

# SW Engineering CSC648/848 FALL 2020

## Let's Donate

Team 3

Milestone 3

November 19, 2020

Team Lead: Himanshu Garg (hgarg@mail.sfsu.edu)

Front-End Lead: Milo Abril (mabril0526@gmail.com)

Back-End Lead: Marlon Bustamante (mbustama@mail.sfsu.edu)

Github Master: Suman Basaula (Sbasaula@mail.sfsu.edu)

Database Manager: Dipendra Dhoj Rana (drana1@mail.sfsu.edu)

Full Stack Developer: Kunyu Ruan (kruan1@mail.sfsu.edu)

Date	Version
Oct 1, 2020	Milestone 1 v1
Oct 8, 2020	Milestone 1 v2
Oct 29, 2020	Milestone 2 v1
Nov 12, 2020	Milestone 2 v2
Nov 19, 2020	Milestone 3, v1

## **Table of Contents**

<b>1. Data Entities and Definitions</b>	<b>3</b>
<b>2. Functional Requirement V3</b>	<b>6</b>
<b>3. UI Mockups</b>	<b>9</b>
<b>4. High level database architecture and organization</b>	<b>16</b>
<b>5. High level diagrams</b>	<b>20</b>
<b>6. Detailed List of Contributions</b>	<b>23</b>

# 1. Data Entities and Definitions

**User or registered User:** Any visitor to Let's Donate website is a user if they own an account on Let's Donate.

Every registered user will have

- a. A name - **name**.
- b. A username - **userName**.
- c. A password - **password**.
- d. A role - **admin, donor, receiver**

**A. Admin:** An incharge of Let's Donate to keep a track of all the activities over the website and tracking of malicious accounts to delete them if required.

The Admin will have

1. Access to suspicious posts - **postSuspiciousList**
2. Access to deleted posts - **postsDeletedList**
3. Access to the list of spam users - **spamUsersList**
4. Access to deleted users - **deletedUserList**.

**B. Donor:** A registered user who will be able to post ads and donate money over the website.

Every donor will have

1. a list of posts uploaded - **postsUploadedList**
2. A list of products donated - **productsDonatedList**
3. A list of non-utilized products - **availableProductsList**
4. Pick up location for the product - **productLocation**

**C. Receiver:** A registered user who can contact the donor and ask for the uploaded product or cash donation.

Every receiver will have

1. A list of products received - **productsReceivedList**

2. A count of threshold limit for products - **thresholdLimit**
3. A count of products picked for the month - **productsReceived**
4. List of donors contacted - **donorsContactedList**
5. List of liked products - **likedProductsList**

**Guest:** An unregistered user who will be able to view the available products posted for donation but need to register to get or post products.

**Post:** Post products like chairs, books, tables, and every other material that you want to donate.

Every Post will have

- a. The product to be given away - **product**.
  - A. **Product:** A product can be anything that any user wants to donate like television, bed, bike, etc.

Every product will

1. have a name - **productName**.
  2. have a description - **productDescription**.
  3. Depict the condition of the product - **productCondition**.
- b. The name of the donor - **donorName**.
  - c. The location of the donor - **donorLocation**.
  - d. Date of posting - **postDate**.
  - e. A picture of the product - **productImage**
  - f. Number of likes - **numInterestedUsers**

**Sign up:** User sign up is required for making and receiving donations.

- a. Username is required - **userName**.
- b. Password is required - **userEmail**.
- c. Full Name is required - **name**.

**Show Interest in Post:** Users can show interest in any of the products mentioned on the website by hitting the like button for whichever product they like.

- A. List of users who liked the post - **interestedUsersList**.
- B. The Product - **product**.
- C. The name of the donor - **donorName**.

**Grab the Product:** Contact the donor for whichever product/posting you like via direct message or email provided by the website when interest is shown.

Every product grabbed will have details about

- a. the donor - **donor**.
- b. the receiver - **receiver**.
- c. the posting - **post**.

**Number of Products Received Limit:** There will be a limit on the number of products a receiver can get i.e. 5 per month.

**Ask for Financial Support:** Write your story about why you need money and attract donors for receiving money donations.

Every fundraiser post will have the following information

- a. A fundraiser title - **fundRaiserTitle**.
- b. Image verification - **verifiedImage**.
- c. Reason for fund requirements - **fundraiserDescription**.
- d. User asking for funds - **fundSeeker**

**Profile:** User profile will show the products they donated or received.

**Product Info:** A little information is required for every product when uploading a post to describe the condition of product and other details.

**Report Spam Posts:** Users can always report spam posts which can be deleted by the Admin.

**Search:** An user can use the search bar to find a product based on product name.

## 2. Functional Requirement V3

Prioritizing requirement according to **Priority 1**

### **Guests:**

- 2.1 Guests shall be able to register themselves in the website
  - 2.2 Guests shall be able to create their own accounts
  - 2.3 Guests shall be able to become users
- 3.1 Guests shall be able to browse donations and fundraisers
  - 3.2 Guests shall be able to view donations page
  - 3.3 Guests shall be able to view fundraisers page
- 5.1 Guests shall be able to view user profiles
  - 5.2 Guests shall be able to view other user posting history
  - 5.3 Guests shall be able to view other user receiving history

### **Users:**

- 1.1 Users shall be able to login to the website
  - 1.2 Users shall be able to input an existing email
  - 1.3 Users shall be able to input an existing password
- 2. Users shall have a username
- 3. Users shall have a password.
- 5.1 Users shall be able to visit their profile
  - 5.2 Users shall be able to check their posting donation history
  - 5.3 Users shall be able to check their receiving donation history
- 9.1 Users shall be able to reset their password
  - 9.2 Users shall be able to get verified via text message
  - 9.3 Users shall be able to get verified via email
- 12.1 Users shall be able to shop
  - 12.2 Users shall be able to search browse items
  - 12.3 Users shall be able to direct message other users
- 26. Users shall be able to endorse fundraisers
- 27. Users shall be able to make payments
- 28. Users shall be able to place items up for donation.
- 29. Users shall be able to set up description of items for donation
- 30. Users shall be able to remove their own posts
- 31. Users shall be able to remove their own fundraisers
- 39. Users shall be able to delete their own profile
- 40. Users shall be able to log out of the website

### **Website:**

- 1.1 The website shall be regularly updated regarding items.
  - 1.2 The Website shall update whenever a new post is made
- 2. The website shall be regularly updated regarding history,

