CHITTAGONG UNIVERSITY OF ENGINEERING & TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Course Code : CSE-252

Course Title : DATABASE MANAGEMENT SYSTEM (Sessional)

FINAL REPORT

Project Title: Zakat And Donation Platform

Presented By :

Md. Akib Hasan (1904015)

K.M Mahabub Hossain (1904017)

Sadman Rahman Ananta (1904020)

Supervised By :

Md. Shafiul Alam Forhad

Md. Atiqul Islam Rizvi

Department of CSE, CUET

Introduction:

The purpose of this application is to collect & distribute zakat & charity centrally. It maintains proper collection & donation in a systematic way. We developed this software to collect zakat in a proper way. It mainly helps the Muslims to donate their portion of wealth in a year which they have to donate if enable, with proper calculative way. By using this app, they will know the minimum amount of money that need to be donated to fulfill the conditions of paying Zakat. This app will calculate their minimum money which they need to donate. After completing the transaction, the amount will be added to his name in our database. Zakat payer is enable to view the amount he has paid. This app is not only to pay zakat but also to donate. This project offers anyone to do contribute for charity which we need to use in case of any emergency. Anyone can donate any amount of money that he desired. After donating, his data will also be updated with that amount in our database. He can also get the view of his donation whenever he wants to . Anyone can check the amount what's now in the fund of zakat & what's now in the fund of charity. We give that options to the viewers to maintain the lucidity.

Our aim is to collect the zakat centrally so that we can make the proper distribution in a regular way without having any unnecessary trouble. This project can also be useful in the necessity of emergency by the charity fund. It will help the donator too as nowadays people lead a life like machine & they have very little time to give for contribution in charity, by using this app they can easily donate & make participation in charity work directly.

Objective:

One of the tenets of Islam, zakat, is not **properly collected** or distributed in our nation. As a result, we devise a plan to handle this centrally in order to fill the gap. People who have been harmed by disasters in our nation don't always receive timely assistance. We'll make sure to gather their information beforehand, get in touch with them right away, and try to limit the harm.

Motivation:

