

SW Engineering CSC648/848 FALL 2020

Let's Donate

Team 3

Milestone 2

October 29, 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
October 1, 2020	Milestone 1 v1
October 8, 2020	Milestone 1 v2
October 29, 2020	Milestone 2 v1
November 12, 2020	Milestone 2 v2

Table of Contents

1. Data Definitions V2	3
2. Functional Requirements V2	6
3. UI Mockups (high level only)	9
3 (cont). Usage Storyboard (high level only)	16
4. High Level Database Organization and Architecture	21
5. High Level API and Main Algorithms	25
6. High Level UML Diagram	27
7. High Level Application Network and Deployment Diagrams	28
8. Identifying Actual Key Risks for the Project	30
9. Project Management	32
10. Detailed List of Contributions	33

1. Data Entities and Definitions V2

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 V2

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 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 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 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 able to add items in their wish list
16. Users shall 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 (high level only)

The image shows four hand-drawn UI mockups on a piece of paper:

- Login Form:** Contains four input fields labeled 'email', 'username', 'name', and 'password'. Below these fields are three buttons: 'Sign Up', 'Sign In', and 'recover password'.
- Report Form:** Contains a large 'Text Form' area and a 'Send Report' button at the bottom right.
- Browsing Drop Down:** A vertical list with two options: 'Products' and 'Fundraisers'.
- User Edit Form:** Contains a 'Banner' section with a crossed-out image placeholder, a 'file.png choose' button, a 'profile-pic' section with a person icon and a 'file.png choose' button, a 'name' input field, and 'save' and 'cancel' buttons at the bottom right.

Forms

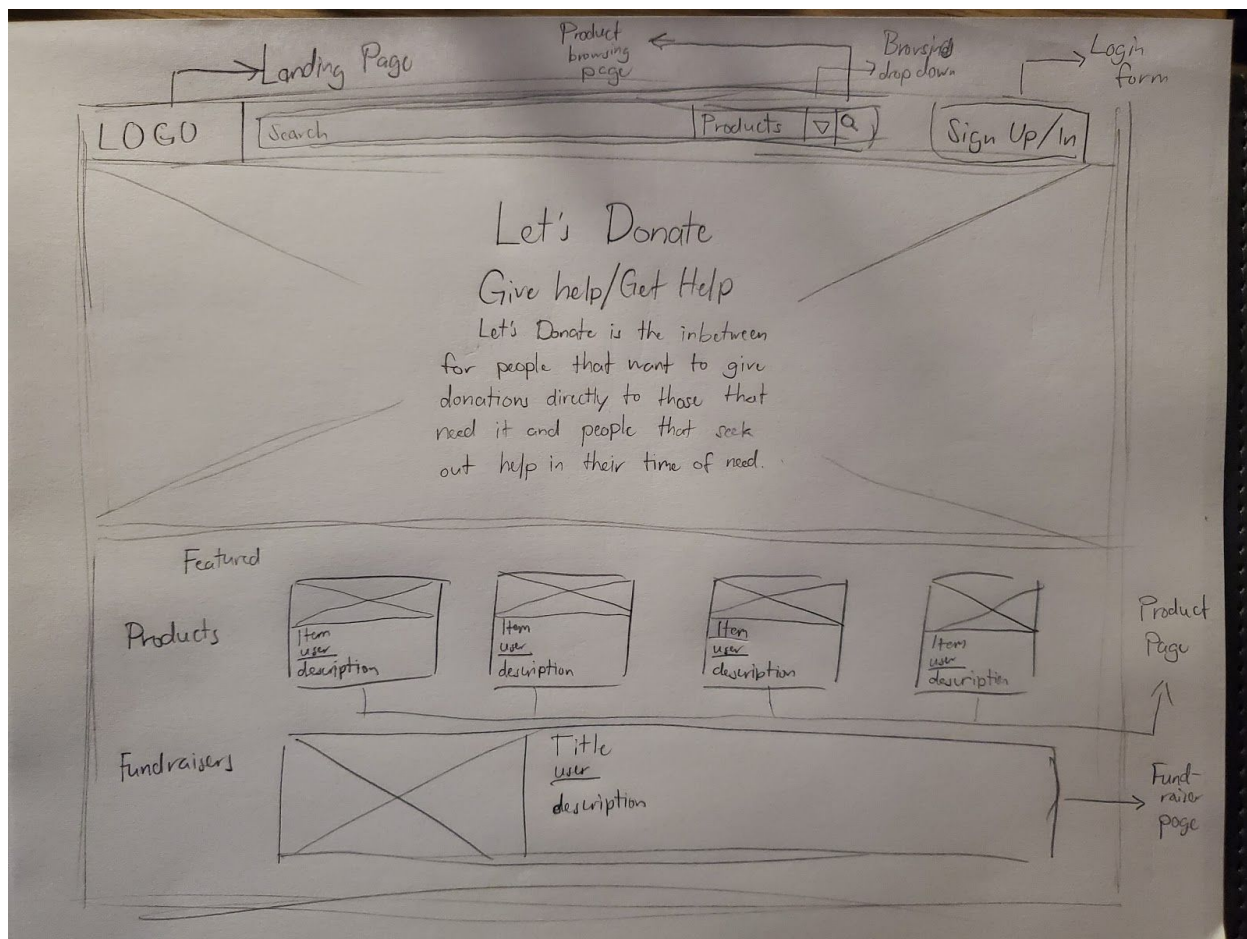
These are the pop ups for forms.

The login form acts as both a sign in and a sign up and will appear when users either click on the button for it or when they try to do something only a logged in account can do (i.e. send a message to another user).

The Browsing drop down determines what type of search or post a user will use.

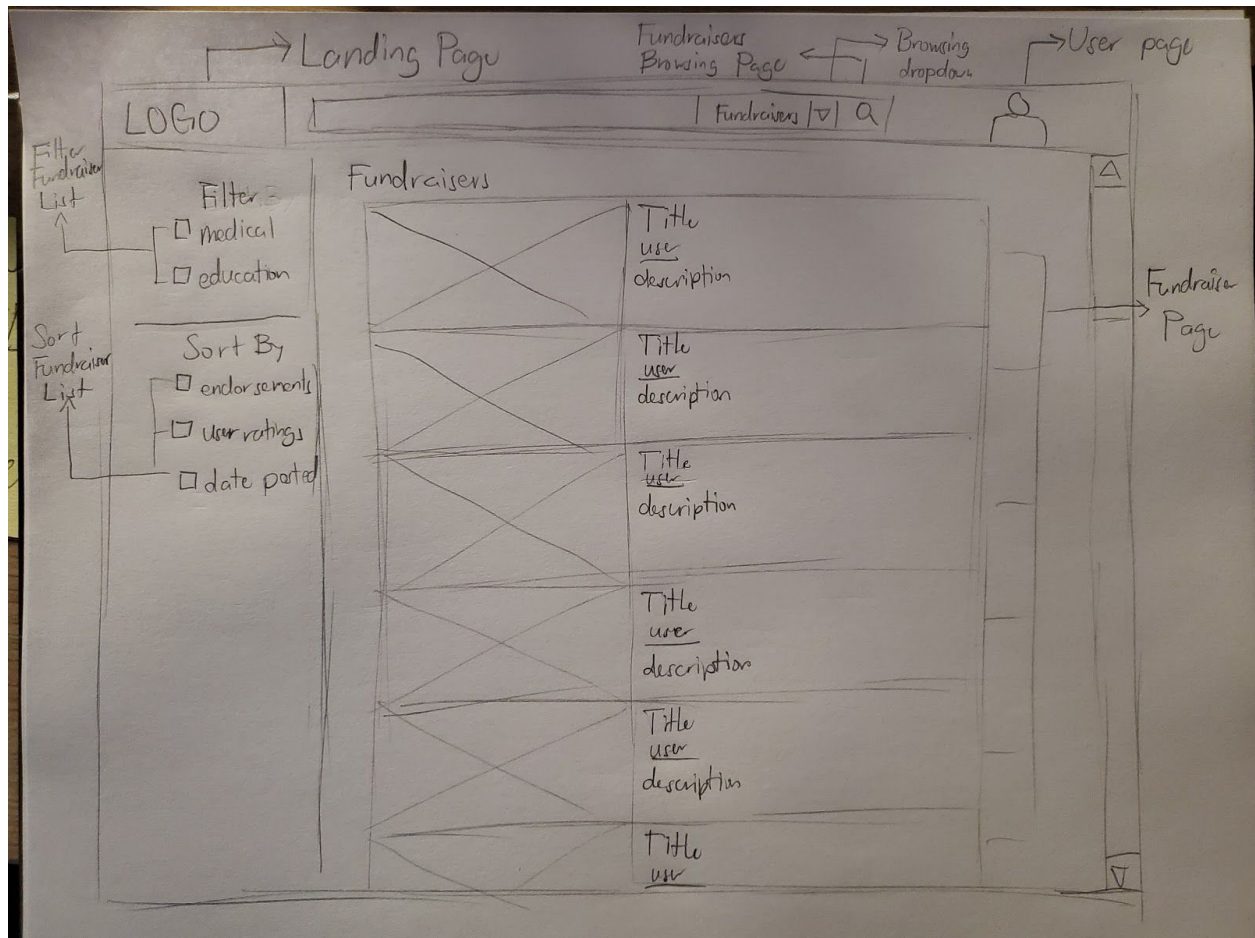
The report form would be a message to an admin about a user or post the form was made on.

