

Organ and Blood Donation System (OBDS)

DBMS Project Report

Team α

Group 23

Link to our android App:

https://drive.google.com/open?id=1r2dWKhx1XpoCHOKyiStL7PzMnV6K_bt

Link to Github repository: <https://github.com/underhood31/App-Team-Alpha/>

Team Members:

Adnan Ali	(2018214)
Akash Sarkar	(2018011)
Manan Jain	(2018294)
Manavjeet Singh	(2018295)
Navneet Agarwal	(2018348)

Work Distribution :

Before Mid-Evaluation

- **Information gathering** related to specific organ matching and donate/receive criterion for creating tables. Also, schema for other necessary tables:
 - Bone marrow + Stem cell (later combined) + Skin transplant: **Manavjeet Singh**
 - Lung + Cornea Transplant + Intestine(later removed) Transplant: **Adnan Ali**
 - Liver + Heart + Pancreas Transplant: **Akash Sarkar**
 - Kidney Transplant + Donor Information + Patient Information: **Manan Jain**
 - Blood Bank, Doctor Information, Hair Bank(Later Removed): **Navneet Agarwal**
- **Table Populating**
Writing scripts to populate tables: **Manavjeet Singh, Manan Jain, Navneet Agarwal**
Data Rectification and Queries: **Adnan Ali**
- **Table Creation**
Sat together and brainstormed the schema of all the tables.

After Mid-Evaluation

- **Normalization**
Navneet Agarwal
- **Finding the attributes to be Indexed**
Manan Jain
- **Pushing database on online server and Maintaining it (Used in GUI)**
Manavjeet Singh
- **Relational Algebraic query**
Manan Jain
Navneet Agarwal
Akash Sarkar
- **Embedded/Dynamic query**

The query used in Front end (Adnan Ali, Manan Jain, Manavjeet Singh and Navneet Agarwal)

Query other than the front end (Manavjeet Singh, Akash Sarkar)

- **ER Diagram**

Adnan Ali

Navneet Agarwal

- **Front end(GUI)**

Login Page : Adnan Ali and Manavjeet Singh

Sign-up Page : Nanveet Agarwal

Doctor's Interface: Manavjeet Singh

Patient's Interface: Manan Jain

Donor's Interface : Adnan Ali

Blood bank Interface: Navneet Agarwal

Organization Interface: Manan Jain, Navneet Agarwal

Compiling all the interfaces together and exporting the application : Manavjeet Singh

OBDS App

OBDS is the backend for an Organ and Blood Donation System which is accessed via an App. It helps in easy and secure donation of organs and blood. It helps in overcoming the problem of shortage of organs/blood and easy access to the records of receivers and donors.

Stakeholders

1. Doctor
2. Patient and the Patient's family
3. Donor and the Donor's family
4. Organisation

More information about the stake holders is in work in progress Document :

https://docs.google.com/document/d/1Bd744UUtlSXtS-Kwm9KDzmIEF5vHPJv_fPnJ36pM-w/edit?usp=sharing

Bonus Feature:

- We have created two events for the bonus feature, Any Blood_Donor who hasn't donated blood in an year is no longer a valid blood_donor, we reduced this work by creating two events to do so, this eliminates the use of manual labour in doing so, the event runs every 24 hours and deletes any inActive blood_donor, the second event removes there account.

Create Event RemoveOld On Schedule Every 24 Hour Do Delete from Blood_donor where datediff(curdate(),Last_Donated) >365;

Create Event RemoveOldAccount On Schedule Every 24 Hour Do Delete from Account where id not in (Select BLD_id from Blood_bank) and id like('BLD%') ;

The above two event schedulers can be turned on and off using this query below :

SET GLOBAL event_scheduler = ON; //To start the even_scheduler

- **We have successfully created an Android based app (Front end) that will facilitate the purpose of centralized organ and blood donation system.**

This helps the user to use the database efficiently, without typing any queries. Users can access the database contents by filling some text fields and pressing some buttons.

The link to download the app is here:

https://drive.google.com/open?id=1r2dWKhx1XpoCHOKyiStL7PzMnV6K_bT

Note:- Please make sure you are connected to a network while using the app.

- **You get pop ups whenever you enter a wrong input like negative value, non integer value, not a valid id etc and you also get hint text like Minimum age can be 18, Input format for contact etc.**

This helps users to enter the correct values so that there is no mistake in the formulation of a query. This can be seen in the Signup page and some functionalities of other pages.

- **While typing the unique id of blood bank, Organs etc you get suggestions as you type.** This is just like the way it happens while searching a contact or finding an account on Facebook, Instagram etc. This feature can be seen in Transfer units utility of Blood bank , Organ History utility of Doctor etc.

- **You also get swipe down to refresh feature for the live view of units available with a blood bank.** It is just like swiping to refresh your posts on Facebook, twitter etc. This feature can be seen in the Blood_banks page.

Note:- All the bonus features related to app cab\n be seen below or you may directly use the app.

Front End Description :

Sample Login Credentials:

Doctor

ID: DOC_636

Password: DOC_636

Patient

ID: PAT_807

Password: PAT_807

Donor

ID: DON_90

Password: DON_90

Blood Donor

ID: BLD_581

Password: BLD_581

Organisation

ID: ORG_214

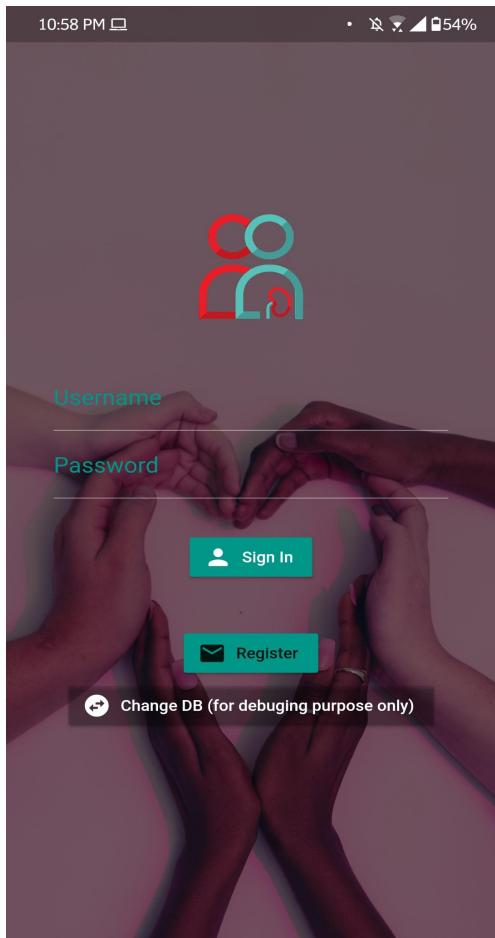
Password: ORG_214

BloodBank

Id: BBANK_999

Password: BANK_999

Login Page



This is the first screen to appear when you open the app. The Login Page asks for user credentials, if the user has not yet registered he can register with the help of a “Register” option. Once the user enters a valid credentials, he/she is taken to their respective page depending on whether the user is a donor, a doctor, an organisation or a patient. The page performs several error-checks for eg. neither of the two fields can be empty, the entered credentials should be valid, etc.

The change DB option will help us to change our database to our backup database. Because free mySQL hostings are not so reliable.

Sign-Up page

This page is used by users to start the process for becoming new donors or new blood donors or both.

In the Login screen there is an option which says Register, If you click on that the screen shown on the side will open.
When you click on any text ,you get some hint text about what should this field contain like pincode format,minimum age etc.
If you leave any mandatory field empty you also get pop ups to show ,what's wrong with the input .
This puts a check that the insert query will pass all the constraints in those tables.If you all fields are correct you get a pop up showing sign up successful.Some screenshots are shown below :-

Here you can see the predictive text for the Age is displayed when you click on it.Similar hint texts are available for all text fields.

10:59 PM 54%
Signup

Name _____

Age _____

Address _____

Pin Code _____ Phone Number _____

Blood Group _____

Do you want to be an Blood Donor

Do you want to be a Organ Donor?

Organ To Donate _____

SIGNUP

Go Back

10:59 AM 49%
Signup

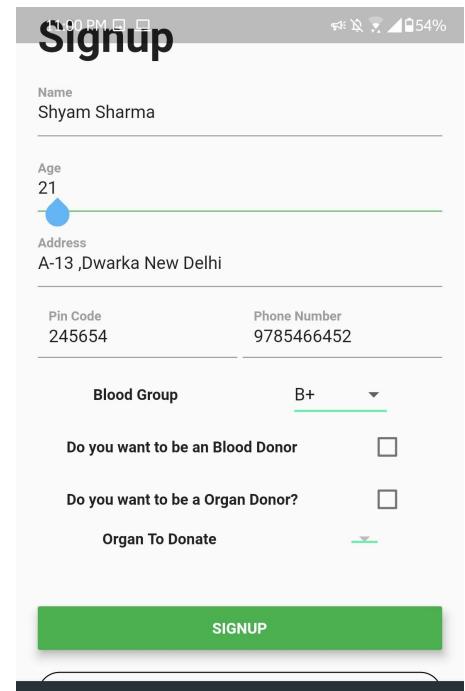
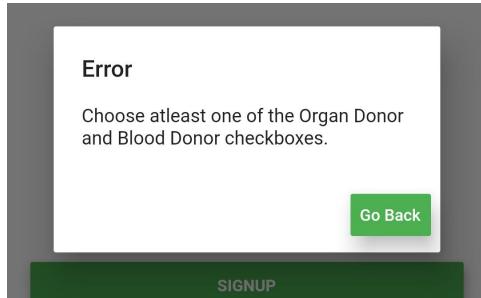
Name
Navneet

Age
You must be atleast 18 years old

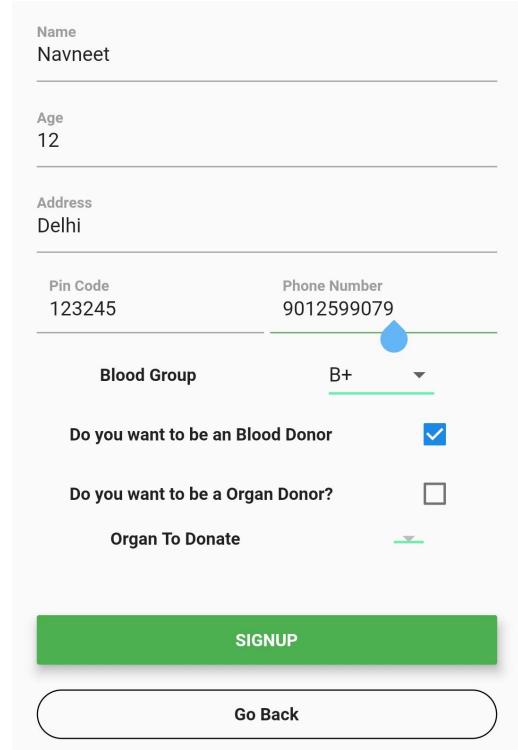
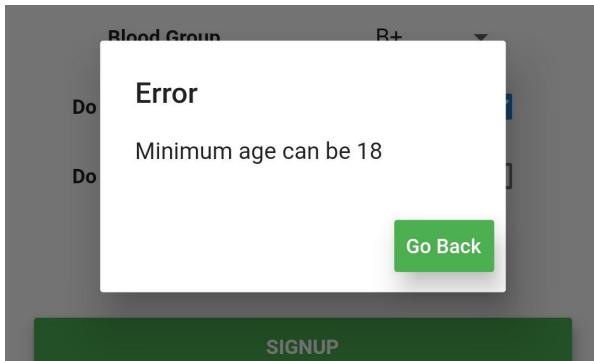
Address _____

Pin Code _____ Phone Number _____

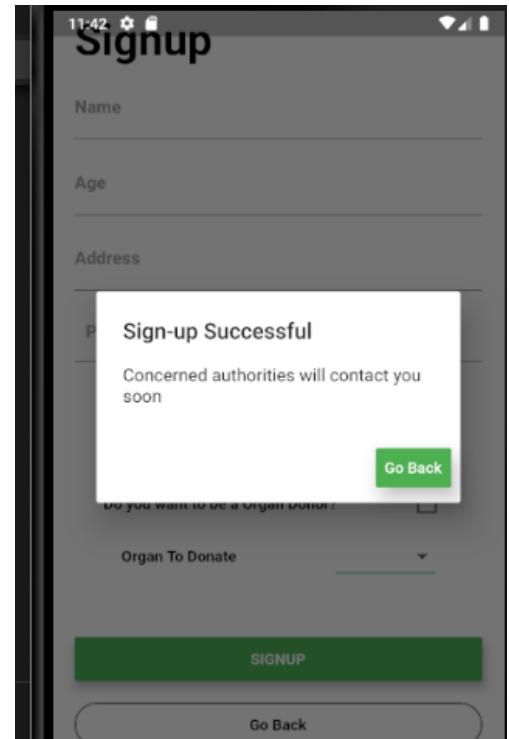
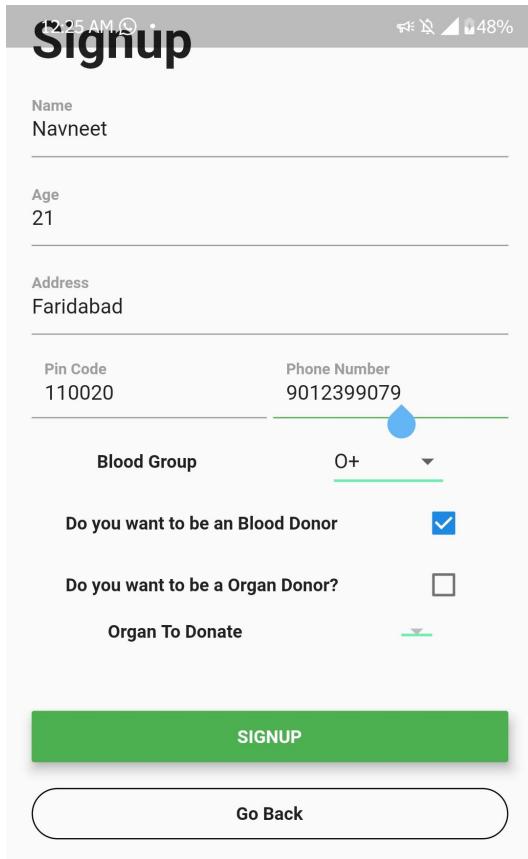
Here, on the right all the fields are correct but user has not tick any of the checkboxes , so you get a pop up, as shown below.Similar pop ups happen if the entry does not match the constraints.



