

DBMS Project Report

Organ Donation Management System

"Saving Lives Through Technology, One Person at a Time"

Semester: Fall 2025

Instructor: Ms. Maria Rahim

Submission Date: 9th Dec 2025

Team Members:

Marium Ali (29183)

Fatima Faisal (28989)

Khadeja Qureshi 27200)

1. Business Scenario

1.1 Domain Overview

The Organ Donation Management System addresses the critical challenge of connecting organ donors with recipients timely in Pakistan and globally. Operating within the healthcare and transplant domain, it provides a centralized platform for donors, recipients, and hospitals, streamlining organ procurement and allocation. By integrating all stakeholders into a single system, it enables real-time coordination, transparent matching, and efficient communication, ensuring that organs reach compatible recipients quickly and fairly.

1.2 Problem Statement and Significance

Currently, organ donation faces inefficiencies due to fragmented systems, manual coordination, limited donor accessibility, and delayed communication, often resulting in missed life-saving opportunities. Our system tackles these issues by automating match discovery, reducing wait times, and ensuring transparent allocation based on medical compatibility and urgency. This not only improves hospital workflow but also increases the chances of successful transplants, saving more lives while providing donors, patients, and administrators with a reliable, efficient, and fair platform.

2. Summary of Interview and Application Analysis

2.1 Expert Interview

Resource Person: Dr. Aun Ali

Position: General Surgeon

Hospital: Bantva Anis Hospital

Relevant Experience: Has assisted in transplant-related surgeries

Key Insights from Interview:

1. Basic Requirements for Matching Donors and Recipients

Dr. Aun explained that the first step in matching donors and recipients is checking medical compatibility. This includes blood group matching, overall health condition, and ensuring that the organ is suitable in size and condition for the recipient. He emphasized that without these basics, the transplant cannot proceed safely.

2. Importance of Geographical Proximity

He highlighted that the distance between the donor's hospital and the recipient's hospital is extremely important. Ideally, both should be in the *same city* to prevent delays and preserve organ quality. If they are in different cities, the hospitals must coordinate beforehand, especially for transportation, availability of surgeons, and timing, because even small delays can affect the success of the transplant.

3. Fairness and Ethical Considerations

He mentioned that organs should always be allocated based on medical need and compatibility, not on wealth, personal connections, or social status. Clear and transparent rules ensure fairness and maintain trust in the system.

4. Need for a Centralized System

From his experience, Dr. Aun noted that information is often scattered across hospitals, leading to delays or missed opportunities. A centralized system would help doctors quickly find potential matches, check compatibility, and coordinate with other hospitals more efficiently.

2.2 Application Analysis

We studied a few existing organ donation and health coordination systems to understand current practices and what we can improve.

Application 1: United Network for Organ Sharing (UNOS) – USA

Strengths:

- Large, nationwide database for all transplant centers
- Strong medical matching criteria (urgency, blood type, waiting time, etc.)
- Real-time tracking of organ movement
- Strong medical matching criteria

Limitations:

- Very complex, it is designed mainly for trained medical staff
- Not friendly or accessible for donors or families

Lessons Learned:

- Different user roles need different interfaces
- A scoring-based matching approach is useful

Application 2: DonateLife – Australia

Strengths:

- Simple and friendly registration process
- Strong awareness and educational material
- Connected to Australia's medical card system, which helps pull verified medical records instantly
- Clear consent options for donors

Limitations:

- Does not show detailed matching information
- Still depends on hospitals manually coordinating

Lessons Learned:

- Good design and easy registration make more people sign up
- Consent and donor preferences should be clear

2.3 Synthesis and Our Approach

Based on the expert interview and application analysis, our Organ Donation Management System incorporates the following unique features:

- **Donor Empowerment:** Unlike existing systems, donors can view all potential matches with compatibility scores and select the best match themselves.

- **Transparent Scoring:** Compatibility and urgency scores are visible to all stakeholders, ensuring fairness and building trust.
- **Multi-Stage Verification:** Both donor acceptance and admin approval are required before final confirmation, ensuring medical verification.
- **Fair Allocation:** Matching considers only medical factors (blood type, size, urgency, proximity) and not socioeconomic status.

3. Business Rules

3.1 Core Business Rules

Rule 1: Donor Registration

- Donors must register with complete personal and medical information (age, blood type, weight, contact).
- Age requirement: 18–75 years.
- An emergency contact is required.
- Registrations must be verified by an admin before listing any organs.
- Donor status options: Active, Inactive, Deceased, Suspended.

Rule 2: Organ Listing

- One donor may list multiple organs.
- Valid organ types include: Heart, Kidney, Liver, Lung, Pancreas, Intestine, Cornea.
- Organ statuses: Available, Unavailable, Matched, Transplanted, Expired.
- Once a match is accepted and approved, the organ becomes Unavailable.

Rule 3: Hospital Registration

- Hospitals must register with a valid license and be admin-verified.
- They must indicate if they have transplant capabilities.
- Only hospitals can register recipients; individuals cannot self-register.

Rule 4: Recipient Registration

- Recipients must be registered by a hospital with complete medical details and a unique CNIC.
- Each recipient requests one specific organ (like you request two hearts, but can do one heart and one kidney)
- They receive a critical level (Critical, High, Medium, Low) and a priority score (0–100).
- Recipient statuses: Waiting, Matched, Transplanted, Deceased, Removed.

Rule 5: Match Generation

- When an organ becomes available, the system automatically finds compatible recipients.
- Matching considers:
 - Blood type (mandatory)
 - Organ type
 - Age/size compatibility

- Geographical proximity
- One organ may have multiple potential matches; one recipient may appear in many.
- Each match includes: Compatibility Score, Urgency Score, Distance, and Estimated Travel Time.

Rule 6: Donor Match Acceptance

- Donors can view all potential matches for their organs.
- A donor may accept only one match per organ.
- Once accepted, all other matches for that organ are cancelled.
- The match becomes pending admin approval.

Rule 7: Admin Approval

- Admins verify medical compatibility, hospital readiness, and record validity.
- Once approved:
 - Organ → **Unavailable**
 - Recipient → **Matched**
 - Match → **Approved**
- Recipient hospital is notified immediately.
- Hospitals receive donor and organ details along with match data.

Rule 9: Blood Type Rules

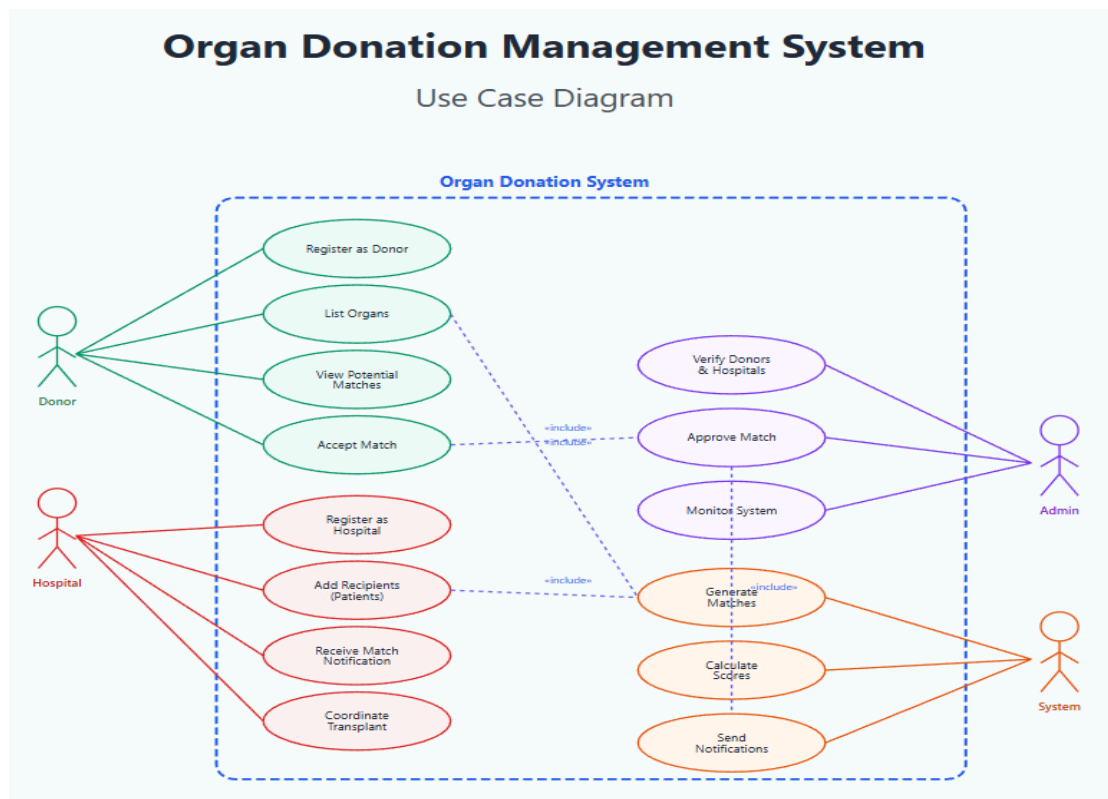
- Blood compatibility follows universal medical standards:
 - **O-** can donate to all blood types (universal donor)
 - **O+** can donate to O+, A+, B+, AB+
 - **A-** can donate to A-, A+, AB-, AB+
 - **A+** can donate to A+, AB+
 - **B-** can donate to B-, B+, AB-, AB+
 - **B+** can donate to B+, AB+
 - **AB-** can donate to AB-, AB+
 - **AB+** can only donate to AB+ (universal recipient)
- Matches with incompatible blood types are never generated

Rule 10: Fair Allocation

- Allocation is strictly based on medical need, compatibility, and location and never social or economic factors.

