

B008 Shaivi Bansal
B067 Khushi Shah
B075 Trisha Tanwar

Requirement Document

Introduction:

The Blood Bank Management System (BBMS) is a programme for storing, processing, retrieving, and analysing data related to administrative and inventory management in a blood bank.

This initiative intends to keep track of all information about blood donors and the various blood groups accessible in the blood bank, as well as to assist them in better managing their operations.

The project's goal is to increase transparency in this industry, making acquiring blood from a blood bank simple and hassle free, and improve the blood bank administration system.

Motivation:

In India, access to safe blood has long been a challenge, but the Covid-19 outbreak and lockdown have made it much worse. Apart from significant medical and surgical treatments that are required to save lives in the event of an illness or accident, blood is also essential during natural disasters and maternal care. Those with thalassemia, who require regular blood transfusions, are particularly susceptible during a blood shortage.

The epidemic of COVID 19 exposed various vulnerabilities in India's healthcare system. The country's hospitals were overburdened, and the system was on the verge of collapsing. Poor management was the biggest crack in the armour.

As a result, we were inspired to build a Blood Bank Management System that was both efficient and simple to use.

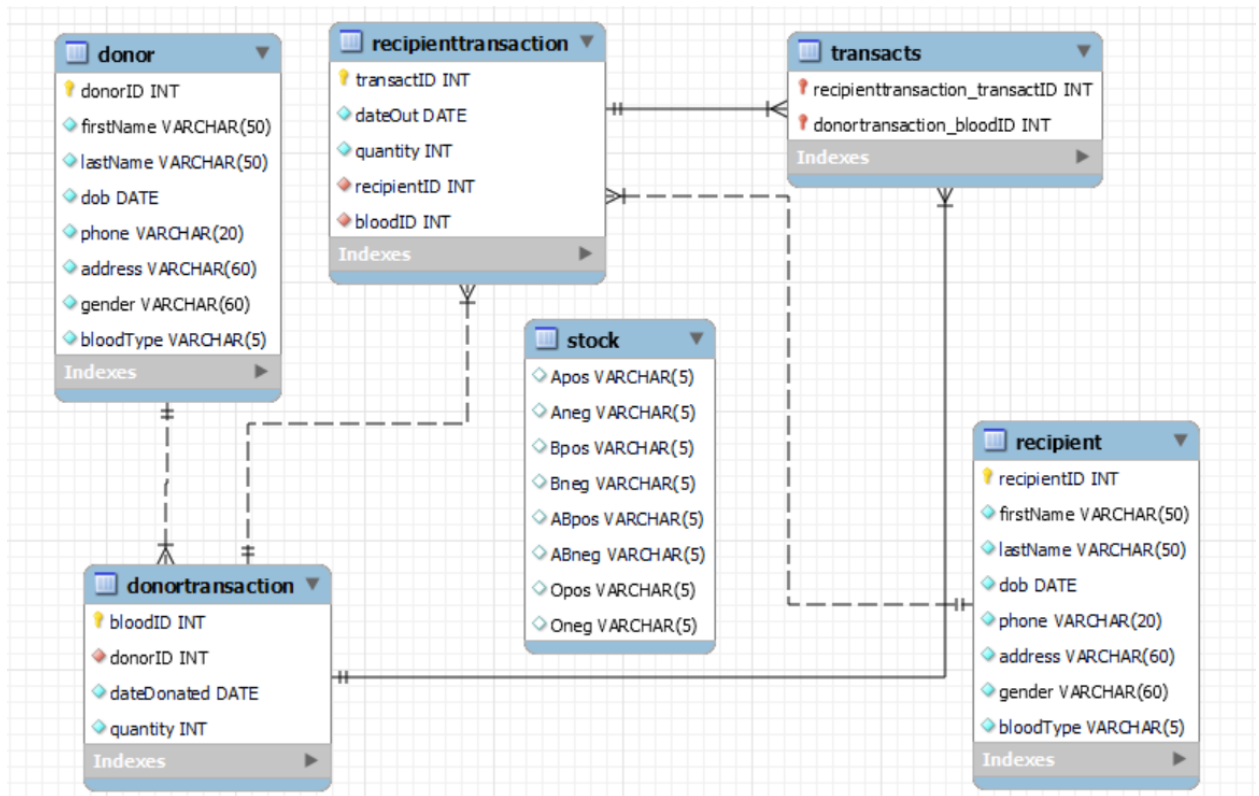
Tools:

- MySQL Workbench
- MySQL Command Line client
- Java JDK 15
- Apache NetBeans IDE 12.5

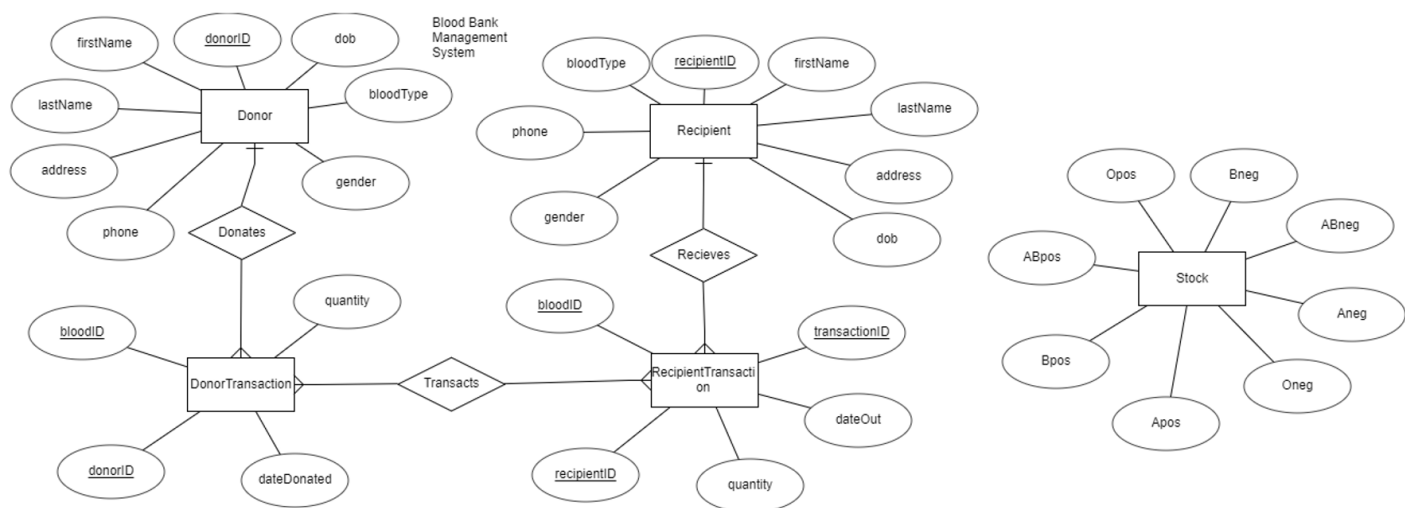
Process:

- Login - Admin login where username and password is asked for
- Home Page - We are directed to the home page from where one can access all our functionalities
- New Donor - We can register a new donor where the donor is assigned a donor id and is asked for: first name, last name, gender, blood group, mobile number, address, date of birth.
- Update Donor details - Any of the above details of an already existing donor can be updated.
- Delete Donor - A donor which exists in the database can be deleted.
- Display Donor Details - Using donor id all details of a particular donor can be displayed.
- Search Donor - All donors of a particular blood group can be searched.
- Display Stock Details - The units of blood present in each blood group can be displayed
- Increment Of Blood units in Stock - When a new donor is registered, the stock is updated by 1 unit of blood for his particular blood group.
- New Recipient - We can register a new recipient where the recipient is assigned a recipient id and is asked for: first name, last name, gender, blood group, mobile number, address, date of birth.
- Update Recipient details - Any of the above details of an already existing recipient request can be updated.
- Delete Recipient - A recipient request which exists in the database can be deleted.