If you click on sign up you will get a pop up that age is not correct



If all the credentials are correct and you click on sign up you will see something like this before and after clicking the button.



Donor Page

1. Organ Donor

The page for organ donor consists of four actions -

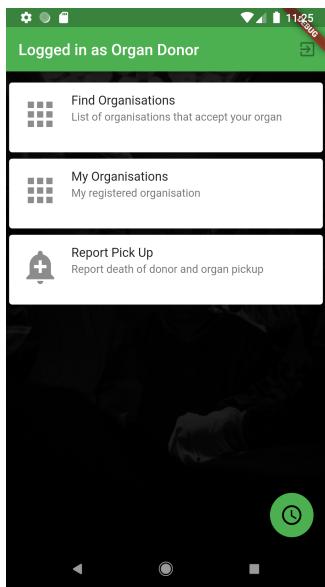
- a. View registered organisation which opens a new page with the information regarding the registered organisation.
- b. View the list of other organisations where they accept the donor's organ which opens a new page with the information regarding those organisations.
- c. Report the death of the donor and organ pick up which opens a page with two fields reporting the cause of death and the time of death of the donor.
- d. View last and next check up dates which opens a pop up window showing the donor's last and next check up dates.

These actions can be performed by the Donor's Family by using the donor's login credentials. Once the donor's death is reported, the Donor page loads up again but this time the page does not consist of the report pick up action.

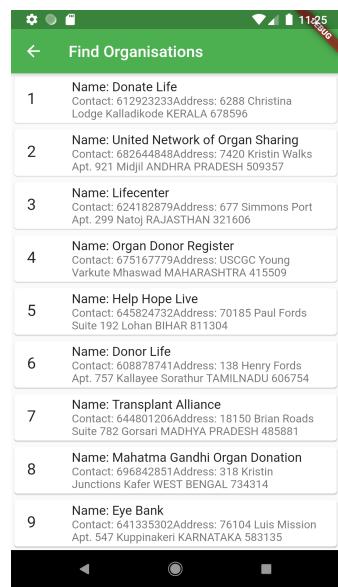
2. Blood Donor

The page for blood donor consists of three actions -

- a. View registered organisation which opens a new page with the information regarding the registered organisation.
- b. View the list of other organisations where the donor can donate which opens a new page with the information regarding those organisations.
- c. View last donated date which opens a pop up window showing the donor's last blood donation date.



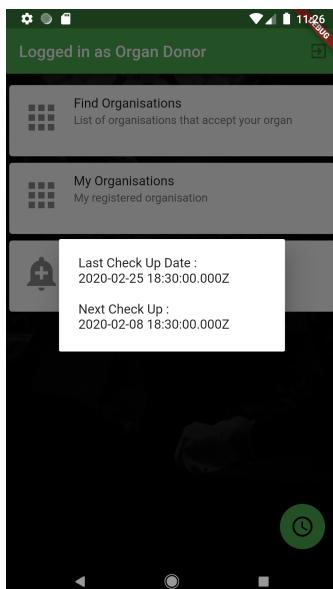
Donor Page



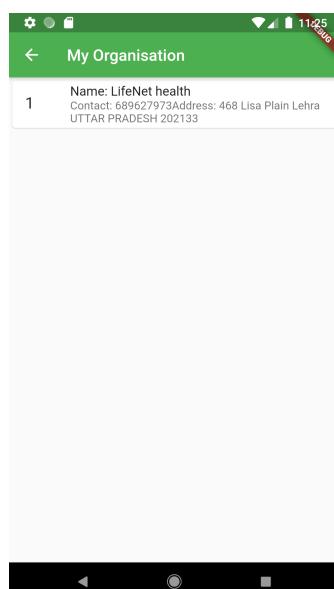
Find Organisations



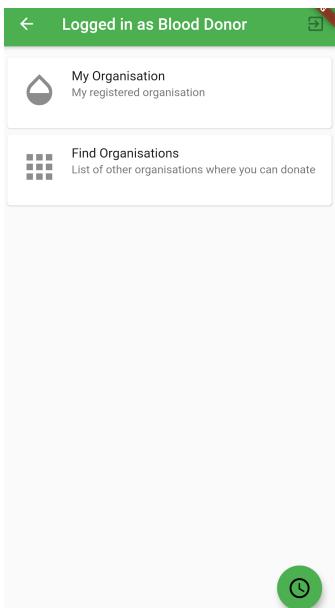
Report Death



Schedule



My Organisation

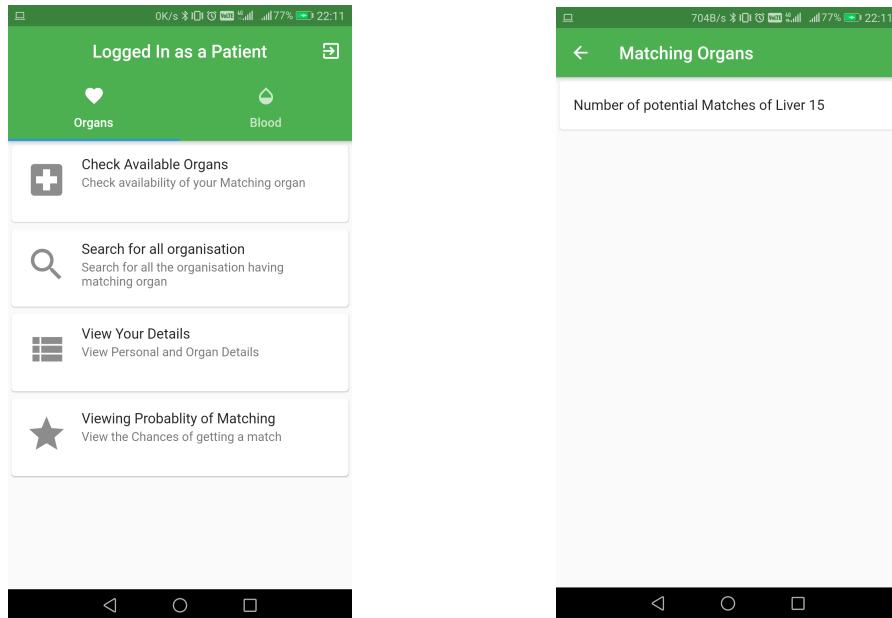


Blood Donor Page

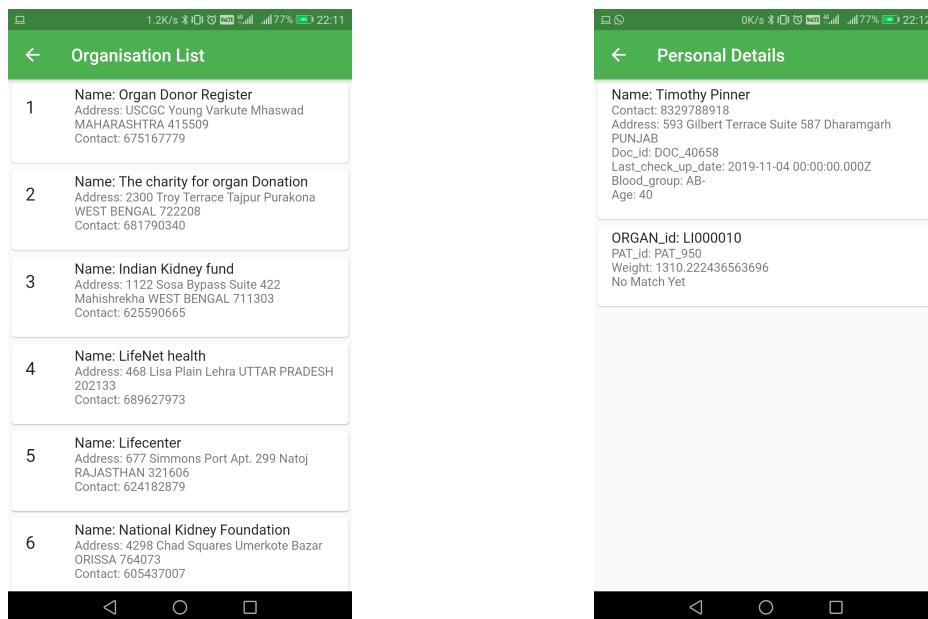
Patient Page

This Page consists of two Section:

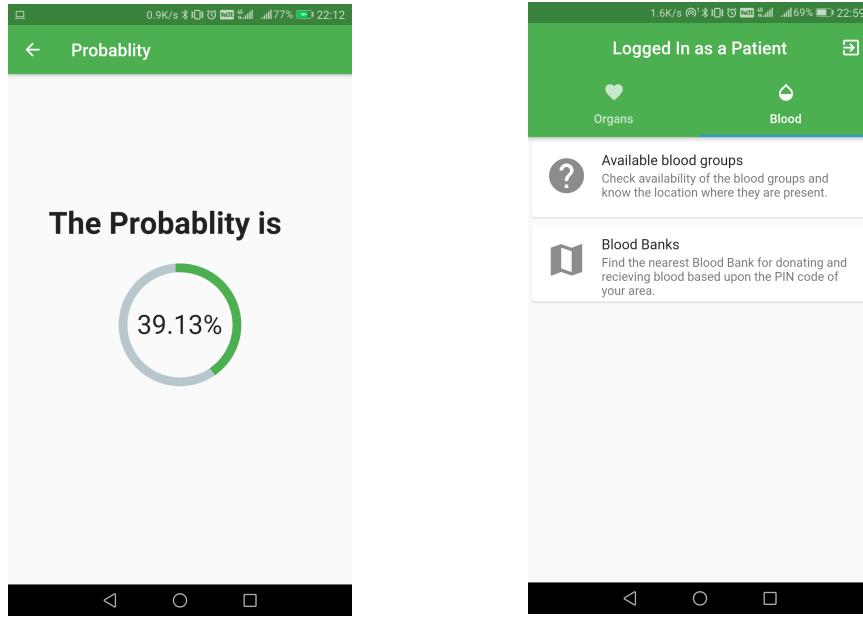
1) Organ-Page:



- a) The first option is viewing the number of potential matches a person can aaget.
- b) The second option shows a list of distinct organisations having these potential matches, this involves addresses, contact information and name of the organisation.

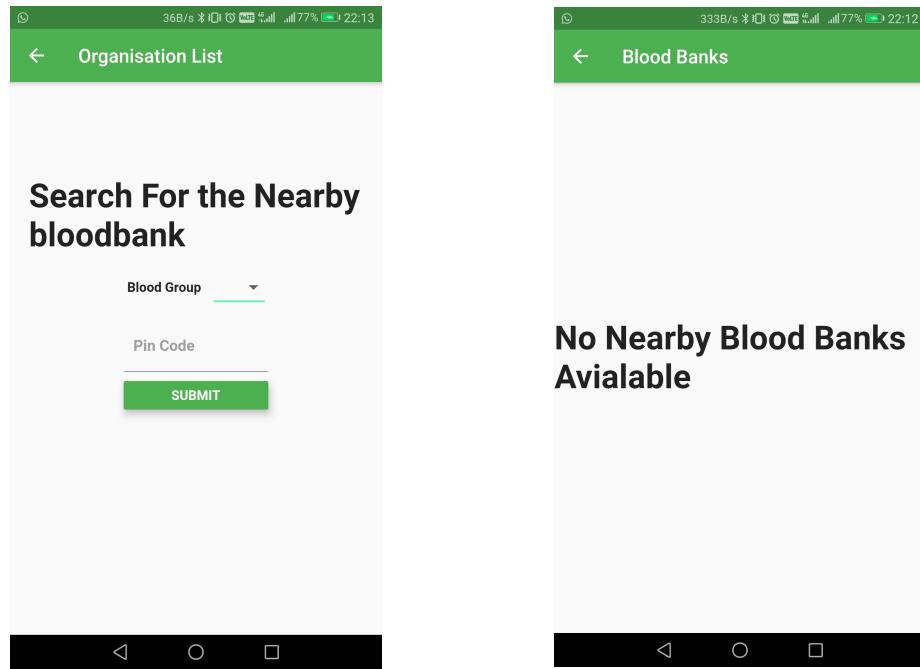


- c) The third option is to view the personal information along with the matched state, organ details, etc. if there is no match yet, it will display that, if there is a match, it will show the match id.
- d) Fourth Option is for Probability of matching, this considers all the possible matching of the Organs and find the approximate probability of matching.