3.2 Use Cases

Use Case Diagram:



Actors:

1. Donor (Green) - 4 use cases
2. Hospital (Red) - 4 use cases
3. Admin (Purple) - 3 use cases
4. System (Orange) - 3 use cases

Donor Actions:

- Register as Donor
- List Organs
- View Potential Matches
- Accept Match

Hospital Actions:

- Register as Hospital
- Add Recipients (Patients)
- Receive Match Notification
- Coordinate Transplant

Admin Actions:

- Verify Donors & Hospitals
- Approve Match
- Monitor System

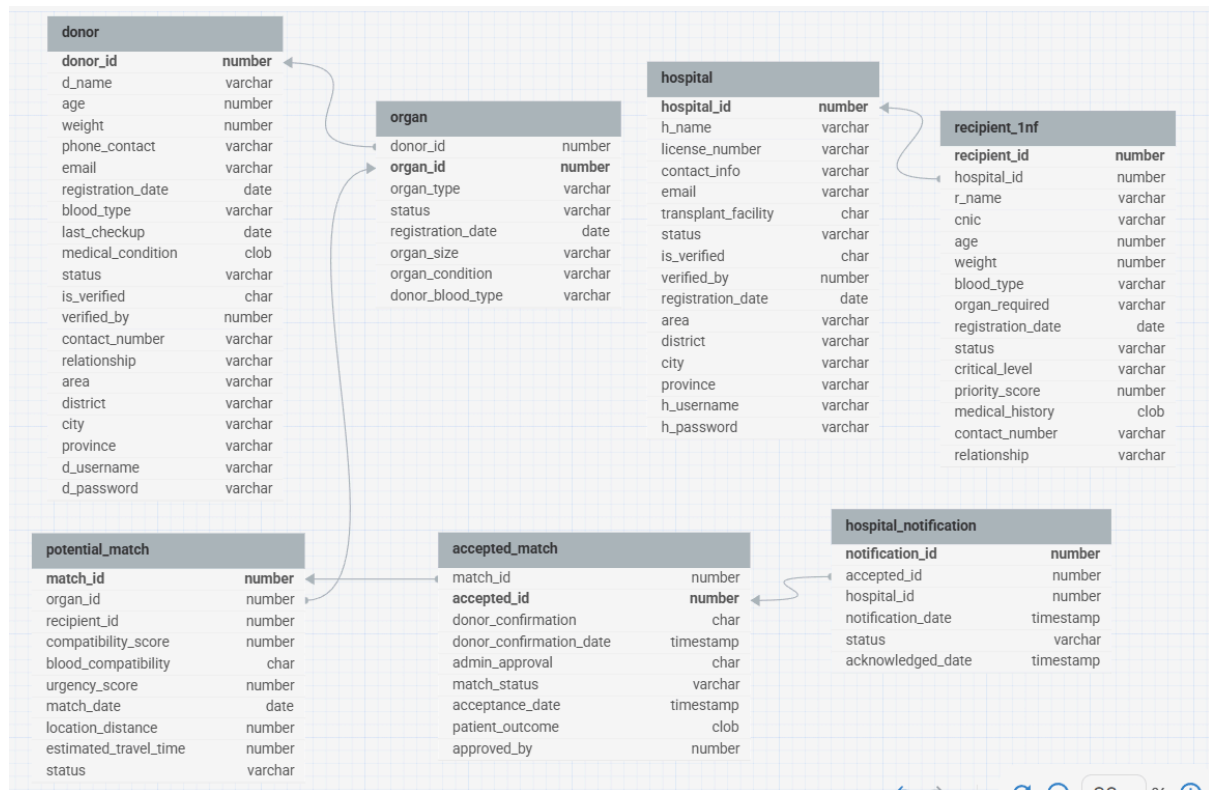
System Actions:

- Generate Matches
- Calculate Scores
- Generate Notification/Alert

4. Entities, Attributes, and Relationships

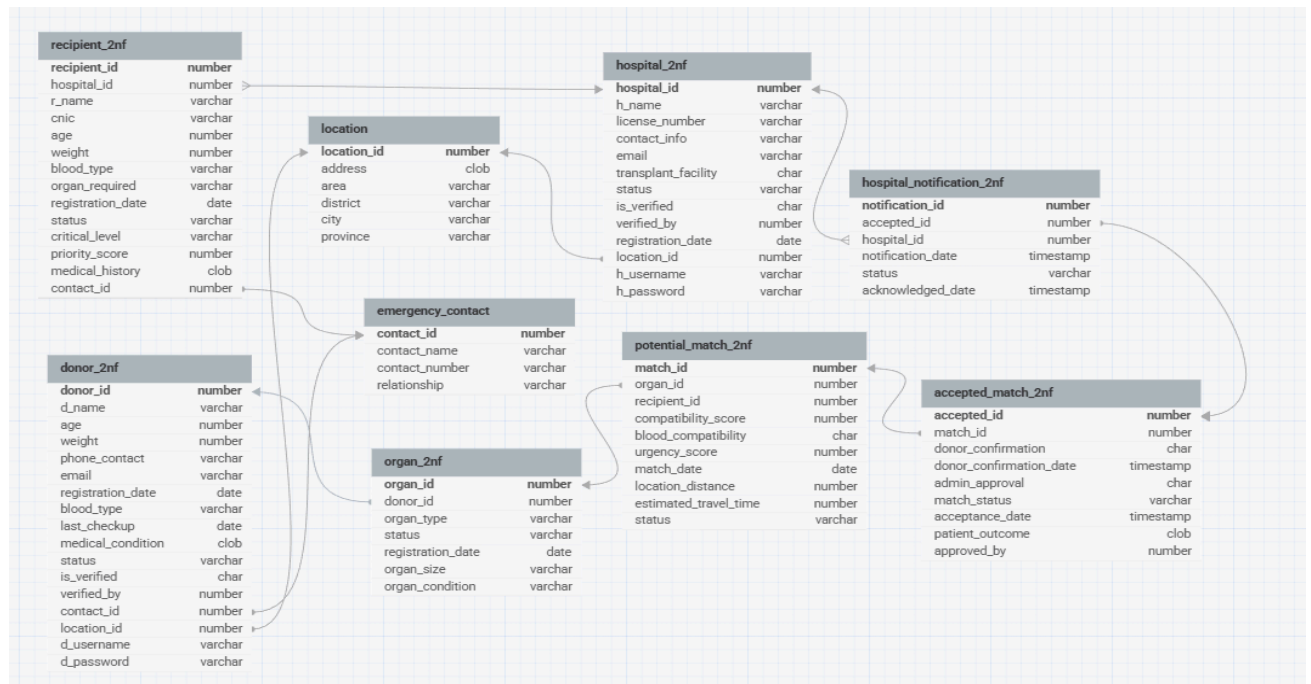
4.1 Database Normalization Process

1NF:



- Emergency contacts and location details are embedded inside donor and recipient tables.
- Donor info is repeated in Organ and Hospital locations are directly in the hospital table
- This has redundancy but all atomic values with no repeating groups.

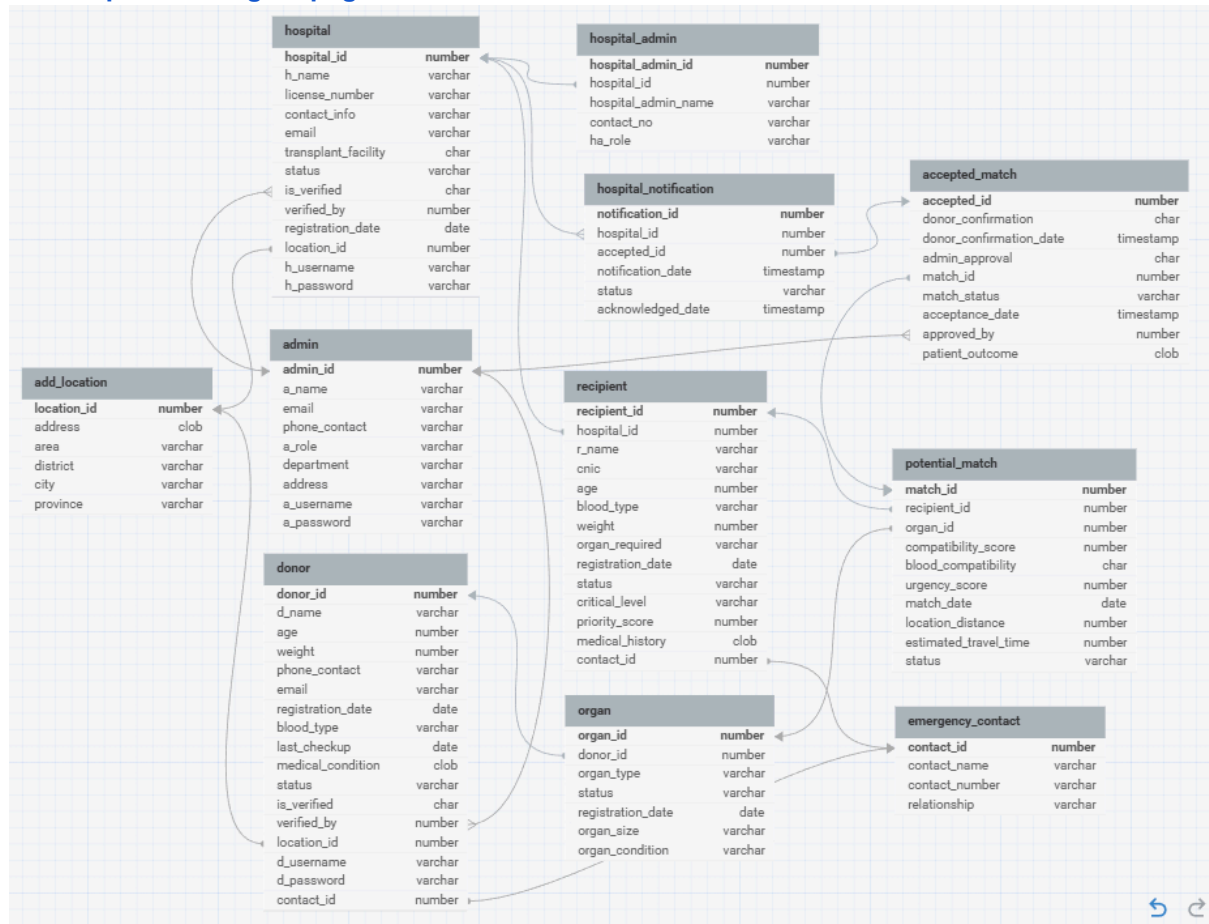
2NF:



Changes from 1NF → 2NF:

- Emergency contact info moved to a separate Emergency_Contact table.
- Location details moved to a separate Location table.
- Repeated donor info removed from Organ table.
- Recipient table references Emergency_Contact via foreign key instead of embedding.
- Hospital locations referenced via Location_Id instead of storing full address.
- All non-key attributes now fully depend on their table's primary key.
- Redundancy reduced, partial dependencies eliminated.