User edit form allows users to edit the banner, profile picture, or name of their account.



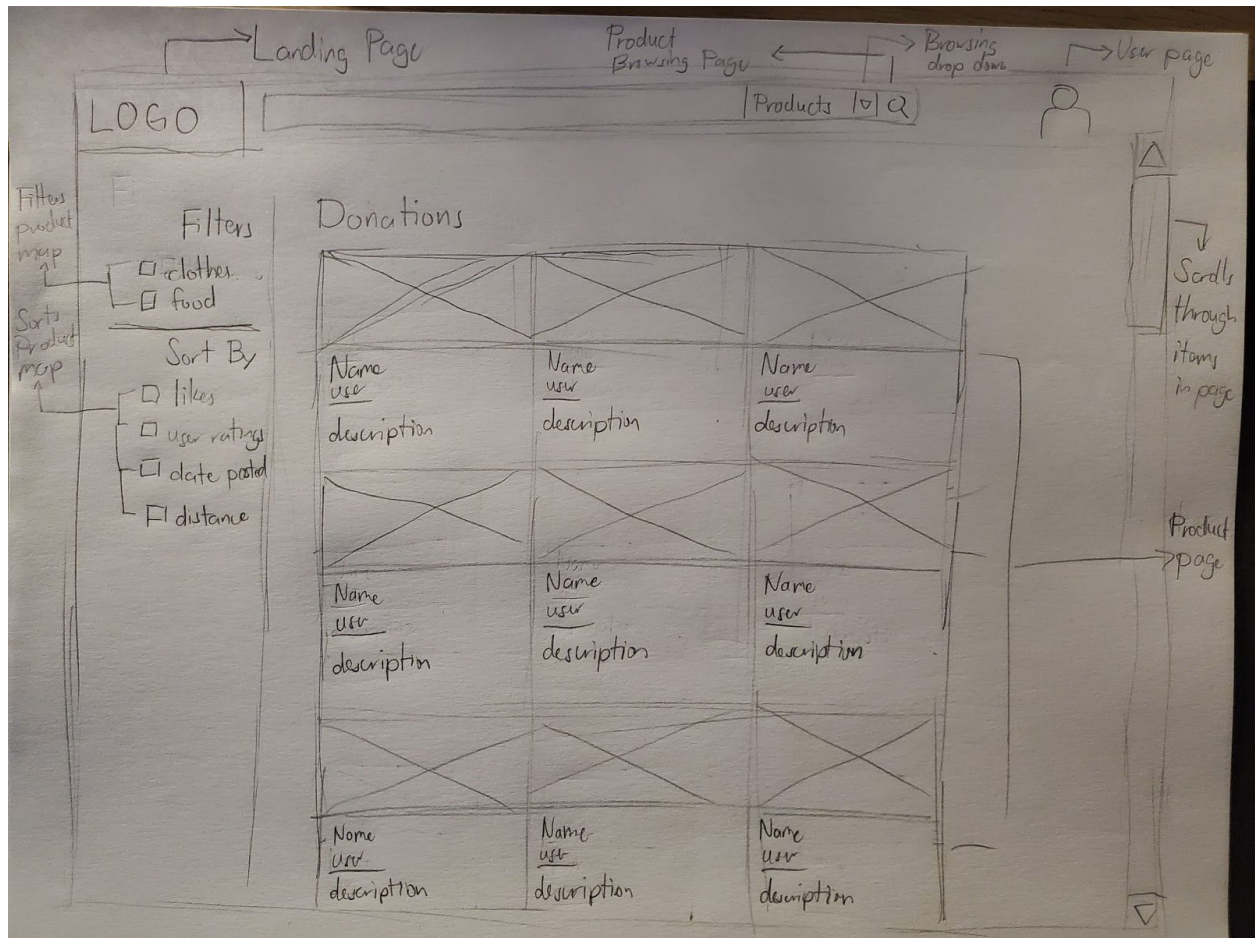
Landing Page

This is the page that all users see first when coming to the website.



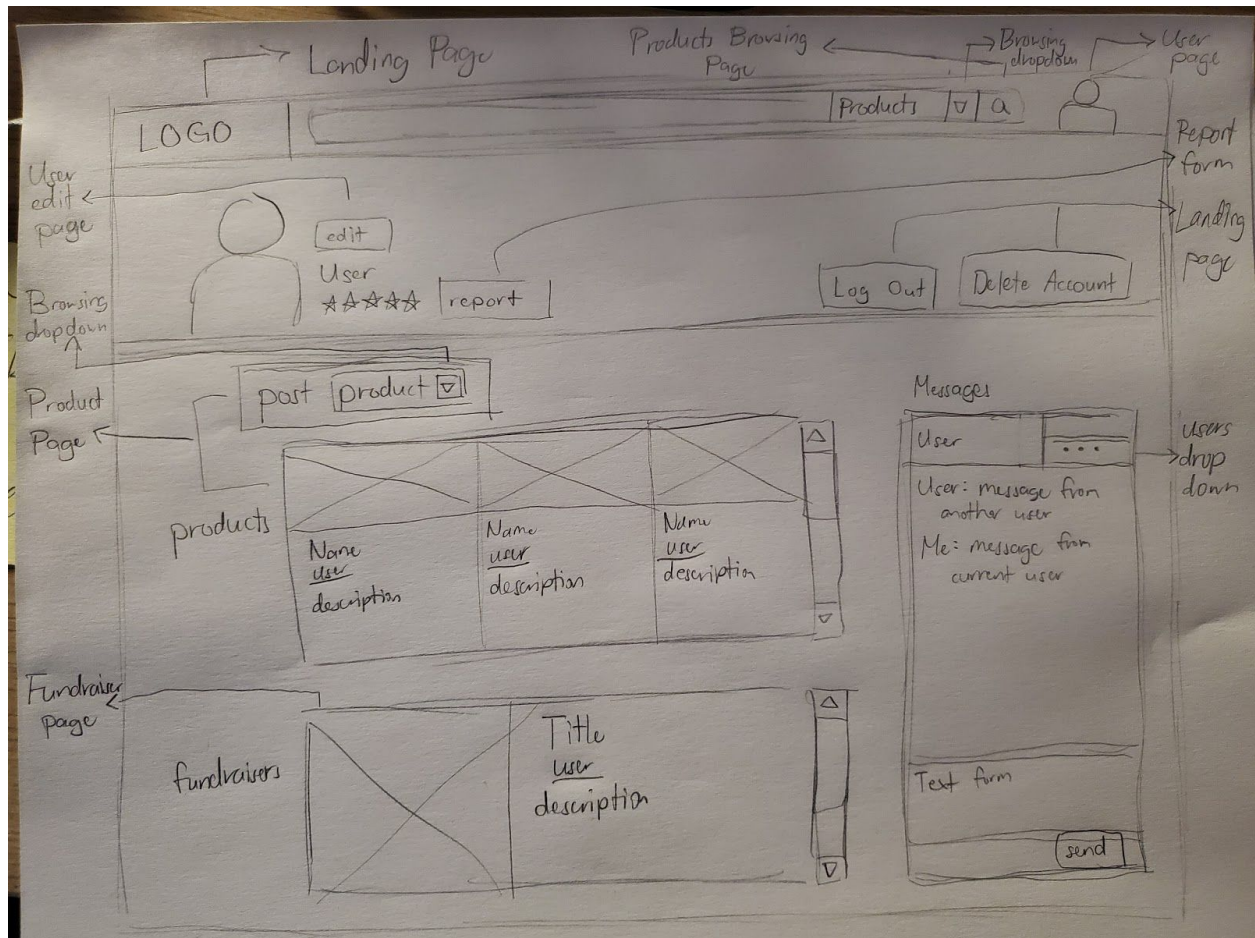
Fundraisers Browsing Page

This page is a list of all the fundraisers users have put up on the website. The list can be sorted and/or filtered by the categories on the left of the page.