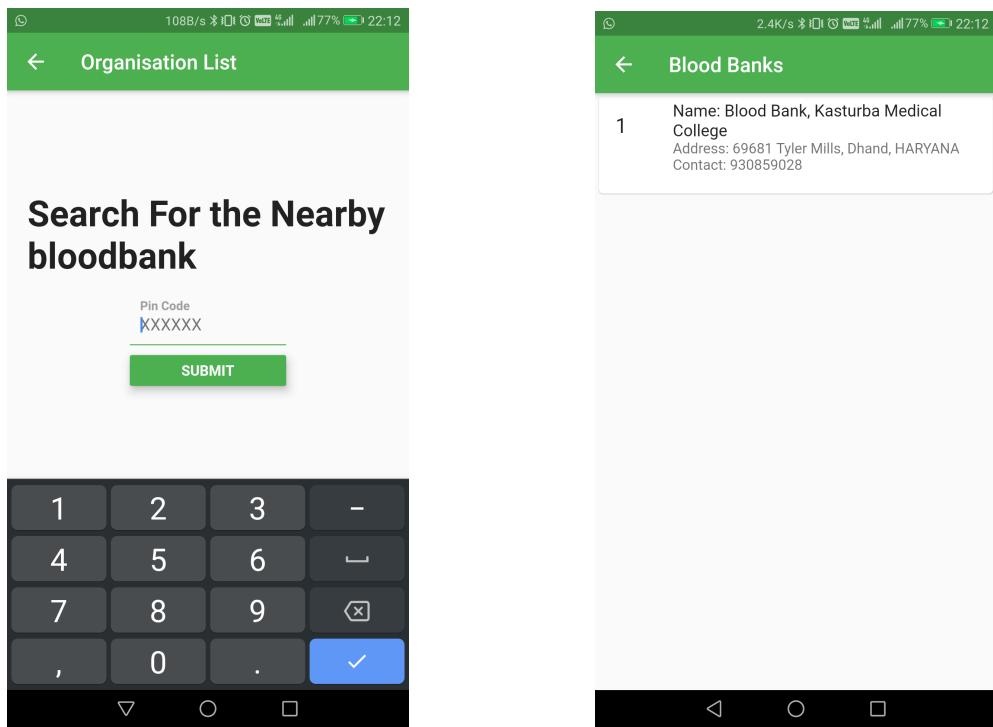
2) Blood-Page:

- a) The first option is to search the nearby blood banks with the particular



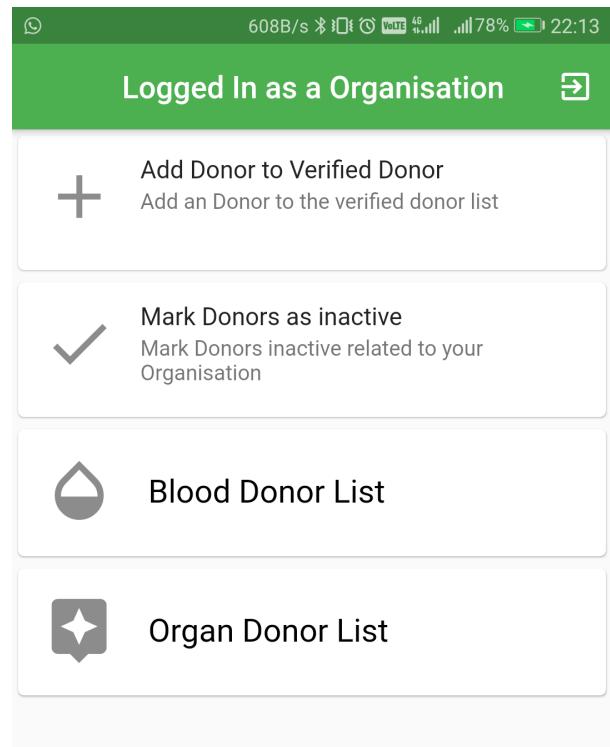
blood type, this uses pincode to find the nearby matching.

- b) The second option is to find the nearby blood banks without considering the number of blood units. In both the options if there is no blood banks available, it shows that.

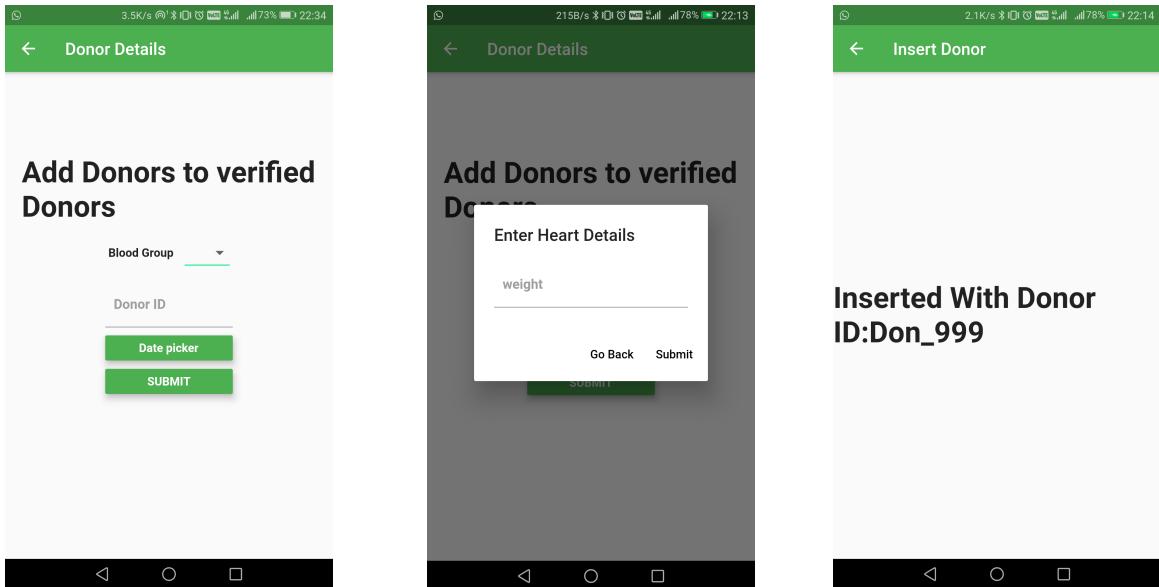


Organisation Page

This page is consist of four options:



- 1) First option is making new donors, as verified donors, User has to enter Donor_id, and other details, if based on organ other details are needed, they are asked in an dialog



box, then this function inserts the data into the Organ_donor table, deletes from new donor, and insert an entry into respective organ table.

- 2) Second option marks the donors, who are inactive, i.e, haven't donated in two months and then displays their list.

Inactive Donors	
1	Age: 45 DON_id: DON_101 Name: Thomas Bennington Address: 0563 Jake Manors Vanike PUNJAB Contact: 7291801309 ORG_id: ORG_971 Status: Dead Last_check_up_date: 2020-02-27 18:30:00.000Z Blood_Group: O+ Pincode: 143108 Active: 0
2	Age: 46 DON_id: DON_35 Name: Christa Mullins Address: 0828 Erika Club Apt. 887 Aurangabad Cantonment MAHARASHTRA Contact: 7193624183 ORG_id: ORG_971 Status: Alive Last_check_up_date: 2020-01-02 18:30:00.000Z Blood_Group: AB+ Pincode: 431002 Active: 0
3	Age: 43 DON_id: DON_769 Name: Mary Robertson Address: 04449 Carroll Views Anandapuram TAMILNADU Contact: 7088275243 ORG_id: ORG_971

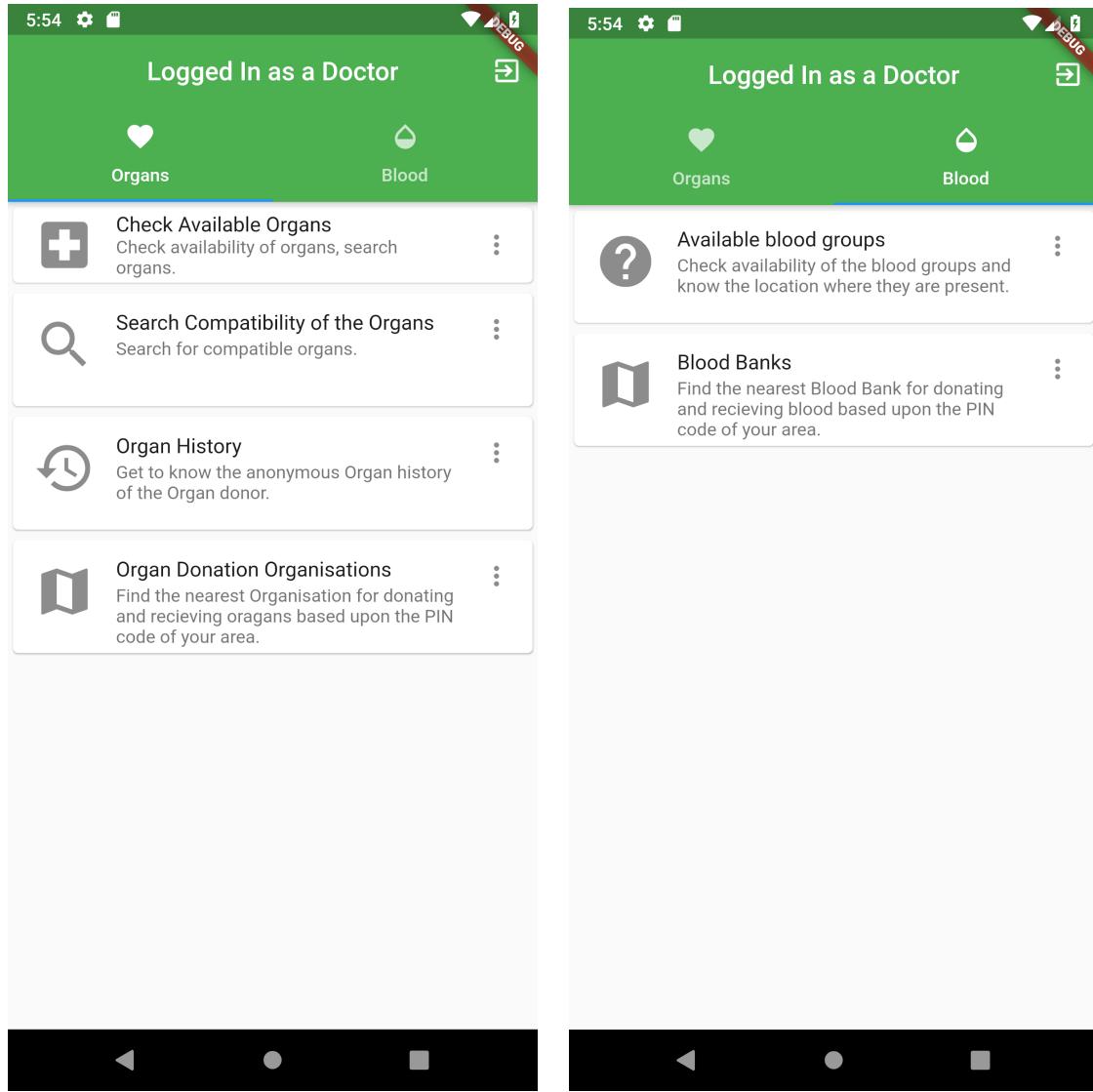
- 3) The third option is to view the Blood Donor list.
 This option can be clicked to see all the Blood Donors registered under this organization.

1	Name: Theresa Sheats Age: 21 Blood_Group: A- Contact: 7477553780 Address: 899 Karen Circle Apt. 784 Dhandwara UTTAR PRADESH, 202522
2	Name: David Gouveia Age: 61 Blood_Group: B- Contact: 7979239793 Address: 09151 Sherry Square Challavarigudem ANDHRA PRADESH, 534447
3	Name: Melvin Dendy Age: 31 Blood_Group: B+ Contact: 7300646745 Address: 6405 Gina Street Gunnotha MADHYA PRADESH, 464331
4	Name: Dorothy Lovisone Age: 34 Blood_Group: O+ Contact: 7371465258 Address: 99388 Campbell Viaduct Suite 331 Gudma MADHYA PRADESH, 481105
5	Name: Lionel Washington Age: 20 Blood_Group: O+ Contact: 7711859341 Address: 478 Kathy Vista Apt. 045 Sangewadi MAHARASHTRA, 413317

1	Name: Thomas Bennington Age: 45 Organ: Liver Contact: 7291801309 Address: 0563 Jake Manors Vanike PUNJAB, 143108
2	Name: Christa Mullins Age: 46 Organ: Kidney Contact: 7193624183 Address: 0828 Erika Club Apt. 887 Aurangabad Cantonment MAHARASHTRA, 431002
3	Name: Mary Robertson Age: 43 Organ: Kidney Contact: 7088275243 Address: 04449 Carroll Views Anandapuram TAMILNADU, 628701
4	Name: navneet123 Age: 23 Organ: Lung Contact: 8912388978 Address: bareilly, 231321
5	Name: manan Age: 23 Organ: Heart Contact: 9875677653 Address: Civil Lines,Delhi , 112021