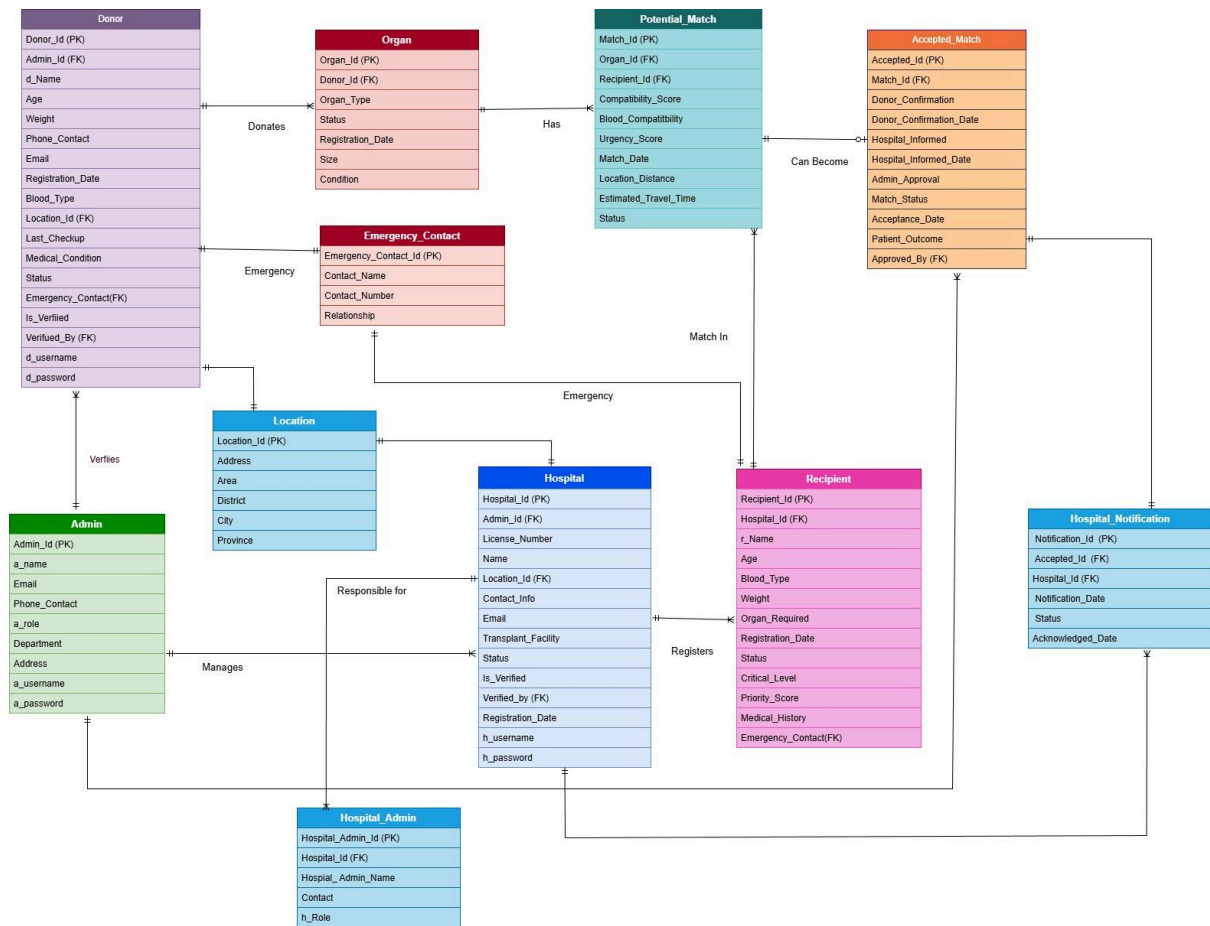
3NF: <https://dbdesigner.page.link/7c7Jva1G2XSd9Gnh9>



Changes from 2NF → 3NF:

- Created a separate Admin table; Verified_By in Donor/Hospital now a foreign key.
- Created Hospital_Admin table to allow multiple admins per hospital.
- Removed transitive dependencies,
- Repeated attributes like Donor_Name or hospital address removed from related tables.
- All non-key attributes now depend only on the table's primary key.
- Redundancy minimized; schema fully normalized to 3NF.

4.2 Entity-Relationship Diagram



The ERD diagram illustrates the complete database schema for the Organ Donation Management System, showing 11 entities and their relationships.

4.3 Detailed Entity Descriptions

Entity 1: ADMIN

Description: System administrators who manage the platform, verify registrations, and approve matches.

Attributes:

- admin_id** (PK): Unique identifier, auto-generated
- a_name**: Administrator's full name
- email**: Unique email address
- phone_contact**: Contact phone number
- a_role**: Role (SuperAdmin, Admin, Coordinator)
- department**: Department (Transplant, Operations)
- address**: Physical address
- a_username**: Unique login username
- a_password**: Login password

Multiplicity Constraints:

- One ADMIN can verify many DONORS (1:N)
- One ADMIN can verify many HOSPITALS (1:N)

- One ADMIN can approve many ACCEPTED_MATCHES (1:N)

Entity 2: DONOR

Description: Individuals who register to donate their organs.

Attributes:

- **Donor_Id** (PK): Unique identifier, auto-generated
- **d_name**: Donor's full name
- **Age**: Age between 18-75 years
- **Weight**: Weight in kilograms
- **Phone_Contact**: Contact phone number
- **Email**: Unique email address
- **Registration_Date**: Date of registration
- **Blood_Type**: Blood group (A+, A-, B+, B-, AB+, AB-, O+, O-)
- **Last_Checkup**: Date of last medical checkup
- **Medical_Condition**: Medical history
- **Status**: Active, Inactive, Deceased, Suspended
- **Is_Verified**: Verification status (Y/N)
- **Verified_By** (FK): Admin who verified
- **Contact_Id** (FK): Emergency contact reference
- **Location_Id** (FK): Location reference
- **d_username**: Login username
- **d_password**: Login password

Multiplicity Constraints:

- One DONOR can donate many ORGANS (1:N)
- Many DONORS can share one EMERGENCY_CONTACT (N:1)
- Many DONORS can be at one ADD_LOCATION (N:1)
- Many DONORS are verified by one ADMIN (N:1)

Entity 3: ORGAN

Description: Individual organs listed for donation.

Attributes:

- **Organ_Id** (PK): Unique identifier, auto-generated
- **Donor_Id** (FK): Reference to donor
- **Organ_Type**: Heart, Kidney, Liver, Lung, Pancreas, Intestine, Cornea
- **Status**: Available, Unavailable, Matched, Transplanted, Expired
- **Registration_Date**: Date organ was listed
- **Organ_Size**: Small, Medium, Large
- **Organ_Condition**: Health condition description

Multiplicity Constraints:

- Many ORGANS belong to one DONOR (N:1)
- One ORGAN can have many POTENTIAL_MATCHES (1:N)

Entity 4: HOSPITAL

Description: Medical facilities that register recipients and coordinate transplants.

Attributes:

- **Hospital_Id** (PK): Unique identifier, auto-generated
- **h_name**: Hospital name
- **License_Number**: Unique government license
- **contact_info**: Contact phone
- **email**: Unique email address
- **transplant_facility**: Has transplant capability (Y/N)

- **status:** Active, Inactive, Suspended
- **Is_Verified:** Verification status (Y/N)
- **Verified_By (FK):** Admin who verified
- **registration_date:** Registration date
- **location_id (FK):** Location reference
- **h_username:** Login username
- **h_password:** Login password

Multiplicity Constraints:

- One HOSPITAL can register many RECIPIENTS (1:N)
- One HOSPITAL can have many HOSPITAL_ADMINS (1:N)
- One HOSPITAL can receive many HOSPITAL_NOTIFICATIONS (1:N)
- Many HOSPITALS are at one ADD_LOCATION (N:1)
- Many HOSPITALS are verified by one ADMIN (N:1)

Entity 5: RECIPIENT

Description: Patients waiting for organ transplants.

Attributes:

- **Recipient_Id (PK):** Unique identifier, auto-generated
- **Hospital_Id (FK):** Hospital managing patient
- **R_Name:** Recipient's full name
- **Cnic:** Unique national ID
- **Age:** Age between 0-120 years
- **Blood_Type:** Blood group
- **weight:** Weight in kilograms
- **Organ_Required:** Type of organ needed
- **Registration_date:** Registration date
- **Status:** Waiting, Matched, Transplanted, Deceased, Removed
- **Critical_level:** Critical, High, Medium, Low
- **priority_score:** Medical priority (0-100)
- **medical_history:** Medical history
- **contact_id (FK):** Emergency contact reference

Multiplicity Constraints:

- Many RECIPIENTS belong to one HOSPITAL (N:1)
- One RECIPIENT can have many POTENTIAL_MATCHES (1:N)
- Many RECIPIENTS can share one EMERGENCY_CONTACT (N:1)

Entity 6: POTENTIAL_MATCH

Description: System-generated matches between organs and recipients.

Attributes:

- **Match_Id (PK):** Unique identifier, auto-generated
- **Organ_Id (FK):** Reference to organ
- **Recipient_Id (FK):** Reference to recipient
- **Compatibility_Score:** Score 0-100 (auto-calculated)
- **Blood_Compatibility:** Y/N (auto-calculated)
- **Urgency_Score:** Score 0-100 (auto-calculated)
- **Match_Date:** Date match was created
- **Location_Distance:** Distance in km (auto-calculated)
- **Estimated_Travel_Time:** Travel time in hours (auto-calculated)
- **Status:** Pending, Accepted, Rejected, Cancelled, Expired

Multiplicity Constraints:

- Many POTENTIAL_MATCHES for one ORGAN (N:1)
- Many POTENTIAL_MATCHES for one RECIPIENT (N:1)
- One POTENTIAL_MATCH can become one ACCEPTED_MATCH (1:1)

Entity 7: ACCEPTED_MATCH

Description: Matches accepted by donors and approved by admins.