Product Browsing Page

This page shows a map of all the products users have posted for donation and the map can be sorted and filtered through the categories on the left of the page.



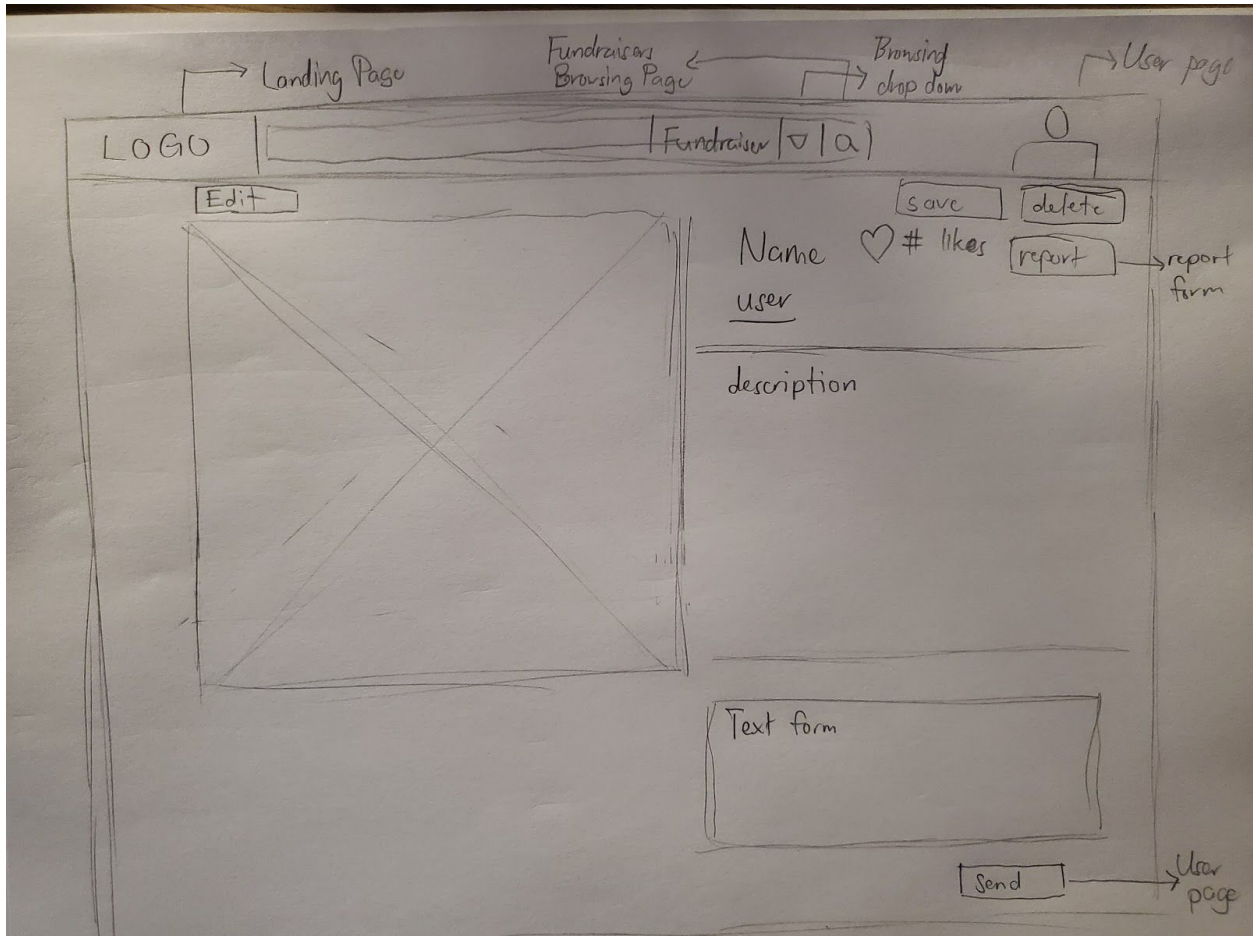
User Page

This page is the dashboard for a user.

If the current user is not signed into the user this page is referencing, they will not be able to see the edit, log out, delete account, or post button along with the messages box, but will have the option to report them or rate them through the stars next to the profile image.

If the current user of the website is signed into the account this page is for, they will see the edit, log out, delete and post buttons along with the messages on the bottom right of the page.

If the current user is an admin, they have the ability to delete their account.



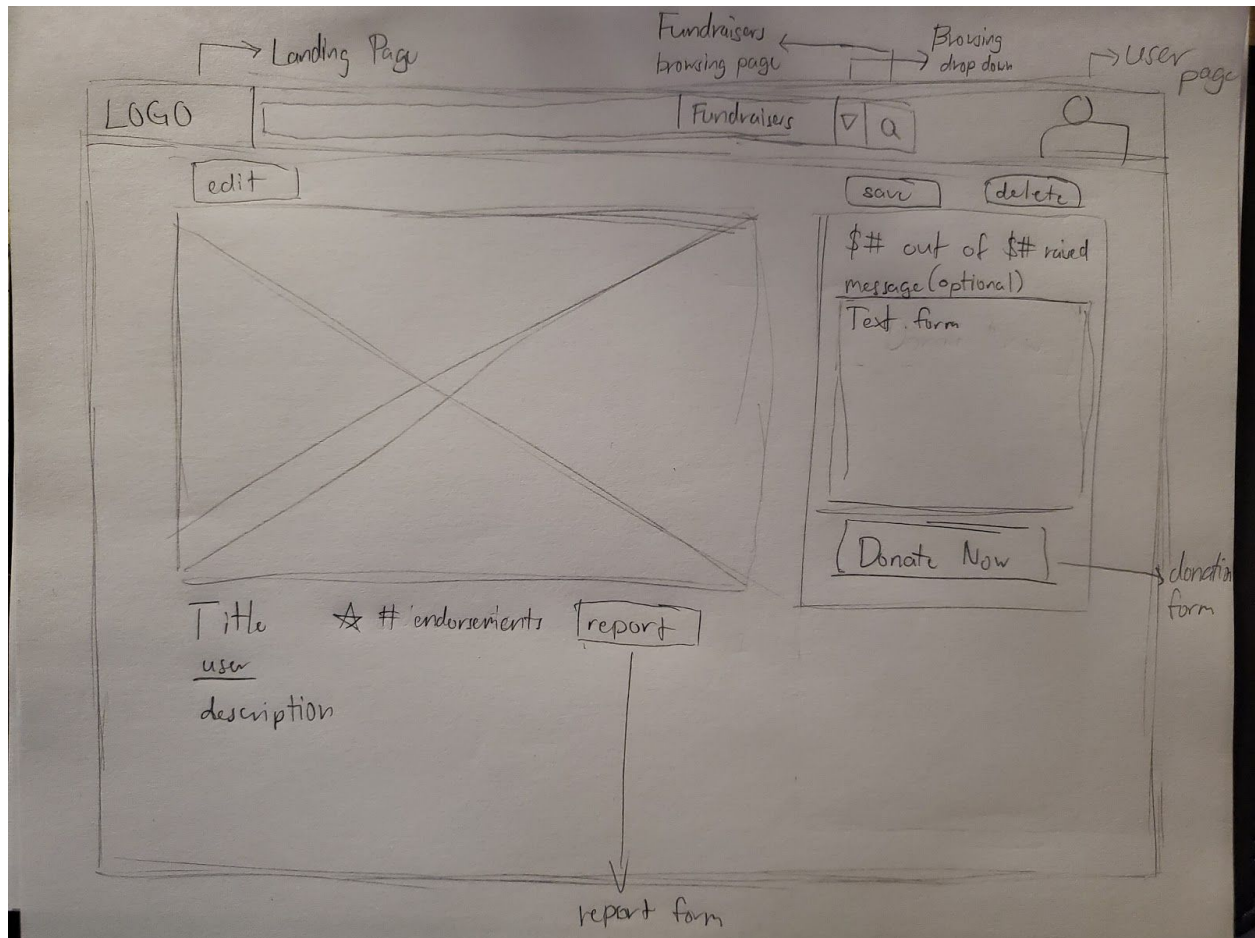
Item Page

This is the page users see for a specific product.

Users that don't own this page can see the likes (the heart) button, report button, text form and send button.

Users that own this page can see the edit, save, and delete button.

Admins can see the delete button.



Fundraiser Page

This is the page users see for a specific fundraiser post.

