

Blood Bank Management System

Mini Project Report -Database Lab (DSE 2260)
Department of Data Science & Computer Applications



B. Tech Data Science
4th Semester – Batch: B2

Submitted By

Shreyas 200968246

K.Sainath Reddy 200968240

Ayushi Singh 200968244

Arihant Gupta 200968248

Mentored By

Vinayak M

Assistant Professor-Senior

DSCA, MIT

Archana H/ Shameem

AssistantProfessor-Senior

DSCA, MIT



MANIPAL INSTITUTE OF TECHNOLOGY
MANIPAL
(A constituent unit of MAHE, Manipal)

Date:

CERTIFICATE

This is to certify that the Shreyas (200968246), K.Sainath Reddy(200968240), Ayushi Singh(200968244), Arihant Gupta(200968248), have successfully executed a mini project titled “Blood Bank management System” rightly bringing fore the competencies and skillsets they have gained during the course- Database Lab (DSE 2262 & DSE), thereby resulting in the culmination of this project.

Vinayak M

Assistant Professor-Senior

DSCA, MIT

Archana H/ Shameem

AssistantProfessor-Senior

DSCA, MIT

ABSTRACT

Blood Bank Management System (BBMS) is a database system to link between the donors and blood banks and act as an interface for the patient to find his/her desired blood in a fast and efficient way. It will make the blood transfusion service and its management more reliable and efficient than the conventional system.

Transfusion of blood and blood components is an established standard way of treating patients who are deficient in one or more blood constituents and is, therefore, an essential part of health care. A blood transfusion service is a complex organization requiring careful design and management. Essential functions of a blood transfusion service are donor recruitment, blood collection, testing of donor blood, component preparation, and supply of these components to the patients. The goal of blood transfusion service is to provide effective blood and blood components that are as safe as possible and adequate to meet the patients' needs.

The basic functions of a blood transfusion center may be listed as follows - Recruitment and retention of voluntary and replacement blood donors, collection, processing, storage, and transportation of blood and its components, laboratory procedures, participation in the clinical use of blood and blood components, teaching and training of personnel, research and development need for Blood Bank.

As a blood transfusion service deals with different functions related to donors and patients, it is imperative to keep in mind the safety of both donors and recipients. The blood transfusion service has to be planned and organized in such a way that it fulfills its ideal aims and objectives i.e. Recruitment of blood donors - voluntary & replacement, care of donor, donated unit and the recipient, maintaining adequate bloodstock, providing clinically effective blood components, optimal use of available blood.

The objectives of the project are to Improve operational & Streamlining operations, maintain all the project, employee, project database, maintain global standards of the project development, reduce Manpower and manual paper works for maintaining the records offline, main the accuracy, integrity and consistency of the data, providing such a mechanism to make the manpower fast to maintain all the information about the project, improved management and control of the inventory, sales, stock.

Contents

1. Introduction	1
2. Synopsis	2
2.1 Proposed System	2
2.2 Objectives	2
3. Functional Requirements	3
4. Detailed Design	7
4.1 ER Diagram	9
4.2 Schema Diagram	10
4.3 Data Dictionary	12
4.4 Relational Model Implementation	13
4.5 Queries	14
4.7 Triggers	..
4.8 Stored Procedures	..
4.9 Stored Functions	
5. Implementation Functional Requirements	
6. Testing No need to write testing	
7. Result	
8. Conclusion and Future Work	20
References not required	

Chapter 1

Introduction

Blood Bank Management is a challenging task in the healthcare industry to link the donors and blood banks and for the patient to find his/her desired blood in a fast and efficient way. The challenge is due to there being a short and limited supply of blood and if a patient cannot find his desired blood type quickly then it may cause harm or even death in some cases.

The organization of a blood transfusion service should receive the utmost attention and care for smooth functioning of various components of the service. The goal of blood transfusion service is to provide effective blood and blood components that are as safe as possible and adequate to meet the patients' needs.

Thus a well-established and efficient Blood Bank Management System needs to be set up to solve the inventory issues at different blood banks and to manage Donors and patients alike who want to donate blood or need blood respectively.