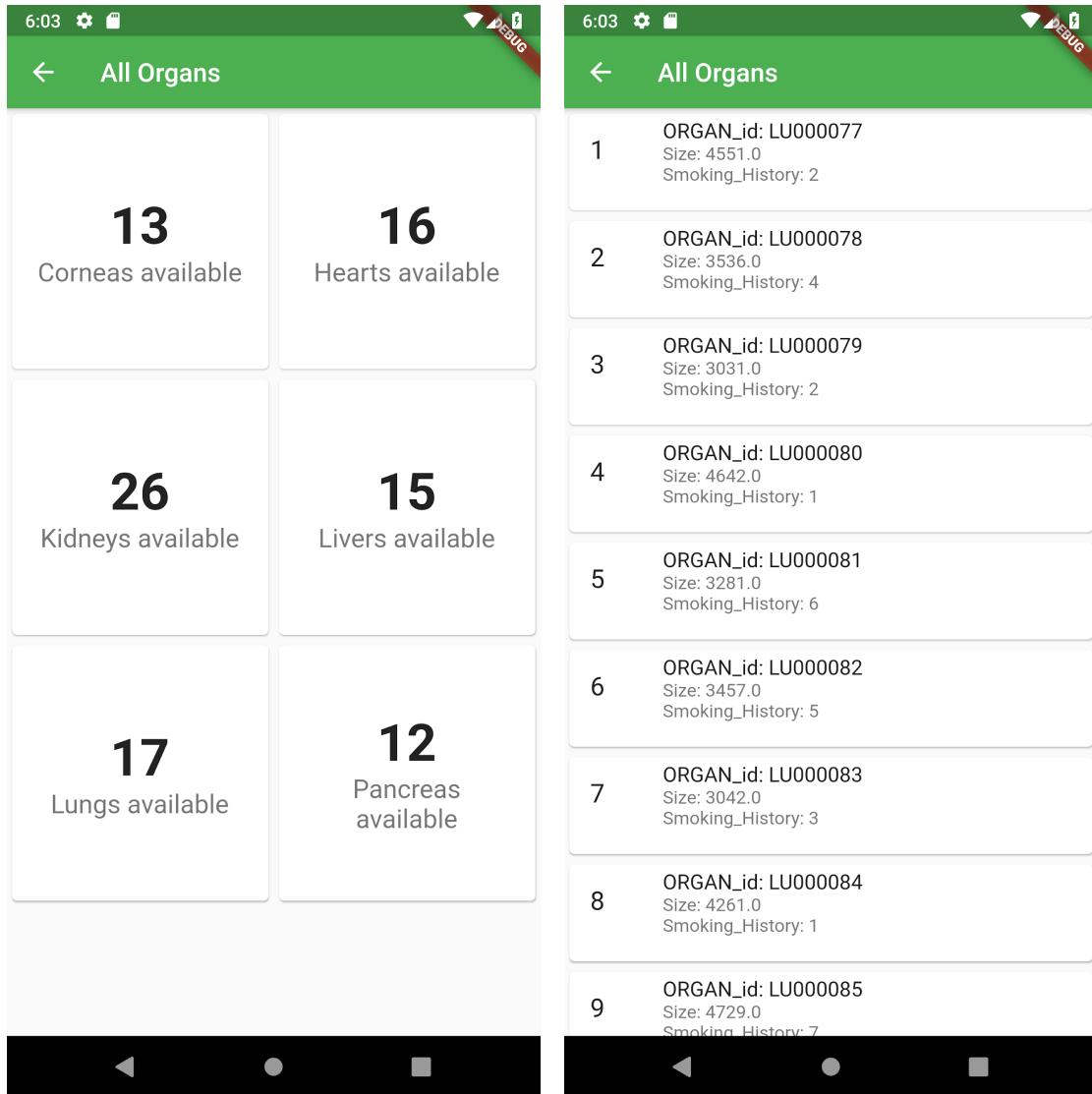
- 4) The third option is to view the Organ Donor list. This option can be clicked to see all the Organ Donors registered under this organization.

Doctor Page



This page is designed to provide optimal functionality to the Doctor. It provides many high lever functionalities which are not given to Patients or Donors. This page is divided into two tabs, one for the functions related to Organs and the other for functions related to blood. The features of the Doctor Page are explained Below.

Check Availability of the organs



This page shows all the available organs in all the Organisations and by clicking on the tile of any organ the user can go to the page showing the relevant matching details of each and every organ of that type.

Search for Compatible Organs

The screenshots show a mobile application interface for searching compatible organs. The first two screens are identical, showing a search form with dropdown menus for organ type and search parameters. The third screen shows the results of the search.

Screenshot 1 (6:35): The screen title is "Search Compatible Organs". It displays a dropdown menu with the following options: "Select Organ to search in" (Cornea, Heart, Kidney, Liver, Lung, Pancreas), "Enter minimum Size of Lungs" (3000), "Enter maximum Size of Lungs" (5000), and "Enter maximum smoking years" (4). A "Submit" button is at the bottom.

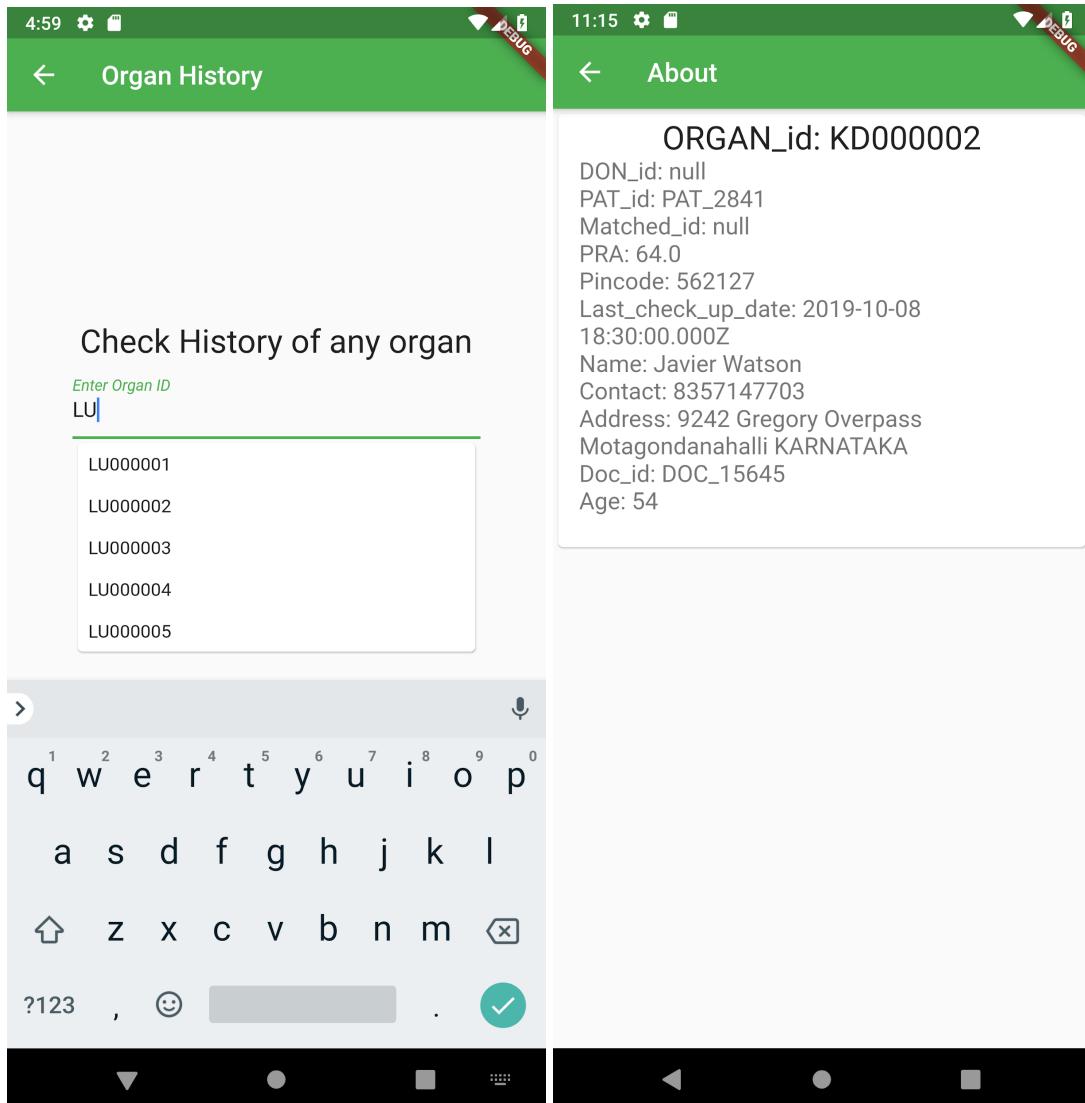
Screenshot 2 (6:36): Similar to Screenshot 1, but the "Select Organ to search in" dropdown is now set to "Lung".

Screenshot 3 (6:37): The screen title is "All Organs". It lists 9 results, each with an ID, size, and smoking history:

Rank	Organ ID	Size	Smoking History
1	ORGAN_id: LU000077	4551.0	2
2	ORGAN_id: LU000079	3031.0	2
3	ORGAN_id: LU000080	4642.0	1
4	ORGAN_id: LU000083	3042.0	3
5	ORGAN_id: LU000084	4261.0	1
6	ORGAN_id: LU000086	4825.0	0
7	ORGAN_id: LU000089	4970.0	2
8	ORGAN_id: LU000090	4484.0	0
9	ORGAN_id: LU000092	3811.0	2

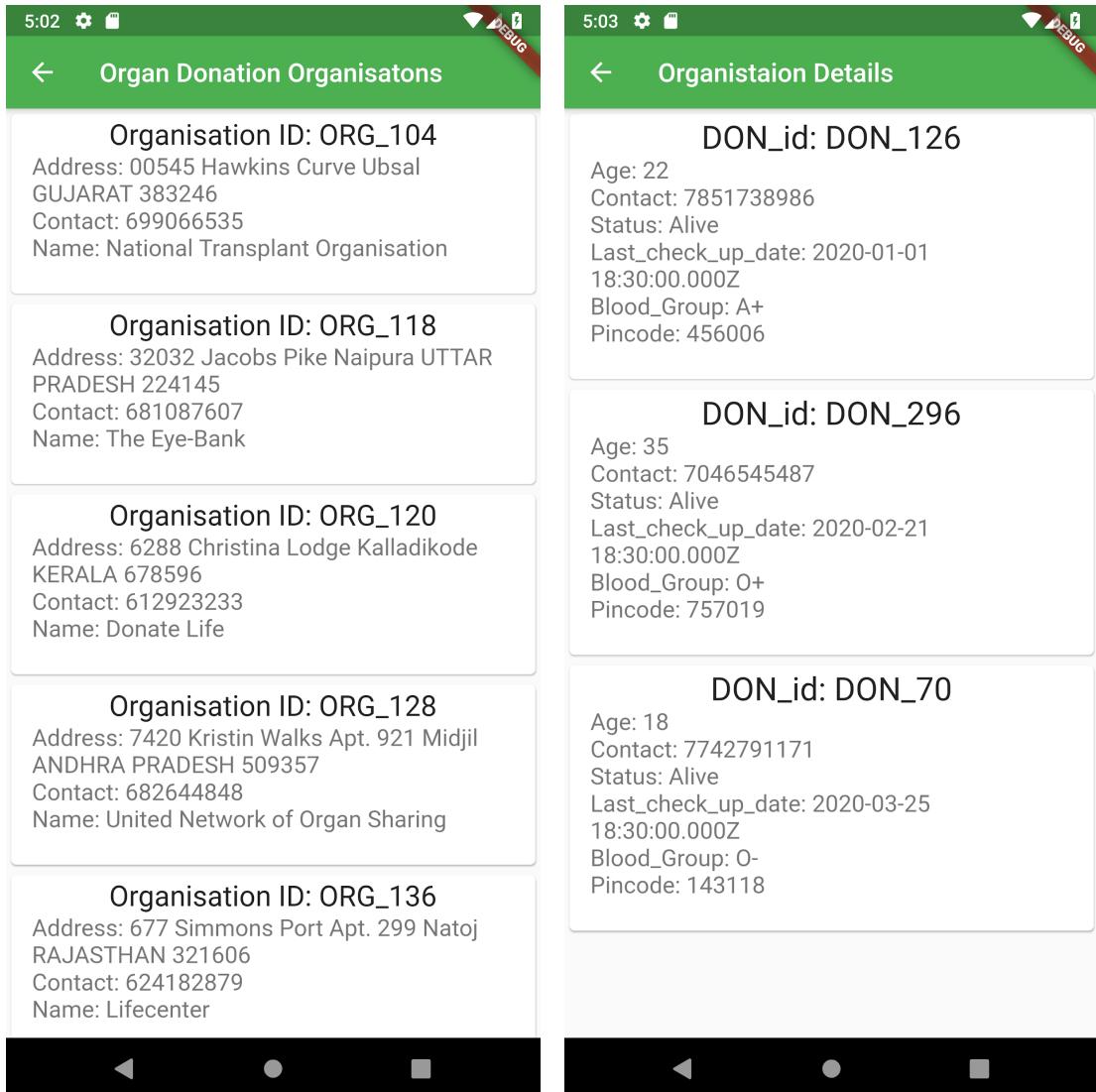
This option helps to search for compatible organs with selected properties and display all of them.

Organ History



Search the details of any organ and donor/patient with the help of Organ ID. It also gives recommendations as the User types.

Organ Donation Organisations



It will show all the Organisations present in the database along with their details such as Address, Contact number, Name and Organisation ID. Tapping on any organisation opens the list of Donors registered with the organisations without disclosing their personal details such as Name and Address.

Available Blood Groups

The option is to search the nearby blood banks with the particular blood type, this uses pincode to find nearby matching.

The image displays two screenshots of a mobile application interface. The left screenshot shows the search interface with fields for Blood Group and Pin Code, and a green SUBMIT button. The right screenshot shows a list of blood banks with details for each.