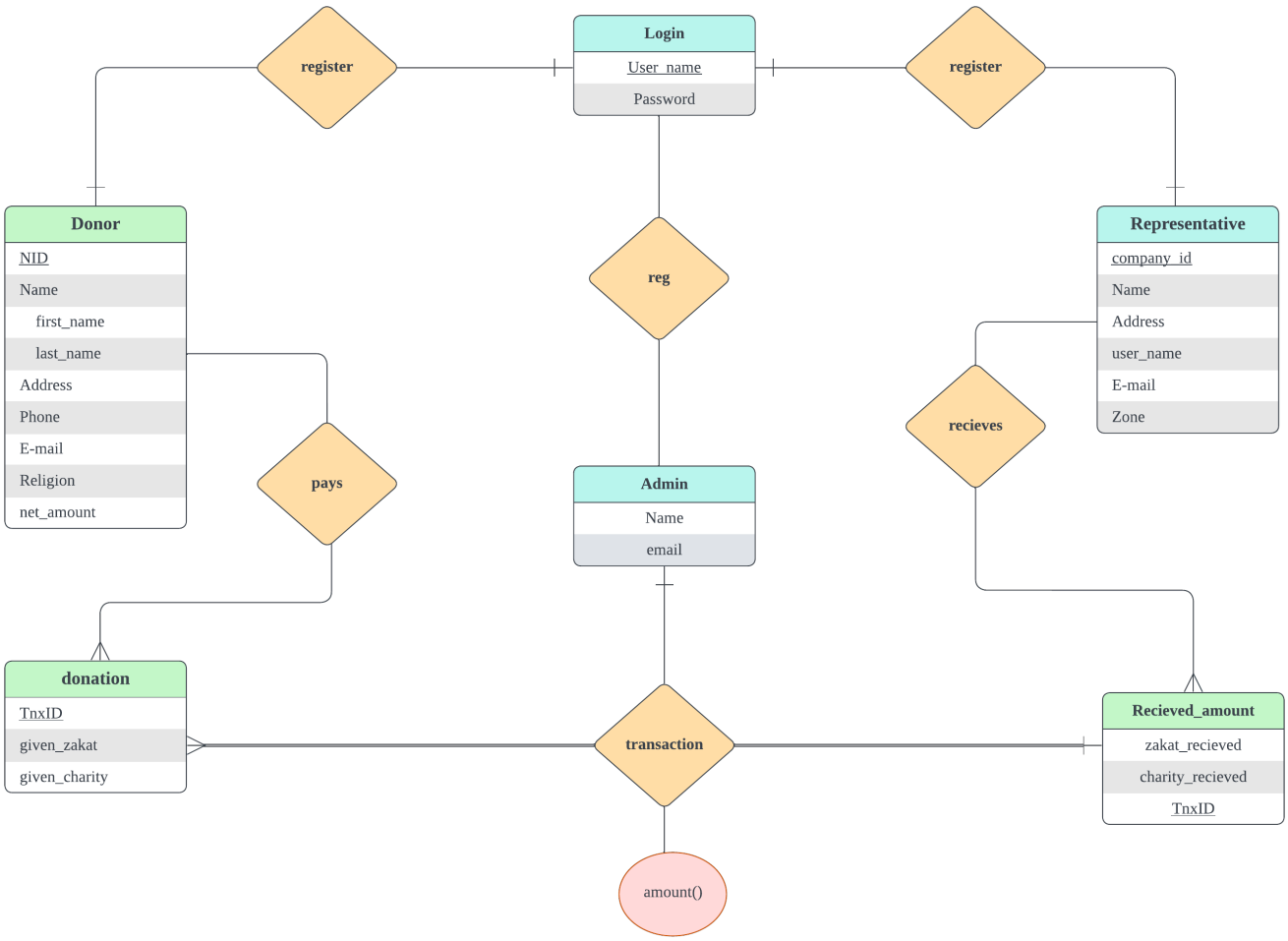
- This project's primary motivation is to improve the practice of Muslim Ummah. One of the tenets of Islam is **zakat**, and those who are able must give up a certain amount of their money each year. There isn't a good structure in place in our nation to gather it **centrally** so that it may be donated properly. The goal of zakat can be fulfilled if we can **distribute the funds** to those who qualify. By upholding zakat in accordance with Shariah, we can accomplish zakat's correct objective, which is to assist the underprivileged in improving their economic situation.
- Every **natural disaster** creates a new predicament. Our country's geographic location makes us vulnerable to natural disasters like cyclones and floods, and there is little we can do to stop them. Affected people don't **quickly receive** relief during that difficult moment. Additionally, in certain areas, the true poor in the rural area don't receive aid, and regularity isn't maintained. So that we can distribute the **relief supplies** in an efficient manner and ensure that the genuine impacted deprived people receive the benefit, we want to create a platform where anyone may donate any amount of money.

Project Description:

In this project, the design of the database schema and its fundamental implementation were our primary concerns. As a result, we began by creating the ER diagram first. On it, we subsequently applied mapping and normalizing. Finally, we used sqlite3 to implement the database schema. Then, to