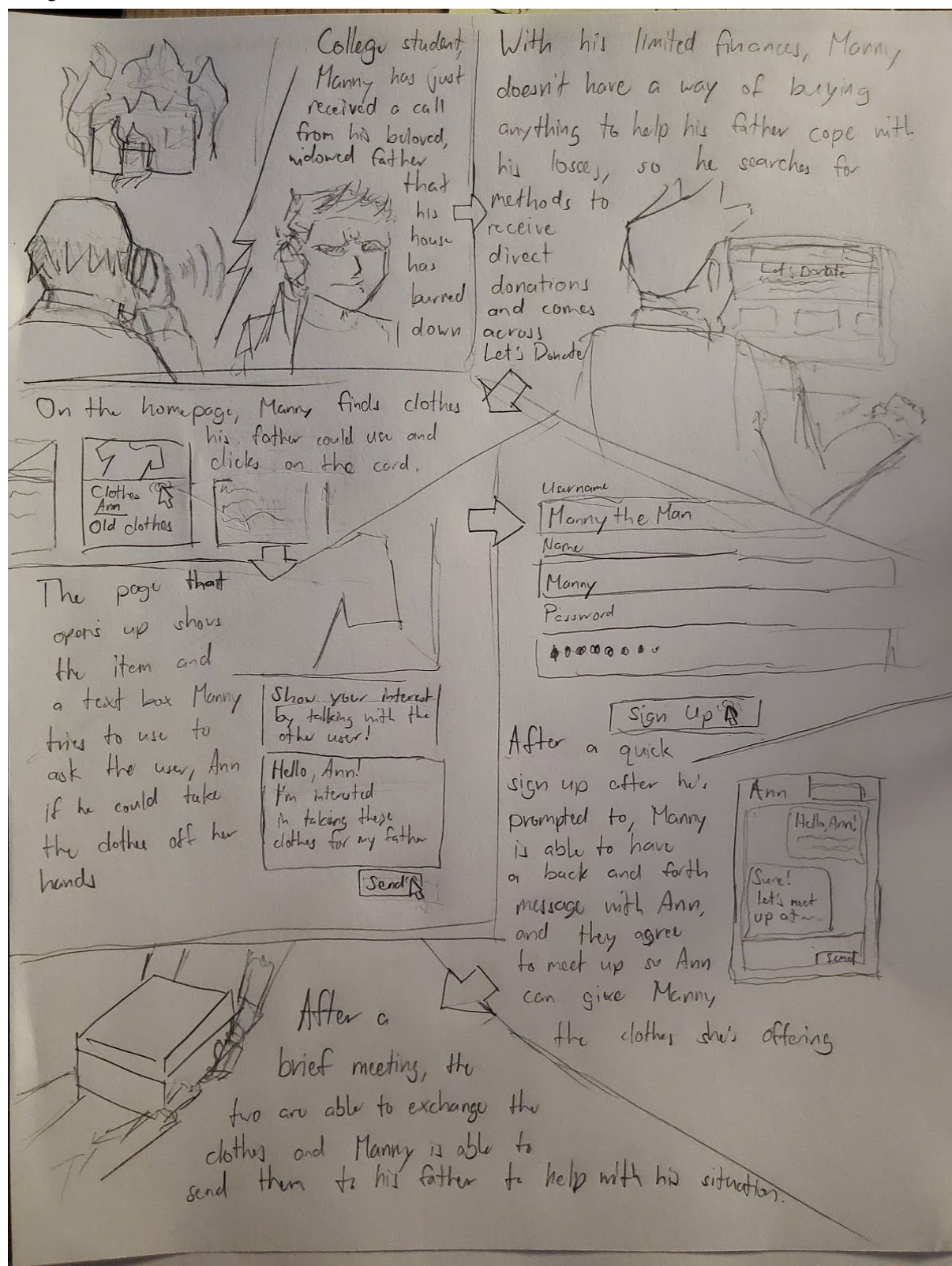
Users that don't own this page can see the endorsement (the star) button, report button, text form and donate now button.

Users that own this page can see the edit, save, and delete button.

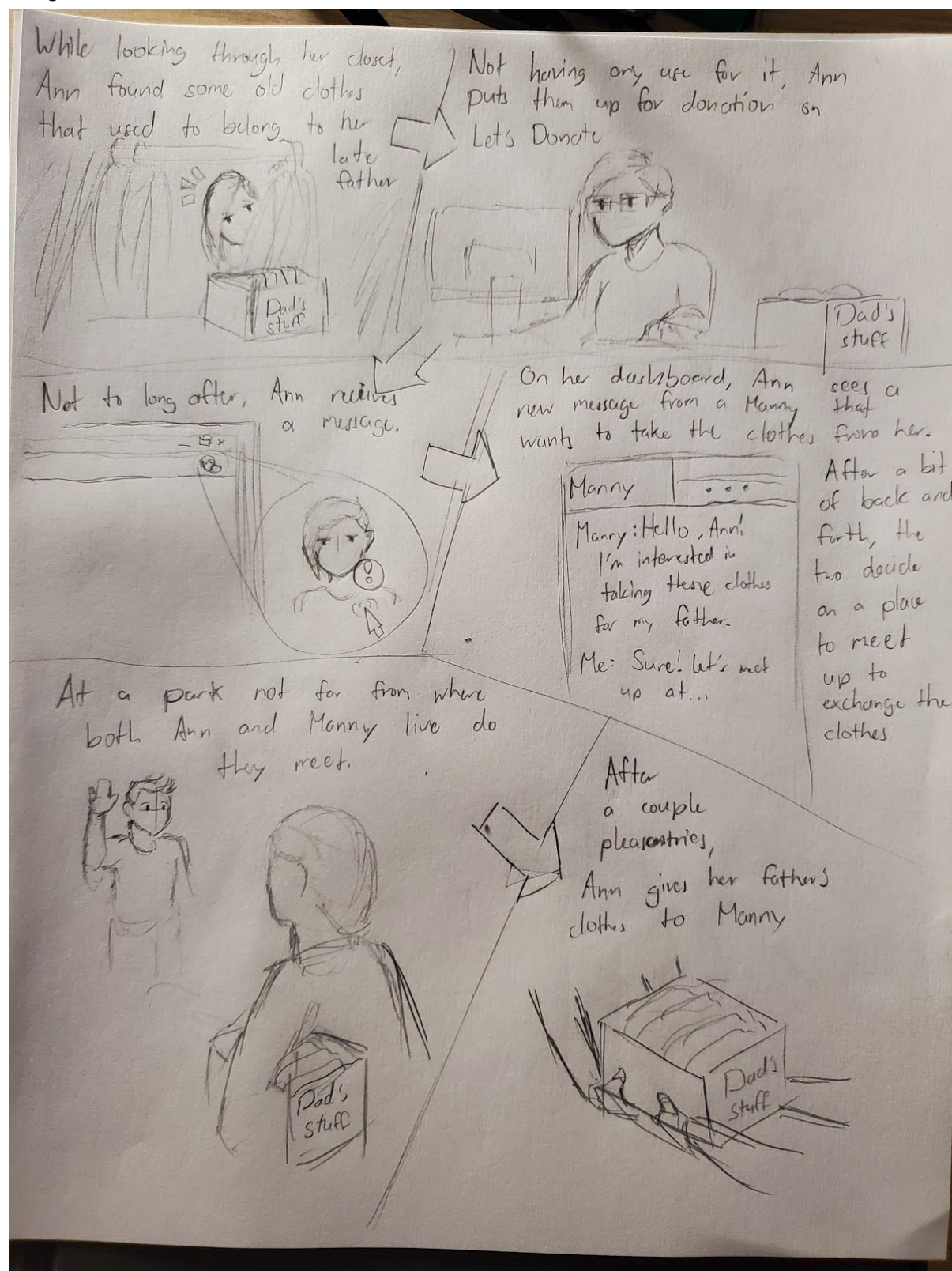
Admins can see the delete button.