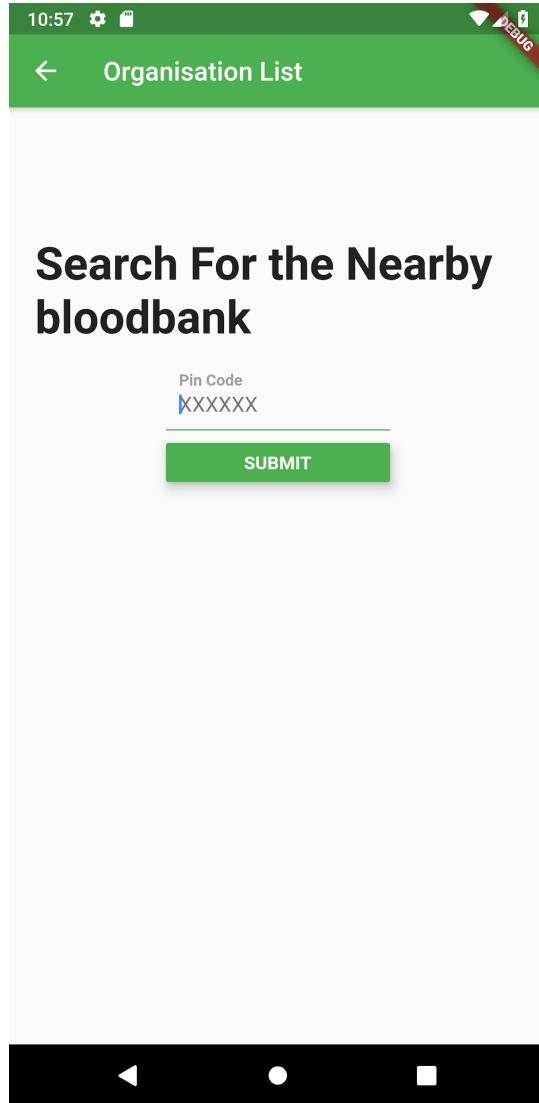
Left Screen (Search Interface):

- 10:43 | DEBUG
- ← Organisation List
- Search For the Nearby bloodbank
- Blood Group: [dropdown menu]
- Pin Code: [text input field]
- SUBMIT

Right Screen (List of Blood Banks):

- 1:37 | DEBUG
- ← Blood Banks
- 1 Name: Blood Bank, Shri Chitra Triun
Institute of Medical Sciences & Technology
Address: 95006 Raven Fork Suite 093, Chiksana,
RAJASTHAN
Contact: 959812485
- 2 Name: Blood Bank, KC General Hospital
Address: 726 Johnson Ways, Kherli Gadasia,
RAJASTHAN
Contact: 999582658

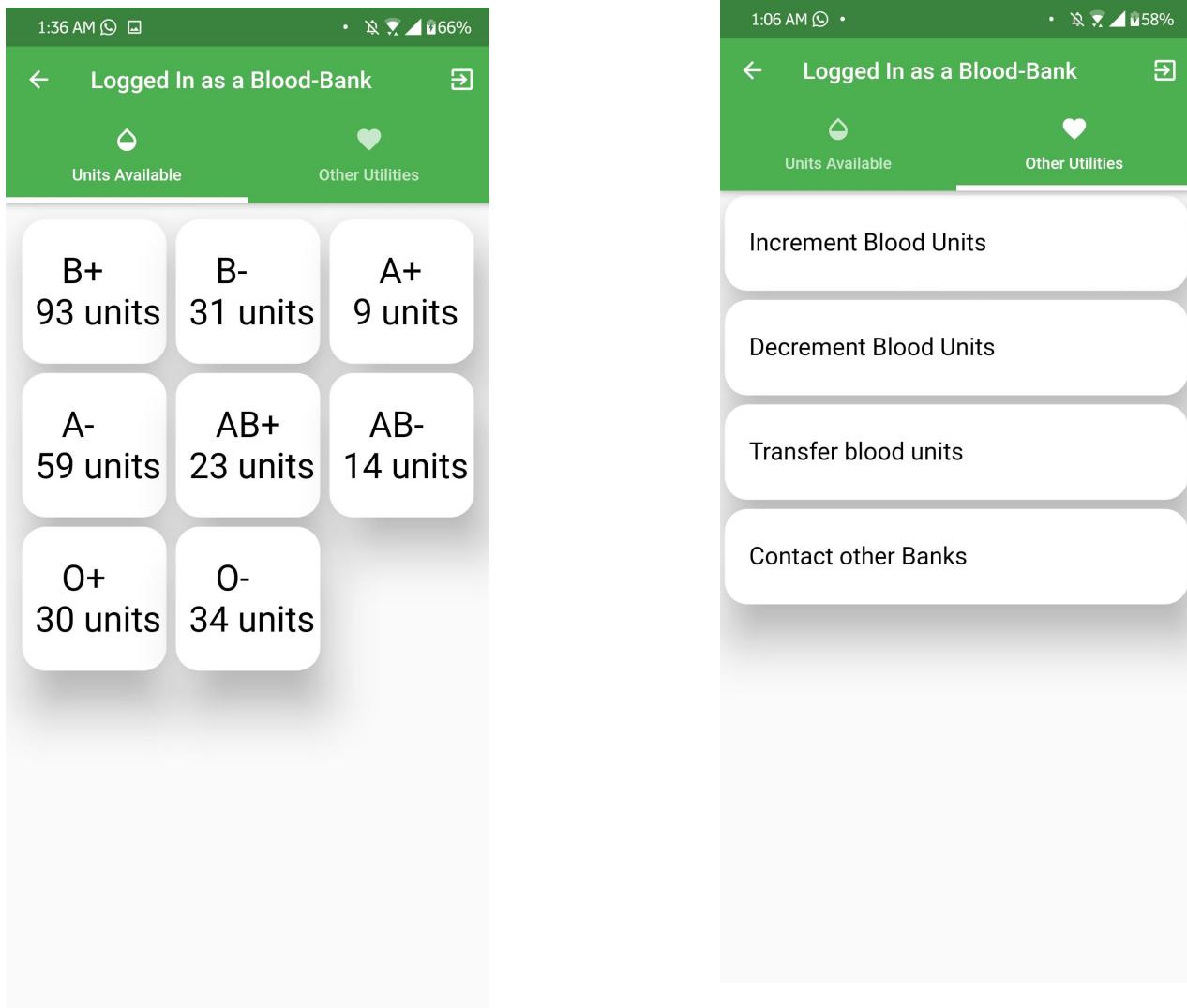
Blood Banks



This page helps the doctor to locate all the blood banks in his/her area using the pincode since the first 2 numbers of a pin code refers to a unique area and subarea.

In all the pages condition checking has been done to ensure that no invalid query is sent to SQL backend.

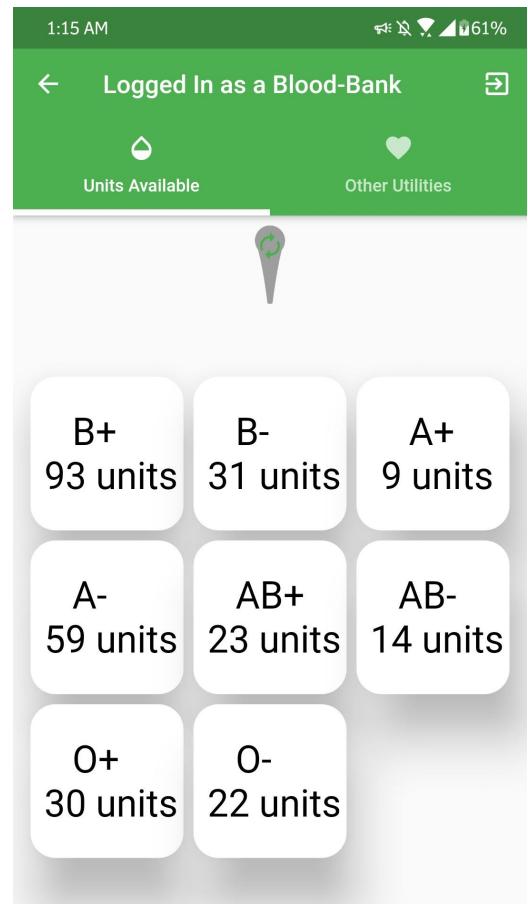
Blood Bank Page



These are the two tabs that come under the blood bank page. Live view of the units available and some utilities that the blood bank can perform.

The live view can be refreshed by swiping the screen down and releasing it. You will get something as shown on the right. It is just like the refresh feature on Instagram, twitter etc

Once the refresh is complete ,the units will get updated and you will be able to see the new values.

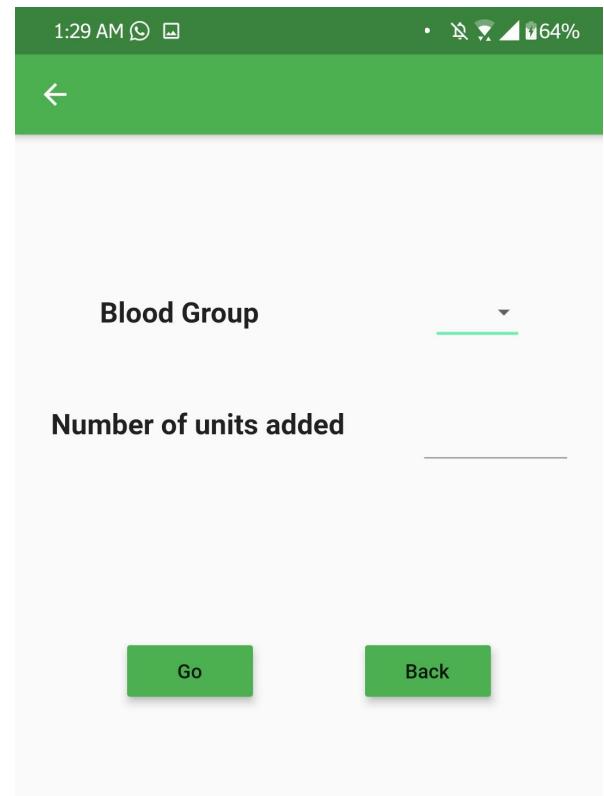


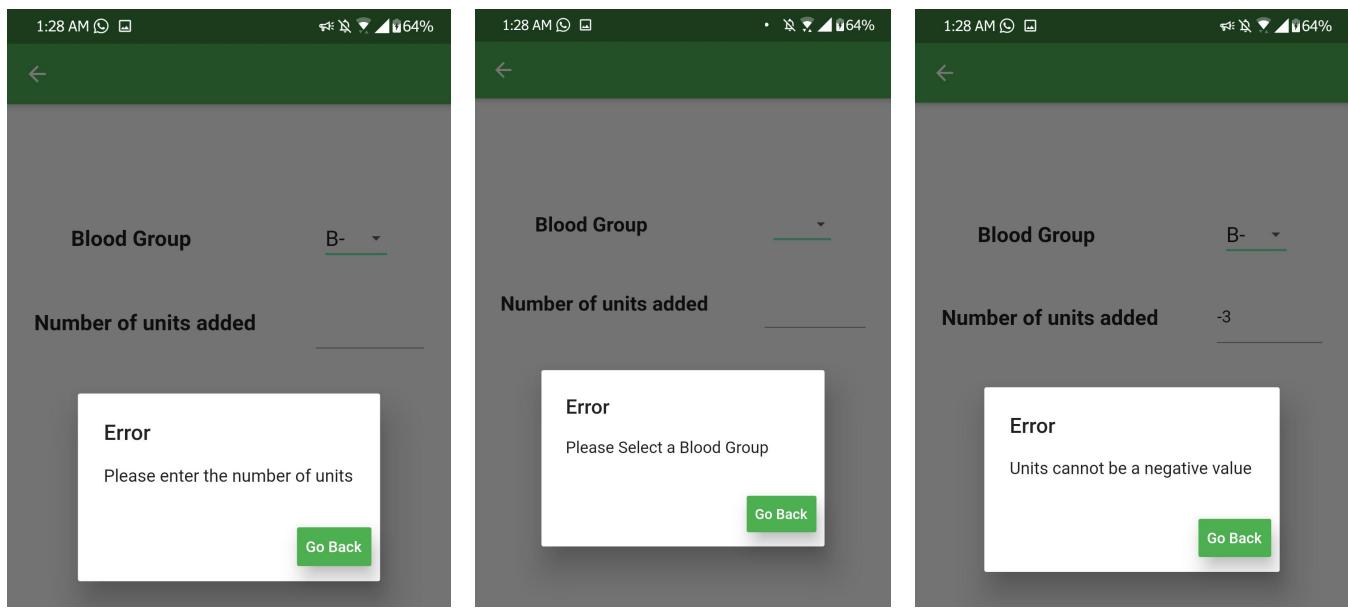
The other utilities include:-

- 1) **Increment of Blood units**:- This option can be used by blood banks when newly donated blood comes to be stored, so they need to update the units of blood of particular blood groups.

This also gives you pop ups for entering negative values or some input that cant be accepted by the query.

Some of the pop ups are shown below:-





If the entries are correct the units will get updated in the live view also.

