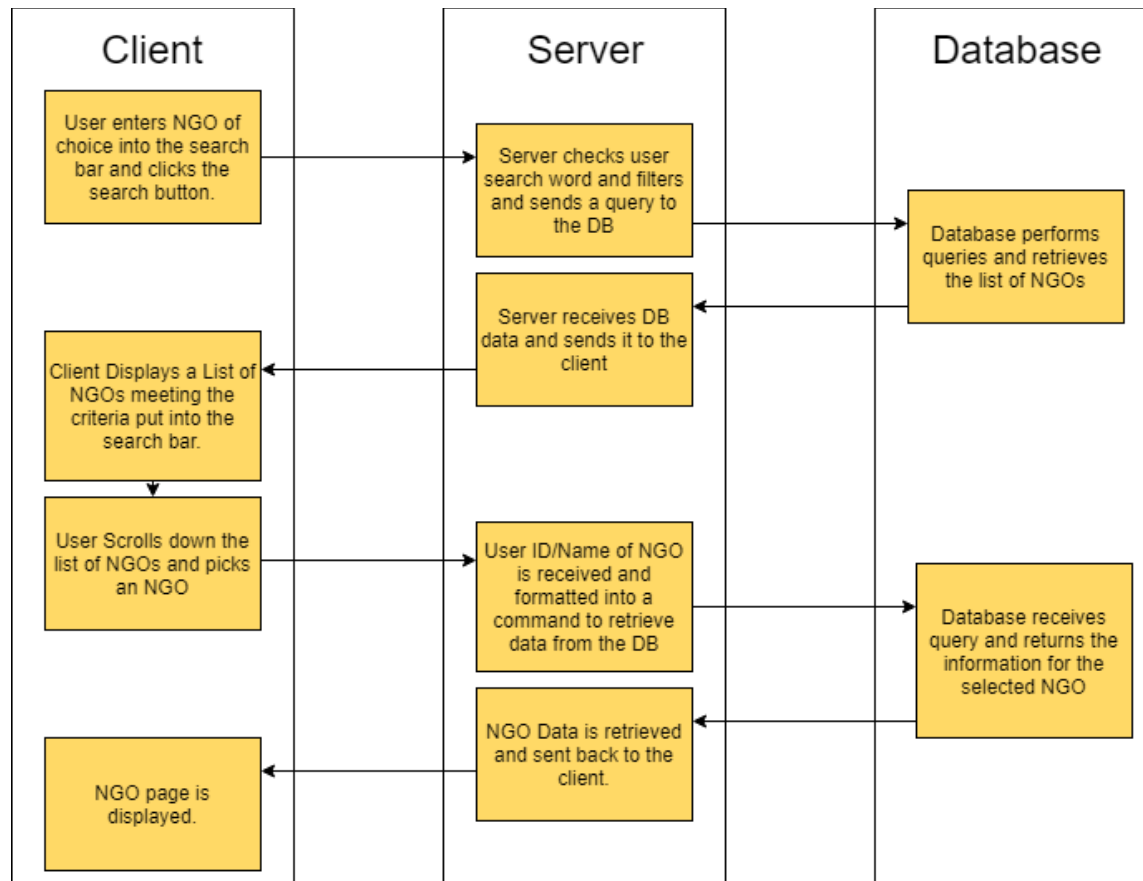
## Sequence Diagrams

### *Register as a User*

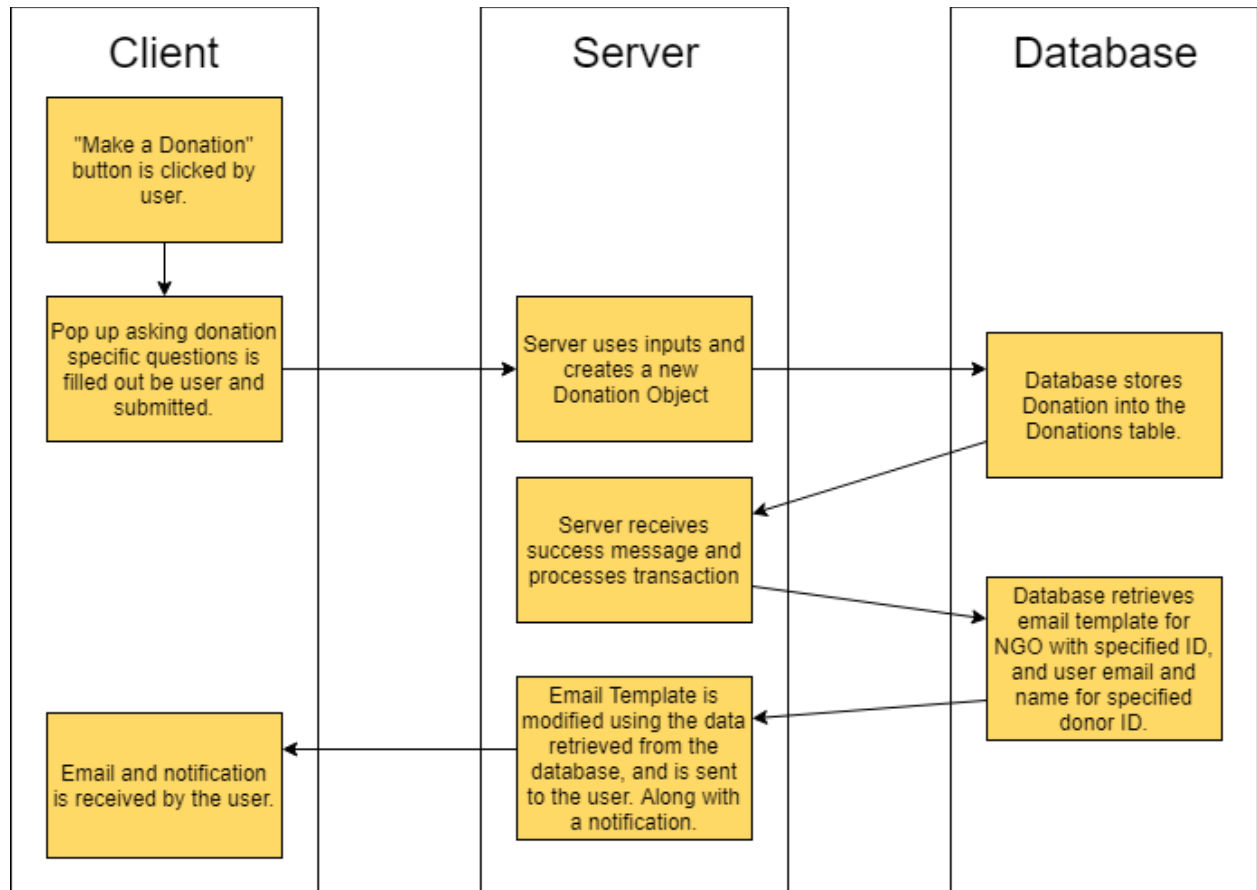




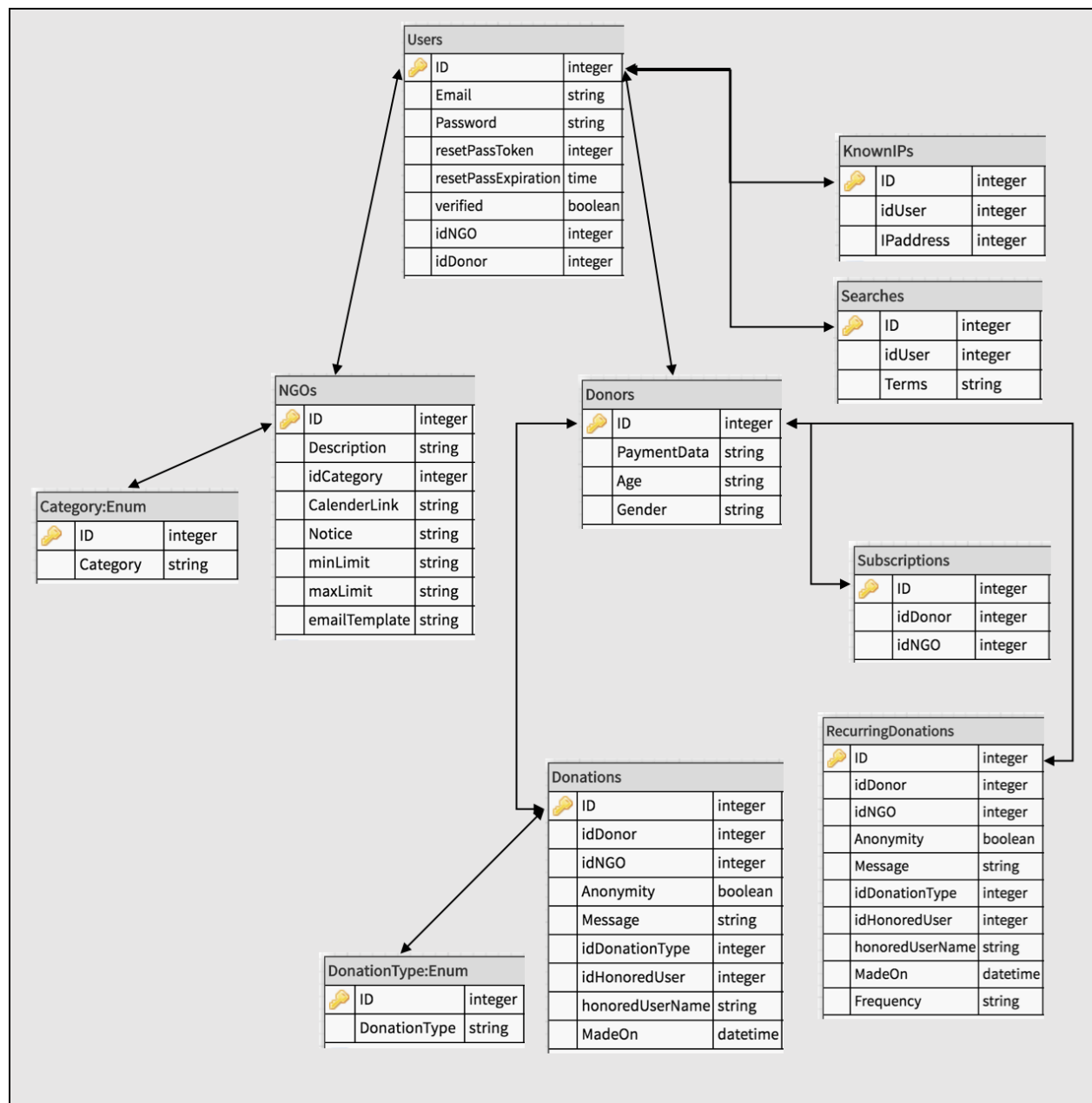
## *Search and View an NGO*



## Making a Donation



## Class Diagram



## ***Users***

- Users is the highest level class for all user accounts. An instance of User is created when a person registers on our system.
- We distinguish users into NGOs and donors. This distinction must be specified during the process of creating a new account.
- This class stores unique information about each account provided during registration, and is to be used to implement process authorization and authentication.

## ***NGOs and Donors***

- NGOs and Donors classes each represent unique users with specific information about their accounts on Glassgift.
- NGOs have profiles on Glassgift, therefore, this class is used to store the necessary details about their page. The user defined as an NGO on our service, will provide additional information such as, description about their organization, category, and more, during profile creation.