Advantages of using BBMS -

1. Blood Balance - It keeps track of different types of blood (A+, B+, AB+, etc.) and helps to figure out exactly what blood group has a short supply right now.
2. Accurate Planning - Using smart Blood Bank management, you can stay ahead of the demand curve, and keep the right amount of Blood types in supply.
3. Time saving - A well designed management system helps to save alot of time on recounting inventory.
4. Blood Tracking - For multiple blood banks, the management system becomes even more important as it would be more tedious to manage all the blood groups without a proper BBMS.

Chapter 2

Synopsis

2.1 Proposed System

The database is divided into four tables. The main tables are the patient, donor and bloodbank tables. The remaining table 'donate' is the relational table which links 'donor' and bloodbank tables with its foreign keys.

Database Table	Description
Patient	Contains all of the patient's information.
donor	Contains all of the donor's information.
BloodBank	Contains all of the blood bank's information.
Donate	Serves as the main linking table for the 'donor' and 'Bloodbank' table by containing the foreign keys of them.

2.2 Objectives

The Main Objective of the work are:

- To develop an application that can handle Blood donation and receiving needs across many blood banks.
- To develop the easy and reliable management of Blood in Blood banks.
- To keep a database of Blood donors and their Blood types.
- Integrity and consistency of the Database.
- To make the Blood supply manageable and simplify the use of inventory in the Blood bank.

Chapter 3

Functional Requirements

Login Form: A Valid Username and Password

Creation Of New Account: Name of the Patient/donor/Blood Bank, Blood Group(for a Patient/donor),Name of the Disease(for a Patient/donor), Address, Contact Number, Email

Update Records: Update/Modify any of these records

Search for desired blood group: Option for the Patient to find the desired blood from the database in different constraints and the quantity of blood available will be shown.

Search for desired blood bank: Option for the Patient to find the desired blood bank in different constraints

Search for the desired donor: Option for the Patient to find the desired donor in different constraints

Data consistency: Check for updated info about any patient, donor or blood bank

3.1 User Registering/Login module

Two lines about module briefly and it supports functionalities- New user registration, Login, Forgot password

3.1.1 New User Registration

The user must be able to create user id and password by supplying appropriate details.

INPUT	New username, Password, phone
Processing	The system must check availability of entered user name. Password must follow criteria- minimum 9 char, at least one capital, one number and one special character. Check for validity of phone number by prompting to enter OTP
OUTPUT	User created Successfully message / highlight the information entered which is wrong and allow to reenter.

3.1.2 Login

The existing user must be able to login upon entering proper user name and password.

INPUT	Username, Password
Processing	Check the user name and password against information stored in data storage
OUTPUT	If user entered correct user name & Password Login successful and open main application menu Else Display Login not successful, retry logging in

