PROJECT REPORT

Final Submission

Subject:- Blood Bank Management System

Course Instructor: Dr.Antriksh Goswami

Course Code: CS262

Group members:-

CHINIGEPALLI CHAKRADHAR SRINIVAS - 201951048

GANUGUCHINTALA RAJASEKHAR - 201951062

MUDADLA SIVA SAI - 201951096

PENUBALA DEERAJ - 201951113

Table Of Contents

Preface

1. We have used mysql workbench of database
2. We have written our code in java language,so we have used eclipse as a compiler and text editor.
3. We used a mysql connector to connect the java file and sql script.

Steps to Execute:-

1. Import the .java file from the zip file to the eclipse folder,then compile and execute.
2. Make sure that you have changed all the paths for images and for databases in sql as per your system requirements.

Note:Please note that we have used java framework for our frontend part,So please make sure that you have imported all the packages which are required to execute.

What is a Blood bank management system?

The blood bank management system is the one which is particularly aimed at saving lives in case of accidents and operations where usually death occurs due to absence of blood, this system helps you to find any blood group at difficult times.

Existing System:-

The operation of the blood bank still now is maintained in the manual system.
The operation is tedious, time-consuming, and space-consuming.
It creates room for errors as the data is entered manually by the persons
It includes the risk of the documents being lost over years and maintenance of the records is difficult.
The data recorded during testing or while acquiring the details of different aspects of the blood bank management system is not so accurate and precise.
Maintaining the stock of blood and daily transactions without computerization also poses a challenge

Why Blood bank management system?

The operation of the blood bank still now is maintained in the manual system. The operation nowadays is time-consuming and space-consuming. It provides a provision for errors as the data is entered manually by humans. In this system there will be a risk of the documents being lost and maintenance of the previous records is difficult as they will be in large size. The data recorded during acquiring the details of different fields required in the blood bank management system is not so accurate and precise and as this will be done by humans so they may possess errors. Maintaining the stock of blood and the daily exchanges and deletions without computerization is a tough challenge so we think it’s better to implement the use of Databases here which makes the system simple and efficient and helps in order to save more lives.

Proposed Solution:-

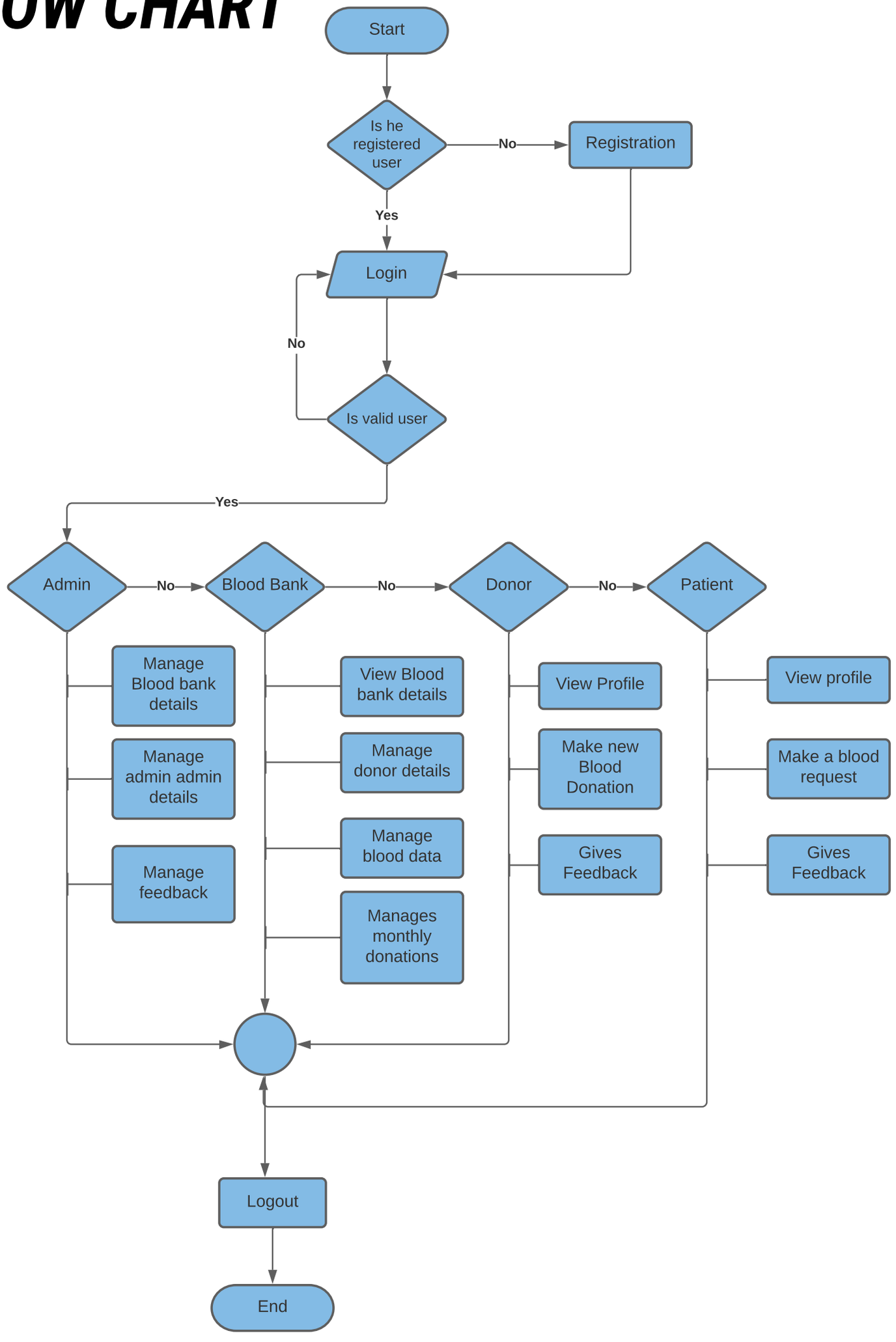
In this project, the following solution is given

- We are providing a better platform for the users to view the nearest blood donors, blood banks anywhere anytime in the world.
- It keeps track of previous donor data which saves lives in absence of a rare blood group. It acts as a connecting bridge between patient and donor and blood bank. It also displays the data required in all possible ways.
- This program has different classes for different users and it also has different operations that they can perform.
- It stores all the data in sql and handles them in no time.
- It stores the data using function handling and also saves human effort and time

Motivation:-

Generally it is highly impossible to maintain all such records in the form of paper or dairy because it is too hectic and it will consume lot of resources for the any management system to maintain such records.So they tend not to keep any database and records with them.With this databases system developed one can easily store information and can perform all kinds of queries easily.Data base management system not only involved in strong.Fetching data from the database but also gave other advantages like user integrity,respective views and organization in a clear path way.

FLOW CHART


Start
Is he registered user
No
Registration
Yes
Login
No
Is valid user
Yes
Admin
No
Blood Bank
No
Donor
No
Patient
Manage Blood bank details
Manage admin admin details
Manage feedback
View Blood bank details
Manage donor details
Manage blood data
Manages monthly donations
View Profile
Make new Blood Donation
Gives Feedback
View profile
Make a blood request
Gives Feedback
Logout
End

Solution In detail

The final end users on the login Frame are :-

1. *Administrator*
2. *Blood bank employee*
3. *Donor*
4. *Patient*

Whenever you choose the option the corresponding class is called and its menu will be popped up as shown below

An Admin will have a menu like (**Screenshot of this menu is in the next pages**)

- ➔ Administrator data
- ➔ Blood bank employee data
- ➔ Feedback
- ➔ Manage his profile

The administrator data option will display all the details of admin data with its corresponding options like Editing data. The Bloodbank employee data option will display all the details of Blood Bank employee data with its corresponding options like Editing data. The feedback option enables one to read the feedback and suggestions given by Donors and patients.

A blood bank employee will have a menu like (**Screenshot of this menu is in the next pages)**

- ➔ Blood bank employee
- ➔ Blood Data
- ➔ Donor data

The Bloodbank employee data option will display all the details of Blood Bank employee data. The Blood data option enables one to view their respective data and their corresponding options like adding, removing, and updating blood and the same with the Donor data option. There is another special option for Blood bank employee which reduces his/her effort is **Update list option.**

The update list option in Blood Data ***removes the spoiled blood that is as per Biological norms if blood is unused for more than 42 days it gets spoiled*** so by choosing this option the unused blood is removed by storing its adopted date in a date column of BLood donor data. This will also ***not accept the donation from a donor if the duration from his last donation is less than 56 days as it is not good for his health*** which can be controlled again using the date column of which stores the last donated date of the donor .The ***Donor discussed here is different from the blood donor details in the Blood bank employee class*** as The donor in the blood bank employee is the one which donated their blood and stored in the blood bank, and the donor details here which we are discussing is are the one who is ready to donate their blood in emergency purposes and we store their each and every particular of Blood donor and Donor.