3 (cont). Usage Storyboard (high level only)

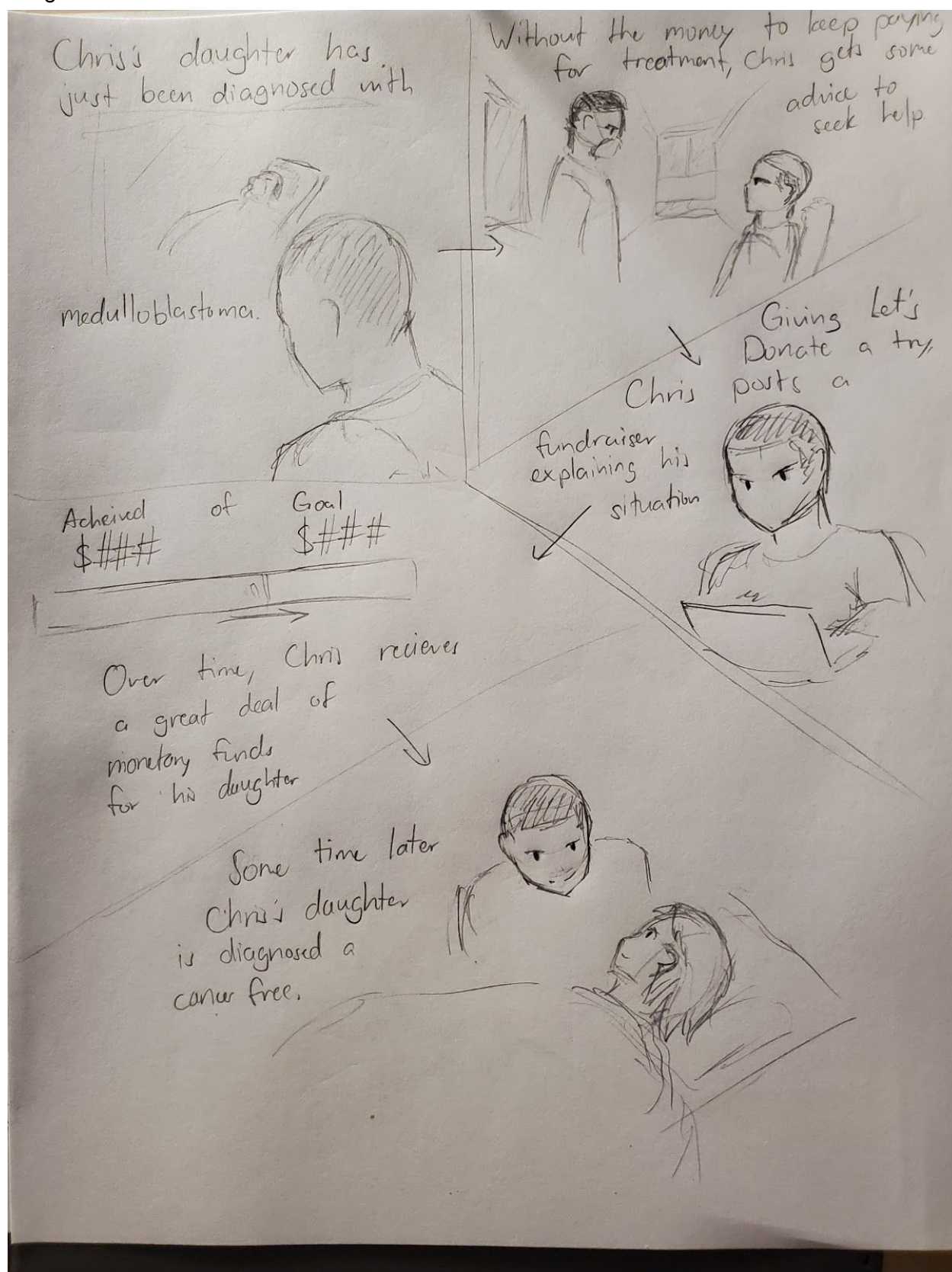
Usage case 1



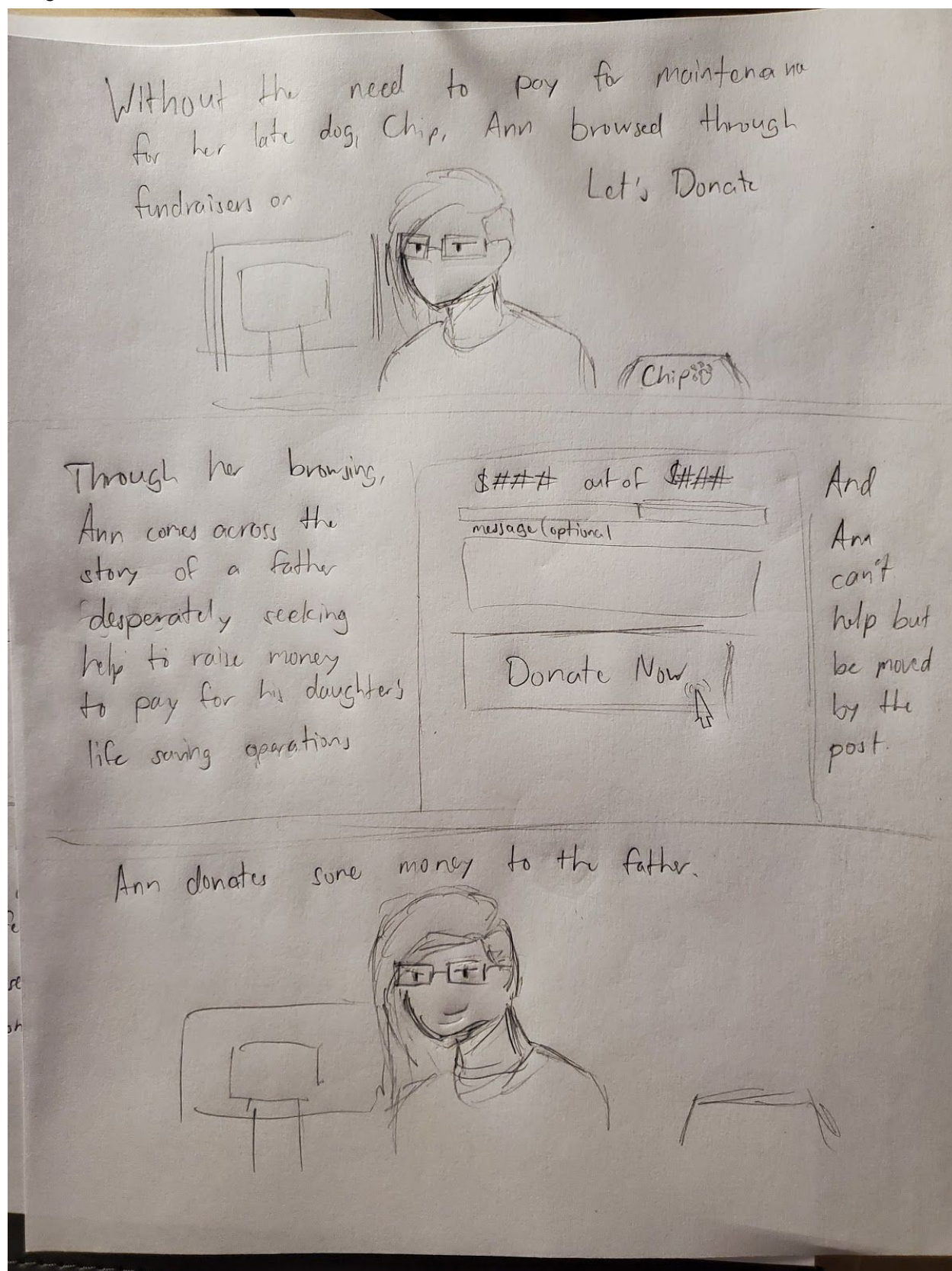
Usage case 2



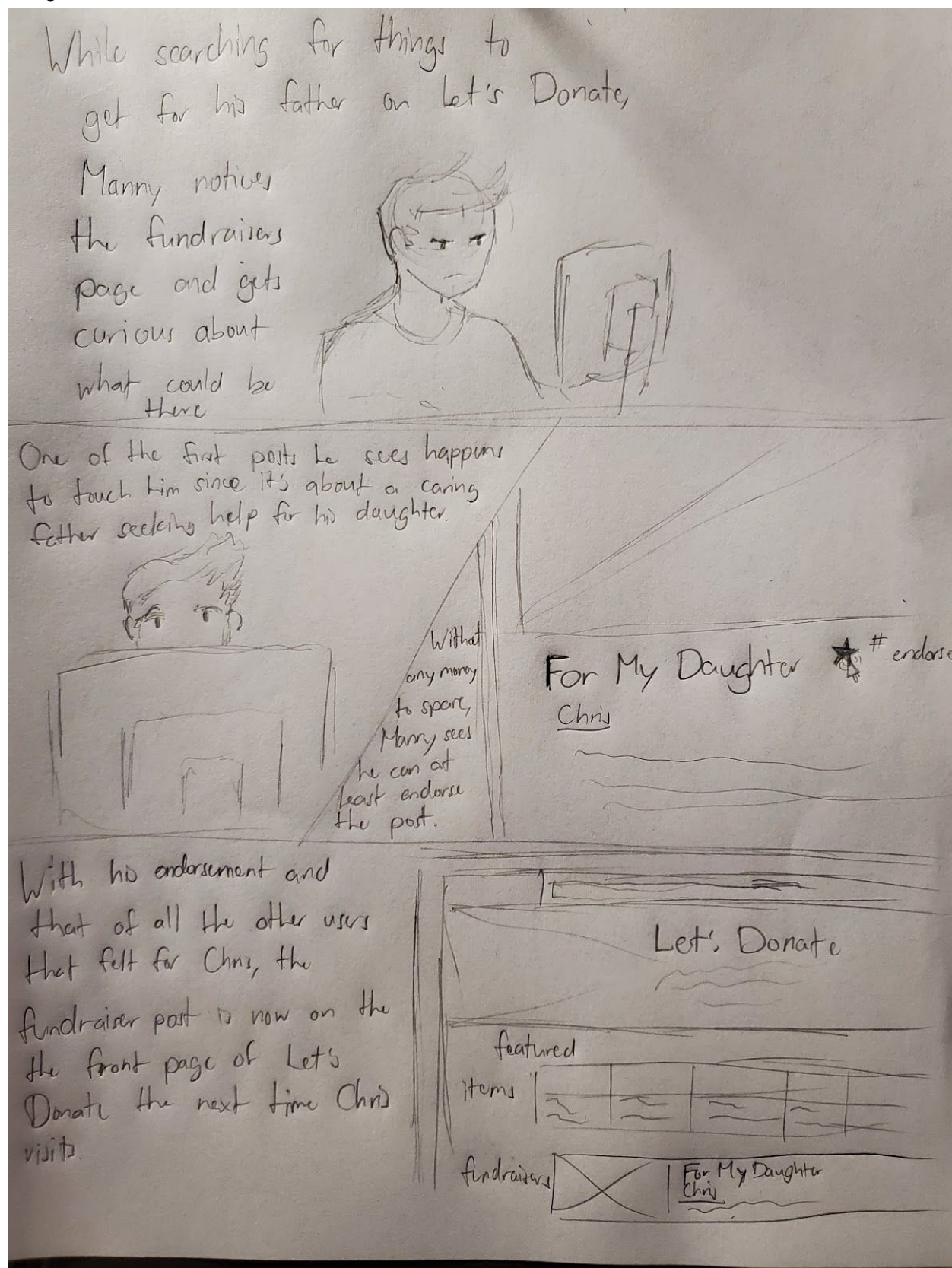
Usage Case 3



Usage Case 4



Usage case 5



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.

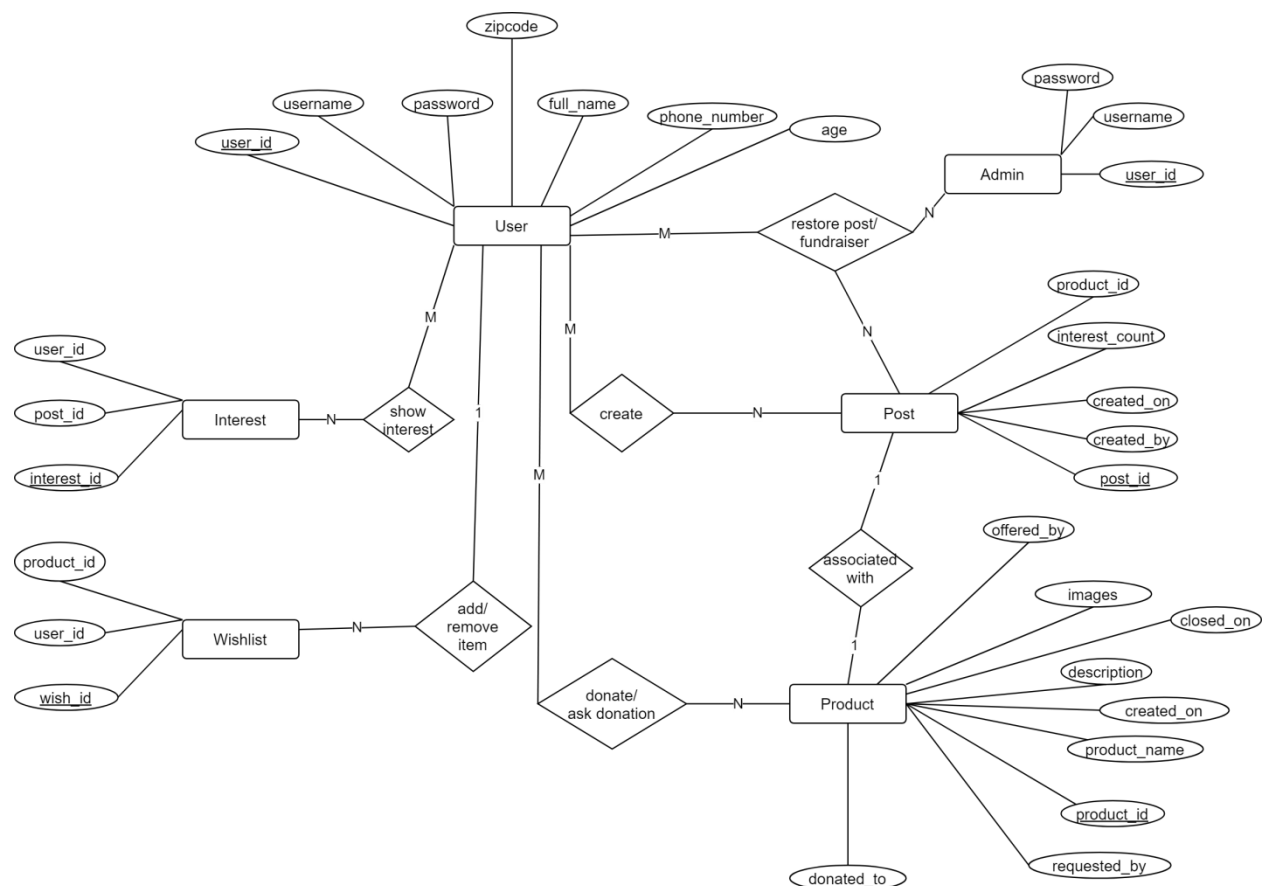


Fig: entity relationship diagram

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.

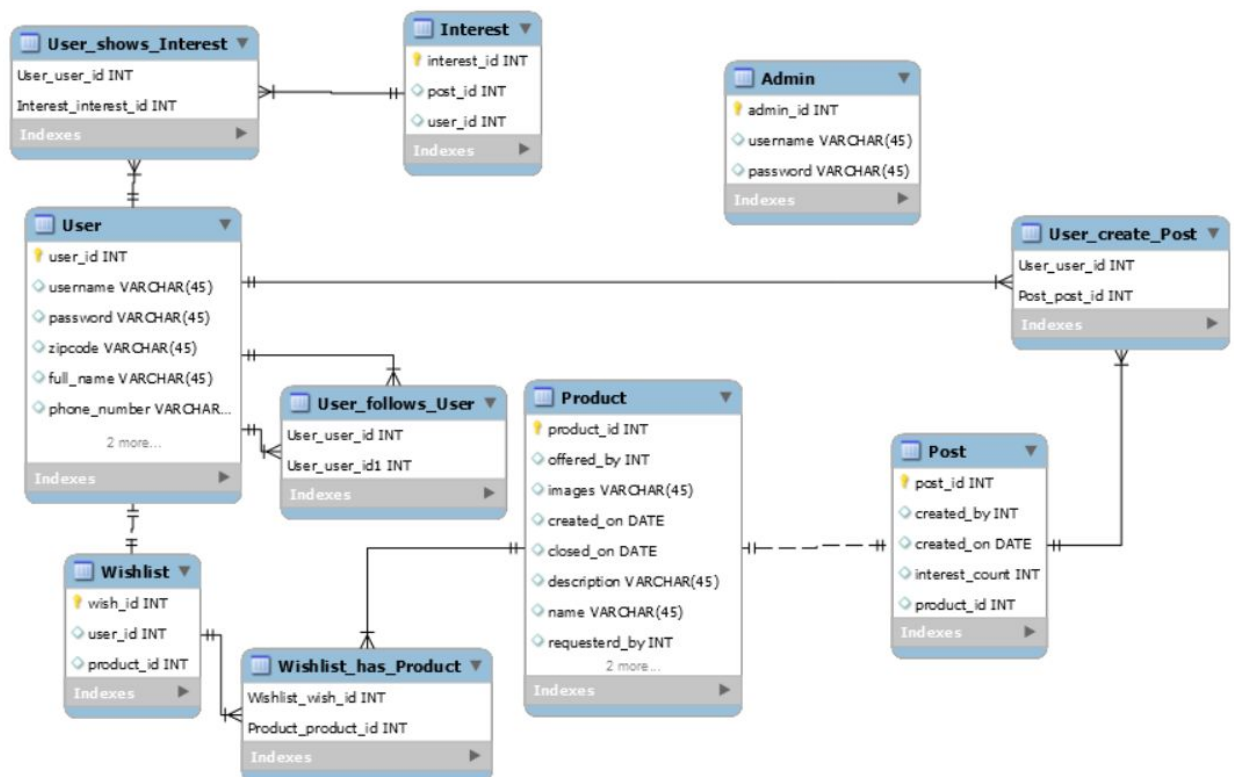


Fig: database model

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 API and Main Algorithms

API's available for every visitor

- **api/signup**
Post request to create users.
- **api/signin**
Post request to allow users to login.
- **api/getPosts**
Post request to get the list of available products.
- **api/getPost?id**
Get request to view the selected product.
- **api/search/product?productId**
Get request to search for a product.

