**Social Financer**

An application to bridge the gap between social inequality and classes formed among people relative to the resources they have access to in their lives. It basically aims to ease the process of finding the rightfully needy people the rich can help/facilitate.

This application is different from other social apps as it focuses on long term commitment between a donor and a consumer. Moreover, it includes transfer of non monetary services as well like education, water etc.

It may be noted here that the application does not perform any maintenance checks on its users and does not take any responsibility towards security or moral ethics of donors or consumers. Users however, can report other users to the application authorities by mail/phone and after verification of the report, the certain user may be removed from the application permanently.

**Features (Consumer, Donor):**

1. Donors and consumers register themselves on the app.
   1. A form is displayed to the user. User enters his basic information.
   2. The longitude and latitude for the user’s address are acquired by google geocoding API and saved in the model at this stage.
   3. User specifies if he is a consumer or donor.
   4. User chooses the categories he wants to participate in. Consumer can choose a single category while donors can choose multiple categories.
2. Once registered, consumer waits till a donor finds him.
   1. Consumer is notified by text message once a donor selects him.
3. Once registered, donor can see his nearby consumers and their categories.
   1. System finds nearby users by comparing their addresses and selects the nearest ones.
   2. Consumers with similar categories are shown at the top.
4. Donor selects consumer(s).
   1. Donor can select one or multiple consumers to start contributing.
5. Once a donor has selected consumers, the consumers’ contact info i.e. phone number and home address are displayed to the donor so he may contact him/her.
6. A user can report another user by mail/phone to the management of the application.
   1. After verification of the report, the user may be permanently removed from the application.

**Features (Admin):**

1. See all registered users and their details.
2. See Donors of Consumers and vice versa.
3. Change personal information of a user like phone number/address.
4. Add user to blocklist with appropriate reason.
5. Remove a consumer from a donors list.

**Technology stack:**

Server: postgresql

Framework: Django 2.1

Language: Python 3

Frontend: ReactJS, Bootstrap

3rd Party: [Google Geocoding Api](https://developers.google.com/maps/documentation/geocoding/start)

References:

* [Geocoding python implementation](https://stackoverflow.com/questions/25888396/how-to-get-latitude-longitude-with-python)
* [User model and form](https://simpleisbetterthancomplex.com/tutorial/2016/07/22/how-to-extend-django-user-model.html)
* [Sending text messages with python (Twilio)](https://stackoverflow.com/questions/430582/sending-an-sms-to-a-cellphone-using-django)
* [Sending text messages with python (Twilio) (2)](https://www.twilio.com/docs/sms/tutorials/server-notifications-python-django)
* [Sample Quizzing Django App](https://simpleisbetterthancomplex.com/tutorial/2018/01/18/how-to-implement-multiple-user-types-with-django.html)

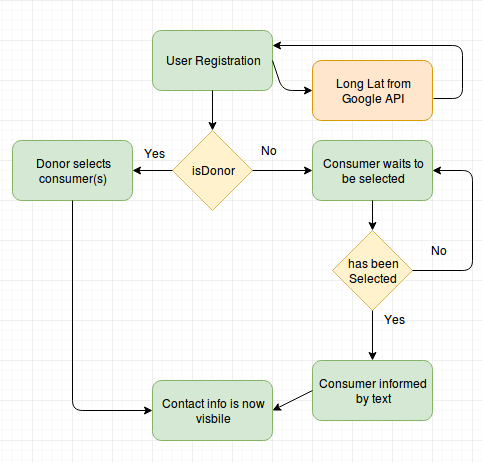
**Flow of application:**

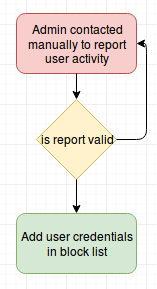
Following is the basic flowchart of the application:

= Actions outside system

\_ = System Actions/Processes

\_ = Conditions

****

****

**Users:**

Admin - Complete control over database and all permissions.

Donor

Consumer

**Tables:**

User

|  |  |
| --- | --- |
| id | pk |
| Name |  |
| Cnic |  |
| Address |  |
| Country |  |
| City |  |
| postal code |  |
| phone |  |
| Role | Donor | Consumer | Admin |
| DonorId | fk, nil for donor |

Category

|  |  |
| --- | --- |
| id | pk |
| Name |  |
|  |  |

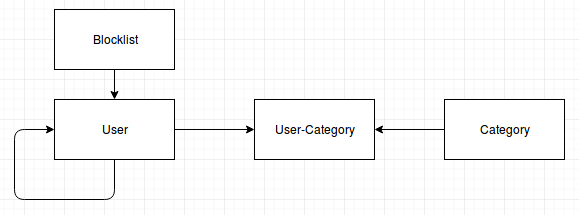
User-Category (Many to Many)

