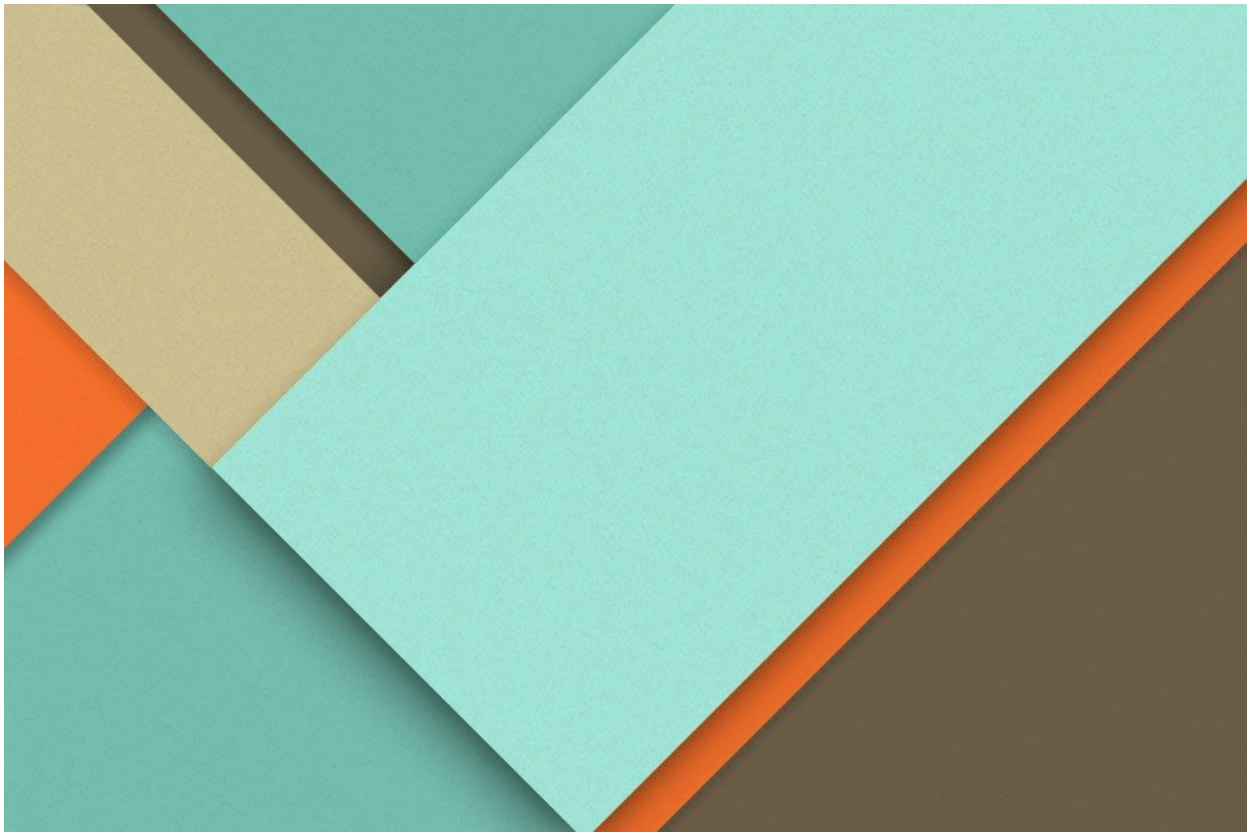


BLOOD BANK MANAGEMENT SYSTEM PROJECT REPORT

DATABASE MANAGEMENT SYSTEMS (CSE2004) - EMBEDDED PROJECT - SLOT L37+L38

SUBMITTED TO DR. J. VELLINGIRI

SUBMITTED BY

AMIT KRISHNA A (19BCE0197)

DIVYA MAHESH (19BCE2519)

ARYAN DOKANIA (19BCE2534)

ABSTRACT

Blood Bank Management System aims at properly storing, updating, analyzing, and retrieving data related to blood donors and the blood banks in a particular city. It ensures that the process of blood donation to a patient in need is completely hassle-free. Details of each donor, as well as the details of the availability of different blood groups in each blood bank, are stored in the database. By using standard queries, these details can be accessed only by the authorized personnel at hospitals and blood banks, in order to ensure data security. The Blood Bank Management System provides a user-friendly interface, which makes it easier for the end-users to access the data.

