# JAVA AWT BASED- INFORMATION RETRIEVAL OF GOOGLE QUERIES OF POSITIVE AND NEGATIVE FEEDBACK- SQL CONNECTIVITY USING JDBC

*A*

*Report*

*Submitted in partial fulfillment of the Requirements for the award of the Degree of*

# BACHELOR OF ENGINEERING

IN

**INFORMATION TECHNOLOGY**

By

Sai Sragvi <1602-19-737-098>

Under the guidance of Ms B. Leelavathy



Department of Information Technology Vasavi College of Engineering (Autonomous) (Affiliated to Osmania University) Ibrahimbagh, Hyderabad-31

# BONAFIDE CERTIFICATE

This is to certify that this project report titled ‘**Finding the Plasma Donor**’ is a project work of Ms.Sai Sragvi bearing roll no. 1602-19-737-098 who carried out the project under my supervision in the IV semester for the academic year 2020- 2021.

Signature Signature

Internal Examiner External Examiner

FINDING THE PLASAMA DONOR

# ABSTRACT

THIS IS THE REAL TIME PROJECT INSPIRED BY THE CURRENT SITUATION IF A DONNIE NEEDS A PLASMA WE WILL PROVIDE THEM A DONOR LIST ,BY MATCHING THEIR BLOOD GROUP AND AVAILABILITY OF LOCATION. “FINDING THE PLASMA DONOR” is a C Project, which is based on the current situation. We all know the important of plasma who are suffering from COVID-19 , our project is mainly about finding the people who are interested to donate plasma .

SAI SRAGVI

1602-19-737-098

**REQUIREMENT ANALYSIS:**

**LIST OF TABLES:**

BRANCH

EMPLOYEE

DONOR

BLOOD\_REQUEST

BLOOD

BRANCH\_HAS\_EMPLOYEE

BRANCH\_HAS\_DONOR

BRANCH\_HAS\_BLOOD\_REQUEST

DONOR\_GIVES\_BLOOD

**List of attributes with their domain**

BRANCH:

* EMAIL
* AREA
* SUB\_AREA
* ID
* B\_NAME
* ADDRESS
* PHONE

EMPLOYEE:

* EMP\_ROLE
* ID
* EMP\_ID
* EMP\_NAME
* EMP\_AREA
* EMP\_ADDRESS
* EMAIL
* PHONE
* EMP\_SALARY

BLOOD\_REQUEST:

* NAME
* ID
* PHONE
* EMAIL
* BLOOD\_AMOUNT
* BLOOD\_GROUP
* AREA
* HOSPITAL
* DELIVERY\_CONFROMATION
* ADDRESS

BLOOD:

* BLOOD\_AMOUNT
* BLOOD\_GROUP
* ID
* PAID\_AMOUNT

HAS:

* B\_ID

GIVES:

* DONOR\_ID

FINDING THE PLASMA DONOR

**AIM AND PRIORITY OF THE PROJECT**

“FINDING THE PLASMA DONOR” is a C Project, which is based on the current situation. We all know the important of plasma who are suffering from COVID-19 , our project is mainly about finding the people who are interested to donate plasma .

FINDING THE PLASMA DONOR

# ARCHITECTURE AND TECHNOLOGY

## Software used:

Java Eclipse, Oracle 11g Database, Java SE version 13, SQL\*Plus.

## Java AWT:

**Java AWT** (Abstract Window Toolkit) is an API to develop GUI or window-based applications in java.

Java AWT components are platform-dependent i.e. components are displayed according to the view of operating system. AWT is heavyweight i.e. its components are using the resources of OS.

The java.awt package provides classes for AWT API such as TextField, Label, TextArea, RadioButton, CheckBox, Choice, List etc.