API's available for registered Users

- **api/user/postProduct**
Post request to add a post.
- **api/user/contactDonor?donorId**
Get request to get contact information of the donor.
- **api/user/updateProduct?productId**
Post request to update the product.
- **api/showInterest/user?userId**
Post request to show interest in a product.
- **api/removeInterest/user?userId**
Post request to remove interest from a product.
- **api/user/makeDonateion**
Post request to allow users to make financial donations.
- **api/user/postFinancialProblem**
Post request to allow users to ask for financial donations.
- **api/user/deleteFinancialPost**
Post request to delete the post related to a financial problem.

- **api/user/updateFinancialPost**

Post request to update the financial post.

- **api/user/shortlistFinancialPost**

Post request to shortlist the financial post.

API's Related to Products

- **api/product/updateImage**

Post request to update the image related to the product.

- **api/product/updateDescription**

Post request to update the information of a product.

- **api/product/updateLocation**

Post request to update the location of product availability.

- **api/product/deleteProduct?productId**

Post request to delete a product with given id.

Main Algorithms

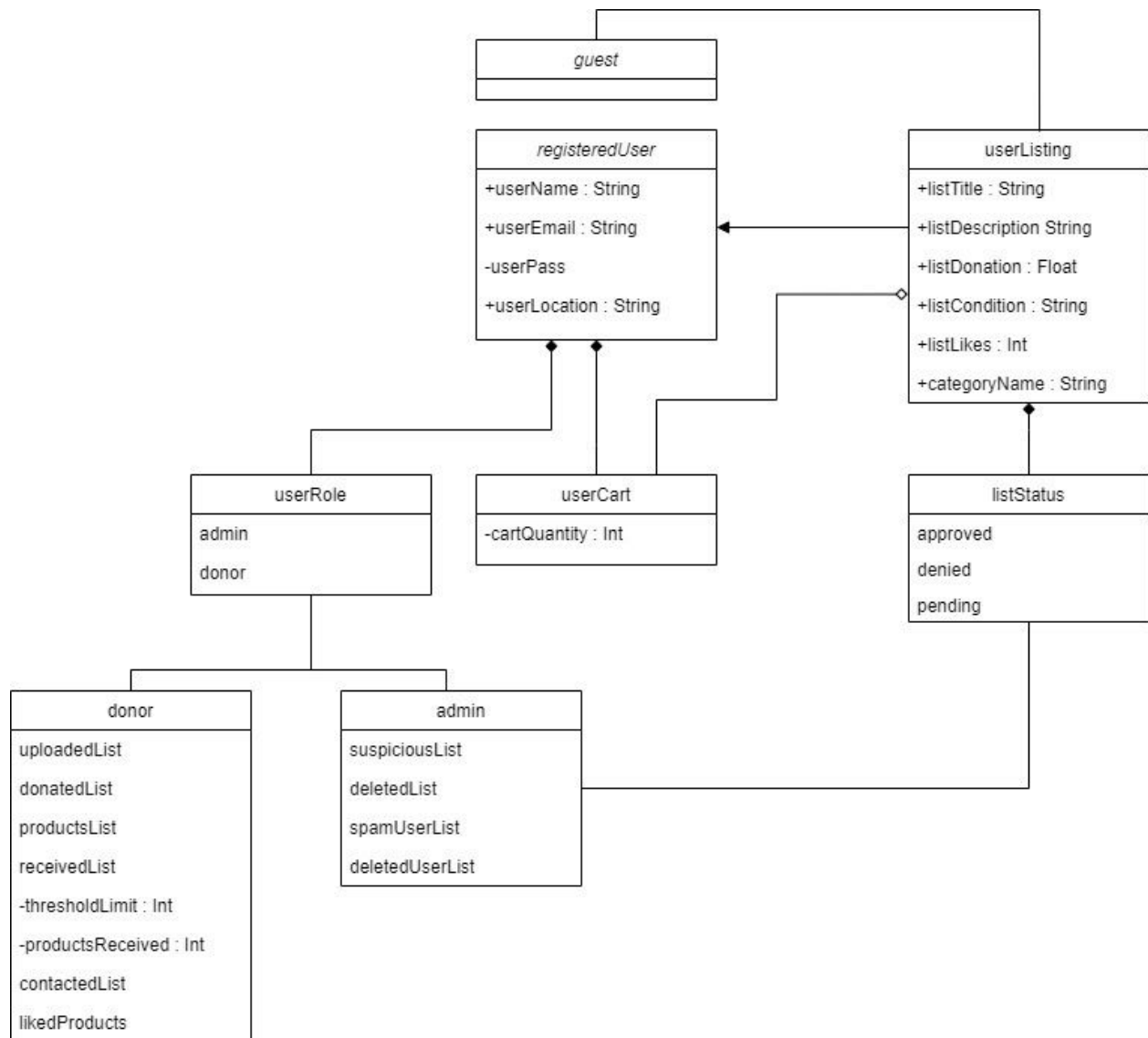
Searching: The information about each product will be stored alphabetically according to the product name and hence when searching for the product, binary search can be implemented to figure out the right product. Whenever there are multiple products with the same name, productid will be used to select the right product.

Rating: Every donor can be rated by different users and to minimize the time to find the average of all the ratings, we will keep a track of the total number of stars given to that donor and the total number of ratings so we can directly divide them both to find the average rating.