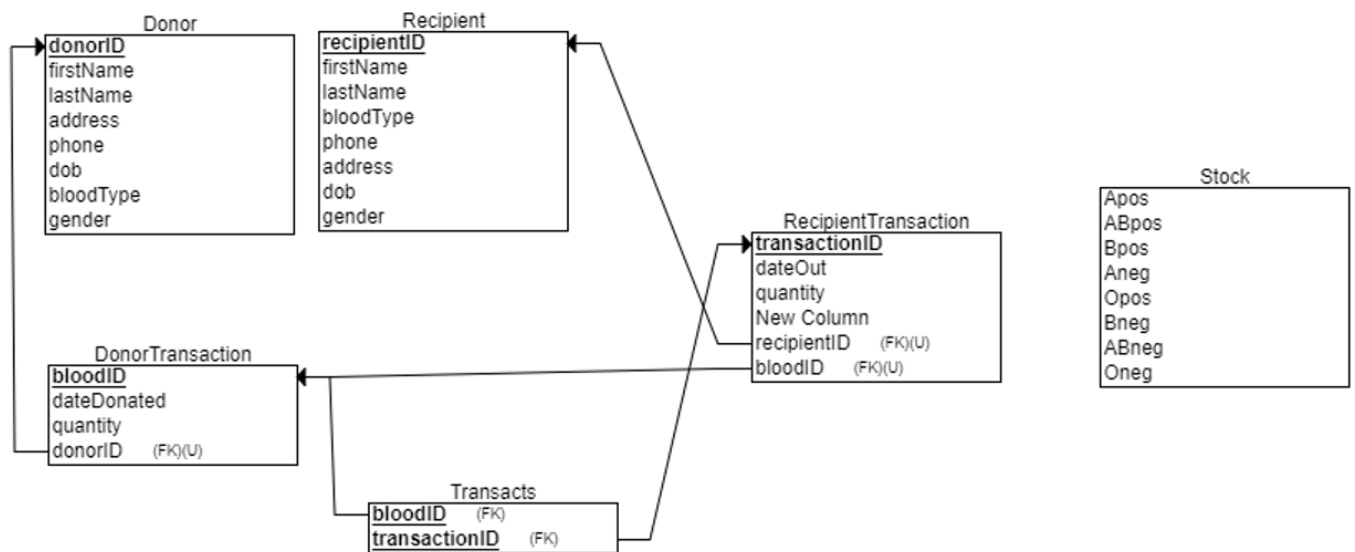
Schema:



ER Diagram:



Relationship Model:



SQL Code:

Create Database:

```
CREATE DATABASE BloodDonation;  
use BloodDonation;
```

Create Tables:

```
CREATE TABLE Donor(  
donorID INT(11) AUTO_INCREMENT NOT NULL ,  
firstName VARCHAR( 50 ) NOT NULL,  
lastName VARCHAR( 50 ) not null,  
dob DATE not null,  
phone VARCHAR( 20 ) not null,  
address VARCHAR( 60 ) not null,  
gender VARCHAR( 60 ) not null,  
bloodType varchar(5) NOT NULL ,  
PRIMARY KEY ( donorID )  
);
```

```
CREATE TABLE Recipient(  
recipientID INT(11) AUTO_INCREMENT  
NOT NULL ,
```

```
firstName VARCHAR( 50 ) NOT NULL,  
lastName VARCHAR( 50 ) not null,  
dob DATE not null,  
phone VARCHAR( 20 ) not null,  
address VARCHAR( 60 ) not null,  
gender VARCHAR(60) not null,  
bloodType varchar(5) NOT NULL,  
PRIMARY KEY ( recipientID )  
);
```

```
CREATE TABLE DonorTransaction(  
bloodID INT( 11 ) AUTO_INCREMENT,  
donorID INT( 11 ) NOT NULL ,  
dateDonated DATE NOT NULL ,  
quantity INT NOT NULL ,  
PRIMARY KEY ( bloodID ) ,  
FOREIGN KEY ( donorID ) REFERENCES  
Donor( donorID )  
);
```

```
CREATE TABLE RecipientTransaction(  
transactID INT( 11 ) AUTO_INCREMENT,  
dateOut DATE NOT NULL ,  
quantity INT NOT NULL ,  
recipientID INT( 11 ) NOT NULL,  
bloodID INT( 11 ) NOT NULL,  
PRIMARY KEY ( transactID ) ,  
FOREIGN KEY ( recipientID ) REFERENCES  
Recipient( recipientID ) ,  
FOREIGN KEY ( bloodID ) REFERENCES  
DonorTransaction( bloodID )  
);
```

```
CREATE TABLE Stock(  
Apos varchar(5),  
Aneg varchar(5),  
Bpos varchar(5),  
Bneg varchar(5),  
ABpos varchar(5),  
ABneg varchar(5),  
Opos varchar(5),  
Oneg varchar(5),  
);
```