## Prioritizing requirement according to **Priority 2**

### **Guest:**

- 4. Guests shall be able to contact administrators

### **User:**

- 4.1 Users shall be given an editable profile
  - 4.2 Users shall be able to edit their user page
- 7. Users shall be able to see other people's donation history
- 8.1 Users shall be able to set their profiles as private
  - 8.2 Users shall be able to prevent guests from seeing their profile page
- 17.1 Users shall be able to search items by distance
  - 17.2 Users shall be able to sort browsed items by closest to farthest distance from their location
- 18.1 Users shall be able to search items by date
  - 18.2 Users shall be able to sort browsed items by date posts are made
- 19.1 Users shall be able to search items by name
  - 19.2 Users shall be able to sort browsed items in alphabetical order of user names
- 32. Users shall be notified if their donating item has been requested by another user
- 44. Users shall be able to donate directly to the product

### **Website:**

- 3. Website will update according to the item's preference set by the user

### **Administrator:**

- 1. Administrator shall receive reports about users
- 2. Administrator shall receive complaints about users

## Prioritizing requirement according to **Priority 3**

### **User:**

- 14. Users shall be able to keep a Wishlist of items
- 15. Users shall be able to add items in their wish list
- 16. Users shall be able to remove items in their wish list
- 20. Users shall sort donations as they wish (i.e. by endorsement)
- 21. Users shall be able to search up other users
- 22. Users shall be able to see other users' items
- 23. Users shall be able to see other users' fundraisers
- 24. Users shall reserve donated items for a certain period of time
- 33.1 Users shall be able to direct message each other
  - 33.2 Users shall be able to message each other through website's messaging system
  - 33.3 Users shall be able to send emails to each other
- 34. Users shall be provided with recommended items based on their view history

- 35.Users shall be able to rate other users on their credibility
- 36.Users shall be able to send complaints about other users to administrators.
- 37.Users shall be able to send reports about other users to administrators.
- 38.Users shall be able to get support from administrators
- 41.Users shall be able to appeal for a deleted post by contacting an Administrator
- 42.Users shall be able to appeal for a deleted fundraiser by contacting an
- 43.Users shall be able to follow other users

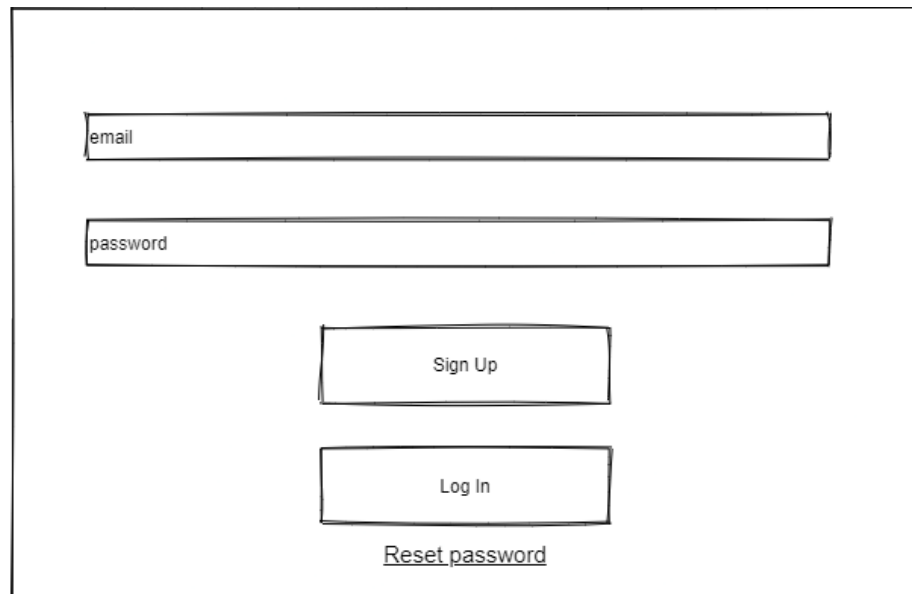
**Administrator:**

- 3. Administrators shall have the ability to ban users from using features of the website
- 5. Administrators shall be able to delete the spam items
- 6. Administrators shall be able to remove posted items
- 7. Administrators shall be able to remove posted fundraisers
- 8. Administrators shall be able to bring back a fundraiser if it has been removed
- 9. Administrators shall be able to bring back a post if it has been removed
- 10.1 Administrators shall be able to send notifications to all users
  - 10.2 Administrator shall be able to send message to users via email
  - 10.3 Administrator shall be able to send message to users via phone message



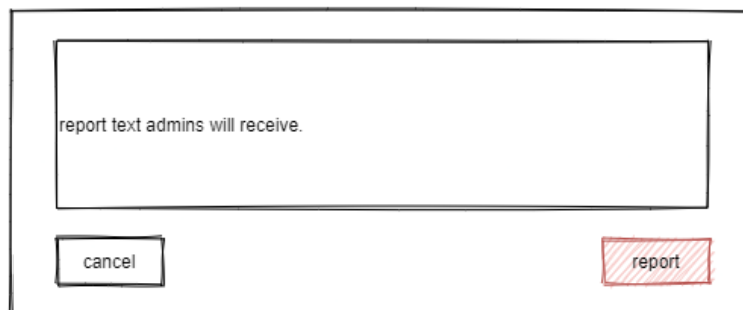
### 3. UI Mockups

#### Login/Sign Up



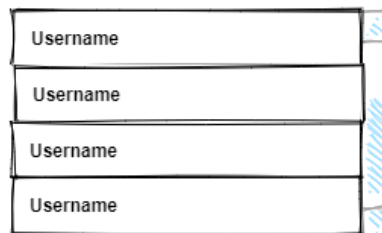
A UI mockup for a login/sign up form. It features two input fields: the top one is labeled 'email' and the bottom one is labeled 'password'. Below the password field are two buttons: 'Sign Up' and 'Log In'. At the bottom of the form is a link labeled 'Reset password'.

#### Report Form



A UI mockup for a report form. It contains a large text area with the placeholder text 'report text admins will receive.'. At the bottom left is a button labeled 'cancel', and at the bottom right is a button labeled 'report' with a red border and diagonal hatching.

#### User Chat Tabs



A UI mockup for user chat tabs. It shows a vertical stack of four tabs, each labeled 'Username'. The right side of the tabs is decorated with blue diagonal hatching.

#### Browsing type



A UI mockup for browsing type. It shows a vertical stack of two tabs: the top one is labeled 'Products' and the bottom one is labeled 'Fundraisers'. The right side of the tabs is decorated with blue diagonal hatching.

## Home Page

This is the landing page of the product, the first thing users see when they visit the website.

