DBMS PROJECT

Organ Donation And Procurement Network Management System

TEAM MEMBERS (A2 BATCH)

Naman Sharma	2021UCP1186
Anshul Sharma	2021UCP1146
Vivek	2021UCP1118
Nischay Gautam	2021UCP1141
Rayansh Yadav	2021UCP1200
Dilip Kumar	2021UCP1131

1. Introduction:

Organ Donation and Procurement Network Management System (ODPNMS):

The primary objective of ODPNMS is to create a centralized platform that facilitates seamless communication and coordination among various entities involved in organ donation and procurement networks. This includes users who update new data of patients such as patients in need of organs, potential donors, medical professionals, healthcare organizations, and regulatory bodies and an admin who administrator who can manage the database.

By leveraging a robust database structure and well-defined relationships between entities, the system ensures accuracy, transparency, and adherence to regulatory standards throughout the process.

2. Features and Functionality:

There are four main stakeholders admin, user, patients, Doctor , Donor , Organization.

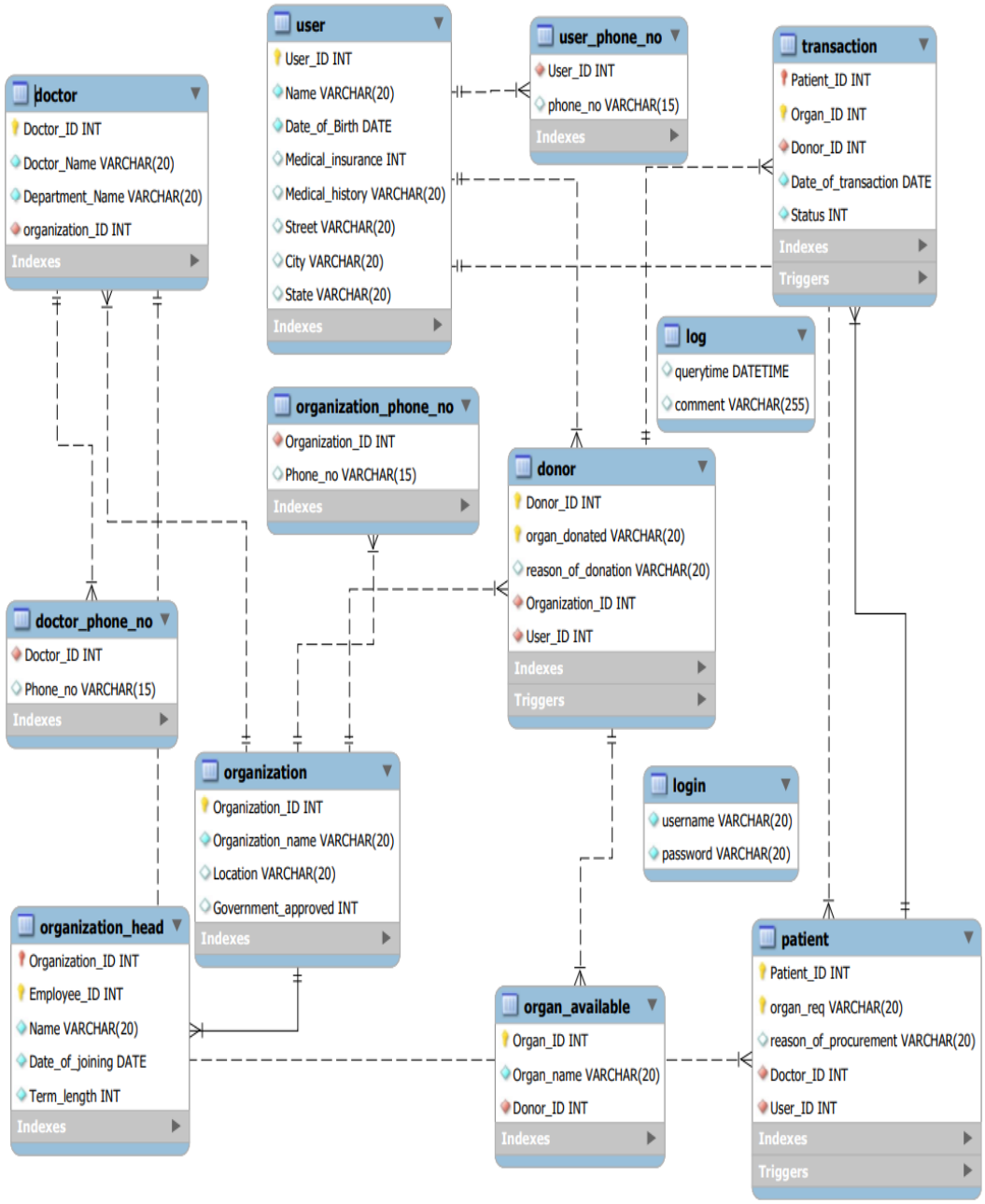
- **Admin**
    - Admin can login.
    - Admin can fetch user details including details of all the patients ,organ donor and transaction all initiated by the given user using its USER_ID .
    - Admin can check the organ available at that moment.
    - Admin can add / update / delete from any table.
    - Search using Patient , User , Donor , Organization , Doctor , Transaction.

- **User**
    - User is the person who are connected to patients , Donor and accomplish transactions of organ to patient.

- **Log**
  - All queries update , delete , insert all are stored into log .