## SQL:

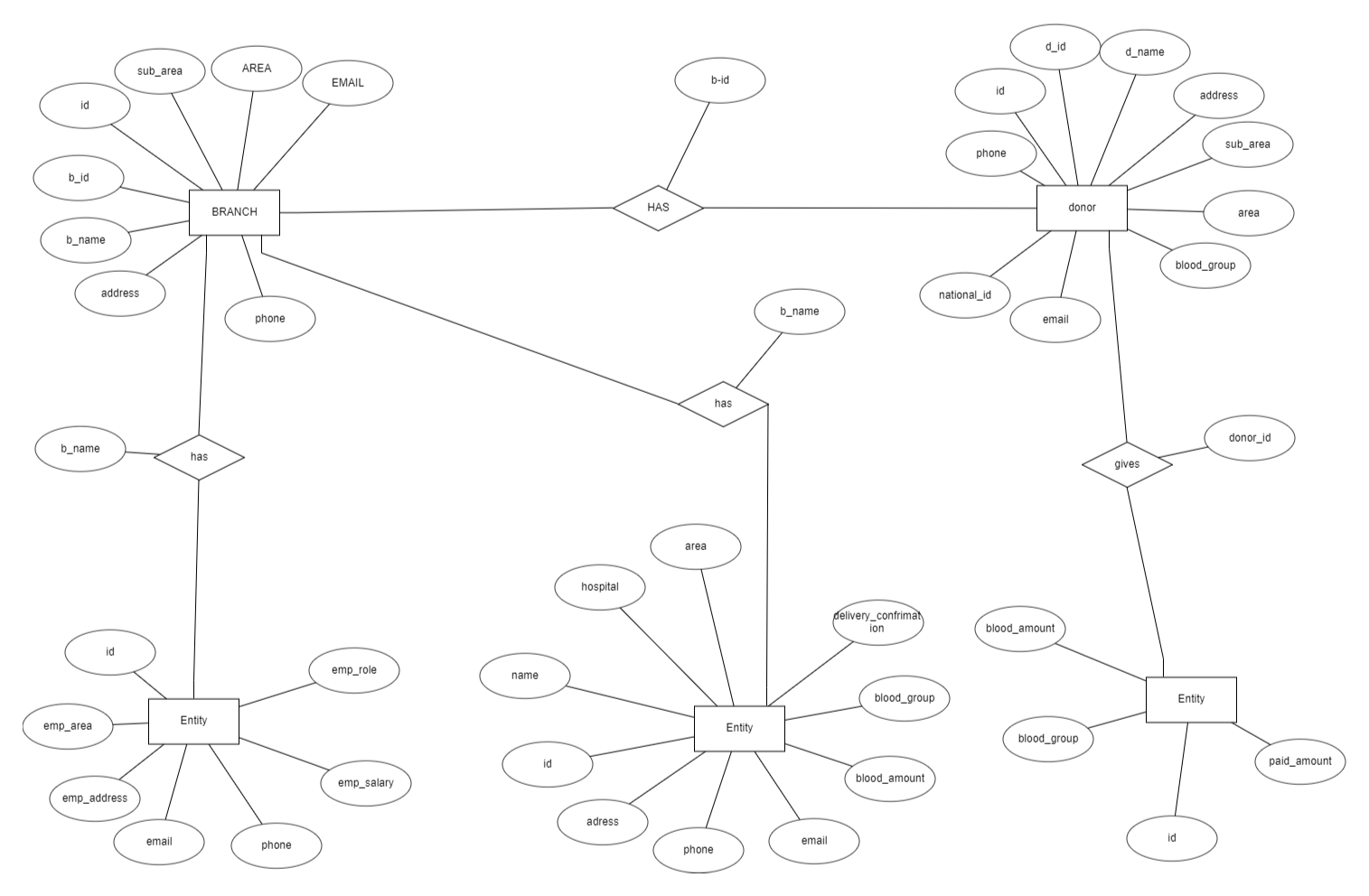
Structure Query Language(SQL) is a database query language used for storing and managing data in Relational DBMS. SQL was the first commercial language introduced for E.F Codd's **Relational** model of database. Today almost all RDBMS (MySql, Oracle, Infomix, Sybase, MS Access) use **SQL** as the standard database query language. SQL is used to perform all types of data operations in RDBMS.