A Donor will have a menu like

- ➔ Manage his profile
- ➔ Make a Donation
- ➔ Feedback

The Manage his profile usually provides users to view and edit their personal profile. The make a Donation option enables the user to change their status that they are willing to donate blood and by changing his status he will be taken into part of consideration while a patient is searching for a particular.

A patient will have a menu like

- ➔ Manage his profile
- ➔ Make a request
- ➔ Feedback

The manage his profile option usually provides users to view and edit their personal profile. ***The request option enables to request blood, the user has an option to request even for him or for his friend or relatives or to anyone***. ***The Donors and Blood details will be displayed in the nearer first this will be achieved by ID of blood bank*** and Closest function is the function used to display the nearer one first.

Some more important points:-

- The feedback and suggestions are taken from the patient and Donor every 30 days and store in a feedback relation. **And we also maintain the feedback anonymously.**
- We have used a class named closest to display the nearer one first to the patient.

Analysis of Our Blood Bank Management System:-

- An efficient blood bank database would aim to easily connect potential donors with blood-banks, blood-banks with potential recipients and vice versa.
- The potential donors would be connected with their nearest blood bank.
- Another Objective of the database is to register and store blood-banks by their location in order for them to be connected to potential donors and recipients with ease.
- The blood banks aim to receive and give out blood from people in need (recipients) closest to them.
- The blood recipient would need to be connected to a bank which possesses his blood in the closest proximity to him.

Technical information & Diagrams

FUNCTION SPECIFICATION

The final end users of our project are

1. *Admin*
2. *Blood bank Employee*
3. *Donor*
4. *Patient*

ADMIN:-

- Manage Registration for Admin and Blood bank employee
- Manage Blood bank and Admin information like (update, delete)
- Manage feedback given by Donor and Patient

BLOOD BANK EMPLOYEE:-

- Blood bank information view/update
- Manage Blood data in their Blood Bank
- View Donor information

DONOR:-

- Manage Donor Information(profile)
- Make new Donation for Blood
- Give feedback

PATIENT:-

- Manage patient information(profile)
- Give the Request for the patient for blood
- Give feedback

Details of end users

Administrator Data

- ID
- Name
- Position
- Mobile No
- Email ID

Blood bank employee Data

- ID
- BloodBank ID
- Name
- Mobile no
- Email ID
- City

Blood Donor Data

- Blood Bank ID
- Blood Bank Donor ID
- Name
- Blood Group
- AGE
- Address
- mobile no
- Gender
- Last donated date

Donor Data

- ID
- Blood Bank ID
- Name
- Blood Group
- AGE
- Address
- mobile no
- Last donated date
- Joined Data
- Feed
- Status

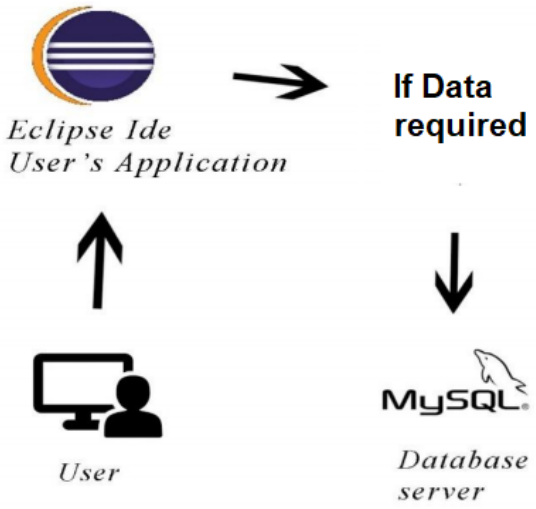
Patient Data

- ID
- Name
- AGE
- Blood Bank ID
- Blood Group
- Address
- mobile no
- JDD
- Feed

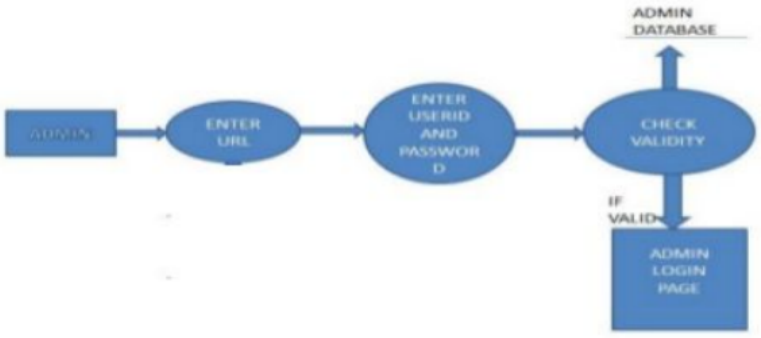
TOOLS/IDES

MySQL Workbench: This tool is used to create the database tables and it is also used to perform various operations which are costly in terms of Programming**(Add,Delete,Update list)** with an efficient way in form queries.

Eclipse: This IDE will be used to connect the database from MySQL to the main program which is written in java language. We have connected the relations in sql and our code in eclipse using JDBC(Java Database Connectivity).

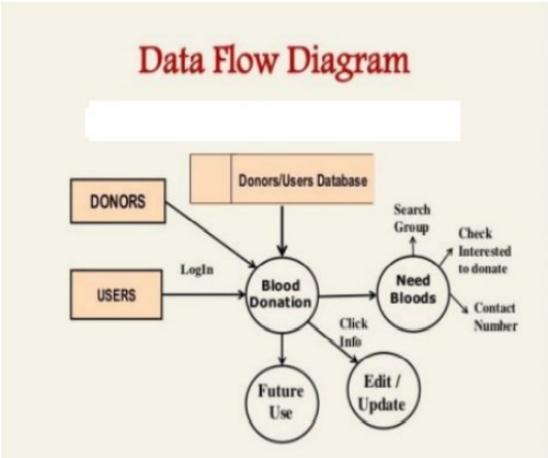


Diagramatic representation of login class



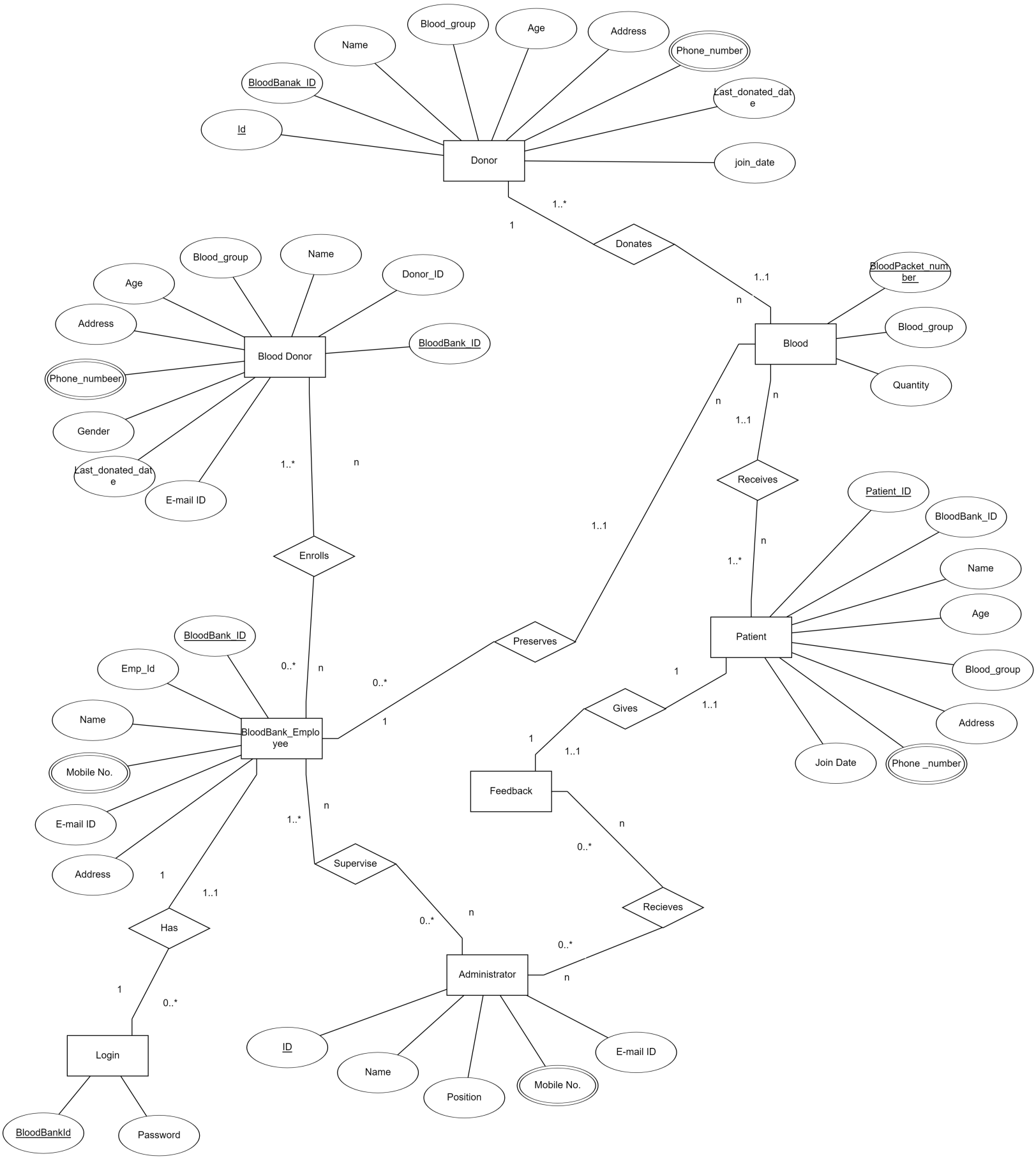
Similarly login class works in the same manner for every one.