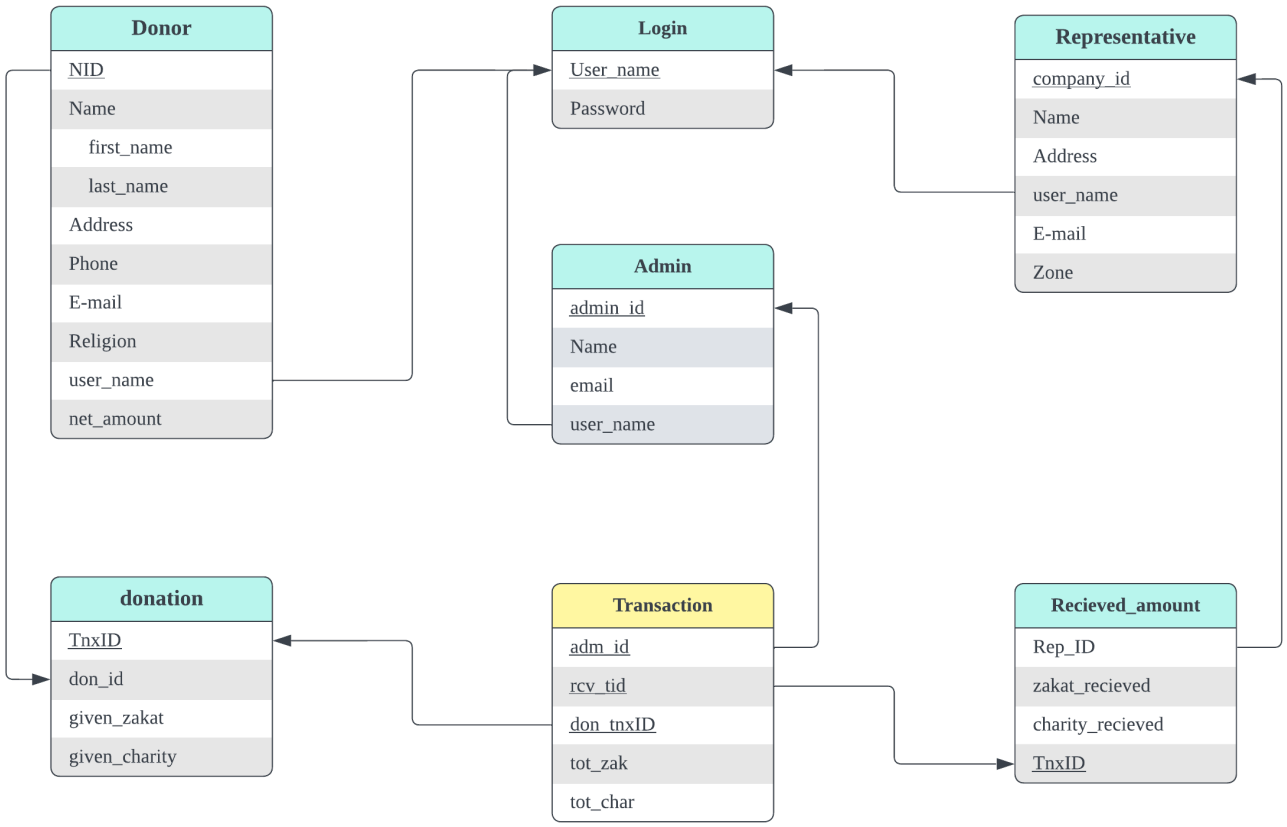
make it simple to retrieve and query data, we used the pyqt base GUI.
Here, A donor, representative, or administrator must log in first using a special username. The donor can then update his information and make his desired zakat or charitable contributions. He can also see how much he has donated (both zakat and charity)
A user can update his information and view the amount made accessible to him by the system as a representative.
Finally, an admin can view the representative's overall allowable budget. He is free to choose the representative and the amount to be paid.

ER Diagram:



This diagram is the basic schema of the project. Here we have six strong entities:
Donor, Donation, Representative, Received_amount, Admin, Login
There are five binary relationship, such as:
Pays: Donor pays charity or zakat to donation
Receives: Representative receives amount in Received_amount
Along with three similar relationship:
Register: Donor, Admin, Representative ‘s login data with Login table
And a ternary relationship:
transaction: between admin, donation and received_amount with amount() as a relationship attribute, where admin creates a transaction from donation to receiver of given amount.

Relational Mapping:



The mapping of the ER diagram is made by converting the entities and relationship consecutively. Here six of the tables are created from the six entities with foreign key.
Three of the entities are converted to database tables or relations with foreign key from login table.