The main background would be a picture that symbolizes what Let's Donate stands for, i.e. two strangers shaking hands/hugging.

Our Logo (should we have one) would appear in the top left of the page.

A search bar would appear near the center of the page to be used to search for specific tags/items. If left empty, the user will be taken to the browsing page.

## Logo button

Clicking on this in any of the pages will bring the user back to the landing page

The wireframe shows a header section with a 'LOGO' placeholder on the left, a 'Search' bar in the center, a 'Product' dropdown menu on the right, and a 'Sign Up / Log In' button on the far right. Below the header is a large hero section with the text 'Let's Donate' and 'Service Statement'. The main content area is divided into two sections: 'Donations' and 'Fundraisers'. The 'Donations' section contains a grid of four item cards, each with a placeholder image, 'Item Name', and a description. The 'Fundraisers' section contains a single card with a placeholder image, 'Fundraiser Name', and a description. Blue arrows indicate the flow from the 'Logo button' text to the 'LOGO' placeholder, from the 'Search Bar' text to the 'Search' bar, from the 'Featured Product List' text to the 'Donations' and 'Fundraisers' grids, and from the 'Sign Up / Log In Form' text to the 'Sign Up / Log In' button.

## Search Bar

Any key words typed here will determine what is displayed in the browsing page. This can be left empty to initially browse without any filters.

Depending on whether the dropdown says Products or Fundraisers, the searchicon will take the user to the Products page or Fundraisers Page.

## Featured Product List

There are two of these lists, both with their own scrollbars.

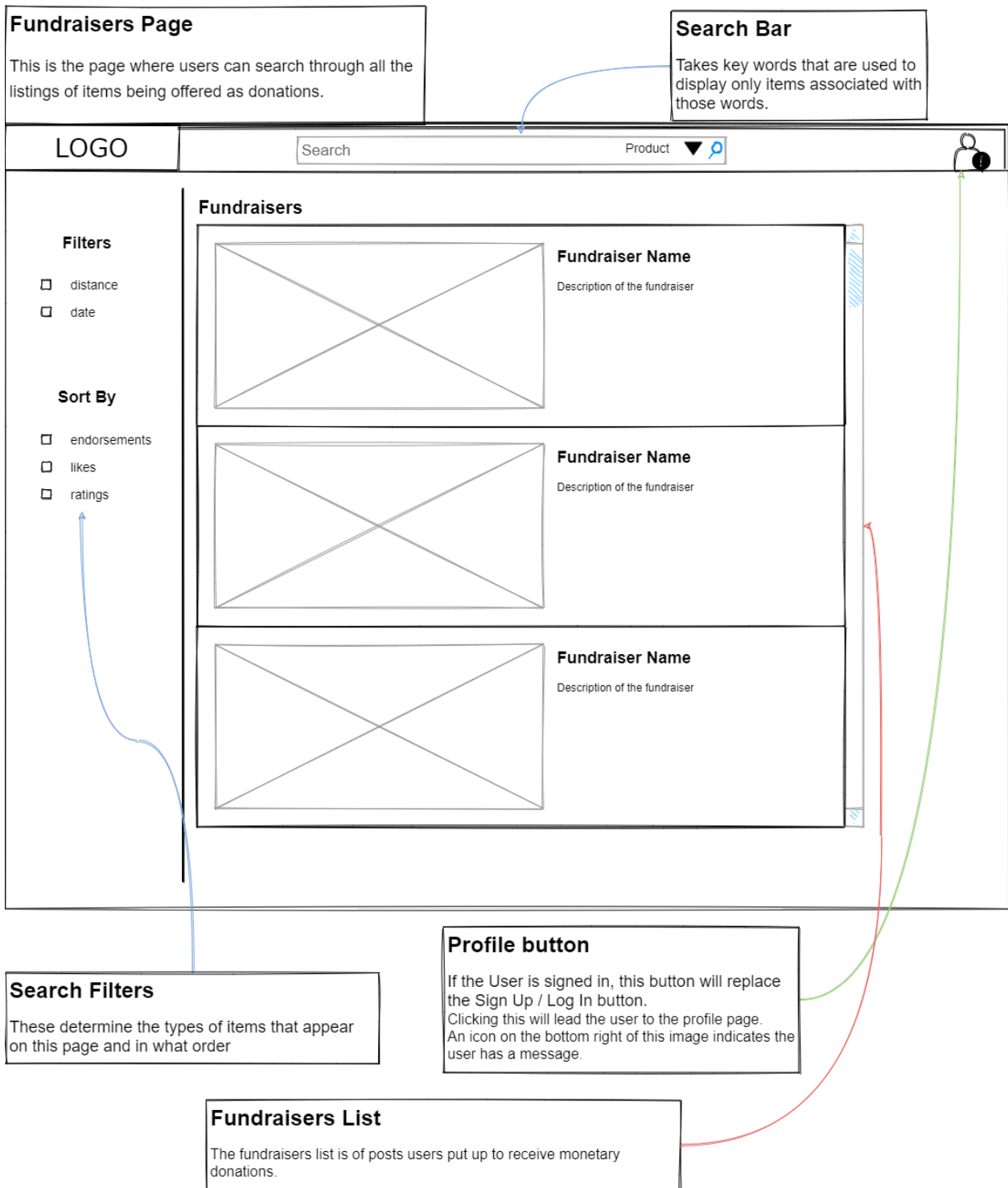
The donations grid holds items that users are giving away while the fundraisers grid is of posts users put up to receive monetary donations.