SAI SRAGVI

1602-19-737-098

**DESIGN**

**Entity relationship diagram**

****

**CONTRAINTS**

Here branch is connected to donor and employee . branch will provide many like email, phone etc and employee is also connected to many like these .donor will connected to blood and also employee will connected to the employee request and that is connected to blood it has blood group, blood amount, id, paid amount

**Database Design:**

DML Operations

**IMPLEMENTATION**

**Front end programs and its connectivity**

**Java Database Connectivity** (**JDBC**) is an application programming

interface (API) for the programming language Java, which defines how a

client may access a database. It is a Java-based data access technology

used for Java database connectivity. It is part of the Java Standard

Edition platform, from Oracle Corporation. It provides methods to query

and update data in a database and is oriented towards relational

databases.

The connection to the database can be performed using Java

programming (JDBC API) as:

**public void** connectToDB()

{

**try**

{

connection

=

DriverManager.*getConnection*("jdbc:oracle:thin:@localhost:1521:xe","eesha","eeshasoham");

statement = connection.createStatement();

}

**catch** (SQLException connectException)

{

System.***out***.println(connectException.getMessage())

;

System.***out***.println(connectException.getSQLState()

);

System.***out***.println(connectException.getErrorCode());

System.*exit*(1);

}

}

Thus, the connection from Java to Oracle database is performed and

therefore, can be used for updating tables in the database directly.

import java.util.\*;

import java.sql.\*;

public class Admin { //code starts

public static void main(String[] args) throws Exception {

String dburl="jdbc:mysql://localhost:3306/blood\_bank\_management\_system";

String uname="root";

String password="swapan";

Class.forName("com.mysql.jdbc.Driver");

Connection con=DriverManager.getConnection(dburl,uname,password);

Scanner input=new Scanner(System.in);

System.out.println("Welcome to BLOOD BANK MANAGEMENT SYSTEM\n");

String ques1,ques2;

String field;

int enter;

do{ //first do

System.out.println("Are you faculty of the blood bank or a visitor? Press \n1. For Faculty \n2. For visitor");

ques1=input.next();

switch(ques1){

case "1":

do{

System.out.println("Enter whether you are an Administrator or a Receptionist?");

ques2=input.next();

switch(ques2){

case "Administrator":

System.out.println("Which dept do you work in? Internal or External?");

ques2=input.next();

switch(ques2){

case "Internal":

System.out.println("Choose one of the following options (Enter Choice No.): "

+ "\n1.Retrieve Blood bank Data"

+ "\n2. Retrieve Blood Inventory details"

+ "\n3.Manage Patient Requests"

+ "\n4.Manage Donor Applications");

ques2=input.next();

String query="";

switch(ques2){

case "1":

query="select \* from Blood\_bank, Administrator where blood\_bank.blood\_bank\_id=Administrator.bank\_id";

Statement st = con.createStatement();

ResultSet rs = st.executeQuery(query);

ResultSetMetaData rsmd=rs.getMetaData();

int c=rsmd.getColumnCount();

int i=c;

while(c!=0){

c--;

System.out.format("%16s",rsmd.getColumnName(i-c));

}

System.out.println();

while(rs.next()){

c=i;

while(c!=0){

c--;

System.out.format("%16s",rs.getString(i-c));

}

System.out.println();

}

break;

case "2":

System.out.println("Enter whether you want to see "

+"\n 'All' blood types"

+"\n 'One' Blood Type"

+"\n 'Misc' for One or more blood types");

ques2=input.next();

switch(ques2){

case "All":

query="select \* from Blood\_bank bb, Administrator a, Blood b where bb.blood\_bank\_id=a.bank\_id and bb.blood\_bank\_id=blood.bb\_id ";

st = con.createStatement();

rs = st.executeQuery(query);

rsmd=rs.getMetaData();

c=rsmd.getColumnCount();

i=c;

while(c!=0){

c--;

System.out.format("%16s",rsmd.getColumnName(i-c));

}

System.out.println();

while(rs.next()){

c=i;

while(c!=0){

c--;

System.out.format("%16s",rs.getString(i-c));

}

System.out.println();

}

break;

case "One":

System.out.println("Enter Blood type");

String btype;

btype=input.next();

query="select \* from blood\_bank bb, Administrator a, Blood b where bb.Blood\_bank-id=a.bank\_id and bb.Blood\_bank\_id=b.bb\_id and Blood\_type=?";

do{

System.out.println("Enter Choice No. \n1. All Details \n2.Quantity \n3.Platelet Count\n4.Expiry Date");

ques2=input.next();

switch(ques2){

case "1":

query="select \* from blood where blood\_type=?";

break;

case "2":

query="select blood\_units from blood where blood\_type=?"; ////////check

System.out.println(btype + "has units ");

break;

case "3":

query="select Platelet\_count from blood where blood\_type=?";