Diagramatic representation of Data Flow :-



In the next page of the report, there is an ER diagram that is going to be explained in detail to enhance the understanding of the system whole. While designing the diagram, we were trying to connect it to the system to be done at the end.

ER DIAGRAM


Donor
Name
Blood_group
Age
Address
Phone_number
BloodBanak_ID
Last_donated_date
Id
join_date
Donates
1..*
1
1..1
n
Blood
BloodPacket_number
Blood_group
Quantity
Blood Donor
Blood_group
Name
Donor_ID
Age
Address
BloodBank_ID
Phone_numbeer
Gender
Last_donated_date
E-mail ID
Enrolls
1..*
n
Receives
1..1
n
Patient
Patient_ID
BloodBank_ID
Name
Age
Blood_group
Address
Join Date
Phone _number
Preserves
1..1
n
0..*
BloodBank_Employee
BloodBank_ID
Emp_Id
Name
Mobile No.
E-mail ID
Address
Gives
1
1..1
Feedback
Supervise
1..*
n
0..*
Recieves
Has
1
1..1
0..*
Administrator
ID
Name
Position
Mobile No.
E-mail ID
Login
BloodBankId
Password

Entities

1. *Login*
2. *Blood Data*
3. *Adminstrator*
4. *Blood Bank employee*
5. *Blood Bank Donor*
6. *Donor*
7. *Patient*
8. *Feedback*

- Login tables are used to login into the blood bank management system with their respective accounts.Like administrator,employee,patients and donors all have their own login id and password.
- Blood Data table collects all the data of blood from the different blood banks and stores it corresponding to their system generated ID's and connects them with the blood bank employee,Patient and Donor.
- Administrator is the head of the system and where administrator supervise blood bank employee and stored the personal info of all administrators.
- Blood bank donor table has all the personal data of persons who donated blood to their corresponding Blood bank,Where all the data of blood bank donor is maintained by blood bank employee.
- Donor is connected with the blood bank and this stores all the data of persons who are ready to donate blood directly to the patients in case of emergency.
- Patient is connected with the blood bank and Donor,this stores the data of patients where the patient seeks blood.
- Feedback is the table which stores the feedback given on the blood bank management system,where feedback is given by both patient and donor, and received by the administrator.

CARDINALITIES

ONE TO ONE

Between Administrator and Login
Between BloodBankEmploy and Login
Between Donor and Login
Between Patient and Login.
Between Patient and Feedback.
Between Donor and Feedback.

ONE TO MANY

Between BloodBankEmploy and Blood
Between Donor and Blood

MANY TO MANY

Between Administrator and BloodBankEmploy
Between Administrator and Feedback
Between BloodBankEmploy and BloodDonor
Between Blood and Patient

Normalization

Normalization is an approach that should be done for databases to minimize their redundancy and increase their efficiency. It consists of several steps and has many forms starting from 1NF (First Normal Form) which has basic and logical requirements such as not having two columns having the same name. Steps of normalization continue with one main goal to achieve; Minimize redundancies.

1NF (First Normal Form) Rules
- Each table cell should contain a single value – Atomicity property
- Each record needs to be unique - Uniqueness

2NF (Second Normal Form) Rules
- Rule 1- Be in 1NF
- Rule 2- Single Column Primary Key (or in simpler words there should be no functional dependency that has partial dependency).

3NF (Third Normal Form) Rules
- Rule 1- Be in 2NF.
- Rule 2- Has no transitive functional dependencies

Below representation of tables shows that steps that was taken to achieve the goal. It was noticed while inserting data that different users can have the same blood bank data(i.e like data of blood packets like AB+ -5 blood packets present). Which will cause some redundancy as the same blood bank data is repeated for different Donors. In that case, a lot of time will be wasted editing the data of the same Blood bank data for all donors. Therefore, as a solution for this redundancy, an Blood bank data table was created to save the blood bank data and defining quantity of blood groups.BId as the primary key of the table in order to connect the Blood data to a unique Id. After that, BId was added as attribute in the Blood Donor table as a foreign key connecting the blood donor to their Blood Bank. Now in this case, adding, deleting, and changing data is made easier and time-saving.

BID	BDID	Name	BGrp	Age	Address	Mobile	Gender	LDD	AB+	AB-	B+	B-	O+	O-	HH
		Teja	O+	25	Gujarat	8978572132	male	21-03-2019	2	5	1	2	3	2	0
		Reshma	HH	21	Kerala	9390099852	female	20-02-2018	2	5	1	2	3	2	0

BID	BDID	Name	BGroup	Age	Address	Mobile	Gender	LDD
10001	111	Teja	O+	25	Gujarat	8978572132	male	21-03-2019
10001	112	Yash	HH	21	Kerala	9390099852	female	20-02-2018

BID	AB+	AB-	B+	B-	O+	O-	HH
	2	5	1	2	3	2	0

DATABASE SCHEMA

AdministrationLogin: This table stores login details of Administrator

	ID	PASSWORD
TYPE	CHAR	VARCHAR
KEY	PK	
EXAMPLE	Asunn	sunny0

TABLE 1 : **ADMINISTRATIONLOGIN**

AdministratorData: This table stores data of administrator

	ID	NAME	POSITION	MOBILENO	MAIL
TYPE	CHAR	VARCHAR	VARCHAR	VARCHAR	VARCHAR
KEY	PK,FK				
EXAMPLE	Asunn	chakri	manager	707515	chakri@

TABLE 2 : **ADMINISTRATORDATA**

BemployLogin: This table stores login details of Blood bank Employ

	ID	PASSWORD
TYPE	CHAR	VARCHAR
KEY	PK	
EXAMPLE	Bbunn	Sunny@123

TABLE 3 : **BEMPLOYLOGIN**

EmployData: This table stores Blood Bank Employ Data

	ID	BID	NAME	MOBILENO	MAIL	CITY
TYPE	CHAR	CHAR	VARCHAR	VARCHAR	VARCHAR	VARCHAR
KEY		PK				
EXAMPLE	Bbunn	12312	chakri	707515	2019510@	ongole

TABLE 4 : EMPLOYDATA

BloodData: This table stores the Blood Data of each Blood Bank

	BID	AB+ve	AB-ve	B+ve	B-ve	O+ve	O-ve	HH
TYPE	CHAR	INT	INT	INT	INT	INT	INT	INT
KEY	PK,FK							
EXAMPLE	12312	2	5	4	6	2	3	5

TABLE 5 : **BLOODDATA**

BloodDonorData: This table stores Data of the Donor who donated blood in Blood Bank

	BID	BDID	NAME	BGrp	AGE	ADDRESS	MOBILE	GENDER	LDD	LDD1
TYPE	CHAR	CHAR	VARCHAR	VARCHAR	INT	VARCHAR	VARCHAR	CHAR	DATE	DATE
KEY	FK	PK								
EXAMPLE	12312	123	sunny	AB+ve	15	Flat405	8688377	M	2008-11-11	2008-11-11

TABLE 6 : **BLOODDONORDATA**

DonorLogin: This table stores login details of Donor

	ID	PASSWORD
TYPE	CHAR	VARCHAR
KEY	PK	
EXAMPLE	Bbunn	Sunny@123

TABLE 7 : **DONORLOGIN**

DonorData: This table stores Donor Data

	ID	BID	NAME	Bgrp	AGE	ADDRESS	MOBILE	LDD	JDD	feed	STATUS
TYPE	CHAR	CHAR	VARCHAR	VARCHAR	INT	VARCHAR	VARCHAR	DATE	DATE	int	VARCHAR
KEY	PK,FK	FK									
EXAMPLE	12312	123	sunny	AB+ve	15	Flat405	8688377	2008-11-11	2008-11-11		