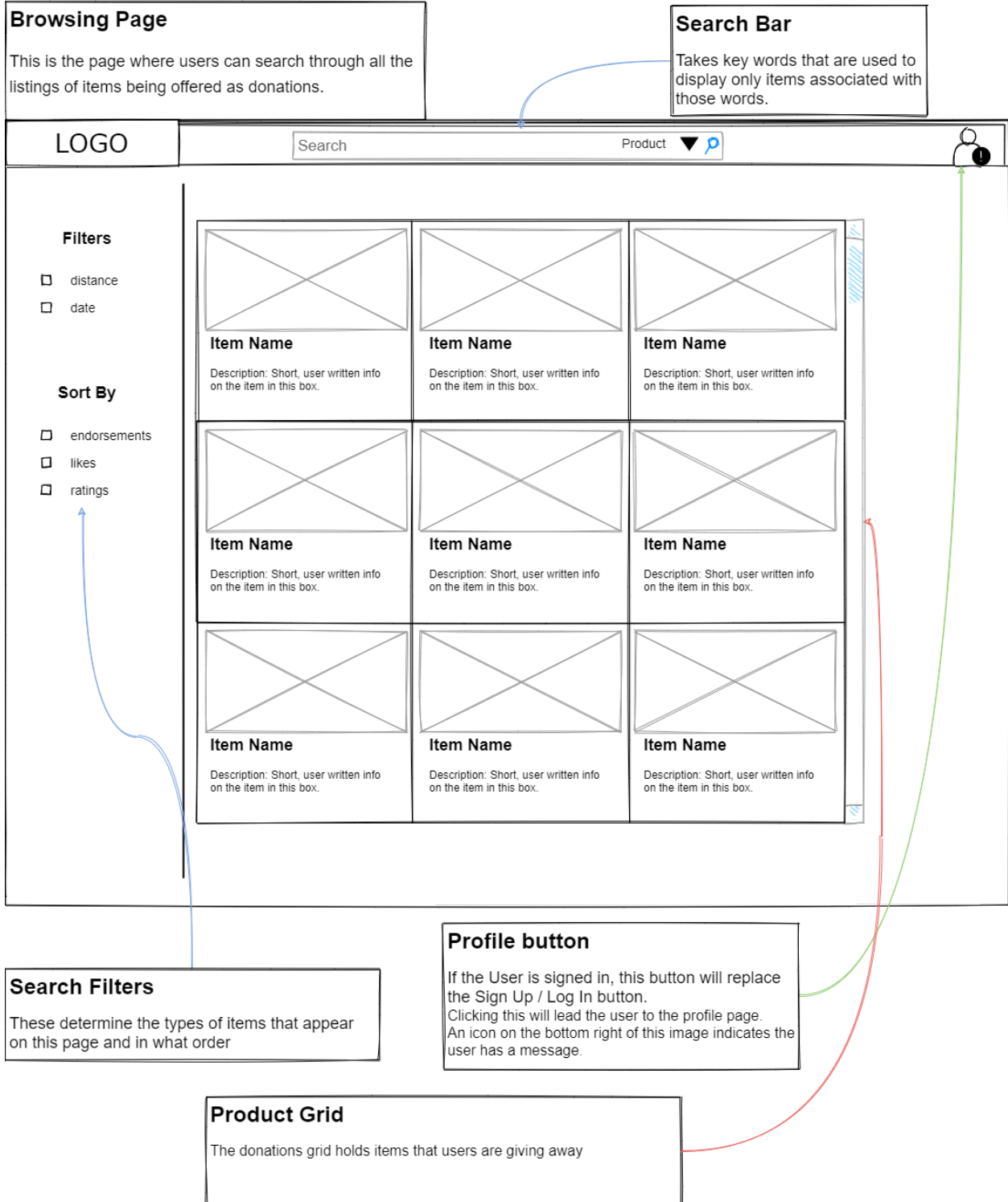
The top items are for users to receive an item, while the bottom grid is for users to give to others.

## Sign Up / Log In Form

This form pops up over the main page when the "Sign Up / Log In" button is clicked.

This is how a user can create a profile or sign into an existing one.



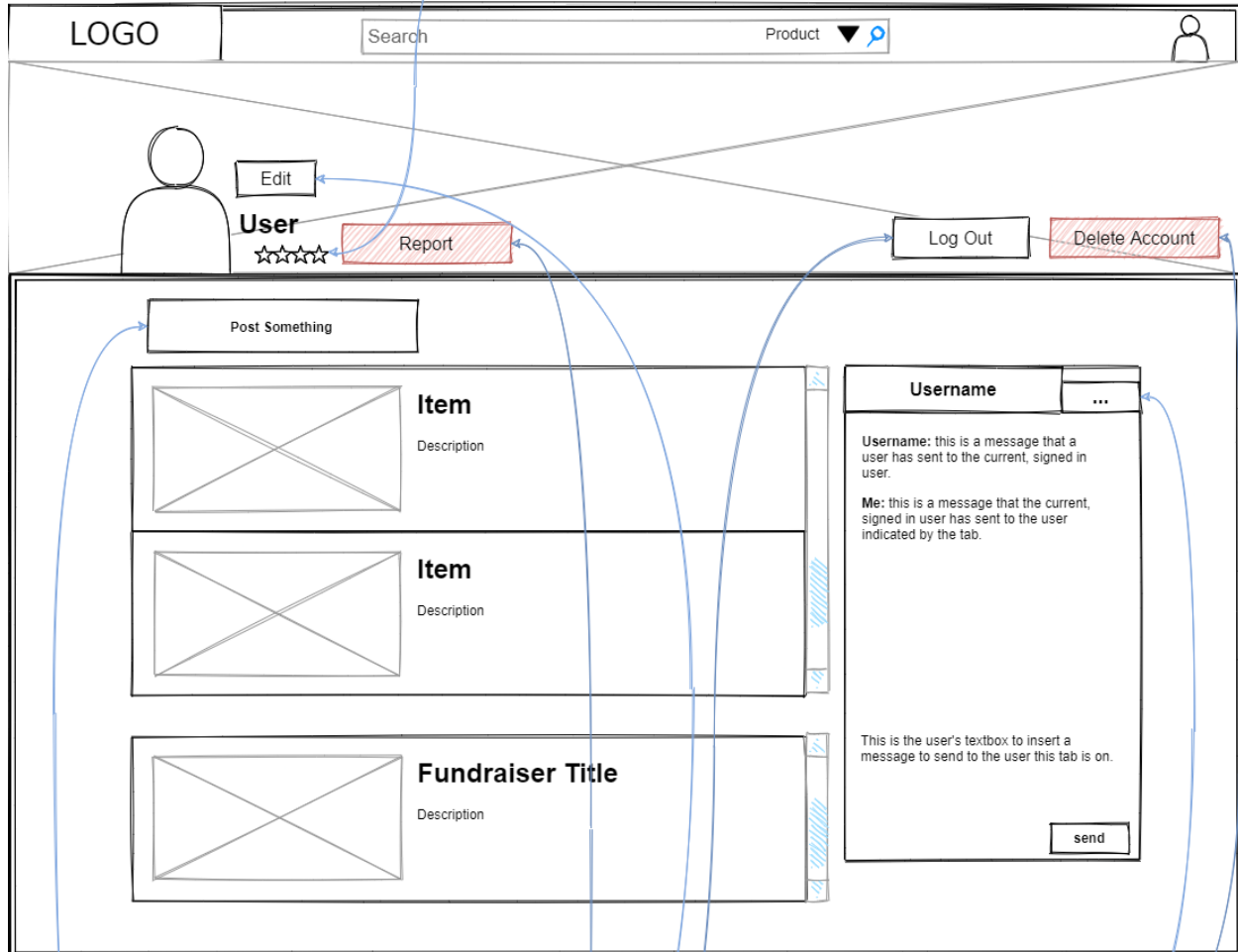


## Profile Page

This displays all the features a user can see about a profile.