System.out.println(btype + "has platelet count");

break;

case "4":

query="select Expiry Date from blood where lood\_type=?";

break;

}

PreparedStatement st1 = con.prepareStatement(query);

st1.setString(1,btype);

ResultSet rs1 = st1.executeQuery();

ResultSetMetaData rsmd1=rs1.getMetaData();

int c1=rsmd1.getColumnCount();

int i1=c1;

while(c1!=0){

c1--;

System.out.format("%16s",rsmd1.getColumnName(i1-c1));

}

System.out.println();

while(rs1.next()){

c1=i1;

while(c1!=0){

c1--;

System.out.format("%16s",rs1.getString(i1-c1));

}

System.out.println();

}

System.out.println("\nDo you want to continue viewing information for blood type " + btype + "?(Yes/No)");

ques2=input.next();

}while(ques2.equals("Yes"));

break;

case "Misc":

System.out.println("Enter your query in MYSQL");

String temp=input.nextLine();

query=input.nextLine();

Statement st2 = con.createStatement();

ResultSet rs2 = st2.executeQuery(query);

ResultSetMetaData rsmd2=rs2.getMetaData();

int c2=rsmd2.getColumnCount();

int i2=c2;

while(c2!=0){

c2--;

System.out.format("%16s",rsmd2.getColumnName(i2-c2));

}

System.out.println();

while(rs2.next()){

c2=i2;

while(c2!=0){

c2--;

System.out.format("%16s",rs2.getString(i2-c2));

}

System.out.println();

}

break;

}

System.out.println("\nDo you want to continue viewing Blood Type information?(Yes/No)");

field=input.next();

}

///////////////////////////////

case "3":

System.out.println("Do you want to see "

+ "\n1.'All' Patient Records? "

+ "\n2. Approve Patient Request "

+ "\n3. Misc");

ques2=input.next();

switch(ques2){

do{

case "1":

query="select \* from Recipient";

Statement st = con.createStatement();

ResultSet rs = st.executeQuery(query);

ResultSetMetaData rsmd=rs.getMetaData();

int c=rsmd.getColumnCount();

int i=c;

while(c!=0){

c--;

System.out.format("%16s",rsmd.getColumnName(i-c));

}

System.out.println();

while(rs.next()){

c=i;

while(c!=0){

c--;

System.out.format("%16s",rs.getString(i-c));

}

System.out.println();

}

break;

case "2":

System.out.println("Enter the id of the patient you want to update requests of");

int id;

enter=input.nextInt();

query="update table Recipient set approved=TRUE where recipient\_id=?";

String query2="update table blood set blood\_units=blood\_units-?";

/////////////////dooooooo

PreparedStatement st1 = con.prepareStatement(query);

st1.setInt(1,id);

ResultSet rs1 = st1.executeQuery();

ResultSetMetaData rsmd1=rs1.getMetaData();

int c1=rsmd1.getColumnCount();

int i1=c1;

while(c1!=0){

c1--;

System.out.format("%16s",rsmd1.getColumnName(i1-c1));

}

System.out.println();

while(rs1.next()){

c1=i1;

while(c1!=0){

c1--;

System.out.format("%16s",rs1.getString(i1-c1));

}

System.out.println();

}

break;

case "3":

System.out.println("Enter your query in MYSQL");

String temp=input.nextLine();

query=input.nextLine();

Statement st2 = con.createStatement();

ResultSet rs2 = st2.executeQuery(query);

ResultSetMetaData rsmd2=rs2.getMetaData();

int c2=rsmd2.getColumnCount();

int i2=c2;

while(c2!=0){

c2--;

System.out.format("%16s",rsmd2.getColumnName(i2-c2));

}

System.out.println();

while(rs2.next()){

c2=i2;

while(c2!=0){

c2--;

System.out.format("%16s",rs2.getString(i2-c2));

}

System.out.println();

}

break;

//some case

case "External":