TABLE 8 : **DONORDATA**

PatientLogin: This table stores login details of Patient

	ID	PASSWORD
TYPE	CHAR	VARCHAR
KEY	PK	
EXAMPLE	Bbunn	Sunny@123

TABLE 9 : **PATIENTLOGIN**

PatientData: This table stores Patient Data

	ID	NAME	AGE	BID	Bgrp	ADDRESS	MOBILENO	JDD	feed
TYPE	CHAR	VARCHAR	INT	CHAR	VARCHAR	VARCHAR	VARCHAR	Date	int
KEY	PK,FK								
EXAMPLE	12312	sunny	15	123	AB+ve	Flat405	8688377	2008-11-11	30

TABLE 10 : **PATIENTDATA**

Feedback: This table stores feedback given by Patient and Donor

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
TYPE	int	int	int	int	int	int	int	varchar
KEY								
EXAMPLE	5	4	3	4	5	4	5	No suggestions

TABLE 11 : **feedback**

SQL Script

DDLs:

```
CREATE TABLE ADMINSTRATIONLOGIN
        (
        ID CHAR(5) NOT NULL,
        PASSWORD VARCHAR(20),
        PRIMARY KEY (ID),
        CONSTRAINT CHECK_AID CHECK (ID LIKE 'A%')
        );
CREATE TABLE ADMINSTRATORDATA
        (
        ID CHAR(5) NOT NULL,
        NAME VARCHAR(30) NOT NULL,
        POSITION VARCHAR(30) NOT NULL,
        MOBILENO VARCHAR(20) NOT NULL,
        MAIL VARCHAR(50) NOT NULL,
        primary key(ID),
        FOREIGN KEY(ID) REFERENCES ADMINSTRATIONLOGIN(ID)
        );
CREATE TABLE  BEMPLOYLOGIN
       (
        ID CHAR(5) NOT NULL,
        PASSWORD VARCHAR(120) NOT NULL,
        PRIMARY KEY(ID),
        CONSTRAINT CHECK_BID CHECK(ID LIKE'B%')
       );
CREATE TABLE EMPLOYDATA
        (
        ID CHAR(5) NOT NULL unique,
        BID CHAR(5) NOT NULL,
        NAME VARCHAR(30) NOT NULL,
        MOBILENO VARCHAR(15) NOT NULL,
        MAIL VARCHAR(50) NOT NULL,
        CITY VARCHAR(30) NOT NULL,
```

```
PRIMARY KEY(BID)
);
```

CREATE TABLE BLOODDATA

```
(
BID CHAR(5) NOT NULL,
ABPOSITIVE INT,
ABNEGATIVE INT,
BPOSITIVE INT,
BENEGATIVE INT,
OPOSITIVE INT,
ONEGATIVE INT,
HH INT,
PRIMARY KEY(BID),
FOREIGN KEY(BID) REFERENCES EMPLOYDATA(BID)
);
```

CREATE TABLE BLODDDONORDATA

```
(
BID CHAR(5) NOT NULL,
BDID CHAR(6) NOT NULL,
NAME VARCHAR(30) NOT NULL,
BGrp VARCHAR(15) NOT NULL,
AGE INT NOT NULL,
ADDRESS VARCHAR(50) NOT NULL,
MOBILE VARCHAR(15) NOT NULL,
GENDER CHAR(1),
LDD DATE ,  #LDD DENOTES LAST DONATED DATE
LDD1 DATE,
ldd2 date,
PRIMARY KEY(BDID),
FOREIGN KEY(BID) REFERENCES EMPLOYDATA(BID),
CONSTRAINT CHECK_GENDER CHECK (GENDER IN('M','F'))
);
```

CREATE TABLE DONORLOGIN

```
(
  ID CHAR(9) NOT NULL,
  PASSWORD VARCHAR(20) NOT NULL,
  PRIMARY KEY(ID),
  CONSTRAINT CHECK_DID CHECK(ID LIKE'D%')
);
```

CREATE TABLE DONORDATA

```
(
 ID CHAR(9) NOT NULL,
 BID CHAR(5) NOT NULL,
 NAME VARCHAR(20) NOT NULL,
 Bgrp VARCHAR(15) NOT NULL,
 AGE INT NOT NULL,
 ADDRESS VARCHAR(50) NOT NULL,
 MOBILE VARCHAR(15) NOT NULL,
 LDD DATE NOT NULL,
 JDD DATE NOT NULL,
 feed int NOT NULL,
 STATUS VARCHAR(3) NOT NULL,
 PRIMARY KEY(ID),
 FOREIGN Key(ID) references Donorlogin(ID),
 FOREIGN KEY(BID) REFERENCES EMPLOYDATA(BID),
 CONSTRAINT CHECK_STATUS CHECK (STATUS IN ('YES','NO'))
);
```

CREATE TABLE PATIENTLOGIN

```
(
 ID CHAR(9) NOT NULL,
 PASSWORD VARCHAR(20) NOT NULL,
 PRIMARY KEY(ID),
 CONSTRAINT CHECK_PID CHECK(ID LIKE 'P%')
);
```

CREATE TABLE PATIENTDATA

```
(
 ID CHAR(9) NOT NULL,
 NAME VARCHAR(30) NOT NULL,
 AGE INT,
 BID CHAR(5) NOT NULL,
 Bgrp VARCHAR(15) NOT NULL,
 ADDRESS VARCHAR(50) NOT NULL,
 MOBILENO VARCHAR(20) NOT NULL,
 JDD Date not null,
 Feed int,
 PRIMARY KEY(ID),
 FOREIGN KEY(ID) REFERENCES PATIENTLOGIN(ID)
);
```

```
create table feedback
 (
   Q1 int,
   Q2 int,
   Q3 int,
   Q4 int,
   Q5 int,
   Q6 int,
   Q7 int,
   Q8 varchar(150)
);
```

Views:

Below are the views for our database system. Our system will have four main users, the administrator, the donors, the blood bank employee and the patients. These users have different needs and as a result would need to access different data but be unable to access others. To get rid of these cases are following views are created which are listed below

```
CREATE VIEW bdonor AS
SELECT BID,BDID,BGrp,Name,AGE,ADDRESS,MOBILE,GENDER,LDD
FROM blodddonordata;
```

```
create view Bloodemploy as
(select *
 from employdata);
```

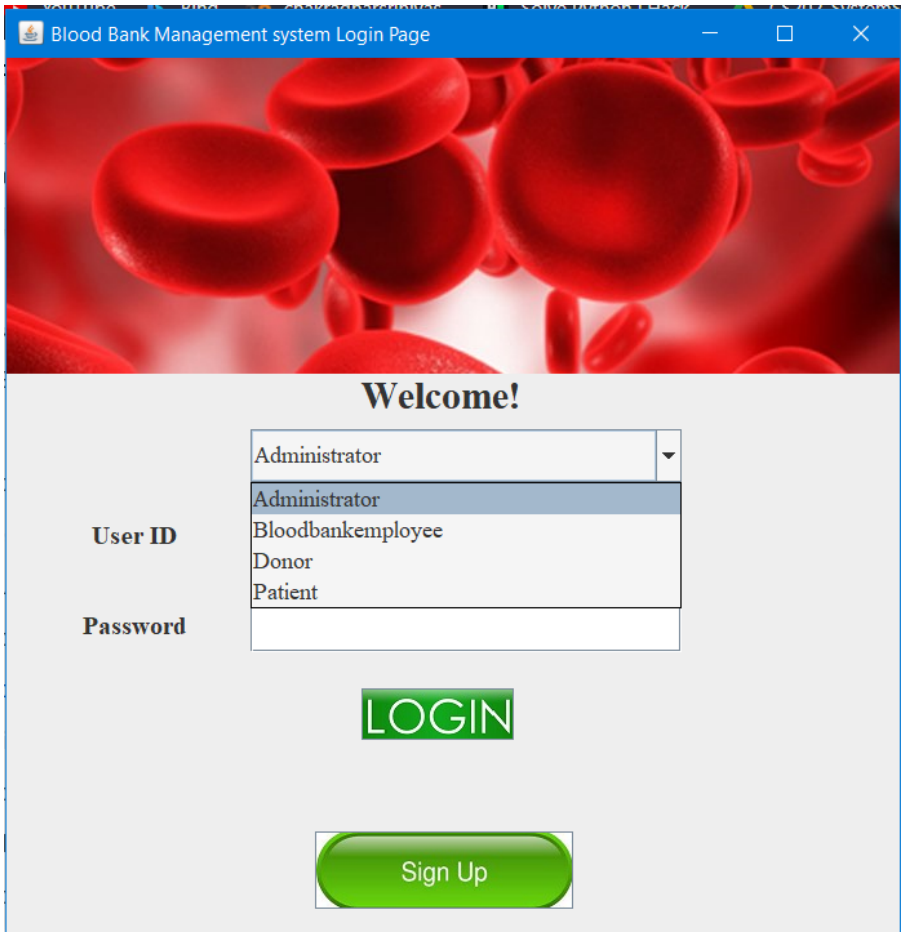
```
CREATE VIEW sdonor AS
SELECT ID,BID,BGrp,Name,AGE,ADDRESS,MOBILE,LDD,status
FROM donordata;
```

```
CREATE VIEW spatient AS
SELECT ID,BID,BGrp,Name,AGE,ADDRESS,MOBILE,LDD,status
FROM donordata;
```