INTRODUCTION

Blood transfusion has been responsible for saving millions of lives each year around the world. Yet the quantity and quality of blood pool available for transfusions is still a major concern across the globe, especially in developing countries.

Many blood banks in India still lack the needed facilities to make blood components and thus most of them issue whole blood; thus, contributing to the shortage of blood and unnecessarily overburdening the patient causes harm at times, as blood transfusion reactions are more common with whole blood transfusions.

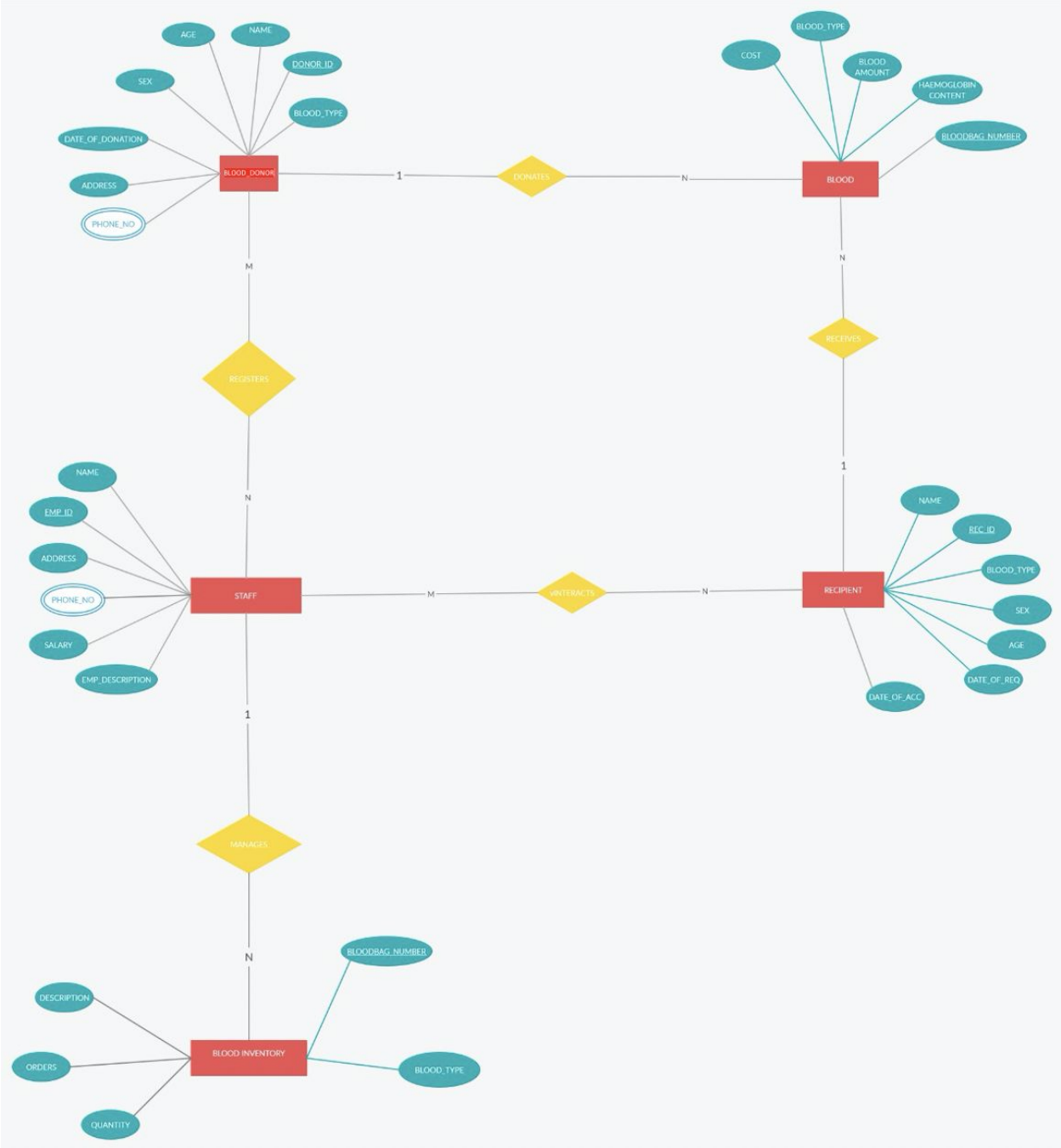
Ideally, there should be enough blood units in a blood bank for the everyday requirements for optimum functioning of the health-care system. However, the non-availability of sufficient blood units is a major problem.