The live view shows the successful addition of units:

Success !!
Added 12 Units of blood to Onegative

Blood Group: B-
Number of units added: 12

Available Units:

Blood Group	Units Available
B+	93 units
B-	43 units
A+	9 units
A-	59 units
AB+	23 units
AB-	14 units
O+	30 units
O-	34 units

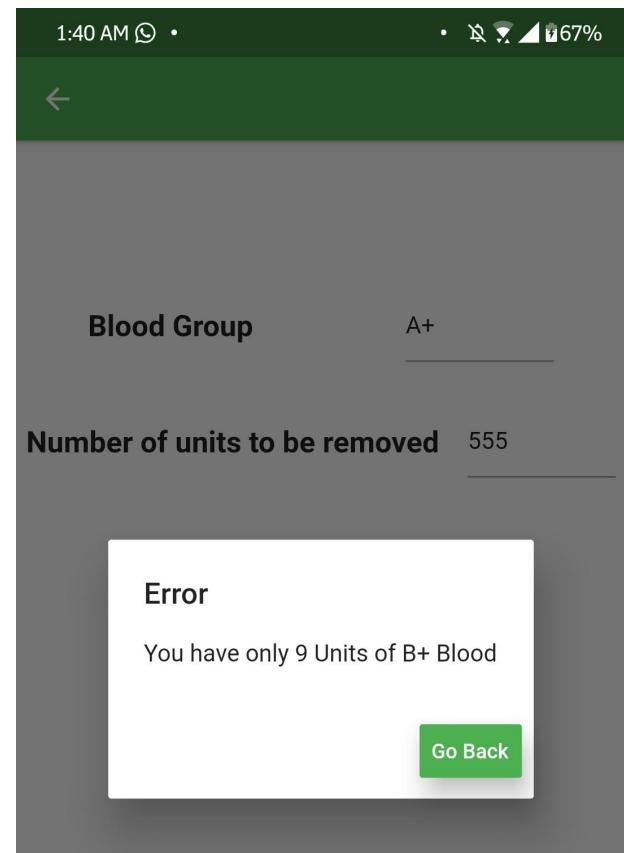
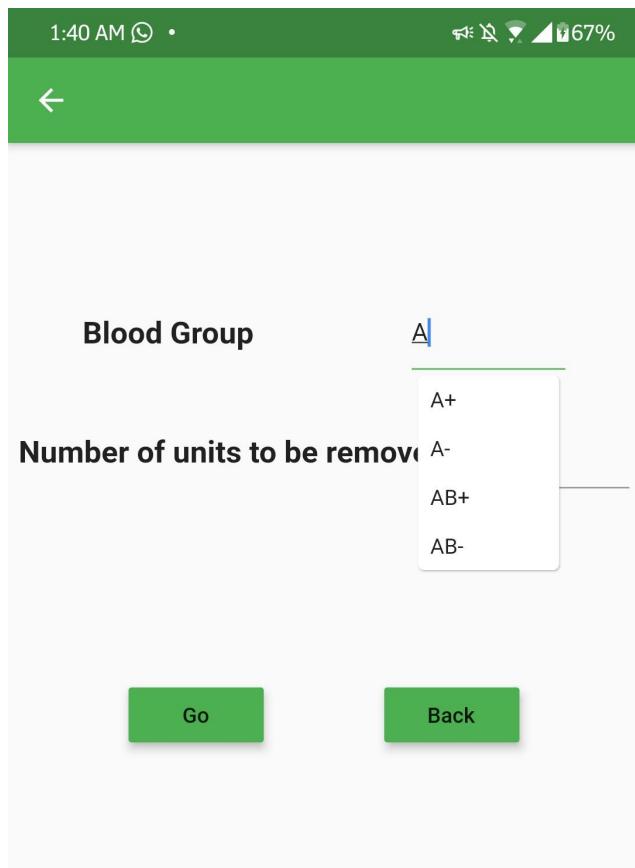
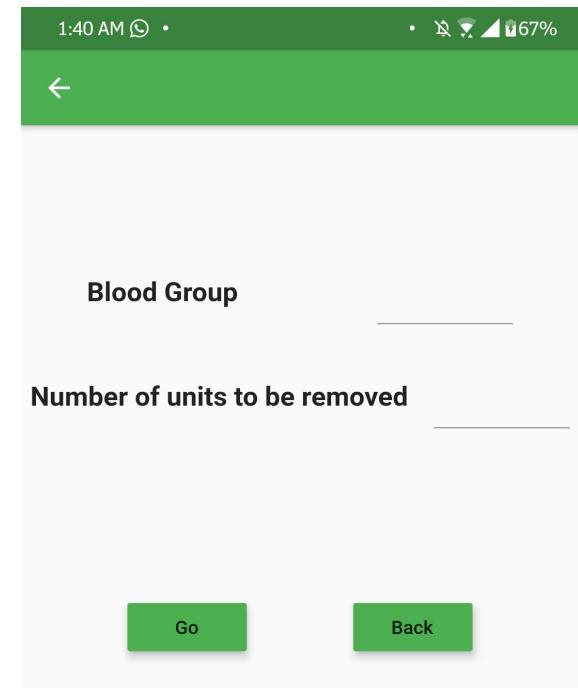
Buttons at the bottom right: Go and Back.

2) Decrement of Blood units

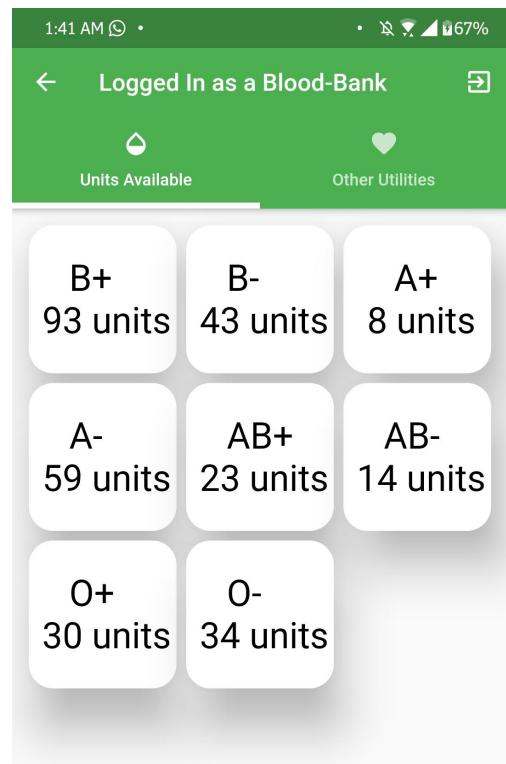
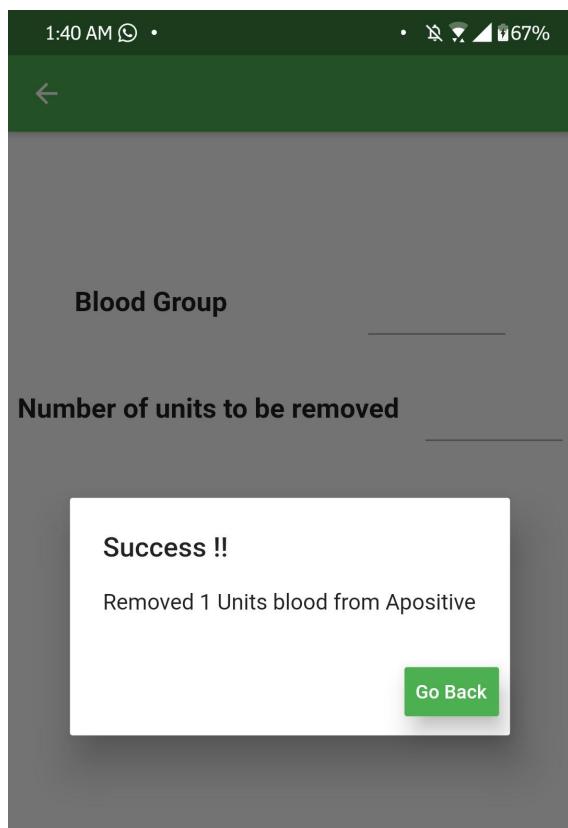
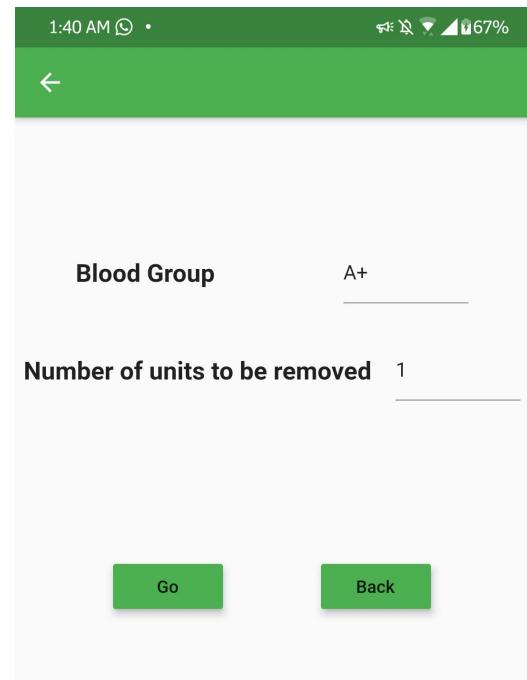
This option is used to reduce the particular blood groups in case they are transferred to any hospital or are consumed by the patients.

Here you get pop ups from entering wrong values and you also get suggestions while typing the blood groups.

Apart from this you get an error if you enter the decrement value more than the available units.

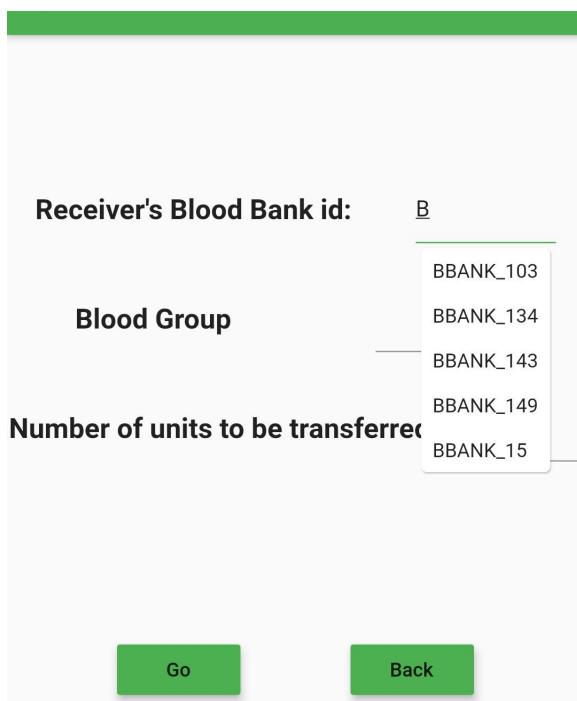
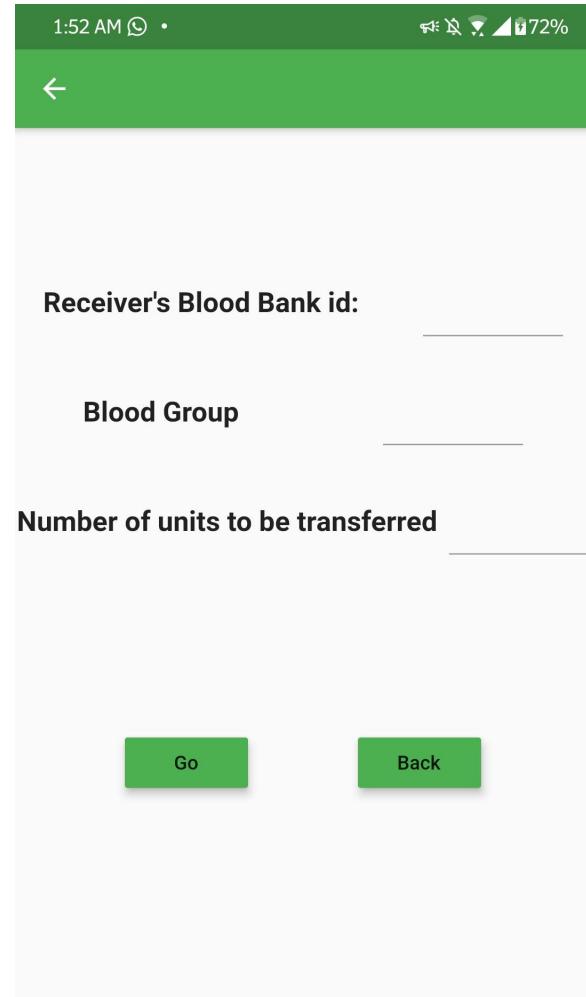


If the entries are correct as on the right.
Then you get a message showing that the units
have been reduced and you can check them in
the live view too.



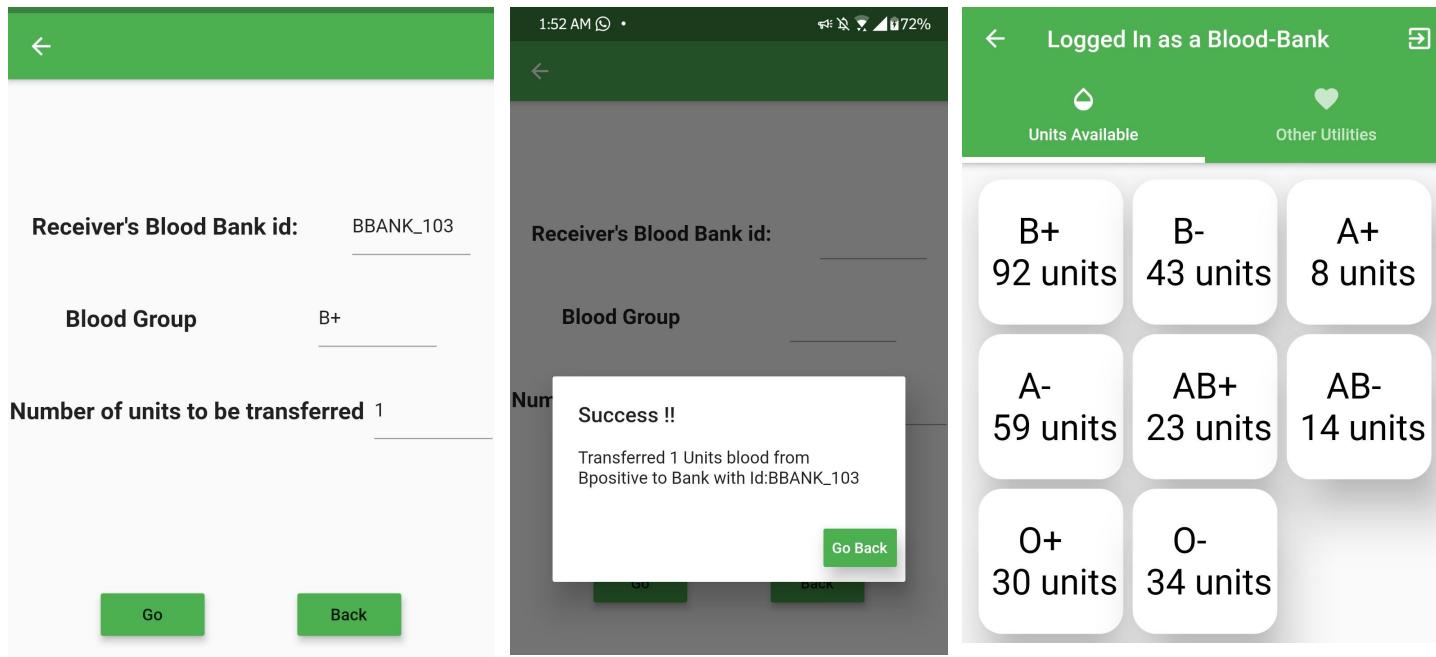
- 3) **Transfer Blood Units:** This option can be used to transfer blood units to other blood bank.