DONOR(NID, first_name, last_name, address,phone, email, relagion, net_amount, user_name)
ADMIN(admin_id, name, email, user_name)
REPRESENTATIVE(company_id, name, address, email, city, user_name)
LOGIN(user_name, password)

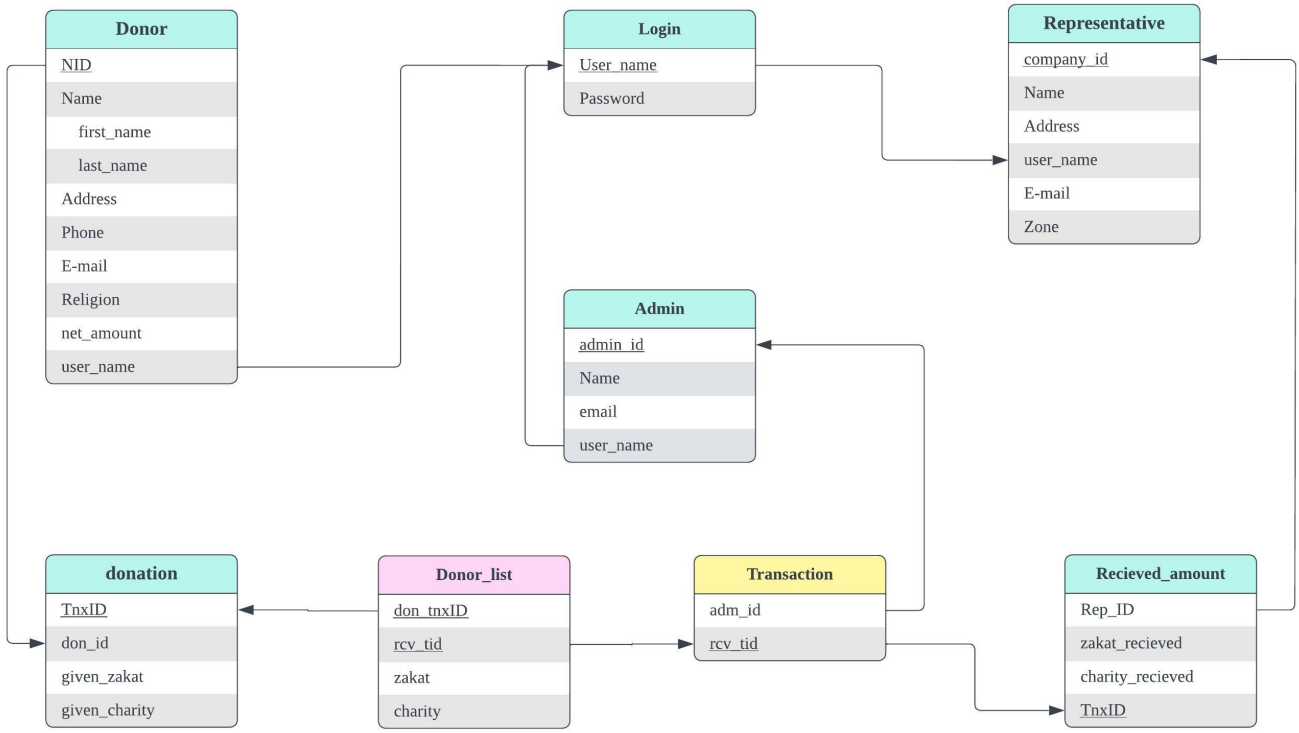
Here, These entities having one to one relationship with login entity. Therefore, for our implementation purpose, we used user_name as foreign key in every table.
Then comes to donation and received entity, which were made table easily like this:

DONATION(TnxID, given_zakat, given_charity, don_id)
RECEIVED_AMOUNT(TnxID, zakat_received, charity_received, rep_id)

These entities having one to many relationship with donor and representative respectively. Therefore we shifted the primary key of one side to many side.
Finally, the ternary relation is converted to a table along with the primary keys of three entities with it's relationship attribute. Such as,

TRANSACTION(adm_id, rep_tid, don_tid, zakat, charity)
That's how the mapping was done from the ER diagram.