Blood donation involves the collection of enormous amounts of donor data, hence there must be an efficient and successful way of managing data. With an increasing population, it has become necessary to ensure that the process of data retrieval is not hindered by conventional manual operator's data entry techniques. The project is carried out on an automated blood bank to solve any issue. We make sure all the pertinent details required

to be kept track of for each blood transfusion process are properly recorded in our database. The main aim of this project will therefore be to find more effective ways of managing

the database of blood banks and blood donors and establish a forum for people in order to connect to potential blood donors in the region.

ER DIAGRAM


AGE
NAME
DONOR_ID
SEX
BLOOD_TYPE
DATE_OF_DONATION
ADDRESS
PHONE_NO
BLOOD_DONOR
1
DONATES
N
BLOOD
COST
BLOOD_TYPE
BLOOD_AMOUNT
HAEMOGLOBIN_CONTENT
BLOODBAG_NUMBER
M
N
REGISTERS
RECEIVES
1
N
NAME
EMP_ID
ADDRESS
PHONE_NO
SALARY
EMP_DESCRIPTION
STAFF
M
INTERACTS
N
RECIPIENT
NAME
REC_ID
BLOOD_TYPE
SEX
AGE
DATE_OF_REQ
DATE_OF_ACC
1
MANAGES
N
BLOODBAG_NUMBER
DESCRIPTION
ORDERS
QUANTITY
BLOOD_INVENTORY
BLOOD_TYPE

ER DIAGRAM TO SCHEMA CONVERSION

BLOOD_DONOR(DONOR_ID , NAME, BLOOD_TYPE, AGE, SEX, DATE_OF_DONATION, ADDRESS)

DONOR_PHNO(DONOR_ID , PHONE_NO)

BLOOD(BLOODBAG_NUMBER, DONOR_ID, REC_ID, HAEMOGLOBIN_CONTENT, BLOOD_AMOUNT, BLOOD_TYPE, COST)

STAFF(EMP_ID, NAME, ADDRESS, SALARY, EMP_DESCRIPTION)

STAFF_PHNO(EMP_ID, PHONE_NO)