Attributes:

- **Accepted_Id** (PK): Unique identifier, auto-generated
- **Match_Id** (FK): Reference to potential match (unique)
- **Donor_Confirmation**: Donor acceptance (Y/N)
- **Donor_Confirmation_Date**: Timestamp of acceptance
- **Admin_Approval**: Admin approval (Y/N)
- **Match_Status**: Pending_Approval, Approved, Rejected, Completed, Failed
- **Acceptance_Date**: Timestamp when created
- **Patient_Outcome**: Transplant outcome description
- **Approved_By** (FK): Admin who approved

Multiplicity Constraints:

- One ACCEPTED_MATCH from one POTENTIAL_MATCH (1:1)
- Many ACCEPTED_MATCHES approved by one ADMIN (N:1)
- One ACCEPTED_MATCH generates one HOSPITAL_NOTIFICATION (1:1)

Entity 8: HOSPITAL_NOTIFICATION

Description: Notifications sent to hospitals when matches are approved.

Attributes:

- **Notification_Id** (PK): Unique identifier, auto-generated
- **Accepted_Id** (FK): Reference to accepted match
- **Hospital_Id** (FK): Reference to hospital
- **Notification_Date**: Timestamp sent
- **Status**: Sent, Delivered, Acknowledged, Failed
- **Acknowledged_Date**: Timestamp acknowledged

Multiplicity Constraints:

- Many NOTIFICATIONS for one ACCEPTED_MATCH (N:1)
- Many NOTIFICATIONS sent to one HOSPITAL (N:1)

Entity 9: EMERGENCY_CONTACT

Description: Emergency contacts for donors and recipients.

Attributes:

- **Contact_id** (PK): Unique identifier, auto-generated
- **contact_name**: Contact person's name
- **contact_number**: Contact phone number
- **relationship**: Relationship to donor/recipient

Multiplicity Constraints:

- One EMERGENCY_CONTACT for many DONORS (1:N)
- One EMERGENCY_CONTACT for many RECIPIENTS (1:N)

Entity 10: ADD_LOCATION

Description: Geographic location information.

Attributes:

- **Location_Id** (PK): Unique identifier, auto-generated

- **Address:** Full street address
- **Area:** Local area/neighborhood
- **District:** Administrative district
- **City:** City name
- **Province:** Province/state name

Multiplicity Constraints:

- One ADD_LOCATION for many DONORS (1:N)
- One ADD_LOCATION for many HOSPITALS (1:N)

Entity 11: HOSPITAL_ADMIN

Description: Hospital administrative staff.

Attributes:

- **Hospital_Admin_Id** (PK): Unique identifier, auto-generated
- **Hospital_Id** (FK): Reference to hospital
- **Hospital_Admin_Name:** Name of administrator
- **Contact_No:** Contact phone
- **ha_Role:** Role (Coordinator, Manager)

Multiplicity Constraints:

- Many HOSPITAL_ADMINS work for one HOSPITAL (N:1)

4.3. Constraints Applied to Tables

1. Primary Key Constraints

All tables have auto-generated primary keys using **GENERATED ALWAYS AS IDENTITY**:

```
CREATE TABLE DONOR (
    Donor_Id NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
```

Ensures each record has a unique identifier that is automatically assigned.

2. Foreign Key Constraints

Example DONOR Table Foreign Keys:

```
CONSTRAINT fk_emergency_contact FOREIGN KEY (Contact_Id) REFERENCES EMERGENCY_CONTACT(Contact_Id) ON DELETE CASCADE,
CONSTRAINT fk_location FOREIGN KEY (Location_Id) REFERENCES Add_Location(Location_Id) ON DELETE CASCADE,
CONSTRAINT fk_d_verified_by FOREIGN KEY (Verified_By) REFERENCES ADMIN(Admin_Id)
```

- Maintains referential integrity between tables
- **ON DELETE CASCADE:** When emergency contact or location is deleted, associated donors are automatically deleted
- No cascade on **Verified_By**: If admin is deleted, donor verification record remains

//have many foreign key constraints in almost all tables, so just included one example snippet

3. UNIQUE Constraints

Ensuring Uniqueness , in example Recipient Table:

```
CREATE TABLE HOSPITAL (
    Hospital_Id      NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
    h_name           varchar2(150) not null,
    License_Number    varchar2(50) unique not null,
```

```
CREATE TABLE ACCEPTED_MATCH (
    Accepted_Id      NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
    Match_Id         NUMBER UNIQUE NOT NULL,
```

- Prevents duplicate emails, usernames, license numbers, and CNICs
- Ensures one accepted match per potential match

4. CHECK Constraints

//only included some important constraint examples in the report

Blood Type Validation:

```
Blood_Type    VARCHAR2(3) NOT NULL CHECK (Blood_Type IN ('A+', 'A-', 'B+', 'B-', 'AB+', 'AB-', 'O+', 'O-')),
```

Organ Type Validation:

```
Organ_Type    VARCHAR2(20) NOT NULL CHECK (Organ_Type IN ('Heart', 'Kidney', 'Liver', 'Lung', 'Pancreas', 'Intestine', 'Cornea')),
Status        VARCHAR2(20) DEFAULT 'Available' CHECK (Status IN ('Available', 'Unavailable', 'Matched', 'Transplanted', 'Expired')),
```

Restricts organ types to medically valid options.

Status Validation:

Organ Status in organ table:

```
Status        VARCHAR2(20) DEFAULT 'Available' CHECK (Status IN ('Available', 'Unavailable', 'Matched', 'Transplanted', 'Expired')),
```

Donor Status in donor table:

```
Status        VARCHAR2(20) DEFAULT 'Active' CHECK (Status IN ('Active', 'Inactive', 'Deceased', 'Suspended')),
```

Enforces valid status transitions and prevents invalid states.

5. NOT NULL Constraints

Critical Fields:

```
Donor_Id      NUMBER NOT NULL,
Organ_Type    VARCHAR2(20) NOT NULL
```

Ex. Organ table

Ensures essential information is always provided, prevents incomplete records.

6. DEFAULT Constraints

Auto-Populated Dates:

```
registration_date date default SYSDATE not null,
```

Automatically captures timestamp when records are created.

Default Values:

```
status          varchar2(20) default 'Active'
```

Sets appropriate initial values/status for fields like status and scores.

4.4 Triggers, Stored Procedures, and Views

4.4.1. Triggers Explanation

Trigger 1: trg_pm_all

Automatically calculates all compatibility metrics when a potential match is created or updated.

What It Does:

1. **Fetches Donor Information:**

- Retrieves donor's age, weight, blood type, and location from DONOR and ORGAN tables
- Gets donor's geographic area, district, city, province from ADD_LOCATION

2. **Fetches Recipient Information:**

- Retrieves recipient's age, weight, blood type, organ required from RECIPIENT table
- Gets recipient hospital's location details from ADD_LOCATION via HOSPITAL

Calculates Blood Compatibility:

```
-- Blood Compatibility
if (
  (v_donor_bt = 'O-' ) or
  (v_donor_bt = 'O+' and v_recipient_bt in ('O+', 'A+', 'B+', 'AB+')) or
  (v_donor_bt = 'A-' and v_recipient_bt in ('A-', 'A+', 'AB-', 'AB+')) or
  (v_donor_bt = 'A+' and v_recipient_bt in ('A+', 'AB+')) or
  (v_donor_bt = 'B-' and v_recipient_bt in ('B-', 'B+', 'AB-', 'AB+')) or
  (v_donor_bt = 'B+' and v_recipient_bt in ('B+', 'AB+')) or
  (v_donor_bt = 'AB-' and v_recipient_bt in ('AB-', 'AB+')) or
  (v_donor_bt = 'AB+' and v_recipient_bt = 'AB+')
) then
  :new.blood_compatibility := 'Y';
  v_score := v_score + 40;
else
  :new.blood_compatibility := 'N';
end if;
```

3. **Calculates Compatibility Score (0-100):**

The trigger calculates a match's compatibility score by checking blood type first (mandatory), then adding points based on age and weight differences, and finally adding points if the organ type matches exactly. The total score is capped at 100, reflecting the best possible match.

4. **Calculates Location Distance:** Based on overlap in donor and hospital location breakdown.

5. **Calculates Estimated Travel Time:** Based on distance with different speed assumptions:

```
-- Location Distance
if v_donor_area = v_recip_area then
:new.location_distance := 5;
elseif v_donor_district = v_recip_district then
:new.location_distance := 15;
elseif v_donor_city = v_recip_city then
:new.location_distance := 50;
elseif v_donor_province = v_recip_province then
:new.location_distance := 100;
else
:new.location_distance := 200;
end if;

-- Estimated Travel Time
if :new.location_distance <= 5 then
:new.estimated_travel_time := round(:new.location_distance / 1.2, 2);
elseif :new.location_distance <= 15 then
:new.estimated_travel_time := round(:new.location_distance / 1.5, 2);
elseif :new.location_distance <= 50 then
:new.estimated_travel_time := round(:new.location_distance / 40, 2);
elseif :new.location_distance <= 100 then
:new.estimated_travel_time := round(:new.location_distance / 60, 2);
else
:new.estimated_travel_time := round(:new.location_distance / 80, 2);
end if;
```

Calculates Urgency Score:

```
--Urgency Score
v_weight := case v_recipient_crit
when 'Critical' then 100
when 'High' then 80
when 'Medium' then 50
when 'Low' then 20
end;
:new.urgency_score := (v_recipient_prio * 0.7) + (v_weight * 0.3);
end;
```

Formula: 70% based on priority_score, 30% based on critical_level

- Eliminates manual calculation errors
- Ensures consistent scoring across all matches
- Updates automatically if match data changes

Trigger 2: trg_accepted_match_organ_unavailable

Purpose: Automatically updates organ status to "Unavailable" when admin approves a match.

What It Does:

- Monitors the match_status field in ACCEPTED_MATCH
- When status changes from anything to 'Approved':
 - The trigger finds the organ for the match, updates its status to "Unavailable," and marks it as reserved so it cannot be matched again.