Normalization:



From the mapping diagram all the six basic relation has simple one primary key with no multi-valued attribute, so these tables can be said at least 3NF form. As per the relation from ternary given zakat and charity is already stored in the received_amount relation using rep_tid. Therefore there it is redundant to store in the transaction relation again, thus they were discarded.
Now the primary key as rcv_tid of transaction relation alone is enough to perform as a primary key, so other keys are made just normal foreign key. Again these keys can be redundant, such as:

If we want to do some transaction:

rcv_tid	adm_id	don_tnxID	zakat	charity
124	1	56	10000	500
124	1	57	0	9500
125	2	57	10000	10000
126	2	57	5000	5000

Where donation relation seems like:

TnxID	given_zakat	given_charity	don_id	tnx_num
56	10000	500	123	abc
57	20000	30000	345	def

Clearly, adm_id and rcv_tid is dependent on each other i.e., for a rcv_tid there is only one adm_id. But don_tnxid can vary to each other which depends on how much a donor donated and how much we want in a transaction. Thus the don_tnxID is placed in a new table with corresponding amount where the primary key is rcv_tid.

Sample Tables:

Here are the sample tables used for query:

Table: admin

	admin_id	name	email	user_name
	Filter	Filter	Filter	Filter
1	1	jim	jim@dundermifflin.com	admin
2	2	dwight	dwight@dundermifflin.com	admin2

Table: donation

	tranx_id	don_id	given_charity	given_zakat	tnx_number
	Filter	Filter	Filter	Filter	Filter
1	3	1234	12000.0	0.0	abcd
2	4	2345	3000000.0	5000000.0	pqrs
3	5	3456	3400000.0	268400.0	*NULL*

Table: donor

	ssn	first_name	last_name	address	email	relegion	user_name	tot_zak	tot_char
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	1234	akib	hasan	CUET	u190401...	islam	akib	3000000	12000
2	2345	mahbub	rabbi	bbh	rabbi128...	islam	rabbi	5000000	4000000
3	3456	sadman	ananta	mes	u190402...	islam	sadman	8000000	56000

Table: received_amount

	tranx_id	rep_id	recv_charity	recv_zakat
	Filter	Filter	Filter	Filter
1	13	cu124	5000.0	30000.0
2	14	nat345	340000.0	23000.0
3	15	nat345	12000.0	22000.0
4	123	cu124	12000.0	16000.0

Table: representative

	company_id	company_name	address	city	email	user_name
	Filter	Filter	Filter	Filter	Filter	Filter
1	cu124	cuet charity	pahartali	ctg	cuet@ma...	cuet_med
2	nat345	national zakat institute	agargaon	dhaka	ntz@mail...	national_zakat

SQL Commands:

** To **insert donation data** in the donation table donated by the person having don_id, we used sql command like this,

```sql
INSERT INTO donation(don_id,
    given_charity, given_zakat, tnx_number)
VALUES(1234,1200, 1200, 'new')
```

Now the donation table seems like this:

Table: donation

	tranx_id	don_id	given_charity	given_zakat	tnx_number
	Filter	Filter	Filter	Filter	Filter
1	3	1234	12000.0	0.0	abcd
2	4	2345	3000000.0	5000000.0	pqrs
3	5	3456	3400000.0	268400.0	*NULL*
4	6	1234	1200.0	1200.0	new

** We can update the personal data of donor, representative such as,

```sql
# update data such as representative relation
UPDATE representative
SET address = 'raozan'
WHERE user_name = 'cuet_med'
```

Then the updated representative table seems as,

Table: representative

	company_id	company_name	address	city	email	user_name
	Filter	Filter	Filter	Filter	Filter	Filter
1	cu124	cuet charity	raozan	ctg	cuet@ma...	cuet_med
2	nat345	national zakat institute	agargaon	dhaka	ntz@mail...	national_zakat

**To show the admin that how much debatable zakat and charity left, we can use,

```
12  # show total amount
13  SELECT sum(given_charity) as charity,
14      sum(given_zakat) as zakat
15  FROM donation
16
```

Then it returns this table,

	charity	zakat
1	6413200.0	5269600.0

```
Execution finished without errors.
Result: 1 rows returned in 15ms
At line 1:
SELECT sum(given_charity) as charity,sum(given_zakat) as zakat
FROM donation
```