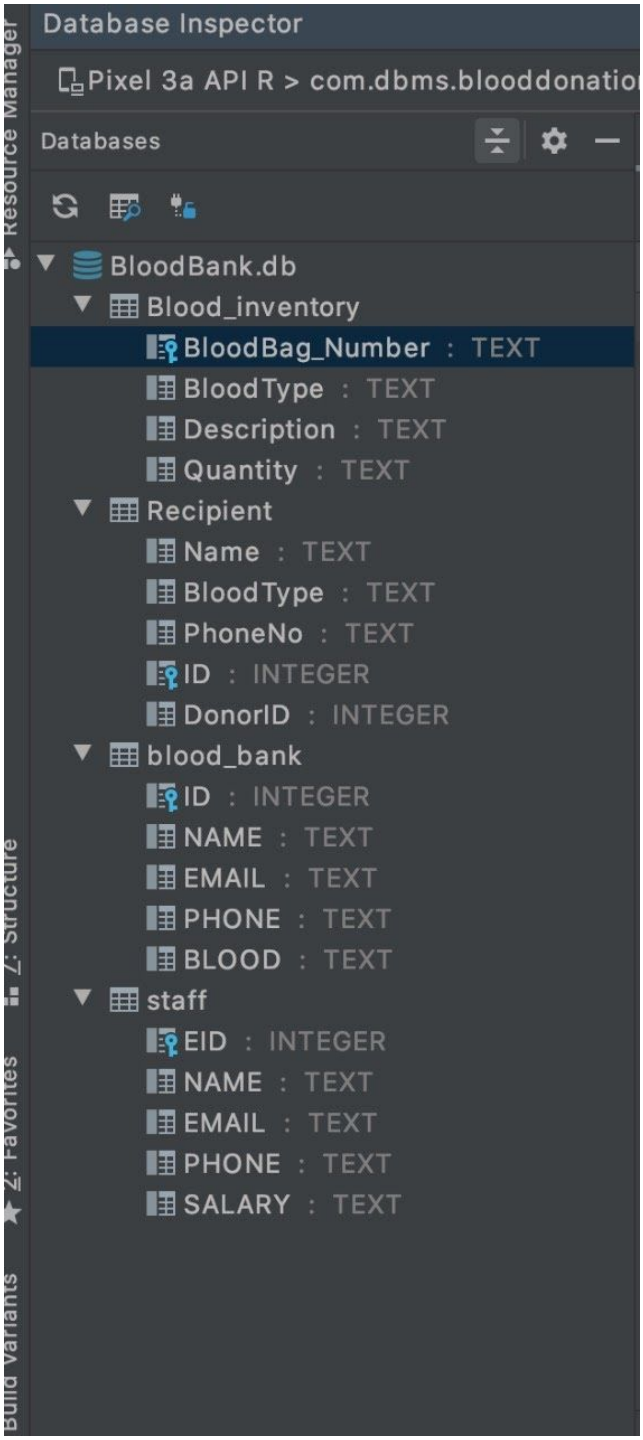
REGISTERS(DONOR_ID,EMP_ID)

RECIPENT(REC_ID, NAME, BLOOD_TYPE, SEX, AGE, DATE_OF_ACCEP, DATE_OF_REQ)

INTERACTS(EMP_ID,REC_ID)

BLOOD_INVENTORY(BLOODBAG_NUMBER, BLOOD_TYPE, DESCRIPTION, ORDERS, QUANTITY, EMP_ID)

CREATION OF NEW RELATIONAL TABLES



NORMALIZATION OF TABLES

① BLOOD_BANK (ID, NAME, EMAIL, PHONE, BLOOD)

Let,

ID ⇔ A

NAME ⇔ B

EMAIL ⇔ C

PHONE ⇔ D

BLOOD ⇔ E

∴ The functional dependencies are :

$A \rightarrow B$

$A \rightarrow C$

$A \rightarrow D$

$A \rightarrow E$

(a) 1NF :

The relation is in 1NF because its attributes contain only atomic values.

(b) 2NF:

ⓐ The relation is in 1NF

ⓑ does not contain any partial dependency.

Hence, it is in 2NF

(c) 3NF:

ⓐ it is 2NF

ⓑ no non-primary key attribute is ~~fully~~ transitively dependent on the primary key.

Hence, it is in 3NF.

(d) BCNF : relation is in BCNF, because :

ⓐ relation is in 3NF

ⓑ for every non-trivial functional dependency:

$A \rightarrow B$, $A \rightarrow C$, $A \rightarrow D$, $A \rightarrow E$

A is the super key of the relation.

② BLOOD-INVENTORY (BLOODBAG-NUMBER, BLOODTYPE, DESCRIPTION, QUANTITY)

Let,

BLOODBAG_NUMBER ⇔ A

BLOODTYPE ⇔ B

DESCRIPTION ⇔ C

QUANTITY ⇔ D

∴ The functional dependencies are :

$A \rightarrow B$

$A \rightarrow C$

$A \rightarrow D$

(a) 1NF:

The relation is in 1NF because its attributes contain only atomic values.

(b) 2NF:

ⓐ relation is in 1NF

ⓑ does not contain any partial dependency, i.e, no nonkey attribute should be functionally dependent on a part of the primary key.