|  |  |
| --- | --- |
| id | pk |
| userId | fk |
| categoryId | fk |

BlockList

|  |  |
| --- | --- |
| id | pk |
| userid | fk |
| reason | string |

**Schema**:

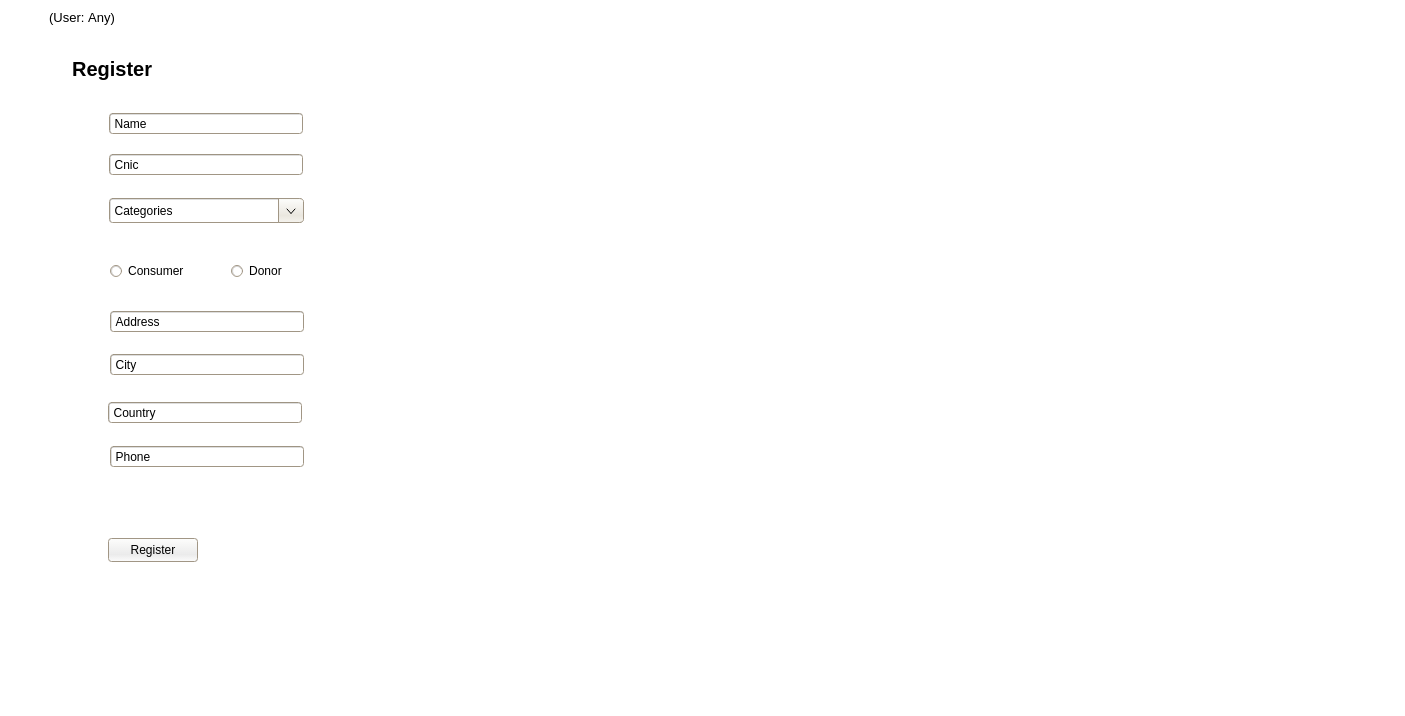


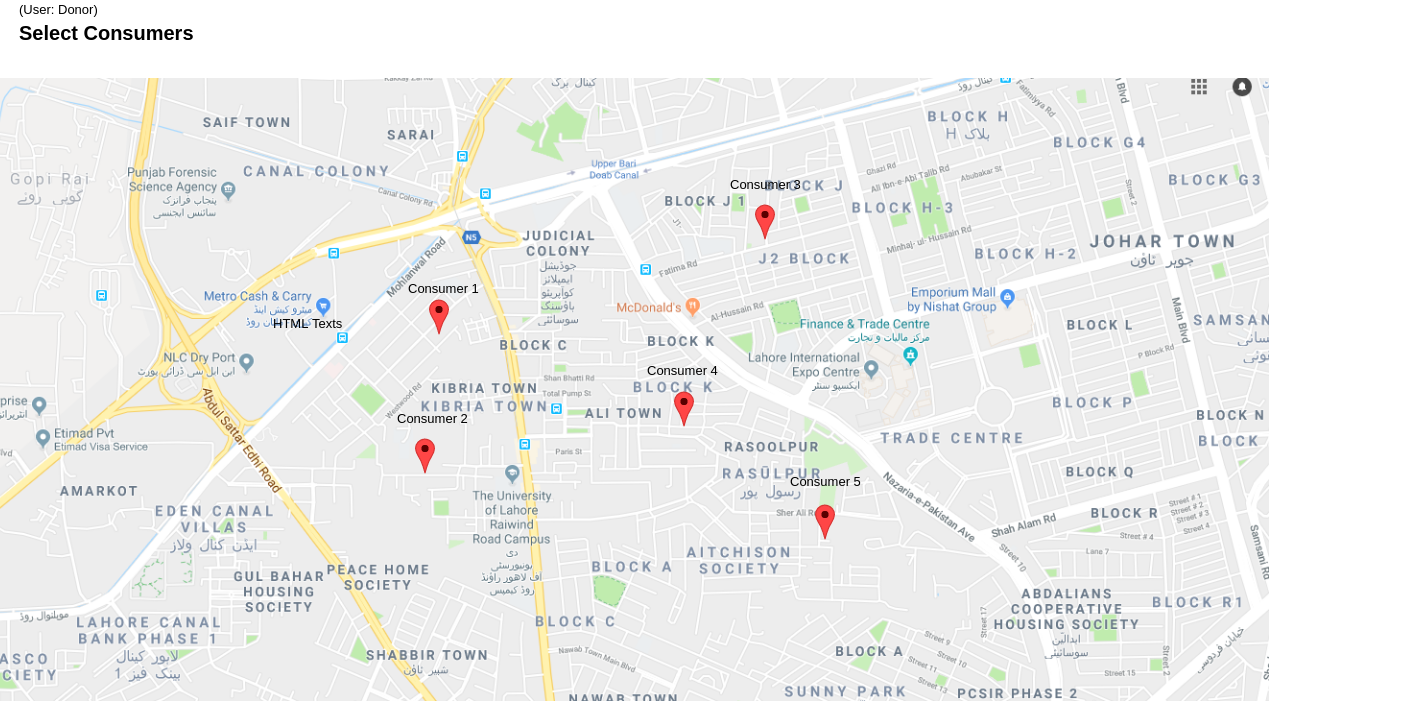
**Implementation Plan:**



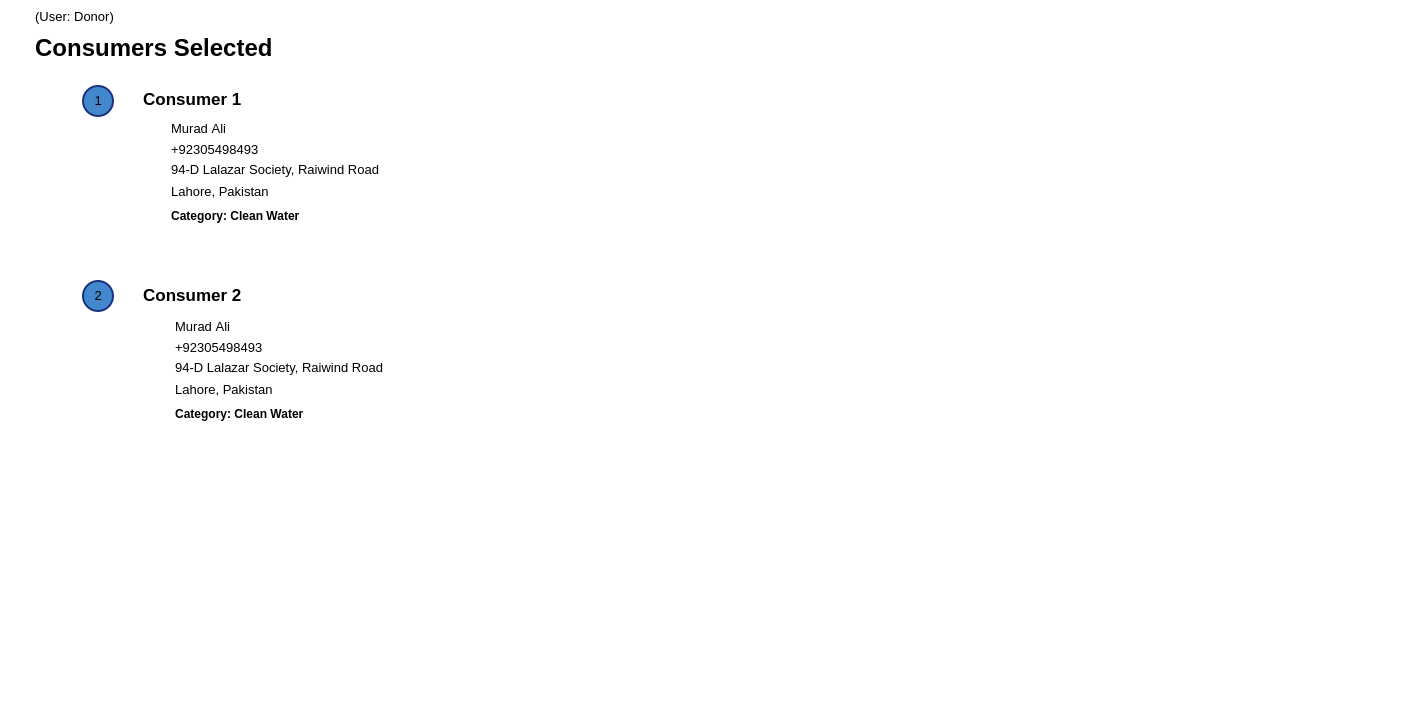
**Wireframes:**

Following is the registration page common to all user types.





Once the donor has selected consumers, the following screen appears, displaying the contact info of all consumers he selected.



**Consumer View:**

After registration, the consumer is redirected to this page.

* If he is selected, his ‘Status’ is set to ‘Selected’ with the donor’s info displaying below.
* If not selected, the ‘Status’ is set to ‘not selected’.