## Ratings Button

Clicking on this will indicate an average rating given by other users. How many stars colored yellow starting from the left indicates the average rating a user has. A user can click on any of the stars and the number of stars from the left is the rating the user gives to the one on the current page.



This button will bring up the item page with the option to edit things so that others may be able to see it as one of the offered items.

Changes Username to text box to be changeable.

Only visible if the current user is the logged into the user for this page. Delete is shown to Admins as well

Logs out of the account

Deletes the account from the website.

Opens a pop up text bot that users fill and when complete is sent to admins as a report of the user on this page. Users that own this page can't see this button.

This is a button that once clicked will bring up a drop down menu if there are other conversations that the current user has open. The "X" next to a tab closes the conversation with that user.

In the case that a user has none of these entities, they will not appear on this page. For example, if a user has not posted any items up for donation, there will be no item grid. Or if this is not the page for the currently signed in user, the current user will not be able to see any of the messages.

## Item Page

This is the page that describes an item a user has up for donation.

## Search Bar

Takes key words that are used to display only items associated with those words.

LOGO

Search

Product ▼

edit

Item Name

# Likes

Username (the one offering the item)

Description: this is the details of the item written by the user offering this item up for donation.

Save

Delete Post

report

This is a text box a user can use to send a message to another user. In the case that nothing has been written here, this text box will be empty. Clicking the "Message" button below will start a conversation with the user that owns this post.

Message

## Editing Buttons

These buttons are only shown to the user that owns this posting and allow said user to edit the parts of this page they are next to. Delete is available to admins as well. If a user is not an owner or an admin, Delete Post will say report. In the case of the "Save" and "Delete Post" buttons, these update or delete the entire post respectively.

This button will take the current user back to their user page to see their conversation with the owner of the post.

Increments this post's like count by 1 if the current user hasn't endorsed the current post before. This button is not available to the owner of the post.

## Fundraiser Page

This is the page that describes an fundraiser a user has up.

## Search Bar

Takes key words that are used to display only items associated with those words.

LOGO

Search

Product

edit

Fundraiser Name

Username (the one offering the item)

Description: this is the details of the item written by the user offering this item up for donation.

Save

Delete Post

##### reached out of #####

Send a message, too (optional)

Text

Donate!

Endorse

edit

## Editing Buttons

These buttons are only shown to the user that owns this posting and allow said user to edit the parts of this page they are next to. Delete is also available to admins  
In the case of the "Save" and "Delete Post" buttons, these update or delete the entire post respectively.

Increments this post's endorsement count by 1 if the current user hasn't endorsed the current post before. This button is not available to the owner of the post.

## Monetary Donations

Will bring up a payment api

## 4. High level database architecture and organization

The business rules that are selected for initial database design are as:-

- 1) Users shall be able to place items up for donation.
- 2) Users shall be able to set up descriptions of items for donation.
- 3) Users shall be able to donate directly to the product.
- 4) Users can show interest in any of the products mentioned on the website by hitting the like button for whichever product they like.
- 5) Users shall be able to endorse fundraisers.
- 6) Users shall be able to keep a wish list of items
- 7) Users shall be able to add items in their wish list.
- 8) Administrators shall be able to bring back a fundraiser if it has been removed.
- 9) Administrators shall be able to bring back a post if it has been removed.

Hence, the entities can be as:-

- 1) User
- 2) Admin
- 3) Post
- 4) Product
- 5) Interest
- 6) Wishlist

(Note: pk- primary key, uk- unique key, fk- foreign key.)

### **User**

The entity user can have attributes such as user\_id (pk), username (uk), password, phone\_number, zip code, full\_name, age.

### **Admin**

The entity user can have attributes such as admin\_id(pk), username (uk), password.

### **Post**

The entity post can have attributes such as post\_id (pk), created\_by(fk), created\_on, interest\_count, product\_id(fk).



## Product

The entity product can have attributes as product\_id(pk), product\_name, description, images, offered\_by(fk), requested\_by(fk), donated\_to(fk), created\_on, closed on.

## Interest

The entity interest can have attributes as interest\_id(pk), post\_id(fk), user\_id(fk).

## Wishlist

The entity wishlist can have attributes such as wish\_id(pk), user\_id(fk), product\_id(fk).

The relationship between these entities will further be shown in the entity relationship diagram below.