Trigger Definition:

```
create or replace trigger trg_accepted_match_organ_unavailable
after update of match_status on accepted_match
for each row
begin
    if :new.match_status = 'Approved' and :old.match_status != 'Approved' then
        update organ o
        set o.status = 'Unavailable'
        where o.organ_id = (
            select pm.organ_id
            from potential_match pm
            where pm.match_id = :new.match_id
        );
    end if;
end;
```

Why It's Important:

- Prevents double-allocation of organs
- Ensures organ availability is always accurate

Trigger 3: trg_cancel_other_matches

Purpose: Automatically cancels all other pending potential matches when one match is accepted.

Trigger Definition:

```
after update of status on potential_match
for each row
when (new.status = 'Accepted')
begin
    update potential_match
    set status = 'Cancelled'
    where organ_id = (select organ_id from potential_match where match_id = :new.match_id)
    and match_id != :new.match_id
    and status = 'Pending';
end;
/
```

What It Does:

- Monitors the status field in POTENTIAL_MATCH
- When any match status changes to 'Accepted':
- Finds all other pending matches for the same organs
- Changes their status to 'Cancelled'.
- Ensures only one match per organ remains active

Why It's Important:

- Enforces business rule: one organ can only be accepted for one recipient
- Prevents confusion with multiple active matches

4.4.2 Stored Procedures Explanation

Procedure 1: sp_create_matches

Purpose: Generates potential matches for a newly listed organ.

Procedure Definition:

```
create or replace procedure sp_create_matches(p_organ_id number) as
v_match_id number;
begin
for r in (select recipient_id
from recipient
where organ_required =
(select organ_type from organ where organ_id = p_organ_id)
and status = 'Waiting')
loop
begin
select match_id into v_match_id from potential_match
where organ_id = p_organ_id and
recipient_id = r.recipient_id;
exception when no_data_found then
insert into potential_match (organ_id, recipient_id)
values (p_organ_id, r.recipient_id)
returning match_id into v_match_id;
end;
end loop;
end;
```

How It Works:

- Takes p_organ_id as input parameter
- Finds the organ type of this organ
- Searches for all recipients who:
- Need this organ type and have status 'Waiting'
- For each matching recipient, it checks if potential match already exists
- If not, creates new POTENTIAL_MATCH record
- Trigger trg_pm_all automatically calculates all scores

Procedure 2: sp_accept_match

Purpose: Handles the complete workflow when a donor accepts a match.

Actions Performed:

- Checks if match exists
- Ensures match is still "Pending" (not already accepted/rejected)
- Verifies organ is still "Available" (not already matched to someone else)
- Creates ACCEPTED_MATCH record with donor confirmation
- Updates POTENTIAL_MATCH status to "Accepted"
- Changes ORGAN status to "Matched"
- Cancels all other pending matches for this organ
- Returns accepted_id and success message

Procedure Definition:

```
begin
p_accepted_id := null; p_message := null;
begin
select organ_id,
status into v_organ_id, v_match_status
from potential_match where match_id = p_match_id;

exception when no_data_found then p_message := 'Match not found';
return;
end;

if v_match_status != 'Pending'
then p_message := 'Match already processed';
return;
end if;

select status into v_organ_status
from organ where organ_id = v_organ_id;
if v_organ_status != 'Available'
then p_message := 'Organ is no longer available';
return;
end if;

insert into accepted_match (match_id, donor_confirmation)
values (p_match_id, 'Y')
returning accepted_id into p_accepted_id;

update potential_match
set status = 'Accepted'
where match_id = p_match_id;

update organ
set status = 'Matched'
where organ_id = v_organ_id;

update potential_match set status = 'Cancelled'
where organ_id = v_organ_id and
match_id != p_match_id and status = 'Pending';

p_message := 'Match accepted successfully';
```

Procedure 3: sp_approve_match

Purpose: Handles admin approval workflow - the final step before hospital notification.

Procedure Definition:

What It Does:

- Fetches match details including recipient and hospital information
- Updates ACCEPTED_MATCH:
- Sets Admin_Approval to 'Y'
- Changes Match_Status to 'Approved'
- Records which admin approved (Approved_By)
- Creates HOSPITAL_NOTIFICATION record
- Updates RECIPIENT status to 'Matched'
- Trigger trg_accepted_match_organ_unavailable fires automatically to update organ status

```

begin
    select pm.match_id, r.recipient_id, r.hospital_id
    into v_match_id, v_recipient_id, v_hospital_id
    from accepted_match am
    join potential_match pm on am.match_id = pm.match_id
    join recipient r on pm.recipient_id = r.recipient_id
    where am.accepted_id = p_accepted_id;
exception
    when no_data_found then
        p_message := 'Accepted match not found';
        return;
end;

update accepted_match
set admin_approval = 'Y',
match_status = 'Approved',
approved_by = p_admin_id
where accepted_id = p_accepted_id;

insert into hospital_notification (accepted_id, hospital_id)
values (p_accepted_id, v_hospital_id);

update recipient
set status = 'Matched'
where recipient_id = v_recipient_id;

p_message := 'Match approved and hospital notified';
exception

```

What It Does:

- Fetches match details including recipient and hospital information
- Updates ACCEPTED_MATCH:
- Sets Admin_Approval to 'Y'
- Changes Match_Status to 'Approved'
- Records which admin approved (Approved_By)
- Creates HOSPITAL_NOTIFICATION record
- Updates RECIPIENT status to 'Matched'
- Trigger trg_accepted_match_organ_unavailable fires automatically to update organ status

4.4.3 Views Explanation

View 1: vw_active_donors_organ

Purpose: Shows all active donors and their listed organs.

Use Case: Admin dashboard showing all active donors and their organ listings.

```

create or replace view vw_active_donors_organ as
select d.donor_id, d.d_name as donor_name, d.age,
d.blood_type, d.status as donor_status,
o.organ_id, o.organ_type, o.status as organ_status
from donor d
join organ o on d.donor_id = o.donor_id
where d.status = 'Active';

```

Sample Output:

DONOR_ID	DONOR_NAME	AGE	BLO	DONOR_STATUS	ORGAN_ID	ORGAN_TYPE
3	Ahmed Malik	40	B-	Active	6	Pancreas
3	Ahmed Malik	40	B-	Active	5	Kidney
1	Ali Raza	30	A+	Active	2	Liver
1	Ali Raza	30	A+	Active	1	Kidney
5	Hassan Raza	35	AB+	Active	9	Liver
4	Marium Ali	32	A+	Active	7	Kidney
4	Marium Ali	32	A+	Active	8	Cornea
2	Sara Khan	28	O+	Active	3	Heart
2	Sara Khan	28	O+	Active	4	Kidney
6	Zainab Shah	45	O-	Active	10	Heart
6	Zainab Shah	45	O-	Active	11	Lung

View 2: vw_waiting_recipients

Purpose: Lists all recipients currently waiting for transplants.

CREATE OR REPLACE VIEW vw_waiting_recipients AS

```
SELECT r.recipient_id, r.r_name AS recipient_name,
       r.age, r.blood_type, r.organ_required,
       r.critical_level, r.priority_score, r.status AS recipient_status,
       h.hospital_id, h.h_name AS hospital_name
FROM RECIPIENT r
JOIN HOSPITAL h ON r.hospital_id = h.hospital_id
WHERE r.status = 'Waiting';
```

Use Case: Shows hospital which patients are still waiting, prioritized by critical level.

Sample Output:

RECIPIENT_NAME	AGE	BLO	ORGAN_REQUIRED	CRITICAL_L	PRIORITY_SCORE	RECIPIENT_ST
Mashif Ali	42	A+	Kidney	Medium	65	Waiting
Bilal Ahmed	45	B-	Kidney	Medium	60	Waiting
Hida Hassan	35	A+	Cornea	Medium	55	Waiting
Sadia Riaz	50	O+	Kidney	Low	45	Waiting
Fatima Malik	38	AB+	Liver	High	85	Waiting
Imran Ali	35	A+	Kidney	High	80	Waiting
Rabia Shah	40	A+	Lung	High	80	Waiting
Tariq Mahmood	55	B+	Pancreas	High	75	Waiting
Usman Tariq	28	AB+	Heart	Critical	98	Waiting
Ayesha Khan	30	O+	Heart	Critical	95	Waiting

View 3: vw_potential_matches

Purpose: Shows all potential matches with compatibility details.

Use Case: Donor views all their potential matches with scores to make informed decision.

```

create or replace view vw_potential_matches as
select pm.match_id, pm.organ_id, o.organ_type,
pm.recipient_id, r.r_name as recipient_name,
pm.compatibility_score, pm.blood_compatibility, pm.urgency_score,
pm.location_distance, pm.estimated_travel_time, pm.status as match_status
from potential_match pm
join organ o on pm.organ_id = o.organ_id
join recipient r on pm.recipient_id = r.recipient_id;

```

Sample Output:

MATCH_ID	ORGAN_ID	RECIPIENT_ID	COMPATIBILITY_SCORE	B	URGENCY_SCORE	MATCH_DATE	LOCATION_DISTANCE	ESTIMATED_TRAVEL_TIME	STATUS
1	1	1	100	Y	80	08-DEC-25	5	4.17	Pending
2	1	3	50	N	57	08-DEC-25	5	4.17	Pending
3	4	1	60	N	80	08-DEC-25	15	10	Pending
4	4	3	100	Y	57	08-DEC-25	15	10	Pending

View 4: vw_accepted_matches

Purpose: Shows all accepted matches with donor and recipient details.