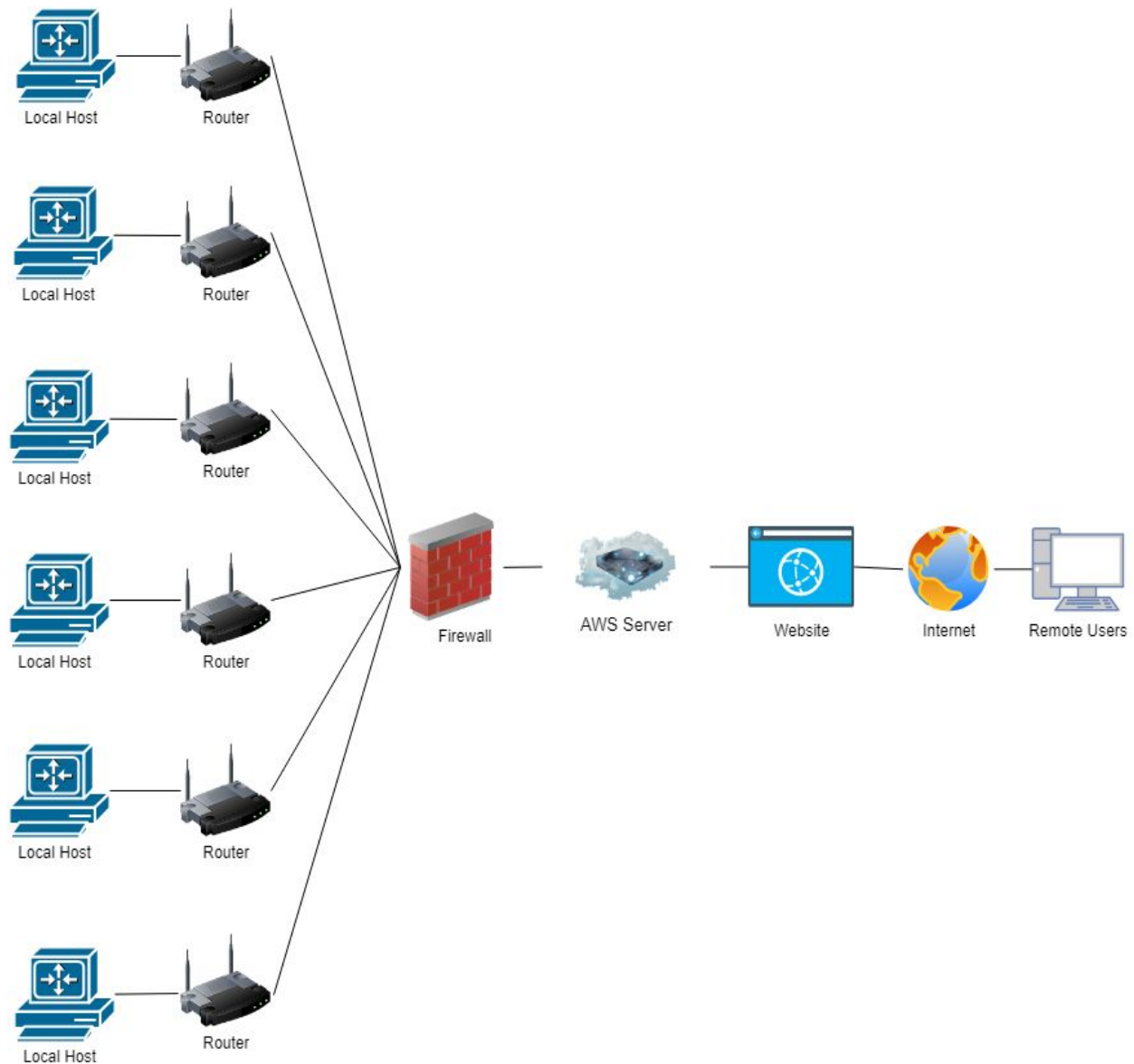
Third Party API

We will be using Stripe's API to make monetary transactions from donor to the receiver.

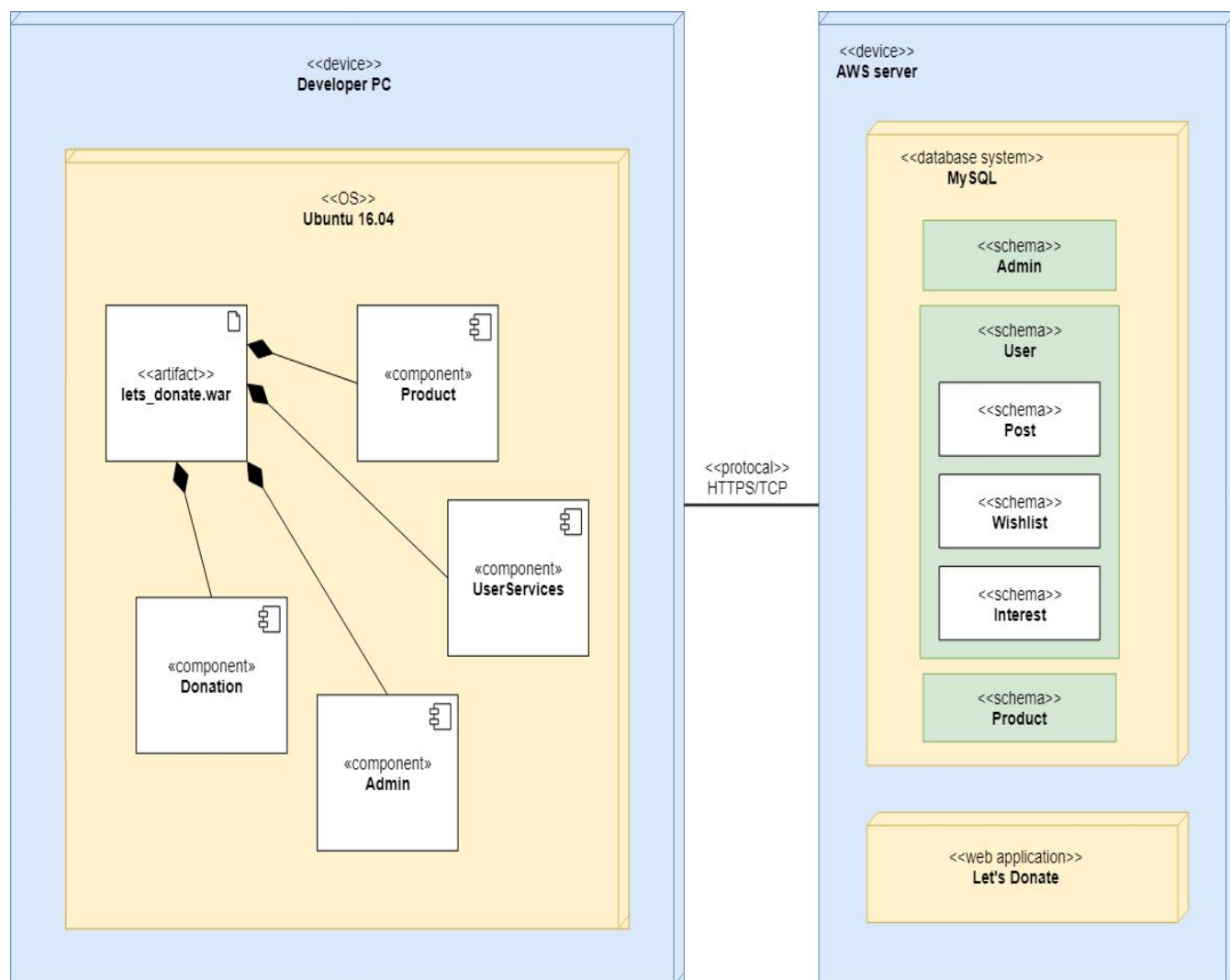
6. High Level UML Diagrams



7. High Level Application Network and Deployment Diagrams



Application Networks Diagram



Deployment Diagram

8. Identifying Actual Key Risks for the Project

Technical Risks:

a. Same user with multiple emails

There will be various receivers willing to receive a product from the donor and the identification used will be their email address and there is a possibility that some users will use multiple email addresses to create an account and therefore receive more than five products per month from the website. It will be hard for the technical team to distinguish the same users with different emails.

Approach towards the risk:

Instead of having a single way of identifying a user, we will have multiple ways to authenticate the user like having both email and mobile number to identify a user which will heavily remove the risk of multiple accounts.

b. Spam Users

There is a possibility of having some spam accounts where a fake donor will post a product to attract different receivers and will try to steal the receiver's personal information like address and other private information.

Approach towards the risk:

The admin will keep a track of the spam users. Whenever there are spam users, the most common way to detect the users who post multiple products for donation consistently to attract more receivers and we will be detecting the number of products posted by each user and therefore analysing fake accounts.

c. Fake Fundraisers

There will be plenty of users who will try to post fake stories in return for getting funds like fake illness or fake financial problems.

Approach towards the risk:

The website shows an image section for the fundraisers where the user needs to upload a trustable paper or certificate like a medical certificate when asking for funds from donors. The users donating the money will be responsible for making any kind of transaction.

Legal Risks:**a. Users uploading stolen products**

There can be scenarios where the users can upload some of the products for donation that don't belong to them. Hence there can be some legal obligations that the receivers might have to face after receiving a property from a false donor.

Approach towards the risk:

The data related to every user will be stored in the database and hence whenever a stolen product is donated, the donor can still be traced.

b. Users donating harmful/deadly products

There can be donors who will post products giving away some harmful weapons or maybe some form of drug. There are definitely chances of interested receivers.

Approach towards the risk:

Every product that appeared to harm anyone in any way will soon be deleted by the admin or by the team of moderators. The website does not intend to deliver any harmful products.

9. Project Management

For Milestone 2, we first divided the tasks according to the proficiency and comfort level of each team member. All team members agreed to work on a google doc so everyone is aware of the status of every part of the milestone and also be able to review or contribute to each other's work. Everyone was given equal workload and we kept the option open to swap some of the work with other team members during the initial phase of the milestone if both team members are comfortable exchanging some of their work portions. The approach seemed to work effectively for most of the part but there will be some changes that we will be making for the upcoming milestones as described below.

Changes for Upcoming Milestones

- Specifying a timeline for each part or some portion of individual work so work progress is smooth.
- More team meetings to discuss team related parts and therefore implementing different portions with least errors.
- Implementing a quick rough outline or a rough draft before implementing/ writing the end product.
- Effectively splitting work among the frontend and backend team so both teams are more aware of their upcoming responsibilities.

10. 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 API and Main Algorithms: Himanshu Garg, Marlon Bustamante.

High Level UML Diagrams: Marlon Bustamante.

High Level Application Network and Deployment Diagrams: Kunyu Ruan

Identifying Actual Key Risks for the Project: Himanshu Garg.

Project Management: Himanshu Garg

Vertical Prototype

Frontend: Milo Abril, Himanshu Garg.

Backend: Marlon Bustamante, Suman Basaula.

Database Management: Dipendra Dhoj Rana.

Deployment: Himanshu Garg.

Team Overview

There have been several team meetings where all team members participated heavily. 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. Everyone did their part with great dedication and proficiency.