Add new donor: INSERT

```
st.executeUpdate("insert into donor  
values("+donorID+", "+firstName+", "+lastName+", "+dob+", "+phone+", "+address+", "+gender  
+", "+bloodType+")");
```

Add new recipient

```
st.executeUpdate("insert into recipient  
values("+recipientID+", "+firstName+", "+lastName+", "+dob+", "+phone+", "+address+", "+gen  
der+", "+bloodType+")");
```

Add new recipient: INSERT

```
st.executeUpdate("insert into recipient  
values("+recipientID+", "+firstName+", "+lastName+", "+dob+", "+phone+", "+address+", "+gen  
der+", "+bloodType+")");
```

Update donor details: UPDATE

```
ResultSet rs=st.executeQuery("select * from donor where donorID="+donorID+"");
```

```
st.executeUpdate("update donor set  
firstName="+firstName+", lastName="+lastName+", dob="+dob+", phone="+mobilen+", addre  
ss="+address+", gender="+gender+", BLOODTYPE="+bloodType+" where  
donorID="+donorID+"");
```

Update Recipient Details: UPDATE

```
ResultSet rs=st.executeQuery("select * from recipient where recipientID="+recipientID+"");
```

```
st.executeUpdate("update recipient set  
firstName="+firstName+", lastName="+lastName+", dob="+dob+", phone="+mobilen+", addre  
ss="+address+", gender="+gender+", BLOODTYPE="+bloodType+" where  
recipientID="+recipientID+"");
```

Search All Donors Of a Particular Blood Type:

```
ResultSet rs=st.executeQuery("select * from donor where BLOODTYPE like  
'%"+BLOODTYPE+"%'");
```

Search All Recipients Of a Particular Blood Type:

```
ResultSet rs=st.executeQuery("select * from recipient where BLOODTYPE like  
'%"+BLOODTYPE+"%'");
```

Display Donor Details: INNER JOIN

```
ResultSet rs=st.executeQuery("select  
d.donorID,b.bloodID,d.firstName,d.lastName,d.gender,d.dob,d.bloodType,b.dateDonated from  
donor d inner join donorTransaction b on d.donorID=b.donorID;");
```

Display Recipient Details: READ

```
ResultSet rs=st.executeQuery("select * from recipient");
```

Delete Donor: DELETE

```
ResultSet rs=st.executeQuery("select * from donor where DonorID='"+DonorID+"'");
```

```
st.executeUpdate("delete from donor where DonorID='"+DonorID+"'");
```

Delete Recipient: DELETE

```
ResultSet rs=st.executeQuery("select * from recipient where RecipientID='"+RecipientID+"'");
```

```
st.executeUpdate("delete from donor where RecipientID='"+RecipientID+"'");
```

```
ResultSet rs=st.executeQuery("select * from stock");
```

Display Stock Details:

```
ResultSet rs=st.executeQuery("select * from stock");
```

Automatically Updates Stock When new Donor Added: UPDATE

```
String bloodType=null;
```

```
int tableunit;
```

```
String q2= "update Stock set "+bloodType.toUpperCase()+"="+bloodType.toUpperCase()+"1  
where "+bloodType.toUpperCase()+"="+tableunit+";";
```

Screenshots:

SQL Screenshots:

```
mysql> show tables;
```

Tables_in_blooddonation	
donor	
donortransaction	
recipient	
recipienttransaction	
stock	

```
5 rows in set (0.03 sec)
```

```
mysql> desc donor;
```

Field	Type	Null	Key	Default	Extra
donorID	int	NO	PRI	NULL	auto_increment
firstName	varchar(50)	NO		NULL	
lastName	varchar(50)	NO		NULL	
dob	date	NO		NULL	
phone	varchar(20)	NO		NULL	
address	varchar(60)	NO		NULL	
gender	varchar(60)	NO		NULL	
bloodType	varchar(5)	NO		NULL	

```
8 rows in set (0.01 sec)
```

```
mysql> desc donortransaction;
```