- Donors, on the other hand, do not have profiles on Glassgift, however, some additional information will be stored such as their age and gender (for analysis of their activity on the app), as well as they will have a convenient option to save their payment data on our platform.
- The IDs on each class are used for referencing information from other classes in our system.

## ***Category***

- This is an enum class which simply contains different types of categories that can be associated with NGO accounts.
- The category options are as follows: Animals; Arts, Culture, & Humanities; Community Development; Education; Environment; Health; Human and Civil Rights; Human Services; International; Religion; Research and Public Policy; Other

## ***Donations***

- This class stores all details about all transactions made on our application, such as, donor ID, NGO ID, message, if donation was anonymous or not, donation type, honored user (if any), honored person name (if any), and when the donation was made on.
- This class will only permit inserts, therefore, no updates and deletes will be made to the information presented in Donations.
- The donation type is another enum class that represents 3 types of donations: On behalf of one's ownself, In honor of someone, and In memory of someone. The donation type must be specified when making a donation on our platform.
- Recurring Donations is a similar class to Donations that stores all the same information expect it also stores the frequency with which a donor wants to make donations to an NGO, for example, monthly, yearly, etc.
- The information stored in Donations is intended to be transparent with exception to the Donor's identity if the anonymity was requested.

## ***Known IPs***

- Known IPs class stores all the IP addresses known for a user.
- This is used for account security, so when a user logs in using an unknown IP address, they will have to first verify their account before gaining access.