Use Case: Admin reviews all accepted matches pending approval.

```

create or replace view vw_accepted_matches as
select am.accepted_id, am.match_id, am.donor_confirmation, am.admin_approval, am.match_status,
d.donor_id, d.d_name as donor_name, o.organ_type,
r.recipient_id, r.r_name as recipient_name, r.organ_required as recipient_organ,
am.acceptance_date, am.approved_by
from accepted_match am
join potential_match pm on am.match_id = pm.match_id
join organ o on pm.organ_id = o.organ_id
join donor d on o.donor_id = d.donor_id
join recipient r on pm.recipient_id = r.recipient_id;

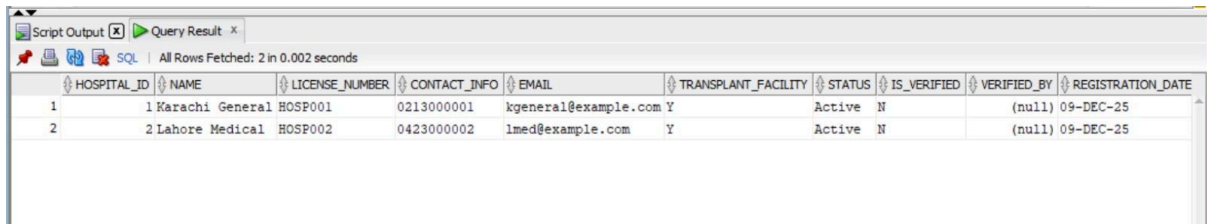
```

Sample Output:

ACCEPTED_ID	MATCH_ID	D	DONOR_CONFIRMATION_DATE	A	MATCH_STATUS	ACCEPTANCE_DATE	PATIENT_OUTCOME
1	1	Y		N	Pending_Approval	08-DEC-25 11.48.45.588000000 AM	

4.5 Sample Data Insertion and Testing

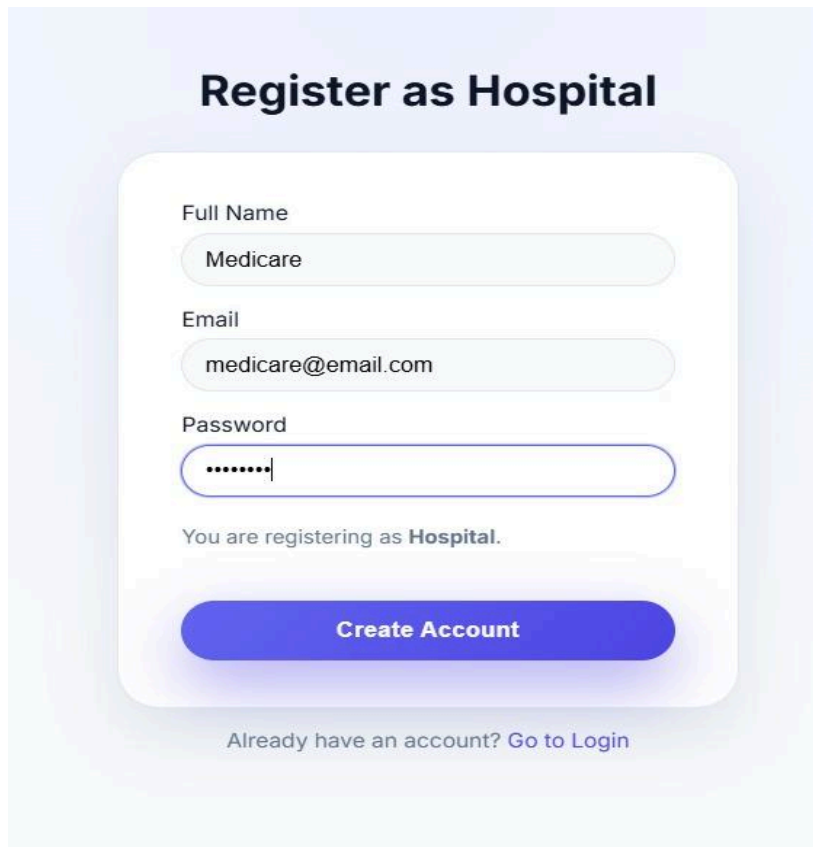
Test Data Insertion in application



Script Output x Query Result x

All Rows Fetched: 2 in 0.002 seconds

HOSPITAL_ID	NAME	LICENSE_NUMBER	CONTACT_INFO	EMAIL	TRANSPLANT_FACILITY	STATUS	IS_VERIFIED	VERIFIED_BY	REGISTRATION_DATE
1	1 Karachi General	HOSP001	0213000001	kgeneral@example.com	Y	Active	N	(null)	09-DEC-25
2	2 Lahore Medical	HOSP002	0423000002	lmed@example.com	Y	Active	N	(null)	09-DEC-25



Register as Hospital

Full Name

Medicare

Email

medicare@email.com

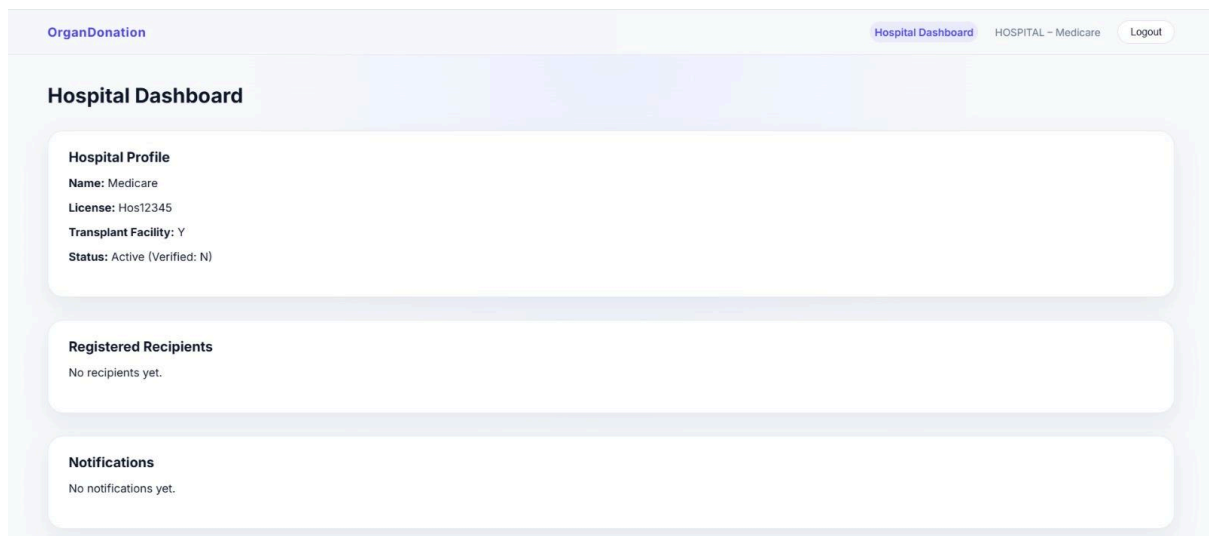
Password

.....

You are registering as **Hospital**.

Create Account

Already have an account? [Go to Login](#)



OrganDonation Hospital Dashboard HOSPITAL - Medicare Logout

Hospital Dashboard

Hospital Profile

Name: Medicare

License: Hos12345

Transplant Facility: Y

Status: Active (Verified: N)

Registered Recipients

No recipients yet.

Notifications

No notifications yet.

Script Output x

Query Result x

SQL

All Rows Fetched: 3 in 0.002 seconds

	HOSPITAL_ID	NAME	LICENSE_NUMBER	CONTACT_INFO	EMAIL	TRANSPLANT_FACILITY	STATUS	IS_VERIFIED	VERIFIED_BY	REGISTRATION_DATE
1	1	Karachi General	HOSP001	0213000001	kgeneral@example.com	Y	Active	N	(null)	09-DEC-25
2	2	Lahore Medical	HOSP002	0423000002	lmed@example.com	Y	Active	N	(null)	09-DEC-25
3	3	Medicare	Hos12345	033112298	medicare@email.com	Y	Active	N	(null)	09-DEC-25

OrganDonation

Admin DashboardADMIN – Usman AliLogout

Admin Dashboard

Pending Donors

Donor	Age	Blood	Status	Verify
Ali Raza	30	A+	Active	Verify
Sara Khan	28	O+	Active	Verify
Ahmed Malik	40	B-	Active	Verify
mariium	30	A+	Active	Verify
test	30	O+	Active	Verify

Pending Hospitals

Hospital	License	Transplant Facility	Status	Verify
Karachi General	HOSP001	Y	Active	Verify
Lahore Medical	HOSP002	Y	Active	Verify
Medicare	Hos12345	Y	Active	Verify

Hospital Dashboard

Hospital Profile

Name: Medicare

License: Hos12345

Transplant Facility: Y

Status: Active (Verified: Y)

Register New Recipient

Each recipient is registered by the hospital, with unique CNIC, one required organ, and medical priority info.

Name*

amna

Blood Type*

O+

Critical Level

Medium

CNIC*

42201-986468964

Weight (kg)

45

Priority Score (0-100)

50

Age*

21

Organ Required*

Kidney

Medical History

kidney failure

Emergency Contact

Name*

amra

Number*

83218225421

Relationship

mother

Register Recipient

Registered Recipients

ID	Name	Organ	Blood	Critical	Priority	Status
4	amna	Kidney	O+	Medium	50	Waiting

RECIPIENT_ID	HOSPITAL_ID	NAME	CNIC	AGE	BLOOD_TYPE	WEIGHT	ORGAN_REQUIRED	REGISTRATION_DATE	STATUS	CRITICAL_LEVEL	PRIORI
1	1	1 Imran Ali	3520212345678	35	A+	72	Kidney	09-DEC-25	Waiting	High	
2	2	2 Ayesha Khan	3520212345679	30	O+	65	Heart	09-DEC-25	Waiting	Critical	
3	3	1 Bilal Ahmed	3520212345680	45	B-	78	Kidney	09-DEC-25	Waiting	Medium	
4	4	3 amna	42201-986468964	21	O+	45	Kidney	09-DEC-25	Waiting	Medium	

OrganDonation

Donor DashboardDONOR - test2Logout

Donor Dashboard

Failed to load donor data

My Donor Profile

Name *

test2

Age *

65

Weight (kg)