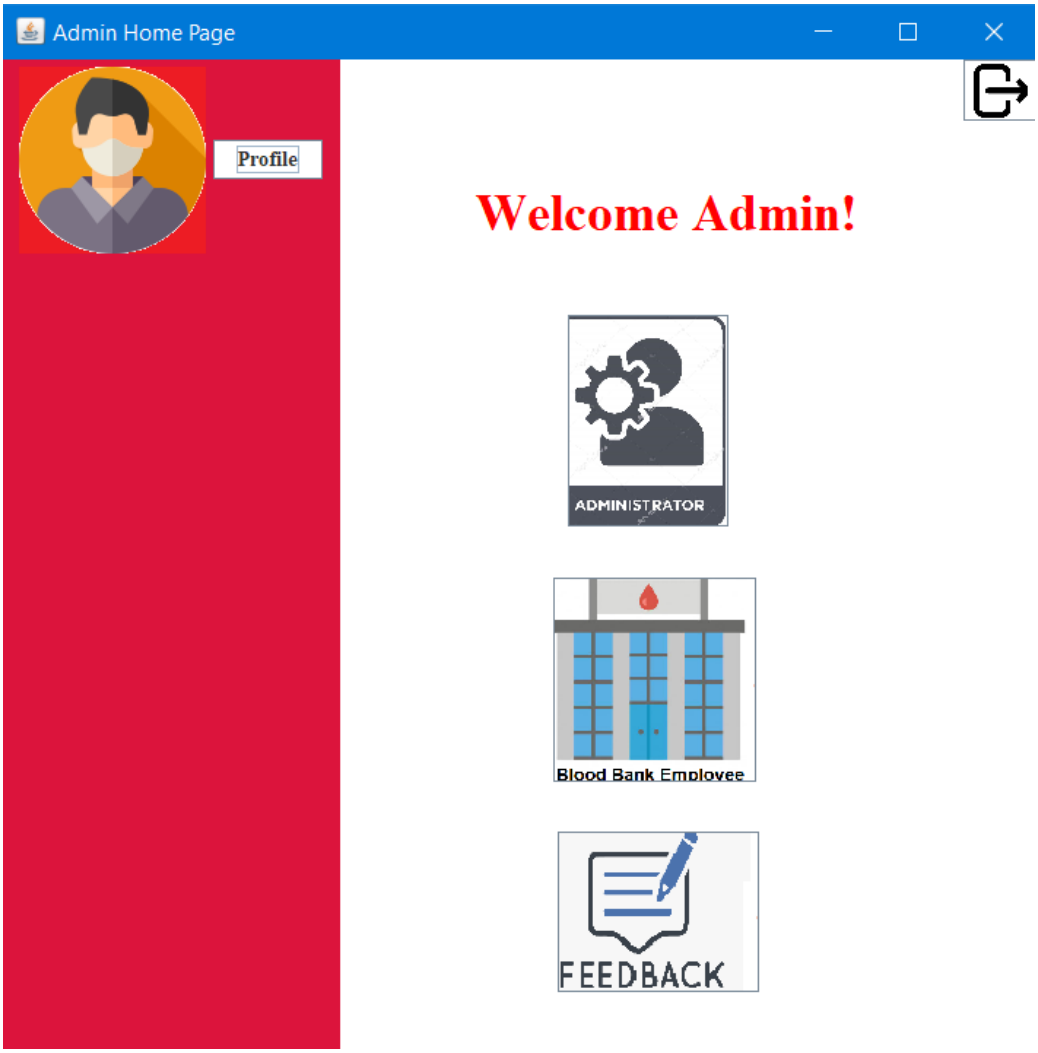
Screenshots of Output

Blood Bank Management system Home Page

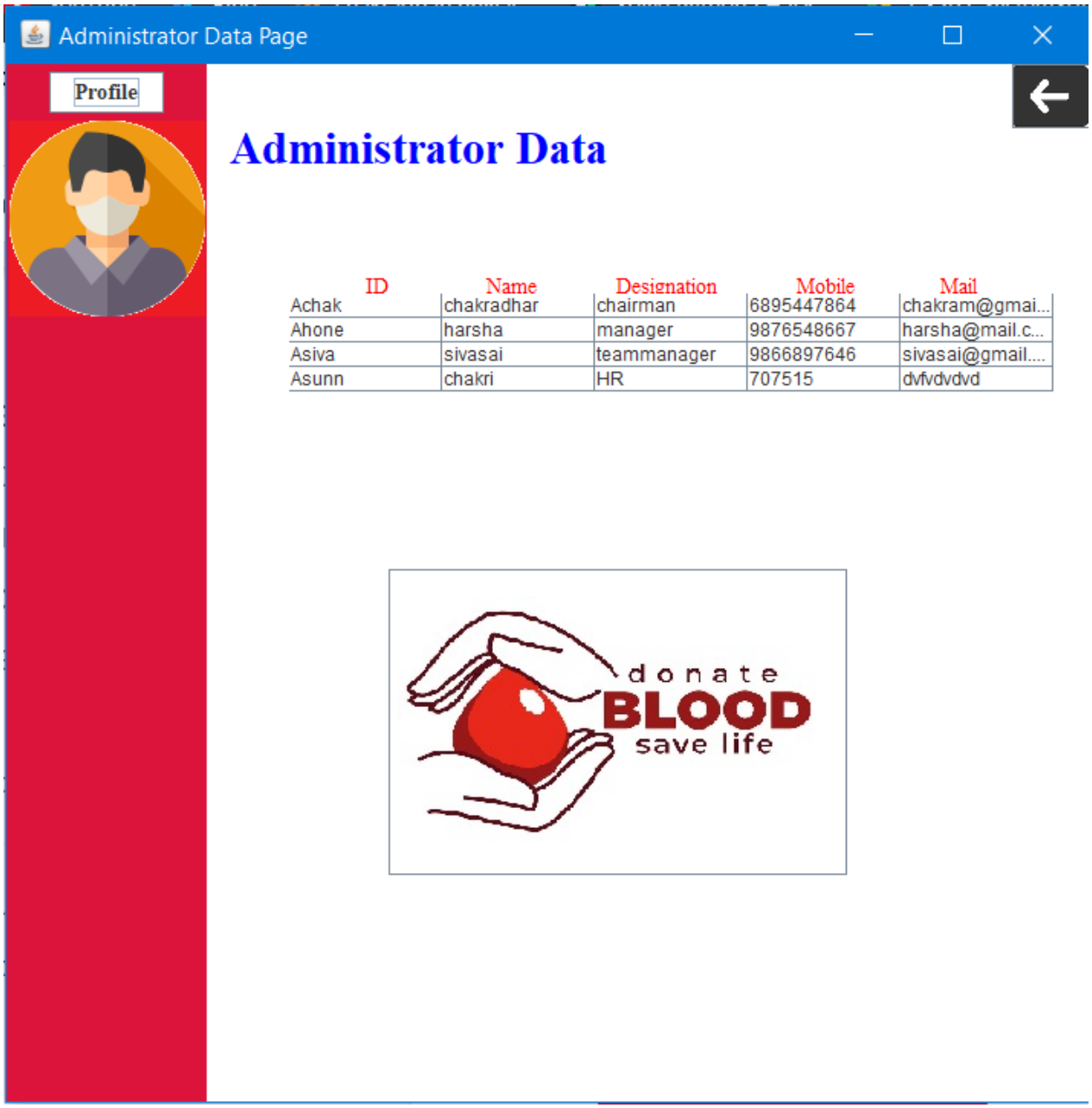
Blood Bank Management System Login Page