Field	Type	Null	Key	Default	Extra
bloodID	int	NO	PRI	NULL	auto_increment
donorID	int	NO	MUL	NULL	
dateDonated	date	NO		NULL	
quantity	int	NO		NULL	

```
4 rows in set (0.00 sec)
```



```
mysql> desc recipient;
```

Field	Type	Null	Key	Default	Extra
recipientID	int	NO	PRI	NULL	auto_increment
firstName	varchar(50)	NO		NULL	
lastName	varchar(50)	NO		NULL	
dob	date	NO		NULL	
phone	varchar(20)	NO		NULL	
address	varchar(60)	NO		NULL	
gender	varchar(60)	NO		NULL	
bloodType	varchar(5)	NO		NULL	

```
8 rows in set (0.00 sec)
```

```
mysql> desc recipienttransaction;
```

Field	Type	Null	Key	Default	Extra
transactID	int	NO	PRI	NULL	auto_increment
dateOut	date	NO		NULL	
quantity	int	NO		NULL	
recipientID	int	NO	MUL	NULL	
bloodID	int	NO	MUL	NULL	

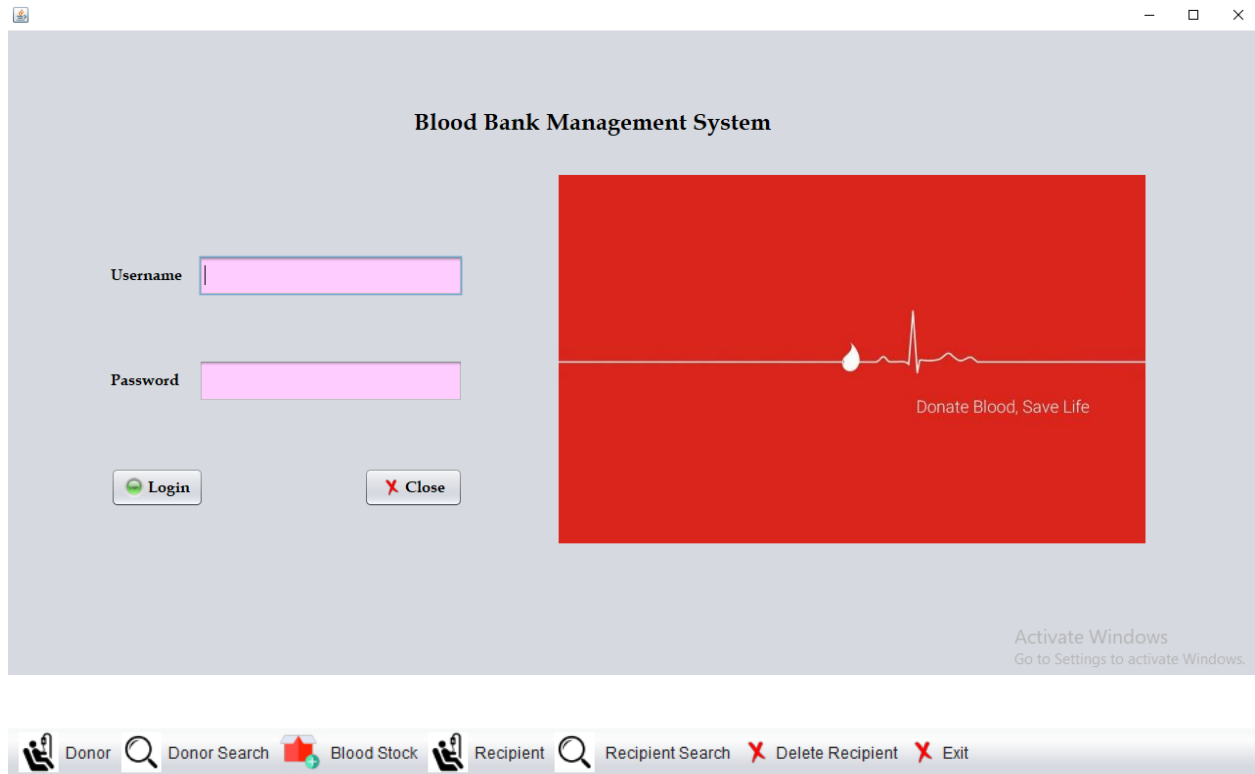
```
5 rows in set (0.00 sec)
```

```
mysql> desc stock;
```