67

Phone

1234567

Blood Type *

A+

Medical Condition

liver damage

Emergency Contact

Name *

namelest

Number *

123456

Relationship

mother

Address

Address *

pakistan

Area

kda

District

east

City

karachi

Province

sindh

Save Profile

Script Output xQuery Result x

All Rows Fetched: 6 in 0.005 seconds

DONOR_ID	NAME	AGE	WEIGHT	PHONE_CONTACT	EMAIL	REGISTRATION_DATE	BLOOD_TYPE	LAST_CHECKUP	MEDICAL_CONDITION	STATUS	IS_VERIFY
1	5 test	30	60	123456789	test1@email.com	09-DEC-25	O+	(null)	none	Active	Y
2	1 Ali Raza	30	70	03001230001	ali@example.com	09-DEC-25	A+	(null)	(null)	Active	N
3	2 Sara Khan	28	65	03001230002	sara@example.com	09-DEC-25	O+	(null)	(null)	Active	N
4	3 Ahmed Malik	40	80	03001230003	ahmed@example.com	09-DEC-25	B-	(null)	(null)	Active	N
5	4 marium	30	70	03001230004	marium@example.com	09-DEC-25	A+	(null)	(null)	Active	N
6	6 test2	65	67	1234567	test2@email.com	09-DEC-25	A+	(null)	liver damage	Active	N

OrganDonation

Admin DashboardADMIN - Usman AliLogout

Admin Dashboard

Pending Donors

Donor	Age	Blood	Status	Verify
Ali Raza	30	A+	Active	Verify
Sara Khan	28	O+	Active	Verify
Ahmed Malik	40	B-	Active	Verify
marium	30	A+	Active	Verify
test2	65	A+	Active	Verify

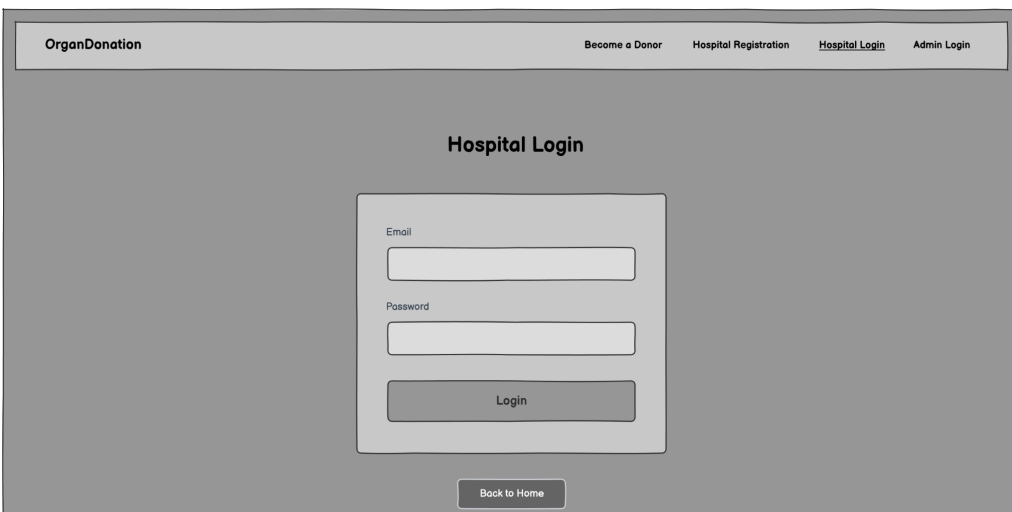
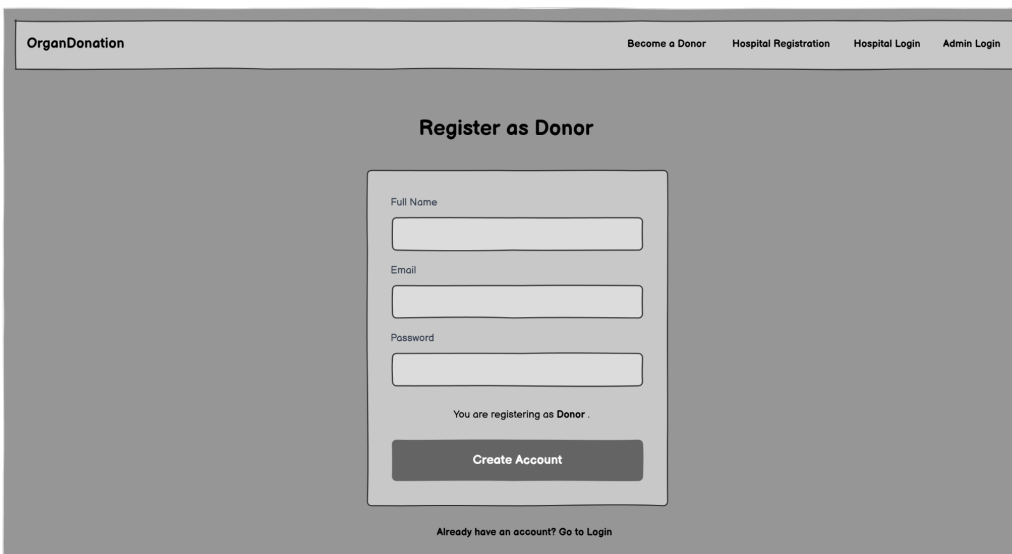
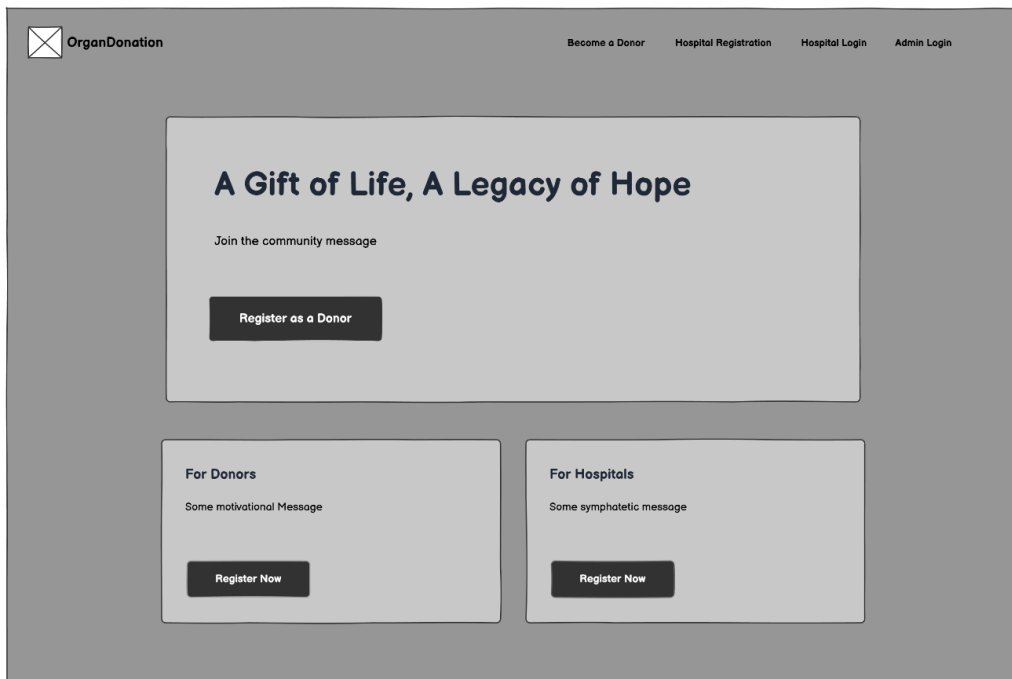
Pending Hospitals

Hospital	License	Transplant Facility	Status	Verify
Karachi General	HOSP001	Y	Active	Verify
Lahore Medical	HOSP002	Y	Active	Verify

Pending Matches for Approval

DONOR_ID	NAME	AGE	WEIGHT	PHONE_CONTACT	EMAIL	REGISTRATION_DATE	BLOOD_TYPE	LAST_CHECKUP	MEDICAL_CONDITION	STATUS	IS_VERIFY
1	5 test	30	60	123456789	test1@email.com	09-DEC-25	O+	(null)	none	Active	Y
2	1 Ali Raza	30	70	03001230001	ali@example.com	09-DEC-25	A+	(null)	(null)	Active	N
3	2 Sara Khan	28	65	03001230002	sara@example.com	09-DEC-25	O+	(null)	(null)	Active	N
4	3 Ahmed Malik	40	80	03001230003	ahmed@example.com	09-DEC-25	B-	(null)	(null)	Active	N
5	4 marium	30	70	03001230004	marium@example.com	09-DEC-25	A+	(null)	(null)	Active	N

5.1 Wireframes



OrganDonation

[Become a Donor](#)[Hospital Registration](#)[Hospital Login](#)[Admin Login](#)

Donor Login

Email

Password

Login

Back to Home

OrganDonation

[Donor Dashboard](#)[DONOR - User](#)

Logout

Donor Dashboard

My Donor Profile

Name

Age

Weight (kg)

Phone

Blood Type

A+ ▾

Medical Condition

Emergency Contact

Name *

Number *

Relationship

Address

Address *

Area

District

City

Province

Save Profile

OrganDonation

[Hospital Dashboard](#)[Hospital - Name](#)

Logout

Register as Hospital

Full Name

Email

Password

You are registering as Hospital .

Create Account

Already have an account? Go to Login

OrganDonation

Hospital DashboardHOSPITAL - omiLogout

Hospital Dashboard

Hospital Registration

Register your hospital with license and address information. An admin must verify your registration before you can register recipients.