3.1.3 Forgot password

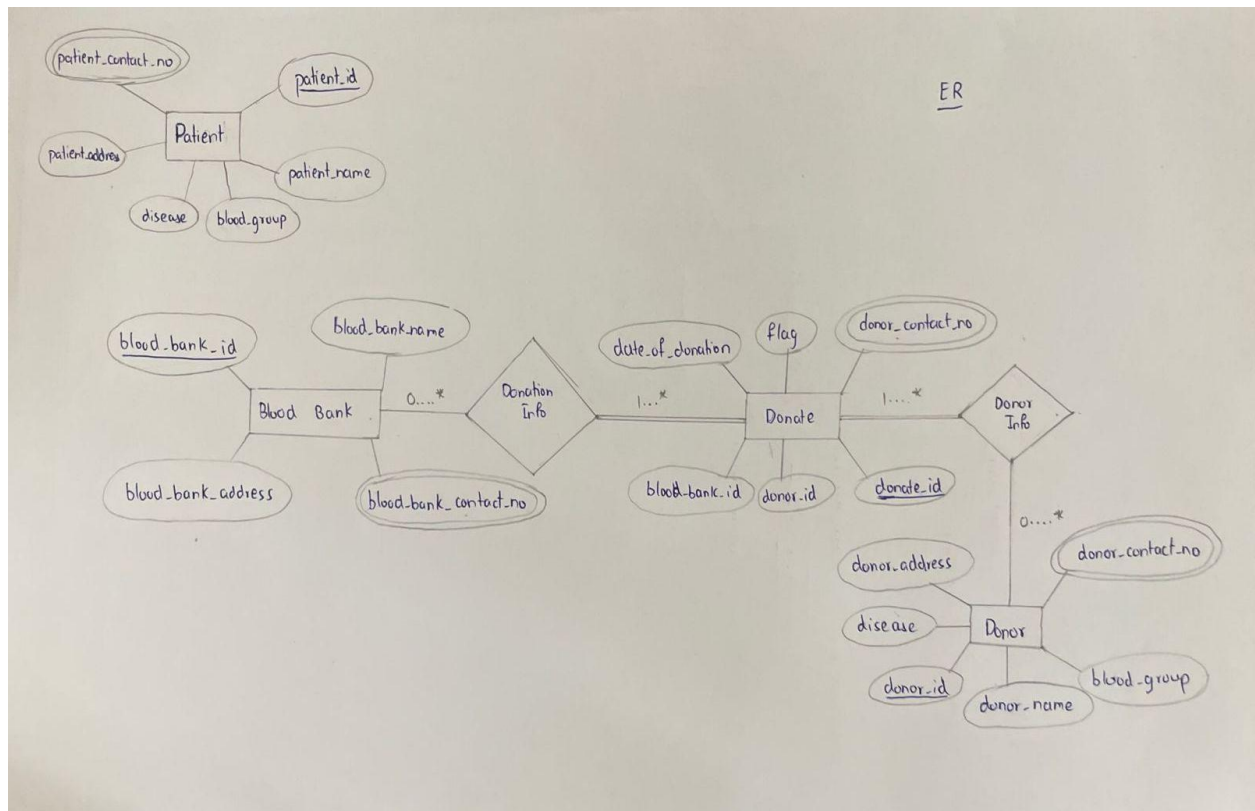
If the existing user name is not able to login, forgot password can be used to reset password.

INPUT	Prompt user to enter username, Phone
Processing	If username and corresponding phone exist in the data storage Send OTP to Phone. Prompt the user to enter OTP If OTP matching Prompt user to change password according to criteria. Else OTP not matching. Else User name and corresponding Phone not existing in the storage
OUTPUT	Password successfully changed / User name, phone not matching

Chapter 4

Detailed Design

4.1 ER Diagram



4.2 Schema Diagram

Patient (patient_id, patient_name, blood_group, disease, patient_address, patient_contact_no)

Donor(donor_id, donor_name, blood_group, donor_contact_no, donor_address, disease)

BloodBank(blood_bank_id, blood_bank_name, blood_bank_address, blood_bank_contact_no)

Donate(donate_id, donor_id, date_of_donation, blood_bank_id, flag integer)

Donor_info(Donor_id , Donate_id)

Donation_info(Donate_id , Blood_bank_id)

(Donate_id references Patient_id,donor_id references donor_id, blood_bank_id references blood_bank_id)

4.3 Data Dictionary

Patient

Column	Data type (size)	Constraint	Constraint Name
patient_id	integer	Primary key	
patient_name	varchar(20)	Not Null	sele
blood_group	varchar(4)		Check in ('O+', 'O-', 'A+', 'A-', 'B+', 'B-', 'AB+', 'AB-')
disease	varchar(20)		
patient_address	varchar(20)		
patient_contact_no	varchar(20)		Like +91-%

Donor

Column	Data type (size)	Constraint	Constraint Name
donor_id	integer	Primary Key	
donor_name	varchar(20)	Not Null	
blood_group	varchar(4)		Check in ('O+', 'O-', 'A+', 'A-', 'B+', 'B-', 'AB+', 'AB-')
donor_contact_no	varchar(20)		
donor_address	varchar(20)		
disease	varchar(20)		

BloodBank

Column	Data type (size)	Constraint	Constraint Name
--------	------------------	------------	-----------------

blood_bank_id	integer	Primary Key	
blood_bank_name	varchar(20)		
blood_bank_address	varchar(20)		
blood_bank_contact_no	varchar(20)		

Donate

Column	Data type (size)	Constraint	Constraint Name
donate_id	integer	Primary Key	
donor_id	integer	foreign key references donor(donor_id)	
date_of_donation	date		
blood_bank_id	integer	foreign key references BloodBank(blood_bank_id)	
flag integer	integer		