System.out.println("Choose one of the following options (Enter Choice No.): "

+ "\n1.View Hospital list"

+ "\n2.Manage hospitals Request"

+ "\n3.Manage Blood Donation drive");

ques2=input.next();

switch(ques2){

case "1":

query="select \* from Hospital";

st = con.createStatement();

rs = st.executeQuery(query);

rsmd=rs.getMetaData();

c=rsmd.getColumnCount();

i=c;

while(c!=0){

c--;

System.out.format("%16s",rsmd.getColumnName(i-c));

}

System.out.println();

while(rs.next()){

c=i;

while(c!=0){

c--;

System.out.format("%16s",rs.getString(i-c));

}

System.out.println();

}

break;

case "2":

System.out.println("Enter ID of the hospital you want to approve the request of");

enter=input.nextInt();

String query2;

query="update Hospital set status='1' where hospital\_id=?";

query2="update Blood set units= blood\_units-? where blood\_type='unitS\_requested";

st = con.createStatement();

rs = st.executeQuery(query);

rsmd=rs.getMetaData();

c=rsmd.getColumnCount();

i=c;

while(c!=0){

c--;

System.out.format("%16s",rsmd.getColumnName(i-c));

}

System.out.println();

while(rs.next()){

c=i;

while(c!=0){

c--;

System.out.format("%16s",rs.getString(i-c));

}

System.out.println();

}

break;

case "3":

System.out.println("Enter 1 to view the list of existing drives "

+ "\n 2 To create an event for blood "

+ "\n 3 To edit existing details");

ques2=input.next();

switch(ques2){

case "1":

query="select \* from Blood\_donation\_drive";

st = con.createStatement();

rs = st.executeQuery(query);

rsmd=rs.getMetaData();

c=rsmd.getColumnCount();

i=c;

while(c!=0){

c--;

System.out.format("%16s",rsmd.getColumnName(i-c));

}

System.out.println();

while(rs.next()){

c=i;

while(c!=0){

c--;

System.out.format("%16s",rs.getString(i-c));

}

System.out.println();

}

break;

case "2":

do{

query2 = "insert into Blood\_donation\_drive values(?,?,?,?)";

PreparedStatement st4 = con.prepareStatement(query2);

System.out.println("Enter Location");

int temp1;

String temp2;

temp2=input.next();

st4.setString(1, temp2);

System.out.println("Enter Number of Donors registered yet");

temp1=input.nextInt();

st4.setInt(2, temp1);

System.out.println("Enter Date(YYYY-MM-DD) of blood donation drive ");

temp2=input.next();

st4.setString(3, temp2);

System.out.println("Enter Blood\_bank\_id of your bank");

temp1=input.nextInt();

st4.setInt(4, temp1);

int count2 = st4.executeUpdate();

if(count2==1)

System.out.println("Successful insertion");

System.out.println("\nDo you want to continue creating new drives?(Yes/No)");

field=input.next();

}while(field.equals("Yes"));

break;

case "3":

do{

System.out.println("Enter the id of the drive you want to edit details of ");

int id=input.nextInt();

System.out.println("Enter 1. to edit Location "

+ "\n2. to update registered number of donors"

+ "\n3. to change date");

enter=input.nextInt();

switch(enter){

case 1:

query2="update blood\_donation\_drive set location=?";

System.out.println("Enter new location");

PreparedStatement st5 = con.prepareStatement(query2);

String temp2=input.next();

st5.setString(1, temp2);

int count2 = st5.executeUpdate();

if(count2==1)

System.out.println("Successful updation");

break;

//////////// while loop lgana hai?

case 2:

query2="update blood\_donation\_drive set registered\_donors=?";

System.out.println("Enter updated number of registered donors");

PreparedStatement st6 = con.prepareStatement(query2);

int temp1=input.nextInt();

st6.setInt(1, temp1);

int count1 = st6.executeUpdate();

if(count1==1)

System.out.println("Successful updation");

break;

case 3:

query=query2="update blood\_donation\_drive set registered\_donors=?";

System.out.println("Enter Changed Date (YYYY-MM-DD)");

st6 = con.prepareStatement(query2);

temp2=input.next();

st6.setString(1, temp2);

count1 = st6.executeUpdate();

if(count1==1)

System.out.println("Successful updation");

break;

}System.out.println("Do you want to continue adding, viewing and deleting donation drives? ");

field=input.next();

}while(field.equals("Yes"));

break;

} // donation drives ka switch case ends ---- break lagana hai?