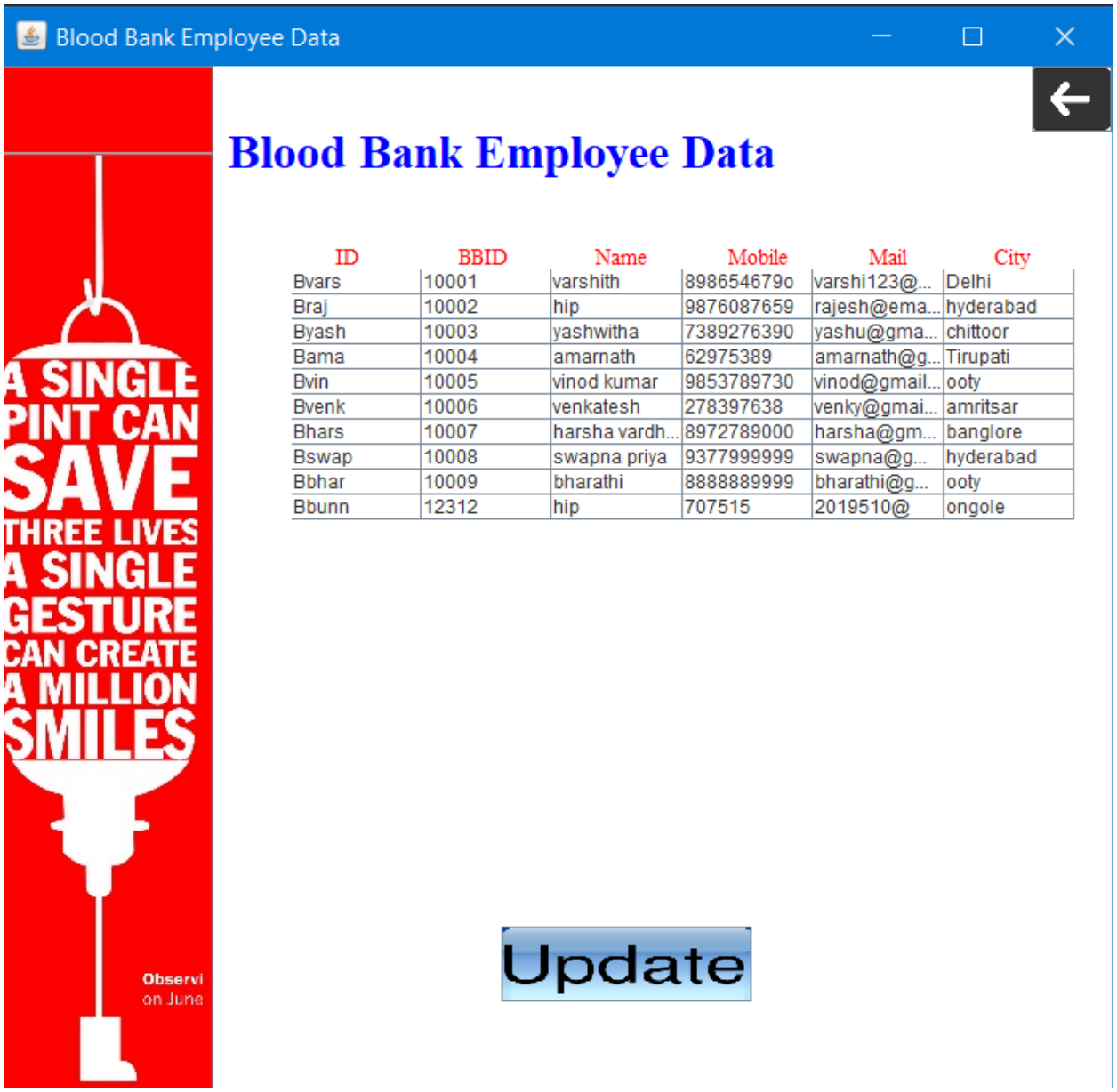
Admin Home Page



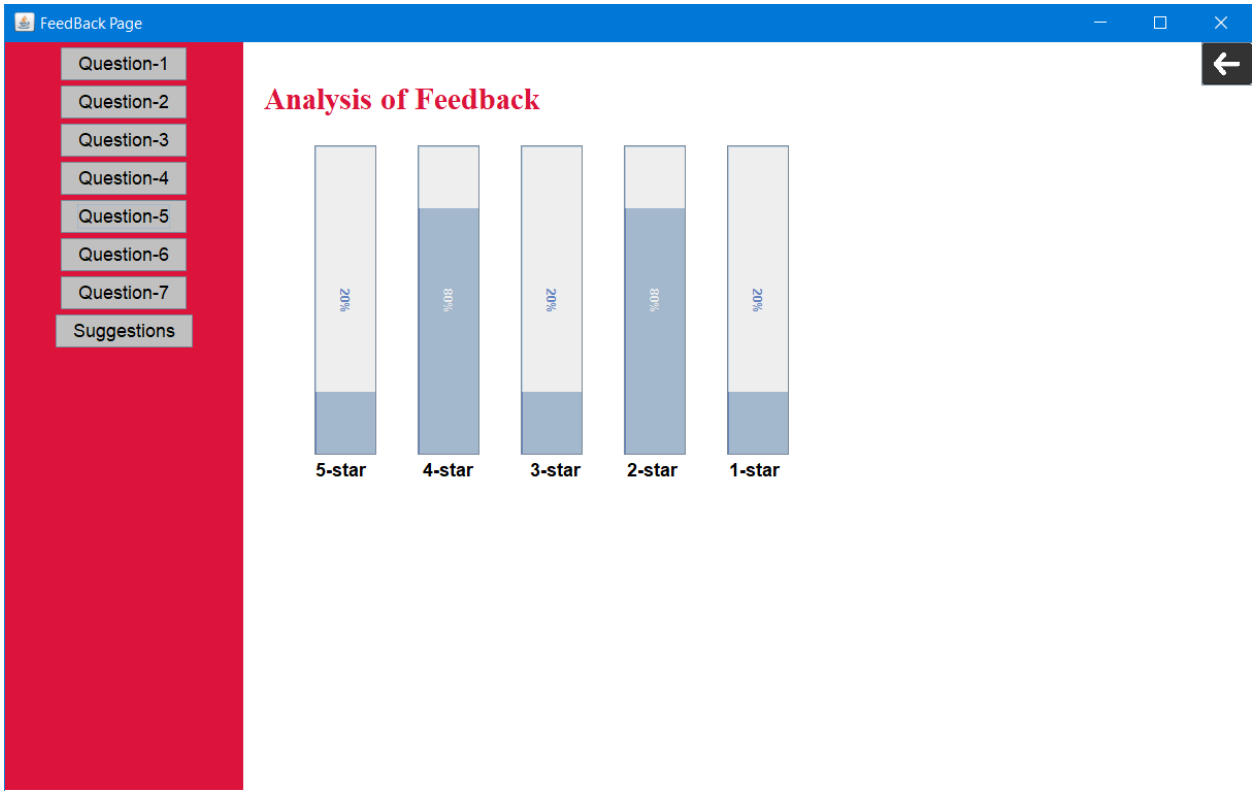
Administrator Data Page



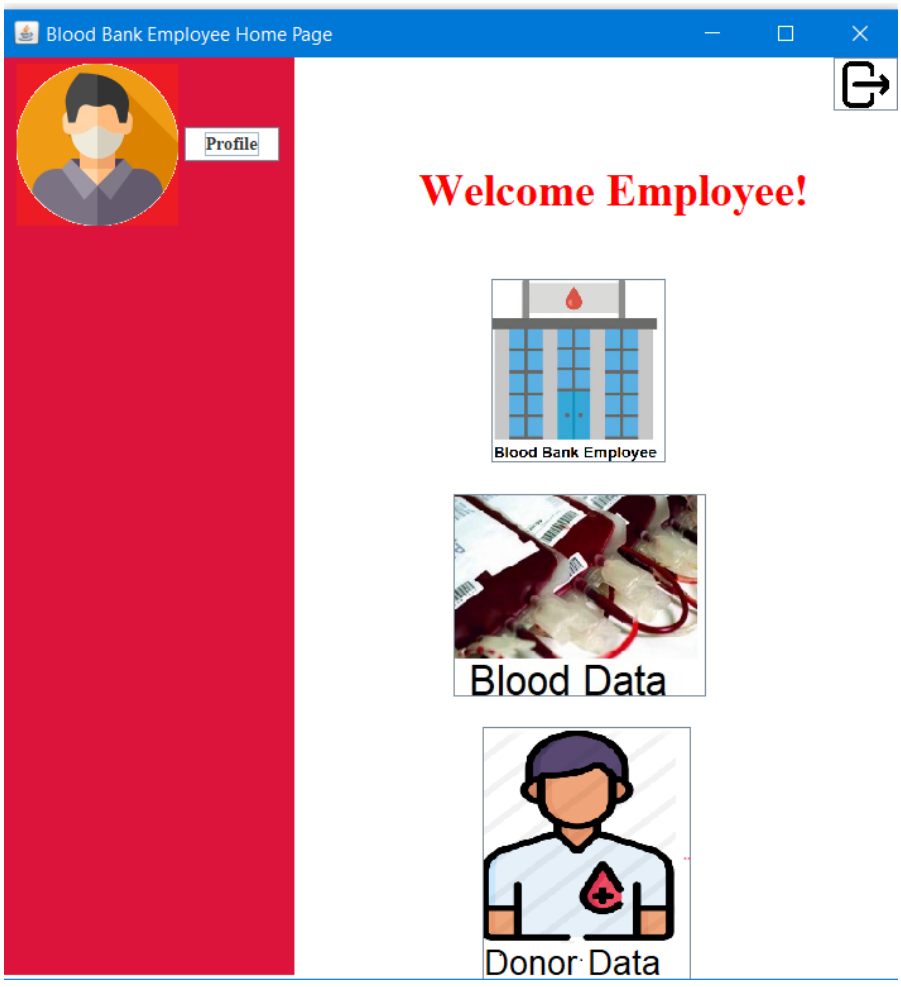
Blood Bank Employee Data