4.4 Relational Model Implementation

```
create table Patient(
    patient_id integer,
    patient_name varchar(20) not null,
    blood_group varchar(4),
    disease varchar(20),
    patient_address varchar(20),
    patient_contact_no varchar(20),
    primary key (patient_id)
);
```

```
create table donor(
    donor_id integer,
    donor_name varchar(20) not null,
    blood_group varchar(4),
    donor_contact_no varchar(20),
    donor_address varchar(20),
    disease varchar(20),
    primary key (donor_id)
```

);

```
create table BloodBank(  
    blood_bank_id integer,  
    blood_bank_name varchar(20),  
    blood_bank_address varchar(20),  
    blood_bank_contact_no varchar(20),  
    primary key (blood_bank_id)  
);
```

```
create table Donate(  
    donate_id integer ,  
    donor_id integer,  
    date_of_donation date,  
    blood_bank_id integer,  
    flag integer default 1,  
    primary key(donate_id),  
    foreign key (donor_id) references donor(donor_id),  
    foreign key (blood_bank_id) references BloodBank(blood_bank_id)  
);
```

4.5 Insertion

BLOODBANK

```
insert into BloodBank values(1,'BBA','Delhi',9114562342);  
Insert into BloodBank values(2,'BBB','Manipal',9114562343);  
Insert into BloodBank values(3,'BBC','Banglore',9114562344);  
Insert into BloodBank values(4,'BBD','Kolkata',9114562345);  
Insert into BloodBank values(5,'BBE','Mumbai',9114562346);  
Insert into BloodBank values(6,'BBF','Hyderabad',9114562347);
```

DONOR

```
Insert into DONOR values(10,'DA','A+',9755598466 ,'Hyderabad',NULL);  
Insert into DONOR values(11,'DB','O-',9154876547,'Manipal',NULL);  
Insert into DONOR values(12,'DD','AB+',9124873598,'Delhi',NULL);  
Insert into DONOR values(13,'DF','B+',9157820498,'Mumbai',NULL);  
Insert into DONOR values(14,'DG','O+',7706482157,'Kolkata',NULL);
```

PATIENT

```
Insert into Patient values(1234,'Aryan','O+', 'hemoglobinopathies', 'Manipal', '8749621587');  
Insert into Patient values(1235,'Ayushi','A+', 'bone_marrow_diseases', 'Delhi', '9784572657');
```

Insert into Patient values(1236,'Arihant','AB+','sickle_cell_anemia','Kolkata','74589658123');
Insert into Patient values(1237,'Shreyas','O-','leukemia','Mumbai','8457894583');
Insert into Patient values(1238,'Sainath','B+','lymphoma','Hyderabad','9874582158');

DONATE

Insert into Donate values(101,11,'27-MAY-2022',2,1);
Insert into Donate values(102,12,'28-MAY-2022',4,1);
Insert into Donate values(103,14,'29-MAY-2022',1,1);
Insert into Donate values(104,13,'24-MAY-2022',5,1);
Insert into Donate values(105,10,'22-MAY-2022',3,1);

4.6 Queries

4.6.1

donor available for a particular patient of specific blood group.

```
select donor_id,donor_name,blood_group,donor_contact_no from donor where donor_id  
in (select donor_id from Donate where donor_id in (select donor_id from donor where  
blood_group in (select blood_group from Patient where patient_id=1234)) );
```

4.6.2

Blood bank where desired blood available for a particular patient

```
select blood_bank_id,blood_bank_name,blood_bank_contact_no from BloodBank where  
blood_bank_id in (select blood_bank_id from Donate where donor_id in (select donor_id  
from donor where blood_group in (select blood_group from Patient where  
patient_id=1235)));
```

4.6.3

Blood bank where desired blood available for a particular patient and the blood bank is located in patient's location

```
select blood_bank_id,blood_bank_name,blood_bank_contact_no,blood_bank_address  
from BloodBank where blood_bank_id in (select blood_bank_id from Donate where  
donor_id in (select donor_id from donor where blood_group = (select blood_group from  
Patient where patient_id=1236)) ) and blood_bank_address in (select patient_address  
from Patient where patient_id=1236);
```