Apart from the regular popups like input in correct format ,max. Number of units and suggestion of blood groups.



We also get suggestion for the receiver's Blood bank id.Suggestion screenshot is shown below

If the input is correct units will be transferred successfully.
Increment units at receiver's end and decrement at sender's end.



4) Contact other Banks:

This shows you the list of all other blood banks that are there, so that you can contact them.

It's a scrollable list that can be scrolled down to see all the Blood_Banks out there.

- 2:11 AM
- 1 Name: Blood Bank, District Hospital
Contact: 937478526
Address: 3494 White Manor Suite 070, Bydanur, KARNATAKA, 561202
 - 2 Name: Blood Bank, Asilo Hospital
Contact: 902358539
Address: 155 Murphy Pike Apt. 795, Pondugula, ANDHRA PRADESH, 522414
 - 3 Name: Blood Bank, Deptt. of Transfusion Medicine
Contact: 904467696
Address: PSC 1701, Box 8051, Kurshi Barawa, BIHAR, 845307
 - 4 Name: Blood Bank, District Hospital
Contact: 909955677
Address: 600 Rivera Mews Suite 875, Lingamguntlapalem, ANDHRA PRADESH, 522212
 - 5 Name: Blood Bank, Kasturba Medical College
Contact: 930859028
Address: 69681 Tyler Mills, Dhand, HARYANA, 125050
 - 6 Name: Blood Bank, HSIS Hospital
Contact: 991405200
Address: Unit 4753 Box 7873, Mallepalli, ANDHRA PRADESH, 515541

Normalization

Before we start the normalization we brainstormed our idea more and came up with some new tables that will help us to implement the front end. So all the tables before Normalization are shown below :-

Doctor: Holds record of all the Doctor

Doctor (Doc_ID, Name, Contact, Speciality);

Patient/Receiver: Holds record of all the Patient

Patient (Pincode, Last_check_up_date, PAT_id, Name , Contact , Address , Doc_id, Organ, Age , Blood_Group)

Organ_Donor: Holds record of all the Organ Donors

Organ_Donor (Age , DON_id , Name, Address, Contact , ORG_id, Status , Last_check_up_date, Blood_Group, Pincode ,Organ,Active);

Blood Donor: Holds record of all the Blood_Donors

Blood_Donor (Age , BLD_id , Name , Address , ORG_id , Blood_Group , Last_Donated , Last_Checkup , Pincode ,Contact)

Organisation: Holds record of all the Organisation

Organisation (ORG_id , Name , Address , Contact ,Organs_accepted ,pincode);

Kidney: Holds record of available and required Kidney

Kidney (ORGAN_id, DON_id , PAT_id , Blood_Group, Matched_id , PRA)

Pancreas: Holds record of available and required Pancreas

Pancreas (ORGAN_id, DON_id, PAT_id , Blood_Group , Matched_id, OPTN)

Heart: Holds record of available and required Heart

Heart (ORGAN_id , DON_id , PAT_id , Blood_Group , Matched_id , Weight)

Liver: Holds record of available and required Liver.

Liver (ORGAN_id, DON_id , PAT_id, Blood_Group, Matched_id , Weight);

Lung: Holds record of available and required Lung.

Lung (ORGAN_id, DON_id , PAT_id , Blood_Group , Matched_id , Age, Size, Smoking_History)

Cornea: Holds record of available and required Cornea.

Cornea (ORGAN_id , DON_id , PAT_id , Blood_Group , Matched_id , Procurement_Timestamp)

Blood_Bank: Holds record of all the Blood Banks

Blood_Bank (BANK_id , Name , Apositive,Anegative,Bpositive,Bnegative,Opositive,Onegative , ABpositive , ABnegative , Address , Contact, pincode)

New Donor: Holds record of people who registered to be organ donors.

New_Donor (Age, DON_id, Name, Address, Contact, Blood_Group, Pincode, Organ)

New Blood Donor: Holds record of people who registered to be blood donors.

New_Blood_Donor (Age, BLD_id, Name, Address, Blood_Group , Pincode, Contact)

Accounts: Holds the login details of different users

Account(id , password , category)

1NF form :-

Organization table has Organs_accepted which contains multiple values.
So to reduce this we add one more table for the mapping of an organ to ORG_ID.
The changes are as follows :-

Organisation

Organisation (ORG_id , Name , Address , Contact , pincode);

Organisation organ mapping table

Organisation (ORG_id ,Organ)

Already in 2NF form :-

3NF form :-

We can remove Blood group from all the organs table.

(If we removing this does not increase the query processing time)

The following changes will be there :-

Kidney

Kidney (ORGAN_id, DON_id , PAT_id , Matched_id , PRA)

Pancreas

Pancreas (ORGAN_id, DON_id, PAT_id , Matched_id, OPTN)

Heart

Heart (ORGAN_id , DON_id , PAT_id , Matched_id , Weight)

Liver

Liver (ORGAN_id, DON_id , PAT_id, Matched_id , Weight);

Lung

Lung (ORGAN_id, DON_id , PAT_id , Matched_id , Age, Size, Smoking_History)

Cornea

Cornea (ORGAN_id , DON_id , PAT_id , Matched_id , Procurement_Timestamp)

We can remove Organ name from all the Patient table and the Donors table.

(If we removing this does not increase the query processing time)

Patient/Receiver

Patient (Pincode, Last_check_up_date, PAT_id, Name , Contact , Address , Doc_id, Age , Blood_Group)

Organ_Donor

Organ_Donor (Age , DON_id , Name, Address, Contact , ORG_id, Status , Last_check_up_date, Blood_Group, Pincode,Active);

Patients_Organ:

Organ_patid_mapping(PAT_id,Organ)

Donors_Organ:

Organ_donid_mapping(DON_id,Organ)

BCNF form :-

Already in BCNF form

Indexing

The primary and foreign keys are indexes by default in MySQL server. Therefore, there is no need to create separate indexes for them. 60% of functions are on primary and foreign keys, which are by default indexes. Other than them we need blood groups and Pin Code for Compatibility, availability and nearest available, therefore we have created indexes on pin code and blood group.

Account:

This table holds the information for the account login details, the primary key is already a index, we can create a secondary index on the password, to speed up the process of password searching.

```
create index password_index on Account(password);
```

Blood_Bank:

This particular table, holds details about the blood_bank address and available blood units, name. Creating index on Pincode will speed up the process of searching the nearby blood_banks. Also the blood_units are frequently used and hence creating a indexes on them will also be useful.

```
create index pincode_index on Blood_Bank(pincode);
create index Apos_index on Blood_Bank(Apositive);
create index Aneg_index on Blood_Bank(Anegative);
create index Bneg_index on Blood_Bank(Bnegative);
create index Bpos_index on Blood_Bank(Bpositive);
create index ABpos_index on Blood_Bank(ABpositive);
create index ABneg_index on Blood_Bank(ABnegative);
create index Oneg_index on Blood_Bank(Onegative);
create index Opos_index on Blood_Bank(Opositive);
```

Blood_Donor:

The blood donor, will store the details of the blood Donors, making Pincode as indexes will fast up the nearest donor searching process. Blood Group are also needed repeatedly to check the availability of a particular blood_group.

```
create index pincode_index on Blood_Donor(pincode);
create index blood_index on Blood_Donor(blood_group);
create index Donated_index on Blood_donor>Last_donated);
```

New_donor:

The pincode and blood_group can be indexed.

```
create index blood_index on New_Donor(blood_group);  
create index pincode_index on New_Donor(pincode);
```

New_Blood_Donor:

The pincode and blood_group can be indexed.

```
create index blood_index on New_Blood_Donor(blood_group);  
create index pincode_index on New_Blood_Donor(pincode);
```

Organisation:

The pincode can be indexed;

```
create index Pincode_index on Organisation(Pincode);
```

Organ_Donor:

The pincode and blood_group can be indexed as blood_group is an essential criteria for matching and pincode as collection has to take place.

```
create index blood_index on Organ_Donor(blood_group);  
create index pincode_index on Organ_Donor(pincode);
```

Patient:

```
create index blood_index on Patient(blood_group);  
create index pincode_index on Patient(pincode);
```

Cornea,Doctor,Donors_organ, Kidney, Liver, Lung,

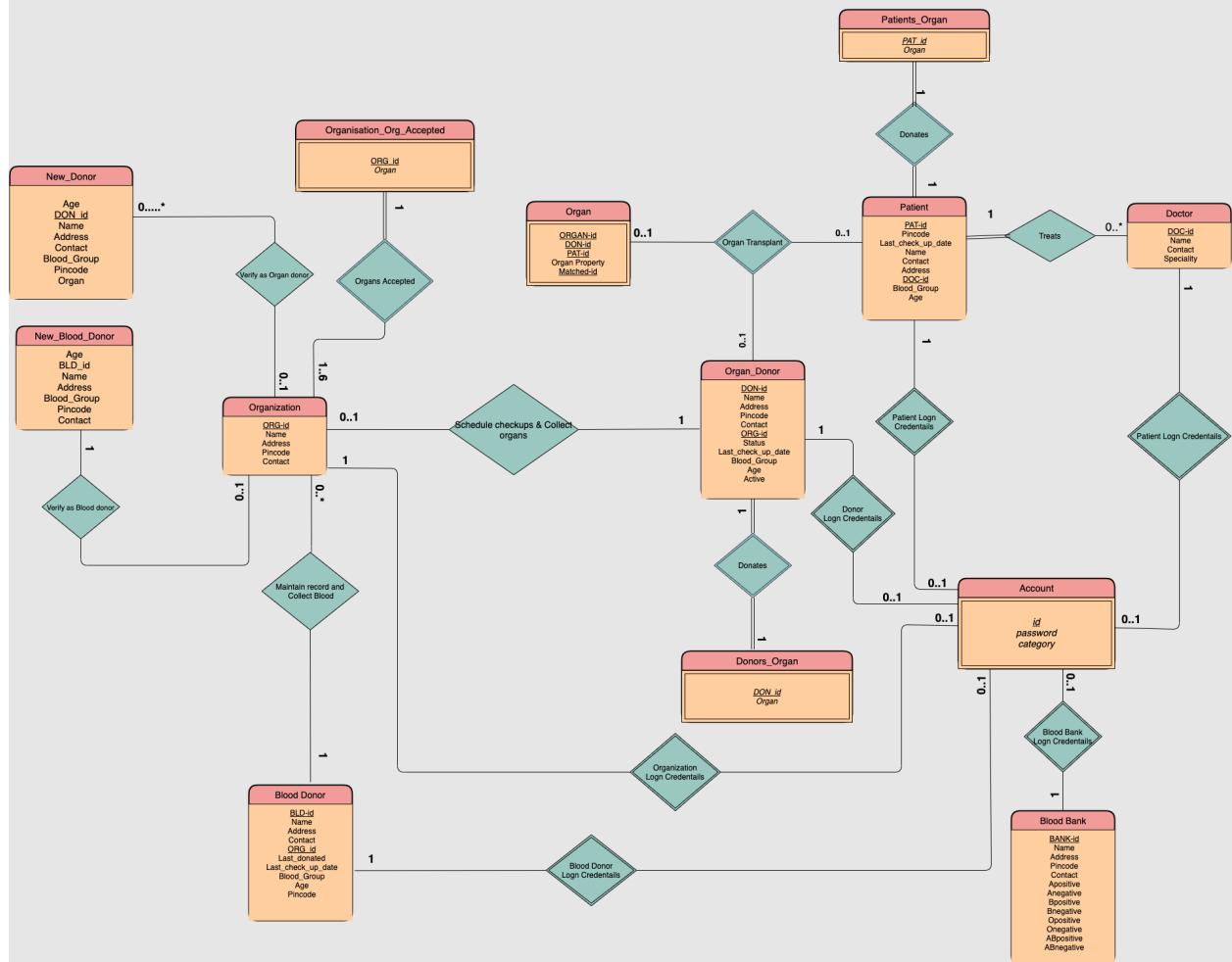
Organisation_Org_Accepted ,Pancreas,Patients_Organ:

All the needed attributes are already indexes,i.e, they are primary or foreign keys.