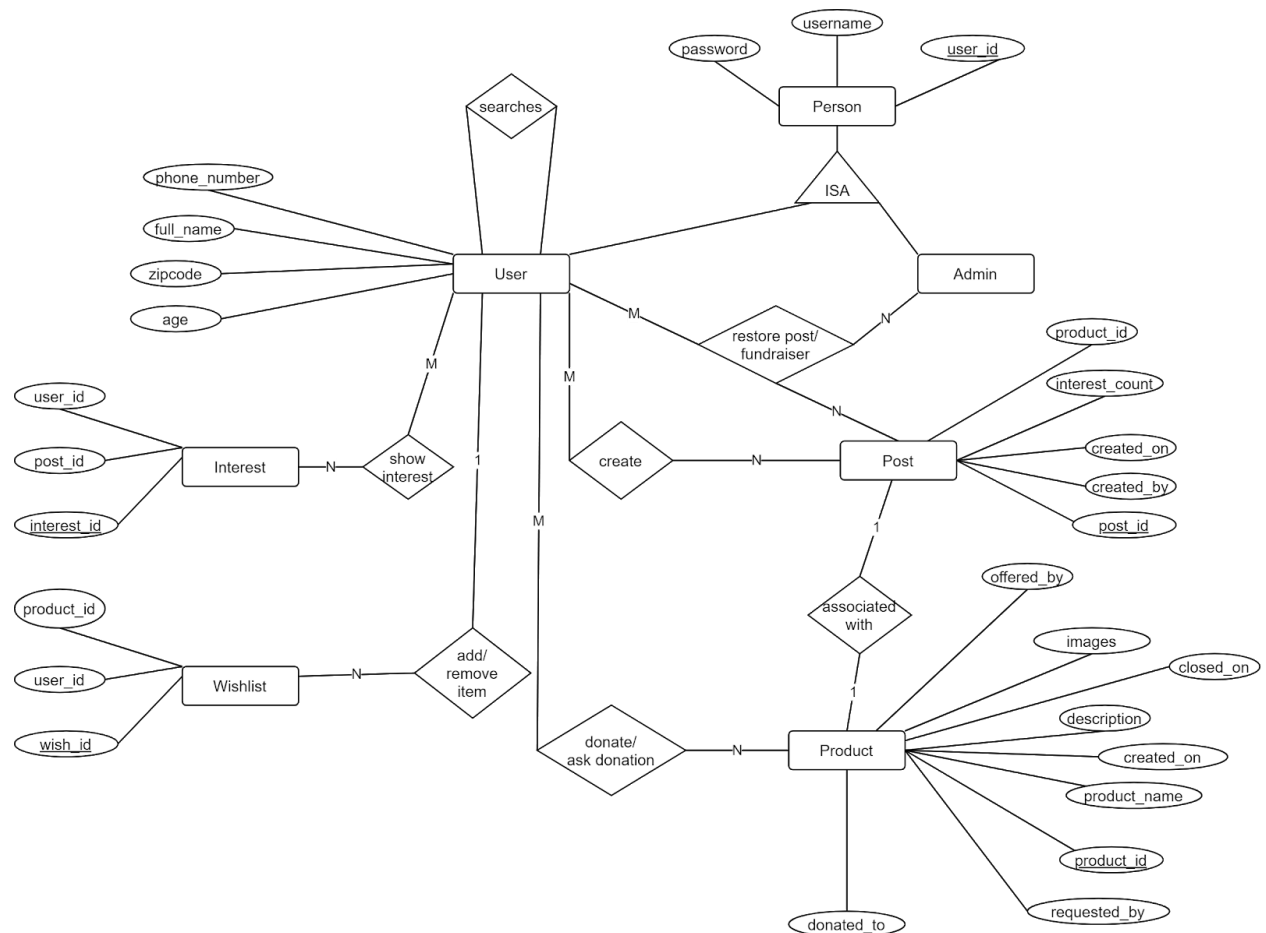


Fig. entity relationship diagram

Here, the entity user had many to many relationships with multiple relationships. Multiple users can create multiple posts, hence users have many to many relationships with entity posts. Similarly, multiple users can show interest in multiple products/posts, hence entity users have many to many relationships with entity interest. Similarly, a user can have multiple items in his/her wish list hence, entity user has one to many relationship with entity wishlist. Multiple users can request admins to restore multiple post/fundraisers hence, entity users have many to many relationships with entity admin. A post can only have only one product associated with it, hence, the entity post has a 1 to 1 relationship with the entity product.

The database model for the project is shown below. The database model is designed using the MySQL Workbench tool.

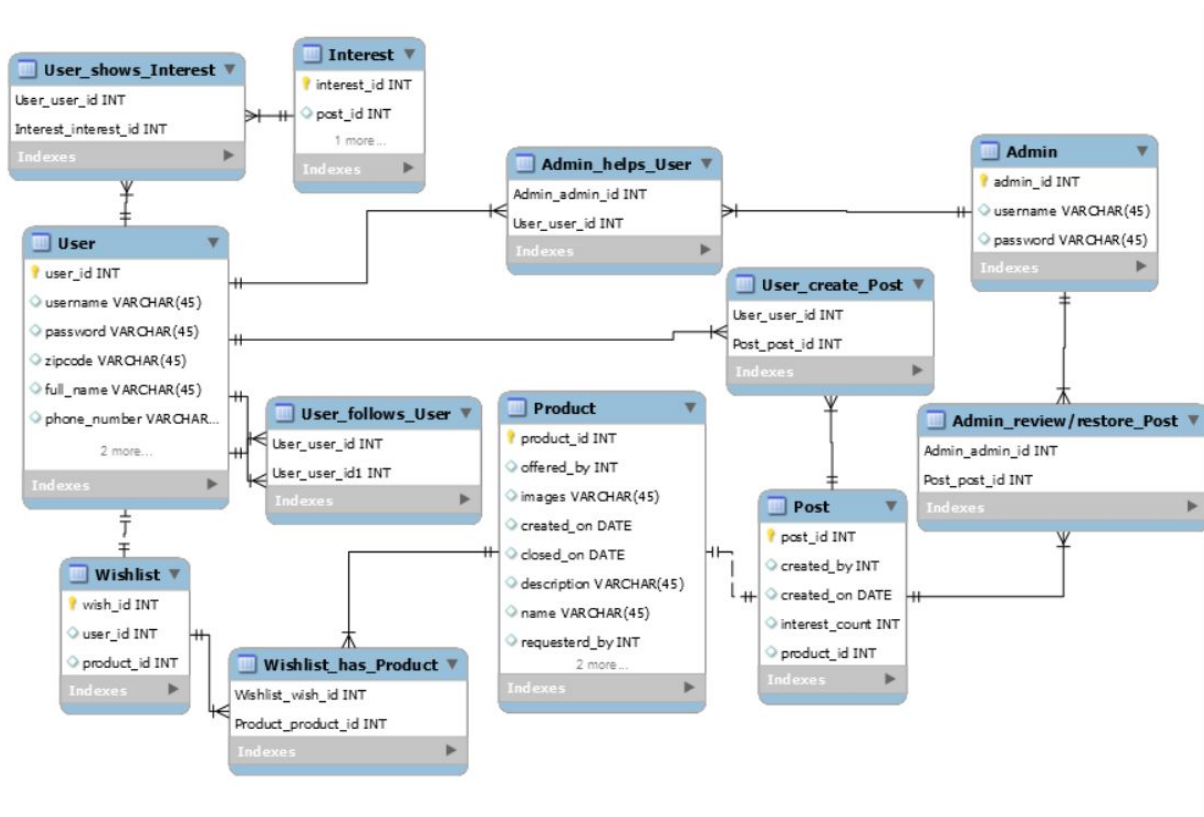


Fig: database model

MySQL will be used for the purpose of the database in this project because it supports relational database models and also MySQL is relatively easier to understand and use, free to use and open source.

## **Media Storage:**

For storing the various multimedia files, the database BLOB will be used. Since, in the project we are only storing the images, which can be of less size, there is no need of using a file directory for storing the images. Images can be easily stored in databases using BLOB hence, database BLOB will be used for storing multimedia files.