Hospital Name*

License Number*

Address*

Contact Info

Transplant Facility
Yes

Area

District

City

Province

Save Hospital Profile

OrganDonation

Become a DonorHospital RegistrationHospital LoginAdmin Login


Admin Login

Email

Password

Login

Back to Home

OrganDonation

Admin DashboardADMIN - Admin1Logout

Admin Dashboard

Pending Donors

Donor	Age	Blood	Status	Verify
Donor1	30	A+	Active	Verify
Donor2	28	O-	Active	Verify
Donor3	40	B-	Active	Verify
Donor4	30	A+	Active	Verify

Pending Hospitals

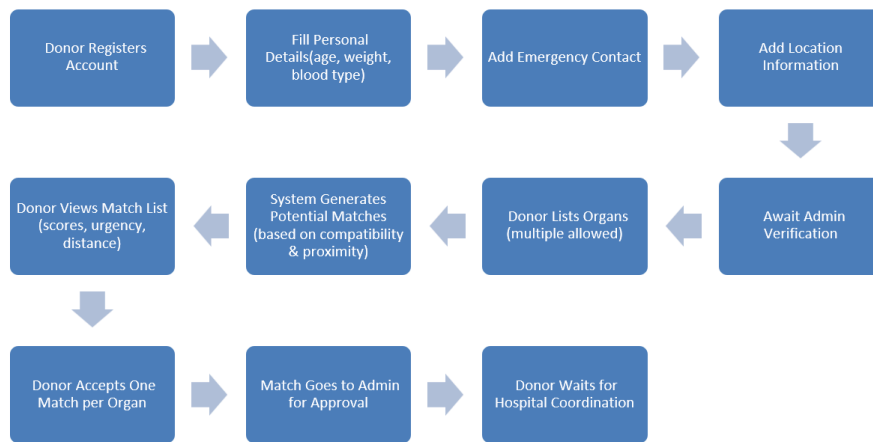
Hospital	License	Transplant Facility	Status	Verify
Hospital1	HOSP001	Y	Active	Verify
Hospital2	HOSP002	Y	Active	Verify

Pending Matches for Approval

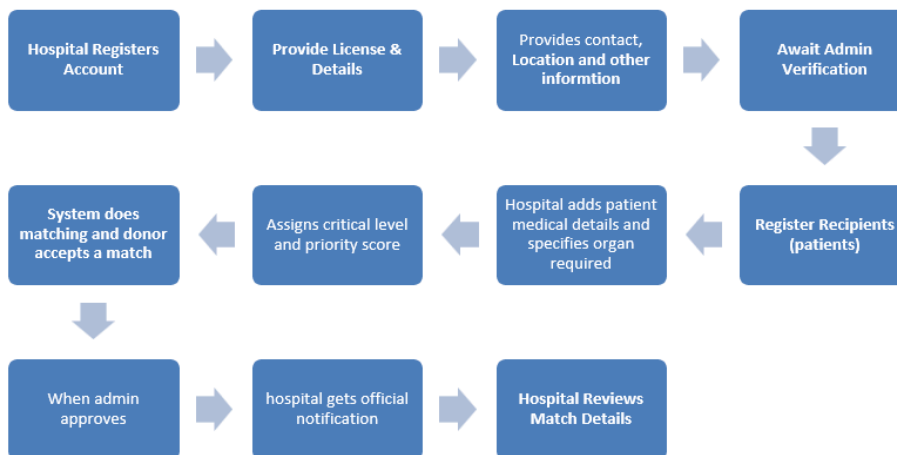
Accepted ID	Donor	Recipient	Organ	Status	Approve
1	Recipient1	Donor1	Kidney	Pending_Approval	Approve

5.2 Flow Diagrams of the Application

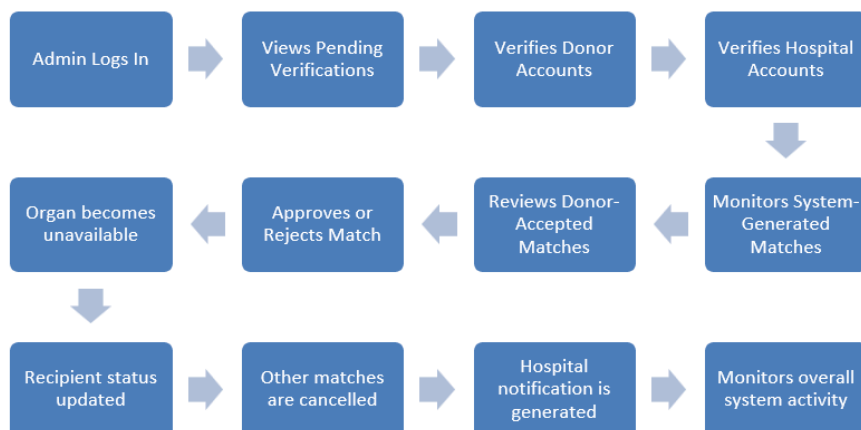
Donor Flow Diagram:



Hospital Flow Diagram:



Admin Flow Diagram:



6. SQL Integration

This section documents the main screens in the Organ Donation Management System and describes the SQL queries that power each screen.

1.1 Admin Login

Purpose: Authenticate admin and load their profile.

SQL Used:

```
SELECT Admin_Id, Name, Email, Phone_Contact, Role, Department
FROM ADMIN
WHERE Email = :p_email;
```

Contribution: Fetches admin identity and role so the system can determine access permissions and load the admin dashboard.

1.2 Admin Dashboard

Purpose: Provide a high-level overview of donors, organs, recipients and matches.

SQL Used (Counts and Summary Data):

```
SELECT COUNT(*) FROM DONOR WHERE Status = 'Active';
SELECT COUNT(*) FROM ORGAN WHERE Status = 'Available';
SELECT COUNT(*) FROM RECIPIENT WHERE Status = 'Waiting';
SELECT COUNT(*) FROM POTENTIAL_MATCH WHERE Status = 'Pending';
```

Contribution: Supplies key statistics for dashboard summary cards.

SQL (Views for Detailed Lists):

```
SELECT * FROM vw_active_donors_organs;
SELECT * FROM vw_waiting_recipients;
```

Contribution: These views display active donors with their organs, and all waiting recipients. They simplify dashboard rendering by gathering joined data in one place.

1.3 Manage Donors

Purpose: Allow admin to view, verify, and update donor information.

SQL Used:

```
SELECT d.*, ec.Contact_Name, al.City
FROM DONOR d
JOIN EMERGENCY_CONTACT ec ON d.Contact_Id = ec.Contact_Id
```

```
JOIN Add_Location al ON d.Location_Id = al.Location_Id;
```

Contribution: Populates the donor table with complete details.

Verification Update:

```
UPDATE DONOR
SET Is_Verified = 'Y', Verified_By = :p_admin_id
WHERE Donor_Id = :p_id;
```

Contribution: Enables admins to verify donor accounts, unlocking their participation in matching.

1.4 Manage Hospitals

Purpose: List hospitals and verify their registration.

Hospital List SQL:

```
SELECT h.Hospital_Id, h.Name, h.License_Number, h.Is_Verified
FROM HOSPITAL h;
```

Verification Update:

```
UPDATE HOSPITAL
SET Is_Verified = 'Y', Verified_By = :p_admin_id
WHERE Hospital_Id = :p_id;
```

Contribution: Ensures hospitals are legitimate before they can register recipients.

1.5 Match Management

Purpose: Allow admin to review, approve, and manage matches.

View Potential Matches:

```
SELECT * FROM vw_potential_matches;
```

Contribution: Displays compatibility score, urgency, and distance to support decisions.

Approve Match (Procedure):

```
EXEC sp_approve_match(:p_accepted_id, :p_admin_id);
```

Contribution:

- Updates match status

- Notifies hospital
- Changes organ and recipient states

Triggers automatically enforce state updates.

2.1 Donor Registration

Purpose: Register a new donor along with emergency contact and location.

SQL Used:

```
INSERT INTO EMERGENCY_CONTACT (...) VALUES (...);
INSERT INTO Add_Location (...) VALUES (...);
INSERT INTO DONOR(...) VALUES (...);
```

Contribution: Creates a complete donor profile following business rules (age limits, required emergency contact, valid blood type).

2.2 Donor Profile

Purpose: Display donor details and allow editing.

SQL Used:

```
SELECT d.*, ec.Contact_Name, al.City
FROM DONOR d
JOIN EMERGENCY_CONTACT ec ON d.Contact_Id = ec.Contact_Id
JOIN Add_Location al ON d.Location_Id = al.Location_Id
WHERE d.Donor_Id = :id;
```

Contribution: Retrieves full donor profile for viewing or editing.

2.3 Donor Organ Management

Purpose: Let donors add organs for donation and view existing organs.

List Organs:

```
SELECT Organ_Id, Organ_Type, Status
FROM ORGAN
WHERE Donor_Id = :p_id;
```

Add Organ:

```
INSERT INTO ORGAN(Donor_Id, Organ_Type, Organ_Size)
VALUES(:p_id, :type, :size);
```


Contribution: Donors supply organ data that is later used for automated matching.

3. Hospital Module

3.1 Hospital Registration

Purpose: Register a hospital with location and transplant capability.

SQL Used:

```
INSERT INTO Add_Location(...) VALUES(...);  
INSERT INTO HOSPITAL(...) VALUES(...);  
INSERT INTO HOSPITAL_ADMIN(...) VALUES(...);
```

Contribution: Creates a complete, normalized hospital record.

3.2 Recipient Registration

Purpose: Hospitals register recipients and assign medical priority.

SQL Used:

```
INSERT INTO RECIPIENT(Name, CNIC, Organ_Required, Critical_Level,  
Priority_Score)  
VALUES(...);
```

Contribution: These attributes are essential for compatibility scoring and urgency.

3.3 Potential Matches

Purpose: Show all matches involving recipients from that hospital.

SQL Used:

```
SELECT vpm.*  
FROM vw_potential_matches vpm  
JOIN RECIPIENT r ON vpm.Recipient_Id = r.Recipient_Id  
WHERE r.Hospital_Id = :p_id;
```

Contribution: Allows hospitals to monitor matches before approval.

3.4 Notifications

Purpose: Hospitals receive notifications after admin approval.

SQL Used:

```
SELECT * FROM vw_hospital_notifications  
WHERE Hospital_Id = :p_id;
```

Contribution: Enables coordination and scheduling after a match is approved.

Contribution Table:

Team Member	Assigned Tasks
Fatima Faisal	<ul style="list-style-type: none">• Frontend: Donor Portal (View Matches, Accept Match)• Backend: Match Generation Logic & Triggers• Database: Potential Match & Accepted Match Tables• Report: Business Rules, Use Cases, ERD Explanation
Marium Ali	<ul style="list-style-type: none">• Frontend UI Design (Landing Page, Login, Dashboard)• Recipient Registration Screens• Backend: Donor Registration & Organ Listing API• Database: Donor, Organ & Emergency Contact Tables• Report: Introduction, Business Scenario, Interview Analysis
Khadeja Qureshi	<ul style="list-style-type: none">• Frontend: Hospital Portal (Register Hospital, Add Recipients)• Backend: Admin Approval Workflow & Notifications• Database: Hospital, Recipient, Location Tables• Report: Normalization, Triggers/Procedures, Conclusion