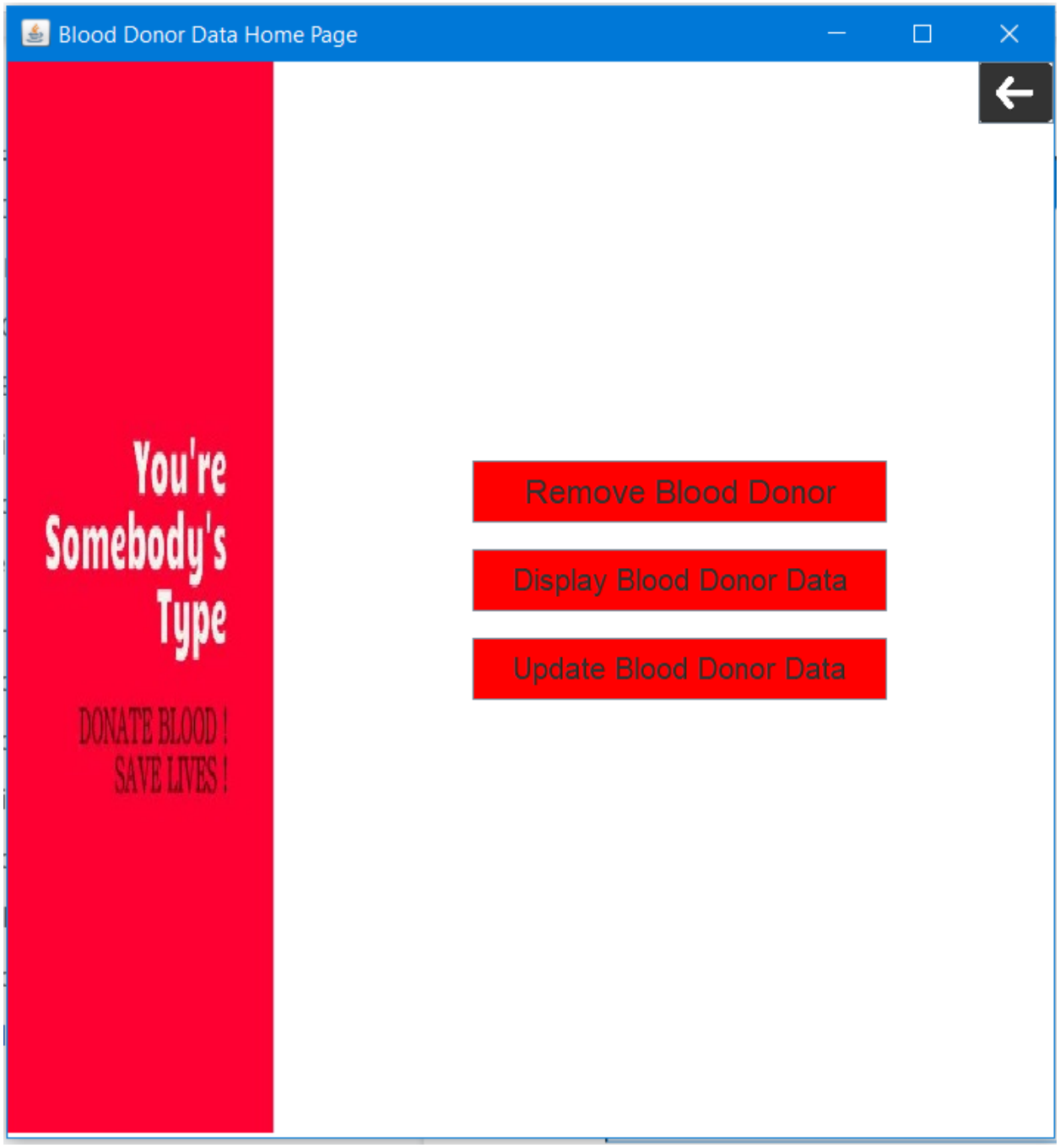
Feedback Page



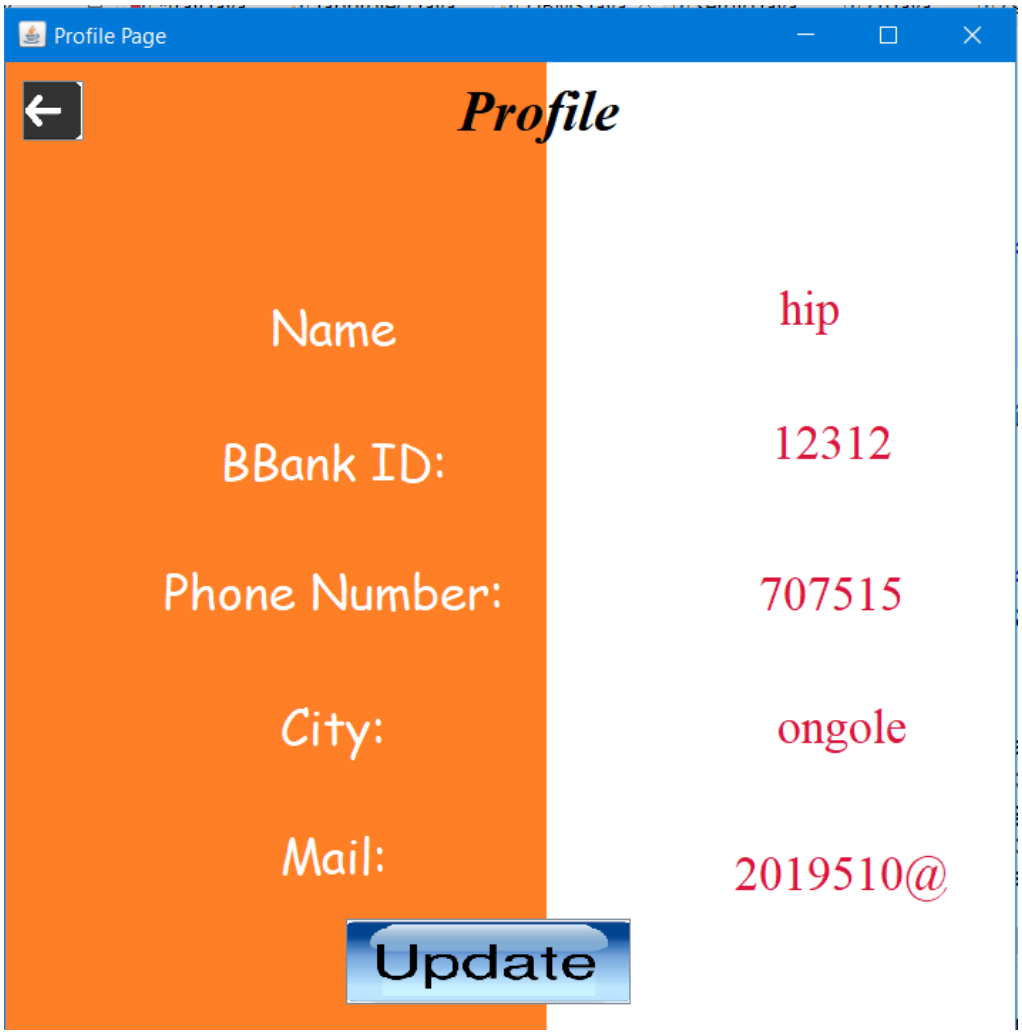
Blood Bank Employee Home Page



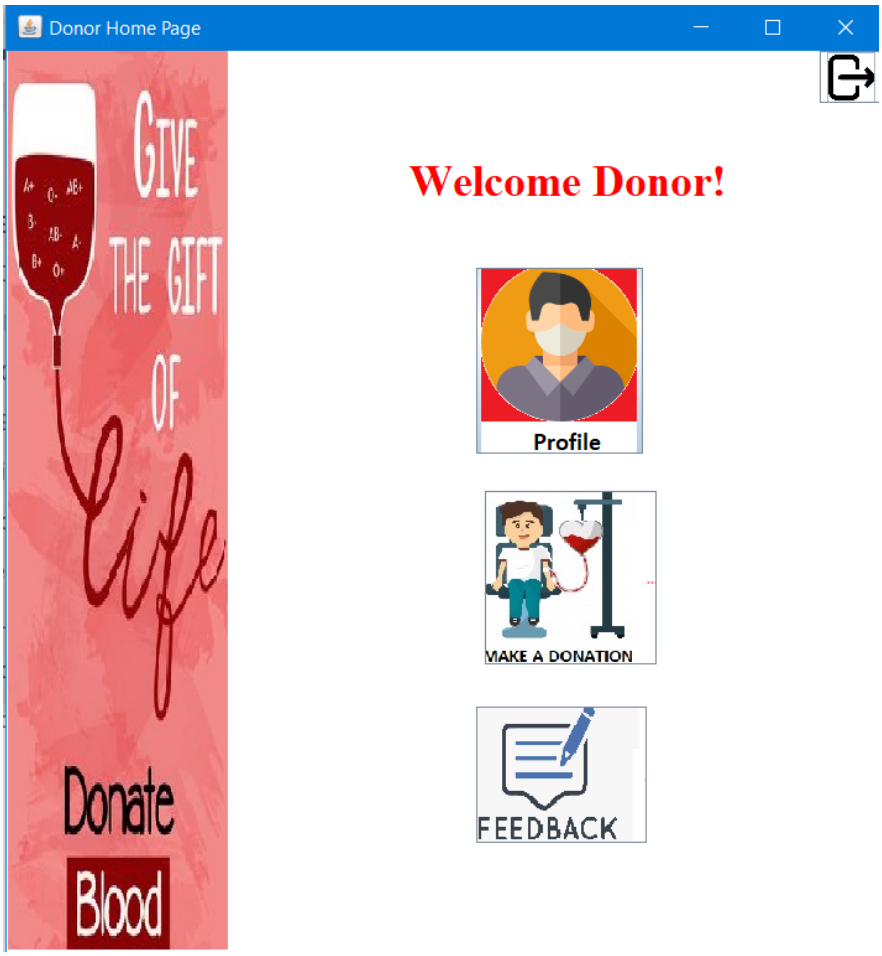