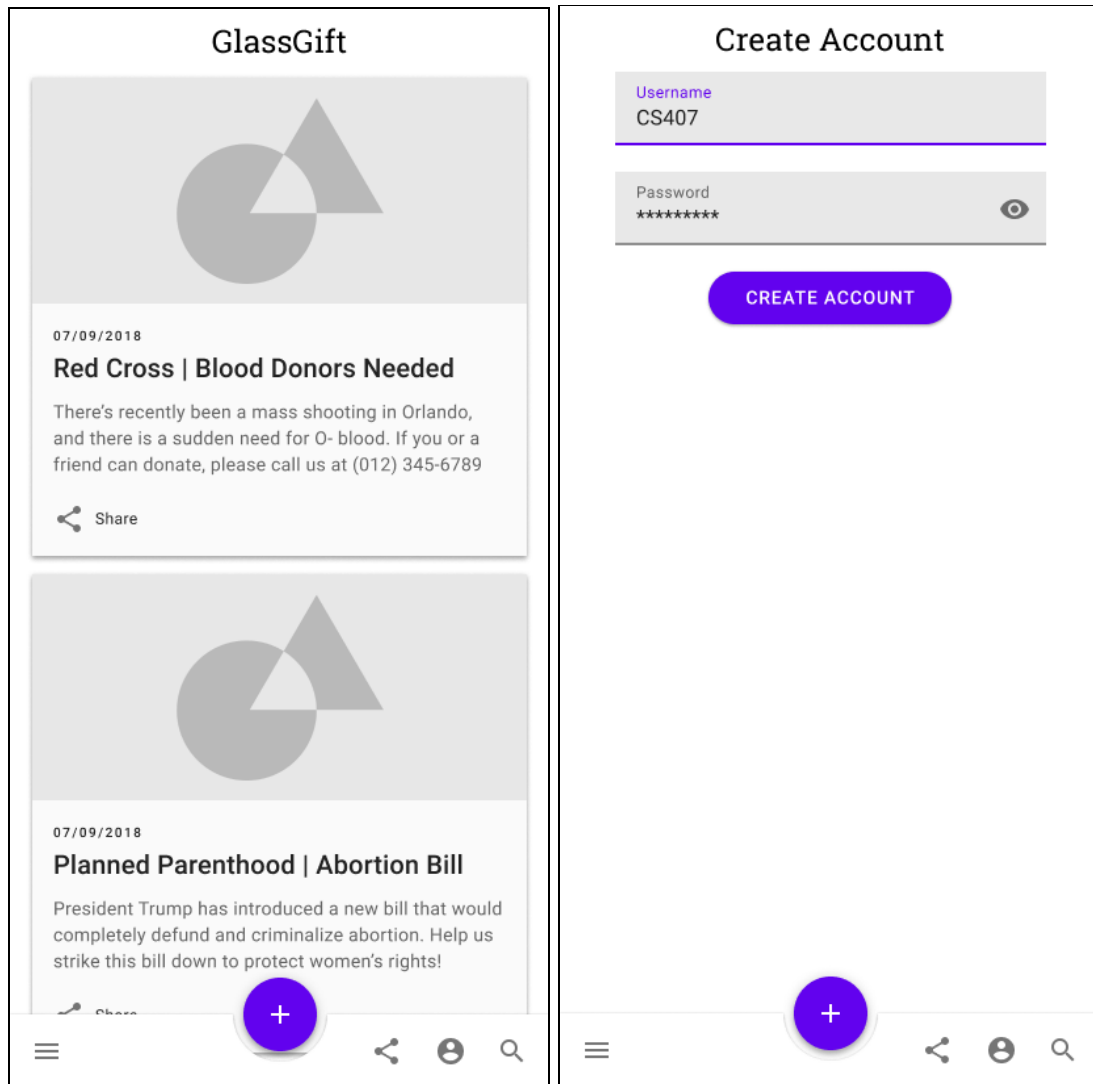
## ***Searches***

- Searches class is used to store the previous searches of a user.
- This class's stored information is used to help user make their searches convenient on the platform.

## ***Subscriptions***

- On Glassgift, a donor will have an option to subscribe to email communications from the NGOs of their choice. The Subscriptions class is used to store this information.

## User Interface Mock-Ups



GlassGift

Search for NGO

Name

Location

☒ Only Verified

SEARCH

Red Cross

support@redcrosstraining.org

Email

Washington, DC, USA

Location

The American Red Cross is a humanitarian organization that provides emergency assistance, disaster relief, and disaster preparedness education in the United States

Description

https://goo.gl/red\_cross\_calendar

Calendar

07/09/2018

Blood Donors Needed

There's recently been a mass shooting in Orlando, and there is a sudden need for O- blood. If you or a friend can donate, please call us at (012) 345-6789

Share

## New Payment

\$00.00

### NGO:

- ☐ On Behalf of Myself
- ☐ In Honor Of Someone
- ☐ In Memory of Someone

☒ Anonymous

- Red Cross
- Doctors Without Borders
- Planned Parenthood
- Red Cross
- United Way