System.out.println("\nDo you want to continue viewing information?(Yes/No)");

field=input.next();

}while(field.equals("Yes"));

break;

case "Receptionist":

System.out.println("Enter your id");

int id=input.nextInt();

System.out.println("Enter"

+ "\n1. For viewing Patients' List"

+ "\n2. For viewing Donors' List"

+ "\n3. For registering Donors"

+ "\n4. For registering Recipients");

enter=input.nextInt();

do{

switch(enter){

case 1:

query="show view receptionist\_donor";

break;

/////////// view wali cheez?

case 2:

query="show view receptionist\_recipients";

break;

case 3:

do{

System.out.println("Enter the id of donor you want to register");

enter=input.nextInt();

query="update donor set registered\_by=? where donor\_id=?";

PreparedStatement st4 = con.prepareStatement(query);

st4.setInt(1, id);

st4.setInt(2, enter);

int count1 = st4.executeUpdate();

if(count1==1)

System.out.println("Successful updation");

System.out.println("Do you want to continuing registering donors?");

field=input.next();

} //// break yahan bhi?

while(field.equals("Yes"));

break;

case 4:

do{

System.out.println("Enter the id of recipient you want to register");

enter=input.nextInt();

query="update recipient set registered\_by=? where recipient\_id=?";

PreparedStatement st4 = con.prepareStatement(query);

st4.setInt(1, id);

st4.setInt(2, enter);

int count1 = st4.executeUpdate();

if(count1==1)

System.out.println("Successful updation");

System.out.println("Do you want to continuing registering recipients?");

field=input.next();

} //// break yahan bhi?

while(field.equals("Yes"));

break;

}System.out.println("Do you want to continuing working?");

field=input.next();

} while(field.equals("Yes"));

break; // switch for Reception ends here

case "2": // visitor

System.out.println("Enter whether you want to donate or recieive blood"

+"\n press D for Donor "

+"\n press R for Recipient");

ques2=input.next();

switch(ques2){

case "D":

System.out.println("Answer the following questions and submit valid proofs of the same:"+

"\n Q1. Is your weight above 47kg? ");

String a1=input.next();

System.out.println("Q2. Are you HIV positive or negative? Enter P or N");

String a2=input.next();

System.out.println("Q2. Is your blood devoid of hard-drugs/alchohol? ");

String a3=input.next();

if(a1.equals("Yes") && a2.equals("N") && a3.equals("Yes")){

String query1="insert into Visitor values(?,?,?,?)";

PreparedStatement st4 = con.prepareStatement(query1);

System.out.println("Enter Aadhar\_id");

int temp1;

String temp2;

temp1=input.nextInt();

st4.setInt(1, temp1);

System.out.println("Enter Name");

temp2=input.next();

st4.setString(2, temp2);

System.out.println("Enter Location");

temp2=input.next();

st4.setString(3, temp2);

System.out.println("Enter Date of Birth(YYYY-MM-DD)");

temp2=input.next();

st4.setString(4, temp2);

int count2 = st4.executeUpdate();

///////////TRIGGGERRRRRRR

String query2="insert into Donor values(?,?)";

PreparedStatement st5 = con.prepareStatement(query2);

System.out.println("Enter Donor\_id");

temp1=input.nextInt();

st5.setInt(1, temp1);

System.out.println("Enter Blood type you want to donate");

temp2=input.next();

st5.setString(2, temp2);

count2 += st5.executeUpdate();

if(count2==2)

System.out.println("Successful insertion");

}

else{

System.out.println("Sorry, you can't donate blood");

}

break;

case "R":

String query1="insert into Visitor values(?,?,?,?)";

PreparedStatement st4 = con.prepareStatement(query1);

System.out.println("Enter Aadhar\_id");

int temp1;

String temp2;

temp1=input.nextInt();

st4.setInt(1, temp1);

System.out.println("Enter Name");

temp2=input.next();

st4.setString(2, temp2);