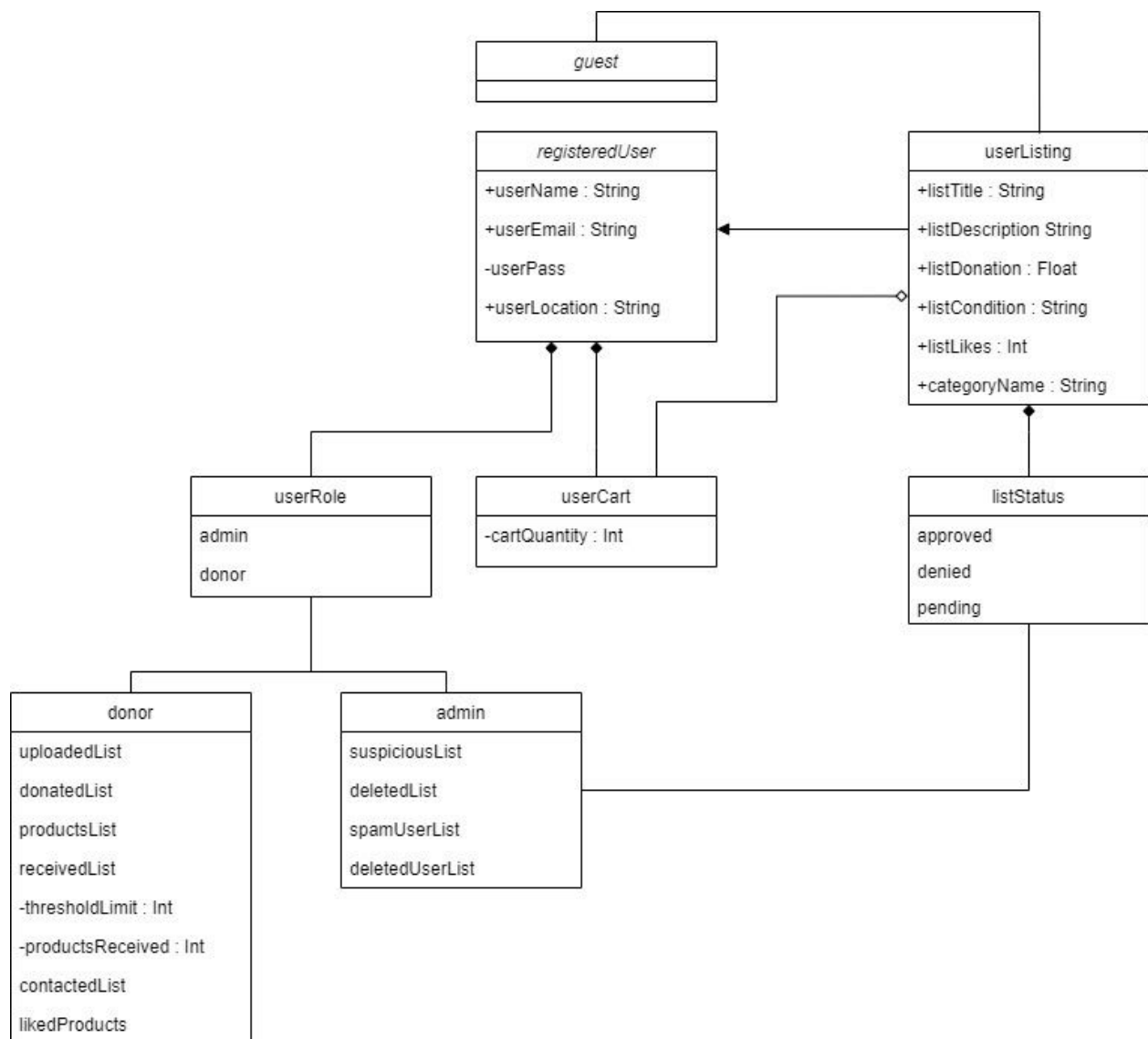
## **Search/filter architecture and implementation:**

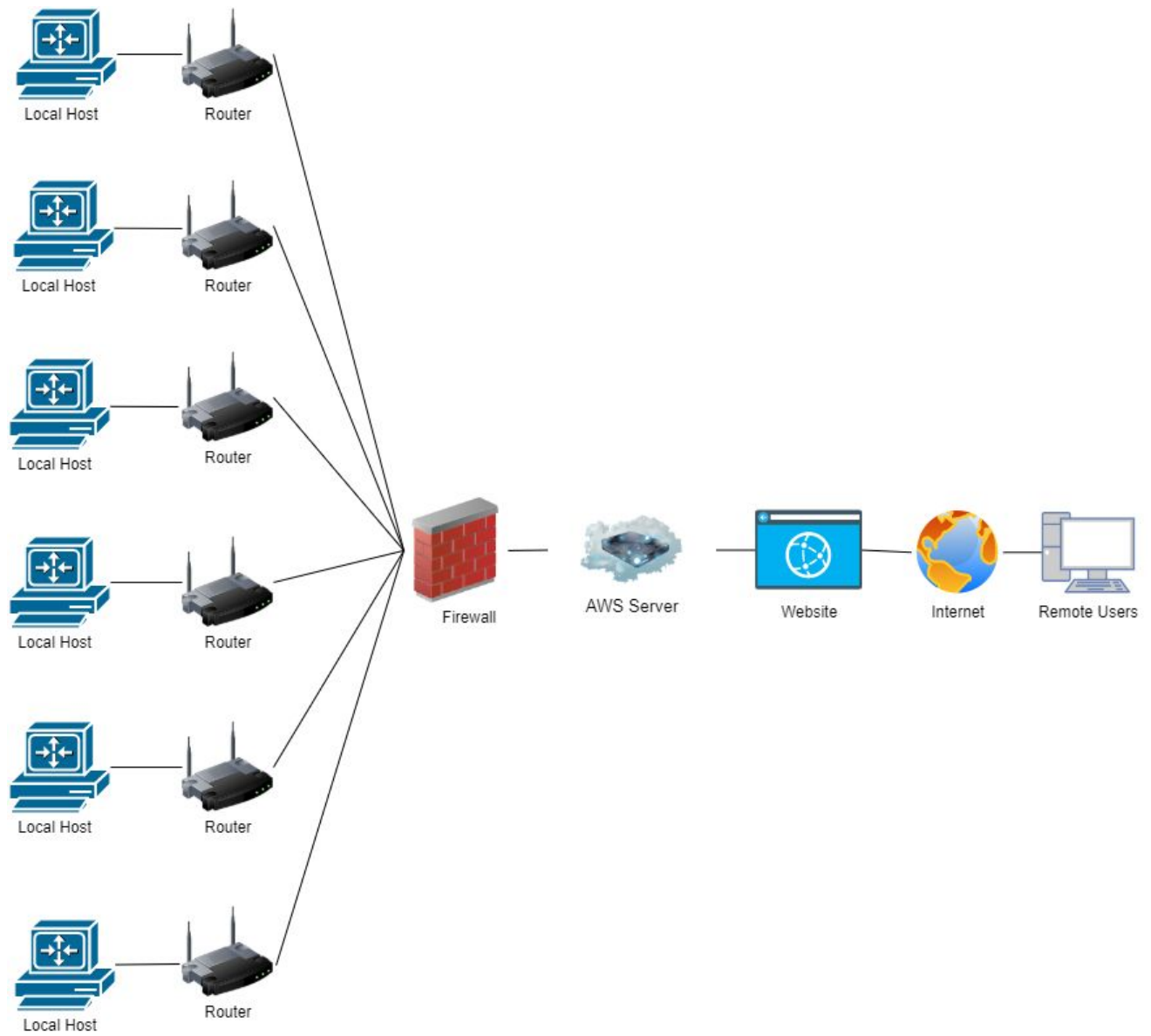
As we can see from the functional requirements, the user shall be able to search for a couple of items that are stored in the database.