** For a particular representative we can see the total amount he received by using command like this

```sql
# show the amount received by an rep
SELECT sum(recv_charity), sum(recv_zakat)
FROM representative
JOIN received_amount
ON representative.company_id = received_amount.rep_id
WHERE company_name = "rep_id"
```

The result shows as,

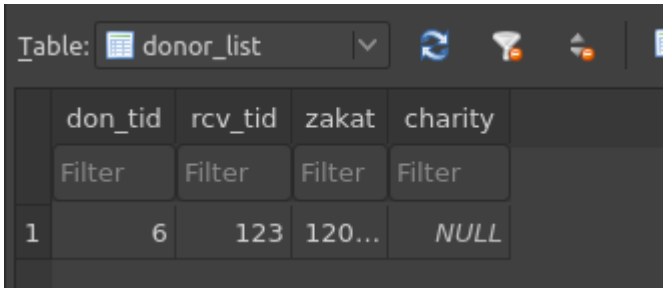
	sum(recv_charity)	sum(recv_zakat)
1	17000.0	46000.0

```
Execution finished without errors.
Result: 1 rows returned in 7ms
At line 1:
SELECT sum(recv_charity), sum(recv_zakat)
FROM representative
JOIN received_amount ON representative.company_id = received_amount.rep_id
WHERE company_name = "cuet charity"
```

An admin can create a **transaction command to give an amount to particular receiver

```sql
# admin creats a transaction to rep such as zakat
# we can search first lowest amount and update it
SELECT tranx_id, given_zakat
FROM donation
WHERE given_zakat > 0
ORDER BY given_zakat asc
LIMIT 1
UPDATE donor_list SET zakat = given_zakat
WHERE rcv_tid = 123 AND don_tid =tranx_id
```