4.6.4

Blood bank where desired blood available

```
select blood_bank_id,blood_bank_name,blood_bank_contact_no from BloodBank where
blood_bank_id in (select blood_bank_id from Donate where donor_id in (select donor_id
from donor where blood_group = 'A+'));
```

4.6.5

Quantity of desired blood in a particular blood bank

```
select count(donate_id) as quantity_of_desired_blood from Donate where
blood_bank_id = 1 and donor_id in (select donor_id from donor where blood_group =
'A+');
select d.donor_name,b.blood_bank_name from donor d,BloodBank b,Donate dd where
dd.donate_id=101 and d.donor_id in (select donor_id from Donate where donate_id =
101) and b.blood_bank_id in (select blood_bank_id from Donate where donate_id =
101);
select donor_name,blood_group,patient_name from donor natural join Patient;
```

4.7 Triggers

Problem 1: In inserting any donor information if he/she has any fatal disease, it will not be inserted:-

```
create or replace trigger disease_trg
before insert on donor
for each row
when(NEW.donor_id>0)
Begin
    if :NEW.disease = 'HepB' then
        dbms_output.put_line('donor with fatal disease!');
        delete from donor where donor_id = :NEW.donor_id;
    end if;
End;
/
```

Problem 2: If there is any update of donor's location it will be displayed:

```
create or replace trigger loc_change
before update on donor
for each row
when(NEW.donor_address!=OLD.donor_address)
declare
Begin
    dbms_output.put_line('Old location : '|| :OLD.donor_address);
    dbms_output.put_line('New location : '|| :NEW.donor_address);
End;
/
```

Problem 3: If any blood bank is given to patient it will be updated & displayed
 create or replace trigger blood_received
 before update on Donate
 for each row
 when(NEW.flag!=OLD.flag)
 Declare
 begin
 dbms_output.put_line('Blood bag given to the patient');
 End;
 /

4.8 Stored Procedures

4.8.1 Procedure for donors for a particular patient:

```
set serveroutput on;
create or replace PROCEDURE donors_for_particular_patient (pid IN Patient.patient_id%type)
IS
BEGIN
  DBMS_OUTPUT.PUT_LINE('List of donors for patient id no. '||pid);
  DBMS_OUTPUT.PUT_LINE('-----');

  FOR cursor1 IN (select donor_id,donor_name,blood_group,donor_contact_no from donor
  where donor_id in (select donor_id from Donate where donor_id in (select donor_id from donor
  where blood_group in (select blood_group from Patient where patient_id=pid)) ))
  LOOP
    DBMS_OUTPUT.PUT_LINE('donor name: ' || cursor1.donor_name ||
      ', blood group: ' || cursor1.blood_group||
      ', contact no: ' || cursor1.donor_contact_no);
  END LOOP;
  DBMS_OUTPUT.PUT_LINE('_____');
END;
/
```

4.8.2 Procedure for bloodgroup quantity in a bank

```
set serveroutput on;
create or replace PROCEDURE blood_group_quantity_in_bank (bank_no IN
BloodBank.blood_bank_id%type) IS
BEGIN
  DBMS_OUTPUT.PUT_LINE('List of blood group with quantity in blood bank no. '||bank_no);
  DBMS_OUTPUT.PUT_LINE('-----');
```

```

FOR cursor1 IN (select blood_group,count(*) as quantity from (select * from Donate natural
join donor) where blood_bank_id = bank_no group by blood_group)
LOOP
    DBMS_OUTPUT.PUT_LINE('blood group: ' || cursor1.blood_group ||
                          ', quantity: ' || cursor1.quantity
                          );
END LOOP;
DBMS_OUTPUT.PUT_LINE('_____');
END;
/

```

5. Functional Requirement Implementation

The database and web application allow technicians to complete the following functions:

- Insert a new patient's information.
- Insert a new donor's information.
- Insert a new blood's bank information.
- Update/modify any of these records.
- Find the desired blood from the database in different constraints.
- Find the quantity of available desired blood in different constraints.
- Find the desired blood bank in different constraints.
- Find the desired donor in different constraints.
- Check any updated information about any patient, donor or blood bank.

8. Conclusion and Future Work

8.1 Conclusion

....

8.2 Scope for future work

....

References