Blood Bank Employee Blood Donor Data Home Page



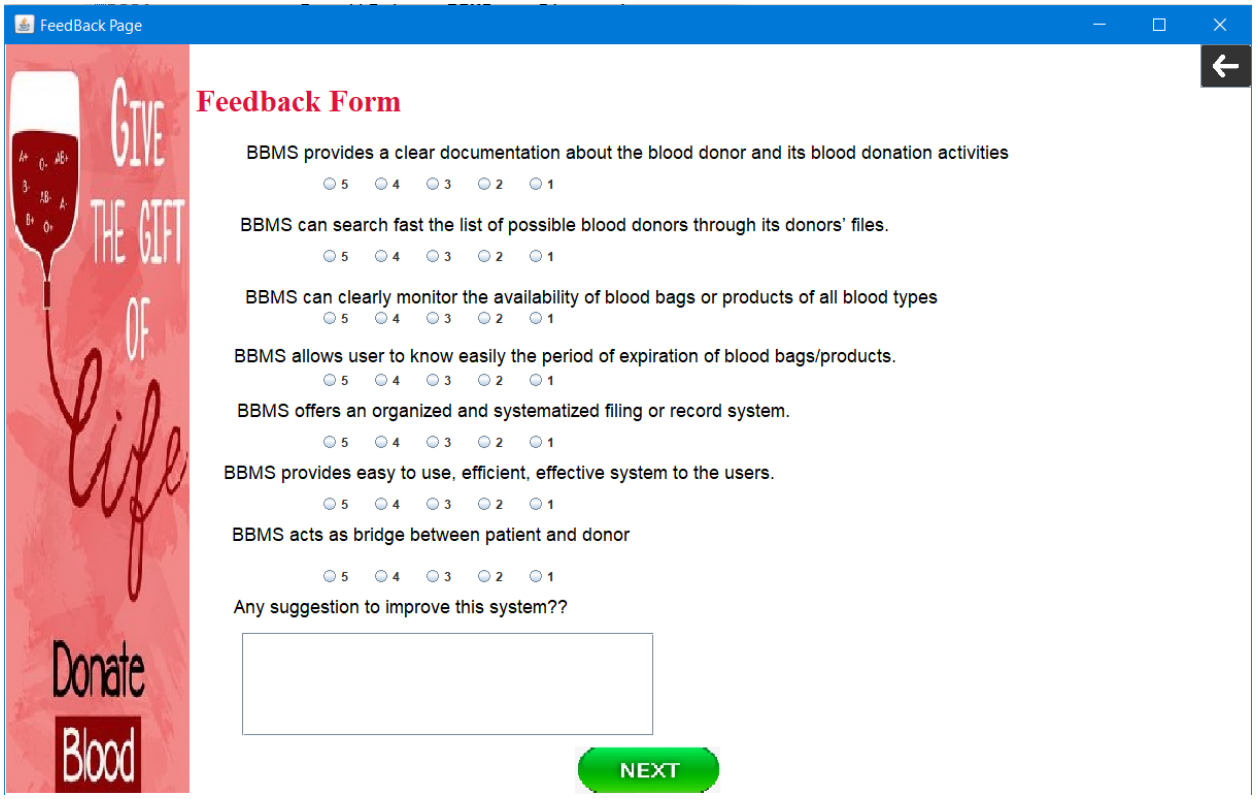
Profile Page



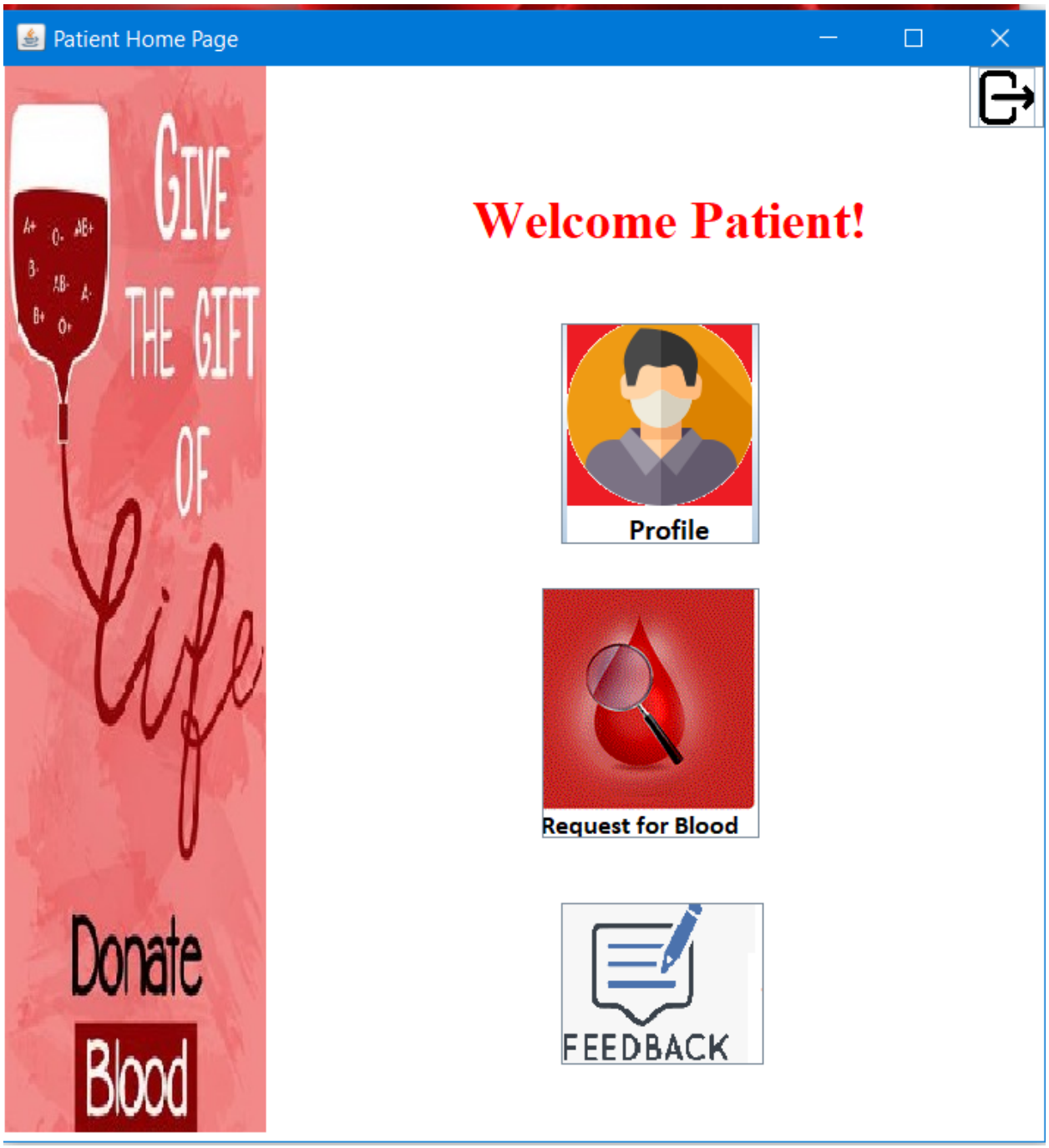
Donor Home Page



Feedback Form



Patient Home Page



THANK

YOU