Entity Relation Diagram

OBDS

Entity Relation Diagram



Link To ER Diagram - <https://drive.google.com/open?id=1HVxuGsv6m8parFk1-DmWvjwzMscH7A74>

Relational Algebraic Queries:

1. Displaying all the banks with particular blood type equal to zero:

$$\prod_{Bank_id, name} \left(\sigma_{apositive=0}(Blood_Bank) \right)$$

=> **SELECT Bank_id, name FROM Blood_Bank WHERE apositive =0;**

2. Check all patients who have not got a match of a particular Organ. Here kidney can be replaced by any organ, i.e, Lung, Liver, Cornea, Heart, Pancreas.

$$\prod_{Pat_id, name, Organ_id} \left(\sigma_{Kidney.pat_id=Patient.pat_id \wedge Matched_id=null}(Kidney \times Patient) \right)$$

=> **SELECT pat_id, name, organ_id FROM Kidney join Patient using(pat_id) WHERE matched_id is not null;**

3. Display details of all patient/donor:

$$\sigma_{Pat_id='Pat_982'}(Patient)$$

$$\sigma_{Don_id='Don_996'}(Organ_Donor)$$

=> **SELECT * FROM Patient; => SELECT * FROM Organ_Donor;**

4. show blood bank with particular blood type available in an area

$$\prod_{name, Address, Apositive} \left(\sigma_{apositive>0 \wedge pincode LIKE('38%')}(Blood_Bank) \right)$$

=> **SELECT name, address, apositive FROM Blood_Bank WHERE apositive > 0 and pincode like('38%');**

5. Display information of kidney with PRA greater then given value, here Kidney can be replaced by liver or pancreas or Heart, with there respective attributes.

$$\sigma_{pra>14}(Kidney)$$

=> **SELECT * FROM Kidney WHERE pra > 14;**

6. Display information for patient with smoking history less then k.

$$\prod_{Patient.*} \left(\sigma_{smoking_history<15 \wedge Patient.pat_id=Lung.pat_id}(Lung \times Patient) \right)$$

=> **SELECT p.* FROM patient p join Lung using(pat_id) WHERE smoking_history < 15;**

7. Display information of all the donors, who want to donate liver.

$$\sigma_{Organ_donor.Don_id=Liver.Don_id}(Liver \times Organ_Donor)$$

=> **SELECT * FROM Organ_Donor join Liver using (don_id);**

8. View Patient's doctor information.

$$\prod_{Doctor.*} \left(\sigma_{Patient.Pat_id='Pat_982' \wedge Patient.Doc_id=Doctor.Doc_id}(Doctor \times Patient) \right)$$

=> **SELECT d.* FROM Doctor d join Patient using(doc_id) WHERE pat_id='pat_982';**

9. View status of Patient organ request, Here Kidney can be replaced by any organ.

$$\prod_{Matched_id} \left(\sigma_{Pat_id='Pat_3307'}(Kidney) \right)$$

=> **SELECT Matched_id FROM Kidney WHERE pat_id='pat_3307';**

10. View details of all the Doctor's Patients.

$$\prod_{Patient.*} \left(\sigma_{Doctor.Doc_id='Doc_42727' \wedge Patient.Doc_id=Doctor.Doc_id}(Doctor \times Patient) \right)$$

=> **SELECT p.* FROM Patient p join Doctor using(doc_id) WHERE doc_id='doc_42727';**

11. View the Blood_donors who haven't donated in 2 months.

$$\sigma_{datediff(curdate(), Last_donated) > 60}(Blood_Donor)$$

=> **SELECT * FROM Blood_Donor WHERE datediff(curdate(), last_donated) > 60;**

12. Find all the Organisations/Blood_bank in your area.

$$\sigma_{pincode LIKE ('62%')}(Organisation)$$

=> **SELECT * FROM Organisation WHERE pincode like "62%";**

13. Find all the Organ Donors Name,Contact,DON_id who donate Kidney FROM 'ORG_951' and Heart FROM ORG_852.

$$\Pi_{Name, Contact, DON_id, Organ, ORG_id} (\sigma_{ORG_id="ORG_951"} \wedge Organ="Kidney" (Organ_Donor X Donors_Organ)) \cup$$

$$\Pi_{Name, Contact, DON_id, Organ, ORG_id} (\sigma_{ORG_id="ORG_854"} \wedge Organ="Heart" (Organ_Donor X Donors_Organ))$$

=> SELECT Name,Contact,DON_id,Organ,ORG_id FROM Organ_Donor join Donors_Organ
using(DON_id) WHERE (ORG_id="ORG_951" and Organ="Kidney") or
(ORG_id="ORG_854" and Organ="Heart");

14. Find the details of all Donors and name of the Organization to which they are linked.

$$\rho_A(\text{Organ_Donor}) \quad \rho_B(\text{Organization})$$

$$\Pi_{A.*, B.Name} (\sigma_{A.ORG_id=B.ORG_id}(AXB))$$

=> select a.* , b.Name from Organ_Donor as a join Organisation as b using(ORG_id);

15. Find the Names of All the donor and patient whose kidney matched. Here kidney can be replaced by any organ, i.e, Lung, Liver, Cornea, Heart, Pancreas. Here kidney can be replaced by any organ, i.e, Lung, Liver, Cornea, Heart, Pancreas.

$$\rho_A(\text{Organ_Donor}) \quad \rho_B(\text{Patient})$$

$$\rho_C(\text{Kidney}) \quad \rho_D(\text{Kidney})$$

$$\Pi_{A.Name, B.Name} (\sigma_{A.DON_id=C.DON_id \wedge B.PAT_id=D.PAT_id \wedge C.Matched_id=D.ORGAN_id}(AXBXCXD))$$

=> select a.Name, b.Name from Organ_Donor as a, Patient as b, Kidney as c, Kidney as d where a.DON_id=c.DON_id and b.PAT_id=d.PAT_id and c.Matched_id=d.ORGAN_id;

16. Find all the Blood Donors with blood group A+ and are from the Organization ORG_951 or ORG_854 .

$$\prod_{Name, Contact} (\sigma_{ORG_id='ORG_951'} \wedge Blood_Group='A+') (Blood_Donor) \cup$$

$$\prod_{Name, Contact} (\sigma_{ORG_id='ORG_854' \text{ and } Blood_Group='A+'}) (Blood_Donor)$$

=> select Name,Contact from Blood_Donor where (ORG_id="ORG_951" and Blood_Group="A+") or (ORG_id="ORG_854" and Blood_Group="A+");

17. Find all the Organ Donors Name,Contact,DON_id who donate Kidney from 'ORG_951' and Heart from ORG_852.

$$\rho_A(\text{Organ_Donor}) \quad \rho_B(\text{Donors_Organ})$$

$$\prod_{Name, Contact, DON_id, Organ, ORG_id} (\sigma_{A.ORG_id='ORG_951'} \wedge B.Organ='Kidney' \wedge B.DON_id=A.DON_id) (AXB) \cup$$

$$\prod_{Name, Contact, DON_id, Organ, ORG_id} (\sigma_{A.ORG_id='ORG_854'} \wedge B.Organ='Heart' \wedge A.DON_id=B.DON_id) (AXB)$$

=> select Name,Contact,DON_id,Organ,ORG_id from Organ_Donor join Donors_Organ using(DON_id)where (ORG_id="ORG_951" and Organ="Kidney") or (ORG_id="ORG_854" and Organ="Heart");

18. Find Name ,Address and Contact of all the Organization who accept Heart and Cornea but not Lungs and Liver.

$\rho_A(\text{Organisation}) \quad \rho_B(\text{Organisation}_\text{Org_Accepted})$

$$\left(\prod_{Name,Address,Contact} (\sigma_{A.\text{ORG_id}=B.\text{ORG_id}} \wedge B.\text{Organ}=\text{"Heart"}) (AXB) \cup \right. \\ \left. \prod_{Name,Address,Contact} (\sigma_{A.\text{ORG_id}=B.\text{ORG_id}} \wedge B.\text{Organ}=\text{"Cornea"}) (AXB) \right) - \\ \left(\prod_{Name,Address,Contact} (\sigma_{A.\text{ORG_id}=B.\text{ORG_id}} \wedge B.\text{Organ}=\text{"Liver"}) (AXB) \cup \right. \\ \left. \prod_{Name,Address,Contact} (\sigma_{A.\text{ORG_id}=B.\text{ORG_id}} \wedge B.\text{Organ}=\text{"Lung"}) (AXB) \right)$$

=> select distinct(Name),Address,Contact,Organ from Organisation join
Organisation_Org_Accepted using (ORG_id) where (Organ="Heart" or Organ="Cornea") and
Name not in (select distinct(Name) from Organisation join Organisation_Org_Accepted using
(ORG_id) where (Organ="Liver" or Organ="Lung"));

SQL Queries

Since mysql doesn't support embedded SQL, we are using Dynamic SQL queries in our project after taking Permission FROM Prof. Md. Shad Akhtar.

Format to access a mySQL database from python code using a connector is :

```
import mysql.connector

mydb = mysql.connector.connect(host='<URL of the host on which
database is present>', user='<your username>', passwd='<Account password
wrt username>', database='<name of the data base to be accessed>')

mycursor = mydb.cursor()

query=<write the query here>
query2=<write another query if it exists>
# .
# ..
queryN=<write N th query if it exists>
mycursor.execute(query1)
mycursor.execute(query2)
```

```

# .
# ..
mycursor.execute(queryN)
# To get the result of the queries in variable->
myresult = mycursor.fetchall()
for x in myresult:
    print(x)
mydb.commit()

```

***Replace all the text in <angle brackets> with actual text.**

1. Mark an organ for use, in this case we are matching a kidney of ORGAN_id "KD000076" of Donor of DONOR_id "DON_463" with kidney of ORGAN_id "KD000074" and Patient ID "PAT_2320". Here kidney can be replaced by Cornea/Heart/Lung/Liver/Pancreas.
 - o START TRANSACTION;
 - o UPDATE Kidney SET Matched_id="KD000076" WHERE PAT_id="PAT_2320";
 - o UPDATE Kidney SET Matched_id="KD000074" WHERE DON_id="DON_463" ;
 - o COMMIT;
2. Details of all the organisations having reserve of an organ, for the sake of example let it be Kidney.
 - o SELECT * FROM Organisation WHERE ORG_id IN (SELECT ORG_id FROM Organ_Donor WHERE DON_id IN (SELECT DON_id FROM Kidney WHERE DON_id IS NOT NULL));
 - o Here kidney can be replaced by Cornea/Heart/Lung/Liver/Pancreas.
3. An Organisation might want to host a medical camp if the number of Inactive Donors exceed 15. The query to check whether they should hold a medical camp or not.
 - o SELECT IF((Select count(*) from Organ_Donor where Active=0 and ORG_id="ORG_951 " GROUP BY Active)>15, "YES", "NO") as "should we host a medical checkup camp" ;
4. To the rank of all the available donations of kidneys on the basis of compatibility with the patient's kidney of ORGAN_ID: "KD000067"
 - o SELECT ORGAN_id,RANK() OVER (ORDER BY abs(PRA-(select PRA from Kidney where ORGAN_id="KD000067"))) as "Rank" from Kidney where DON_id is not NULL;
5. Find Name ,Address and Contact of all the Organization who accept Heart and Cornea but not Lungs and Liver.

- o select distinct(Name), Address, Contact, Organ from Organisation join Organisation_Org_Accepted using (ORG_id) where (Organ="Heart" or Organ="Cornea") and Name not in (select distinct(Name) from Organisation join Organisation_Org_Accepted using (ORG_id) where (Organ="Liver" or Organ="Lung"));

- 6. Query to SELECT All Organs FROM in all the organisations with their ORGAN_id, MATCHED_id, DONOR_id and PAT_id
 - o SELECT ORGAN_id, PAT_id, DON_id, MATCHED_id FROM Cornea union SELECT ORGAN_id, PAT_id, DON_id, MATCHED_id FROM Heart union SELECT ORGAN_id, PAT_id, DON_id, MATCHED_id FROM Pancreas union SELECT ORGAN_id, PAT_id, DON_id, MATCHED_id FROM Kidney union SELECT ORGAN_id, PAT_id, DON_id, MATCHED_id FROM Lung union SELECT ORGAN_id, PAT_id, DON_id, MATCHED_id FROM Liver;

- 7. To get the number of matched Organs FROM all type of organs.
 - o SELECT count(*)/2 FROM (SELECT ORGAN_id, PAT_id, DON_id, MATCHED_id FROM Cornea union SELECT ORGAN_id, PAT_id, DON_id, MATCHED_id FROM Heart union SELECT ORGAN_id, PAT_id, DON_id, MATCHED_id FROM Pancreas union SELECT ORGAN_id, PAT_id, DON_id, MATCHED_id FROM Kidney union SELECT ORGAN_id, PAT_id, DON_id, MATCHED_id FROM Lung union SELECT ORGAN_id, PAT_id, DON_id, MATCHED_id FROM Liver) as A WHERE A.MATCHED_id is not NULL;

- 8. Query to find the list of organisations that accept the organ that a particular donor wishes to donate. In this case the donor is the donor with DON_id = DON_101.
 - o SELECT A.Name, A.Contact, A.Address FROM Organisation as A WHERE A.ORG_id in (SELECT C.ORG_id FROM Donors_Organ as B, Organisation_Org_Accepted as C WHERE B.DON_id = "DON_101" and B.Organ = C.Organ);

- 9. Query to find percentile ranks of lungs organs by their size.


```
SELECT ORGAN_id, size,
         ROUND(PERCENT_RANK() over(order by size desc), 2) percentile_rank
      FROM Lung;
```

- 10. Query to give sequential integer numbers to each row by their Weight FROM Liver Table.

```
SELECT ROW_NUMBER() over (order by Weight) row_num,  
ORGAN_id,Weight  
FROM Liver;
```

11. Give the Number of blood donations for each date.

- o Select last_donated,count(Name) over (order by last_donated rows between 0 preceding and 0 following) as Number_of_Donations from Blood_Donor;

12. Check every 24 hours, if last donated date of blood donor is one year old then delete the donor.

- o Create Event RemoveOld On Schedule Every 24 Hour Do Delete from Blood_donor where datediff(curdade(),Last_Donated) >365;

13. Check every 24 hours, that if there exists an account of a dismissed blood donor then delete it.

- o Create Event RemoveOldAccount On Schedule Every 24 Hour Do Delete from Account where id not in (Select BLD_id from Blood_bank) and id like('BLD%') ;