Field	Type	Null	Key	Default	Extra
Apos	varchar(5)	YES		NULL	
Aneg	varchar(5)	YES		NULL	
Bpos	varchar(5)	YES		NULL	
Bneg	varchar(5)	YES		NULL	
ABpos	varchar(5)	YES		NULL	
ABneg	varchar(5)	YES		NULL	
Opos	varchar(5)	YES		NULL	
Oneg	varchar(5)	YES		NULL	

```
8 rows in set (0.00 sec)
```

GUI Screenshots:



New Donor Details

Donor ID	<input type="text" value="1"/>	Mobile Number	<input type="text" value="8291137488"/>
First Name	<input type="text" value="Shaivi"/>	Address	<div><div>Kharghar, Navi Mumbai</div></div>
Last Name	<input type="text" value="Bansal"/>		
Gender	<div>Female</div>	Date of Birth	<div>DDMMYYYY</div> <div><div>18</div><div>1</div><div>2002</div></div>
Blood Group	<div>Apos</div>		

Save Details

Clear

Exit

Update Donor Details

Donor ID

Search

First Name	<input type="text" value="Shaivi"/>	Address	<div><div>Kharghar, Navi Mumbai</div></div>
Last Name	<input type="text" value="Bansal"/>		
Date of ...	<input type="text" value="2002-01-18"/>	Gender	<input type="text" value="Female"/>
Mobile Number	<input type="text" value="8291137488"/>	Blood Group	<input type="text" value="Apos"/>

Save Details

Clear

Exit

Search for Donor

Blood Group

Apos

Search

donorID	firstName	lastName	dob	phone	address	gender	BLOODTY...
1	Shaivi	Bansal	2002-01-18	8291137488	Kharghar, Navi Mu...	Female	Apos

Clear

Close

Donor Details

donorID	bloodID	firstName	lastName	gender	dob	bloodType	dateDonated
1	10	Shaivi	Bansal	Female	2002-01-18	Apos	2021-09-25
2	11	Niyati	Mehta	Female	2000-07-20	Aneg	2021-09-28
3	12	Rutvi	Mehta	Female	1995-10-24	Bpos	2021-10-15
4	13	Vidhi	Goyal	Female	1992-11-05	Bneg	2021-10-23
5	14	Stuti	Govil	Female	1984-07-12	Opos	2021-10-24
6	15	Damni	Narang	Female	1992-03-04	Oneg	2021-10-13
7	16	Aryan	Gupta	Male	2001-12-03	ABpos	2021-06-14
8	17	Chirag	Adhikari	Male	1993-09-15	ABneg	2021-05-12

Close



Blood Stock

APOS	ANEG	BPOS	BNEG	ABPOS	ABNEG	OPOS	ONEG
2	1	1	1	1	1	1	1

Close



New Recipient Details

Recipient ID

5

Mobile Number

7483648789

First Name

Angella

Address

Sector 23, Noida

Last Name

Arora

Blood Group

Opos

Gender

Female

Date of Birth

4

7

1980

Save Details

Clear

Close

Search for Recipient

Blood Group

Apos

Search

recipientID	firstName	lastName	dob	phone	address	gender	bloodType
1	Aastha	Sur	1998-08-06	8756912367	Powai, Mumbai	Female	Apos

Clear

Close

Delete Recipient

Recipient ID

1

Search

First Name

Aastha

Address

Powai, Mumbai

Last Name

Sur

Date Of Birth

1998-08-06

Gender

Female

Mobile Number

8756912367

Blood Group

Apos

Delete

Clear

Close

Future Scope:

- System gives a warning once stored blood expires.
- Any quantity of blood can be donated or requested when registering donor/recipient (currently only 1 unit).
- Categorize the blood based on how old it is.
- System gives warning when there is shortage in the stock of blood in a particular blood group.
- When there is a shortage of blood the system recommends potential donors based on the previously registered donors in the database

Conclusion:

The project was a great learning experience for all of us. We enhanced our technical knowledge by learning how to create a GUI using JAVA. We also strengthened the DBMS concepts we had learned in class through practical implementation.

We learned several other values such as teamwork, task management and time management which will help us in our future endeavours.