- ☐ Users shall be able to search items by date  
The user will be able to search the items on the basis of date hence, the database will be searched on the basis of created\_on and closed\_on attributes of the product and post entity.
- ☐ Users shall be able to search items by name  
The user will be able to search the items on the basis of the product name, hence, the product\_name attribute of the entity product will be used on this search.
- ☐ Users shall be able to search up other users  
The user will be able to search other users on the basis of their full name or the username of the other user. Hence, the attributes username and full\_name will be used for this search.

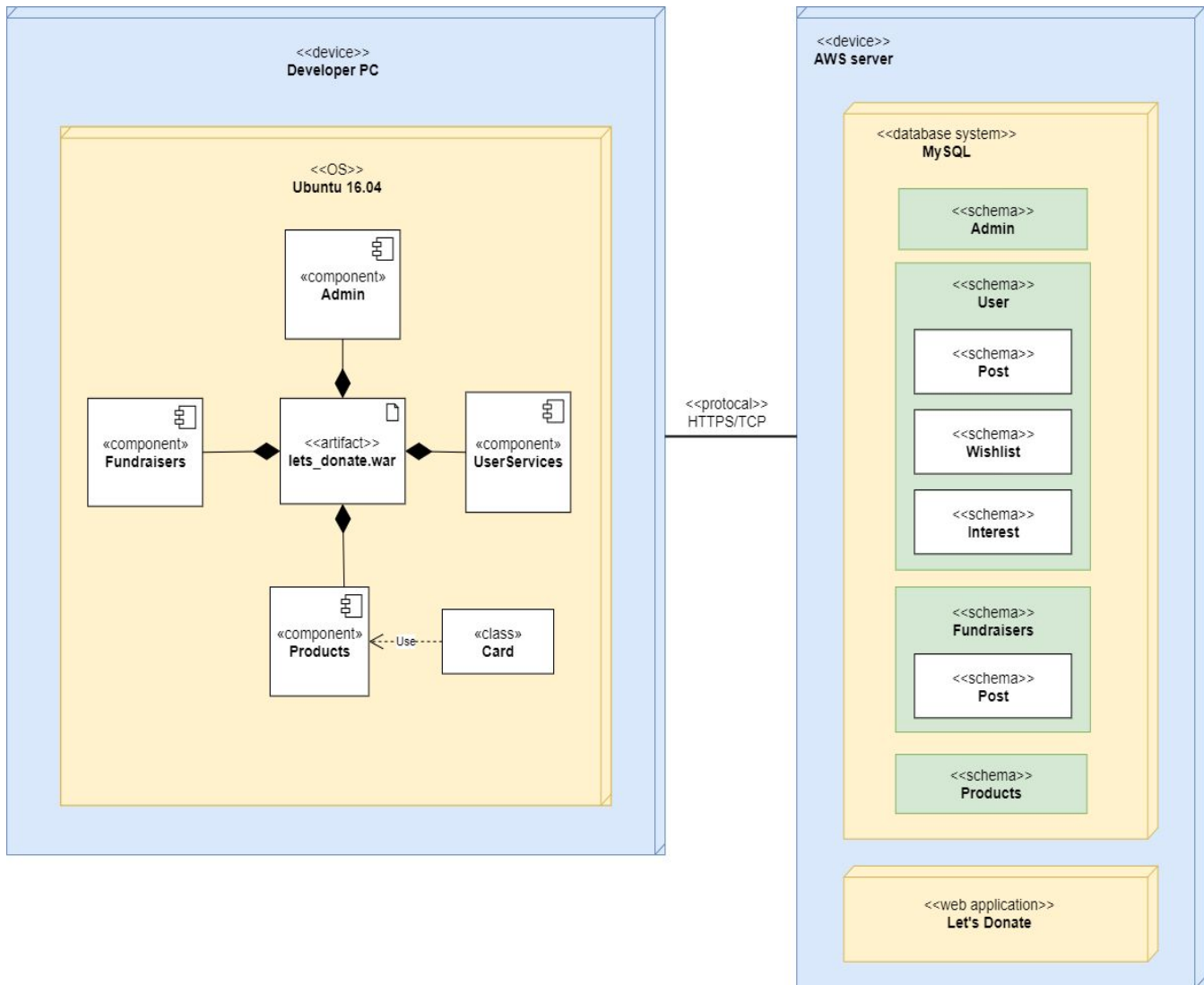
No special algorithm will be used for the searching purposes. Since, only simple %like query will be able to give out the desired output, there seems to be no need for an advanced algorithm for searching purposes.

## 5. High level diagrams





Application Networks Diagram



Deployment Diagram

## 6. Detailed List of Contributions

### Document

**Data Definitions:** Himanshu Garg, Milo Abril, Marlon Bustamante.

**Functional Requirements:** Dipendra Dhoj Rana (Team Discussions Involved.)

**UI Mockups:** Milo Abril (Team discussions involved.)

**High Level Database Organization and Architecture:** Suman Basaula

**High Level UML Diagrams:** Marlon Bustamante.

**High Level Application Network and Deployment Diagrams:** Kunyu Ruan

### Vertical and Horizontal Prototype

**Frontend:** Milo Abril, Himanshu Garg, Marlon Bustamante.

**Backend:** Marlon Bustamante, Dipendra Dhoj Rana.

**Database Management:** Himanshu Garg.

**Deployment:** Himanshu Garg.

### Team Overview

Suman, Dipendra → 2 Team meetings missed.

Kunyu → One Team meeting missed.

There have been several team meetings with great discussion and ideas being shared.

We consistently discussed different parts of the document and helped each other with their sections and collectively took decisions for some sections of the document.