System.out.println("Enter Location");

temp2=input.next();

st4.setString(3, temp2);

System.out.println("Enter Date of Birth(YYYY-MM-DD)");

temp2=input.next();

st4.setString(4, temp2);

int count3 = st4.executeUpdate();

String query2="insert into Donor values(?,?,?)";

PreparedStatement st5 = con.prepareStatement(query2);

System.out.println("Enter Recipient\_id");

temp1=input.nextInt();

st5.setInt(1, temp1);

System.out.println("Enter Blood type you want to recieve");

temp2=input.next();

st5.setString(2, temp2);

count3 += st5.executeUpdate();

if(count3==2){

System.out.println("Successful insertion");}

else{

System.out.println("Sorry, Please Try again");}

break;

}

}

}

}

}

System.out.println("Do you want to continue working in blood bank management system? Yes or No");

ques1=input.next();

}while(ques1.equals("Yes")); // first do ends

System.out.println("ThankYou!"); //main ka hi part

input.close();

con.close();

} //main ends

} // Code Ends

Thus, the connection from Java to Oracle database is

performed and therefore, can be used for updating tables in

the database directly.

**SOFTWARE USED:**

*Java Eclipse, Oracle 11g Database, Java SE version 7,*

*SQL\*Plus.*

**Java AWT:**

*Java AWT (Abstract Window Toolkit) is an API to develop*

*GUI or window-based applications in java. Java AWT components are platform-dependent i.e.*

*components are displayed according to the view of operating*

*system.*

*AWT is heavyweight i.e. its components are using the*

*resources of OS. The java.awt package provides classes for*

*AWT API such as TextField, Label, TextArea, RadioButton,*

*CheckBox, Choice, List etc.*

***SQL:***

*Structure Query Language(SQL) is a database query*

*language used for storing and managing data in Relational*

*DBMS. SQL was the first commercial language introduced*

*for E.F Codd's Relational model of database. Today almost all*

*RDBMS (MySQL, Oracle, Infomix, Sybase, MS Access) use*

*SQL as the standard database query language. SQL is used to*

*perform all types of data operations in RDBMS.*

***GITHUB LINK:***

*https://github.com/Sragvi13/DBMS-project*

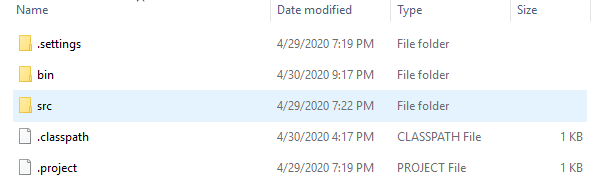
*FOLDER STRUCTURE:*

*This project contains a folder named src in which it has 5*

*different folders for different purposes each folder has 3 codes*

*such as to make insert, delete, update. By this we can navigate easily to reach code and we can make many changes*

*as we can want easily.*



**TESTING**

*The program executes three basic operations those are insert*

*update and delete on 5 different tables. Along with this, it*

*also has an output column which gives information about*

*how many rows have been edited. Errors syntactical or*

*exceptional will be shown if occurred.*

**DISCUSSION AND FUTURE WORK**

So far this project has helped us retrieve the information about Google

queries through the feedback and rating given by the users. The

feedback eventually is the crucial aspect here as it is the parameter that

rates the query and its solution. Qualitative feedbacks give a room for

improvising things so that one can come up with a better solution.

In future the most probable aspects could be involving analyzers with a

high intellect merely to improve the feedback quality. The query also

could be analyzed based on how difficult or easy its solution could be.

The feedback system could be made more comprehensive rather than

just being stuck to either positive or negative feedbacks. Lastly, there

could be a room for including advanced software and other technologies

that could make the project more purposeful and better for future use.

**REFERENCES**

https://www.oracle.com/in/database/technologies/112010-

win64soft.html

https://www.youtube.com/watch?v=fMp63HsIRbc&t=107s

https://mail.google.com/mail/u/0/#all/FMfcgxwHMGDXGBJRHPNwxk

ZSqQDLgPMR

https://mail.google.com/mail/u/0/#search/sam/FMfcgxwHMZGcqMdcj

VXHfcLtWgdqPZdK