3. Database Design:

ER-DIAGRAM



4. Tables and their Functional Dependencies :-

1) **User**(User_ID, Name, Date _of_birth, Medical_Insurance, Medical_History, Street, City, State)

FD={User_ID → Name, Date _of_birth, Medical Insurance, Medical History, Street, City, State**}**

2) **User_phone_no**(User_ID, phone_no)

FD={User_ID -> phone_no**}**

{User_ID**}** is foreign key constraint

3) **Patient**(Patient_ID, organ_req, reason_of_procurement, Doctor_ID, User_ID)

FD={Patient_ID, organ_req -> reason_of_procurement, Doctor_ID, User_ID**}**

{User_ID, Doctor_ID} are foreign key constraints

4) **Donor**(Donor_ID, organ_donated, reason_of_donation, Organization_ID, User_ID)

FD={Donor_ID, organ_donated -> reason_of_donation, Organization_ID, User_ID**}**

{User_ID, Organization_ID**}** are foreign key constraints

5) **Organ Available**(Organ_ID,Organ_name, Donor_ID)

FD={Organ_ID -> Organ_name,Donor_ID**}**

{Donor_ID**}** is foreign key constraint

6) **Transaction**(Patient_ID, Organ_ID, Donor_ID, Date_of_transaction, Status)

FD={Patient_ID, Organ_ID -> Donor_ID,Date_of_transaction, Status}

{Patient_ID, Donor_ID} are foreign key constraints

7) **Organization**(Organization_ID, Organization_name, Location, Government_approved)

FD={Organization_ID -> Organization_name, Location, Government_approved**}**

8) **Organization_phone_no**(Organization_ID, phone_no)

FD={Organization_ID -> phone_no**}**

{Organization_ID**}** are foreign key constraints

9) **Doctor**(Doctor_ID, Doctor_name, Department_name, Organization_id)

FD={Doctor_ID -> Doctor_name, Organization_id**}**

{Organization_ID**}** is foreign key constraint

10) **Doctor_phone_no**(Doctor_ID, phone_no)

FD={Doctor_ID -> phone_no**}**

{Doctor_ID} is foreign key constraint

11) **Organization_head**(Organization_ID, Employee_ID, Name, Date_of_joining, Term_length)

FD={Organization_ID, Employee_ID -> Name, Date_of_joining, Term_length**}**

5.Triggers

The following triggers are added to create a log of actions done on database. The logsare added to the log table.

1) Trigger for adding Donor information to Log table.

```
delimiter //
create trigger ADD_DONOR_LOGafter insert
on Donor
for each rowbegin
insert into log values
(now(), concat("Inserted new Donor",cast(new.Donor_Id as char)));
end // delimiter ;
```

2) Trigger for adding “Update” action information in Log table.

```
create trigger UPD_DONOR_LOGafter update
on Donor
for each rowbegin
insert into log values
(now(), concat("Updated Donor Details",cast(new.Donor_Id as char)));
end // delimiter ;
```

3) Trigger for adding “Delete” action information in Log table.

```
create trigger DEL_DONOR_LOGafter delete
on Donor
for each rowbegin
insert into log values
(now(), concat("Deleted Donor ",cast(old.Donor_Id as
char))); end //
delimiter ;
```

4) Trigger for adding “Add patient” action information in Log table

```
create trigger ADD_PATIENT_LOGafter insert
on Patient for each
rowbegin
insert into log values
(now(), concat("Inserted new Patient", cast(new.Patient_Id as
char))); end //
delimiter ;
```

5) Trigger for adding “Update information” action information in Log table

```
create trigger UPD_PATIENT_LOGafter update
on Patient for each
rowbegin
```

```
insert into log values
(now(), concat("Updated Patient Details", cast(new.Patient_Id as
char)));
end // delimiter ;
```

6) Trigger for adding “Delete information” action information in Log table

```
create trigger DEL_PATIENT_LOGafter delete
on Donor
for each rowbegin
insert into log values
(now(), concat("Deleted Patient ",cast(old.Donor_Id as char)));
end // delimiter ;
```

7) Trigger for adding “Add transaction” action information in Log table

```
create trigger ADD_TRASACTION_LOGafter insert
on Transactionfor each row
begin
insert into log values
(now(), concat("Added Transaction :: Patient ID : ",
cast(new.Patient_ID aschar), "; Donor ID :
" ,cast(new.Donor_ID as char)));end //
delimiter ;
```

8) Trigger to calculate the Organ available after adding a Donor and after the transaction of a organ available at that time.

```
delimiter //
```

```
create trigger ADD_DONOR
after insert
on Donor
for each row
begin
insert into Organ_available(Organ_name, Donor_ID)
values (new.organ_donated, new.Donor_ID);
end//
delimiter ;
delimiter //
create trigger REMOVE_ORGAN
after insert
on Transaction
for each row
begin
delete from Organ_available
where Organ_ID = new.Organ_ID;
end//
delimiter ;
```

5.Technologies Used:

- **MYSQL**
- **HTML**
- **CSS**
- **Python**
- **Flask**

6. Acknowledgement:

A special acknowledgement goes to our project guide, Sir Dinesh Tyagi, for invaluable guidance and expertise. His insights and encouragement significantly improved the project's quality. We also appreciate the support and insightful discussions from our peers, enhancing the collaborative spirit during development.

7.Conclusion:

In conclusion, **Organ Donation and Procurement Network Management System** serves as a pivotal tool for various stakeholders involved in organ donation networks, including patients, potential donors, medical professionals, healthcare organizations, and regulatory bodies. By providing a centralized platform for information exchange and coordination, the system ensures that critical data, such as donor and patient profiles, organ availability, and transaction details, is managed with precision and accessibility.