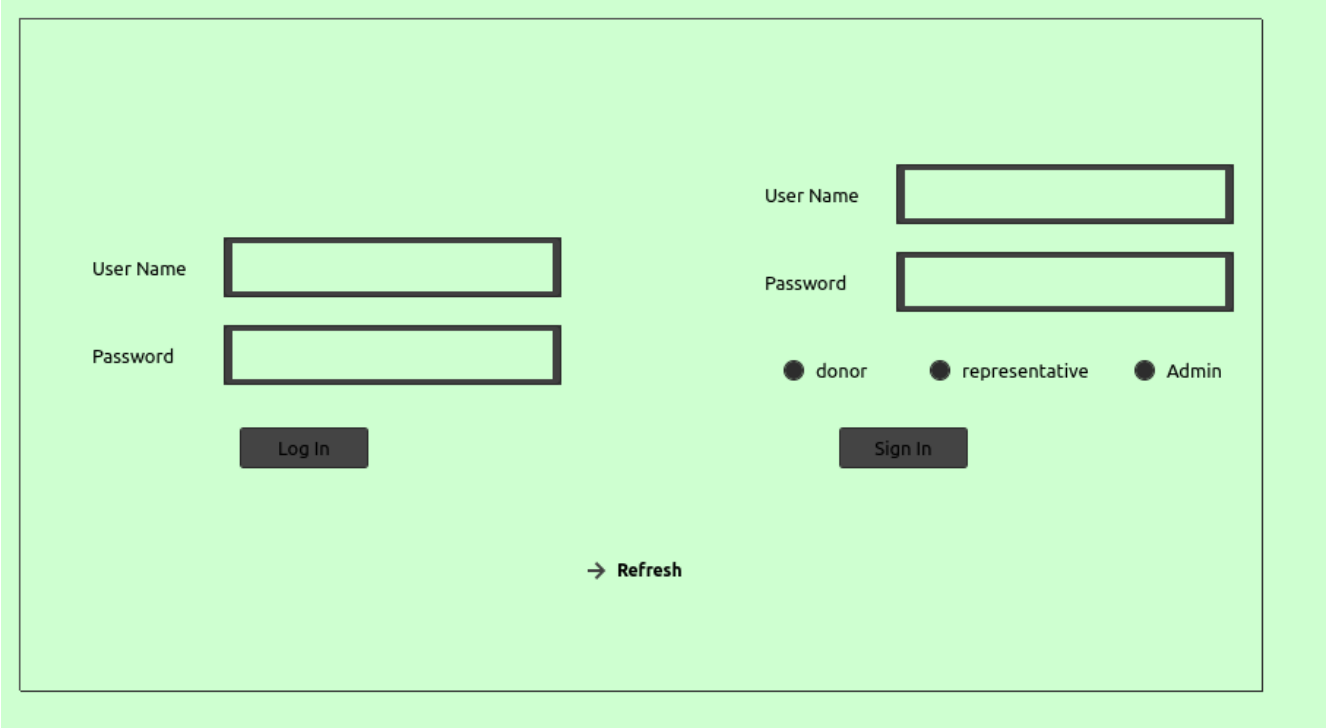
Then the donor_list becomes,



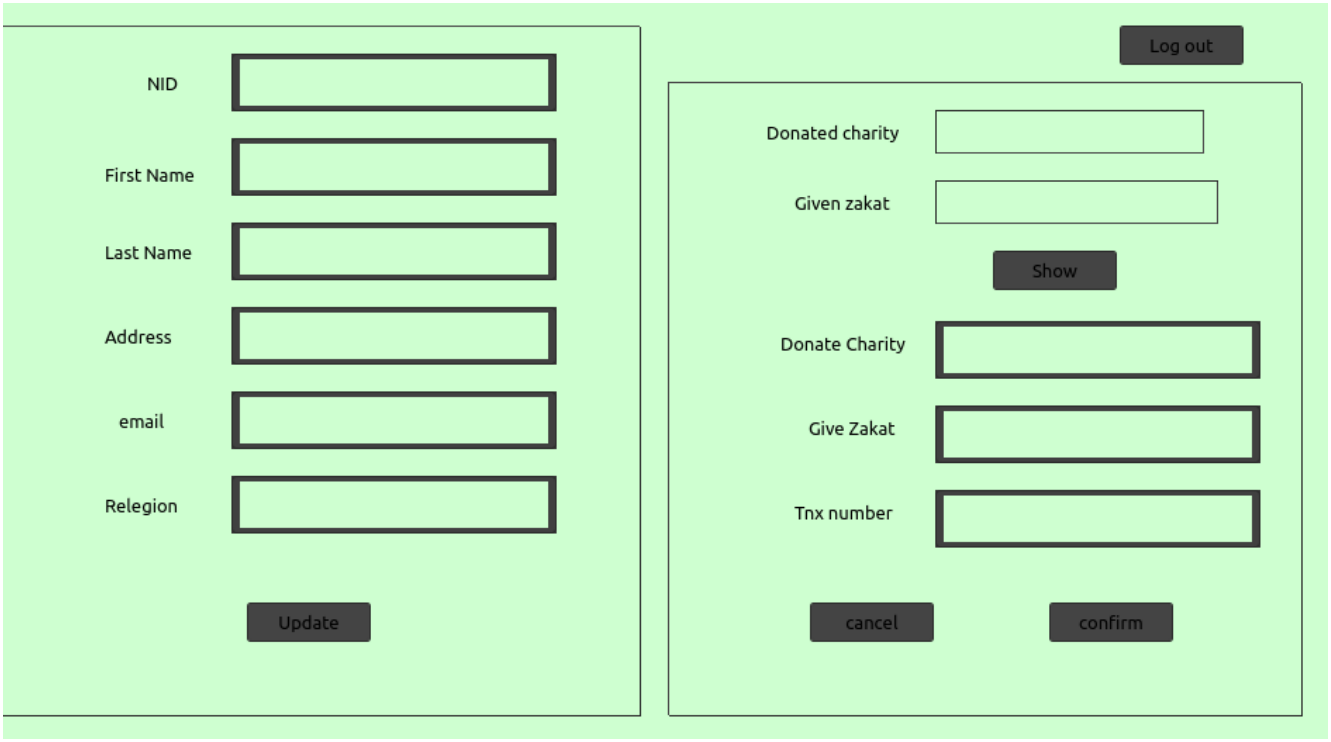
GUI implementation:

To implement these commands we used a GUI created using pyqt and database with python's integrated sqlite3. Some of the interfaces are,

Login Page:

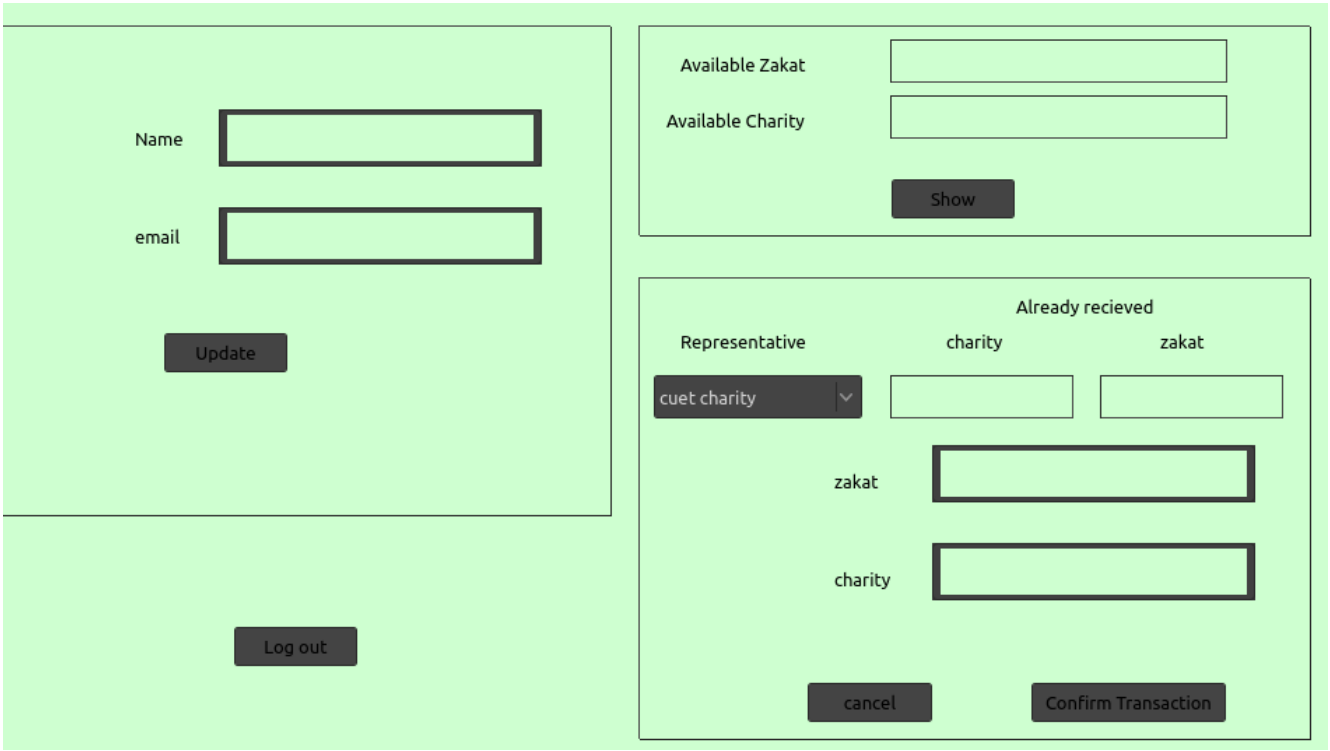


Donor Page:



Representative Page:

Admin Page:



Conclusion:

One of the tenets of Islam, zakat, is not properly collected or distributed in our nation. As a result, we devise a plan to handle this centrally in order to fill the gap. People who have been harmed by disasters in our nation don't always receive timely assistance. We'll make sure to gather their information beforehand, get in touch with them right away, and try to limit the harm.