Hence, it is in 2NF

(c) 3NF:

ⓐ it is in 2NF

ⓑ There is no transitive dependency of non-key attribute on a primary key.
Hence, it is in 3NF

(d) BCNF: it is in BCNF, because:

ⓐ it is in 3NF

ⓑ for every non-trivial functional dependency:

$A \rightarrow B$

$A \rightarrow C$

$A \rightarrow D$

A is a super key of the relation.

③ RECIPIENT (NAME, BLOODTYPE, PHONENO, ID, DONORID)

Let,

ID ⇔ A

NAME ⇔ B

BLOODTYPE ⇔ C

PHONENO ⇔ D

DONORID ⇔ E

∴ The functional dependencies are:

$A \rightarrow B$

$A \rightarrow C$

$A \rightarrow D$

$A \rightarrow E$

(a) 1NF:

The relation is in 1NF because its attributes contain only atomic values

(b) 2NF:

ⓐ relation is in 1NF

ⓑ does not contain any partial dependency, i.e. no non key attribute is functionally dependent on a part of the primary key.

Hence, it is in 2NF

(c) 3NF:

ⓐ it is in 2NF

ⓑ there is no transitive dependency of non-key attributes on a primary key.

Hence, it is in 3NF.

(d) BCNF:

ⓐ it is in 3NF

ⓑ for every non-trivial functional dependency:

$A \rightarrow B$, $A \rightarrow C$, $A \rightarrow D$, $A \rightarrow E$

A is a super key of the relation.

Hence it is in BCNF.

④ STAFF (EID , NAME , EMAIL , PHONE , SALARY)

Let,

EID ⇔ A

NAME ⇔ B

EMAIL ⇔ C

PHONE ⇔ D

SALARY ⇔ E

∴ The functional dependencies are:

$A \rightarrow B$

$A \rightarrow C$

$A \rightarrow D$

$A \rightarrow E$

(a) 1NF:

The relation is in 1NF because its attributes contain only atomic values.

(b) 2NF:

ⓐ the relation is in 1NF.

ⓑ does not contain any partial dependency, ie:, no nonkey attribute is functionally dependent on

SAMPLE CODE FOR DATABASE AND FRONT END CONNECTIVITY

```
package com.dbms.blooddonation;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.util.Log;

import androidx.annotation.Nullable;

public class DatabaseHelper extends SQLiteOpenHelper {
    public static final String DB_NAME = "BloodBank.db";
    public static final String TABLE = "blood_bank";
    public static final String TABLE1 = "staff";
    public static final String TABLE2 = "Blood_inventory";
    public static final String TABLE3 = "Recipient";
    public static final String COL_1 = "ID";
    public static final String COL_2 = "Name";
    public static final String COL_3 = "Email";
    public static final String COL_4 = "Phone";
    public static final String COL_5 = "Blood";
    public static final String COL_6 = "EID";
    public static final String COL_7 = "Name";
    public static final String COL_8 = "Email";
    public static final String COL_9 = "Phone";
    public static final String COL_10 = "Salary";
    public static final String COL_11 = "Bloodbag_Number";
    public static final String COL_12 = "BloodType";
    public static final String COL_13 = "Description";
    public static final String COL_14 = "Quantity";
    public static final String COL_15 = "Name";
    public static final String COL_16 = "PhoneNo";
    public static final String COL_17 = "DonorID";
    public static final String COL_18 = "BloodType";

    public DatabaseHelper(@Nullable Context context) {
        super(context, DB_NAME, null, 3);
    }

    @Override
    public void onCreate(SQLiteDatabase sqLiteDatabase)
    {
        sqLiteDatabase.execSQL("CREATE TABLE " + TABLE + "(ID INTEGER PRIMARY KEY
AUTOINCREMENT, NAME TEXT, EMAIL TEXT, PHONE TEXT, BLOOD TEXT)");
        sqLiteDatabase.execSQL("CREATE TABLE " + TABLE1 + "(EID INTEGER PRIMARY KEY
```

```
AUTOINCREMENT , NAME TEXT, EMAIL TEXT, PHONE TEXT, SALARY TEXT)");
        sqLiteDatabase.execSQL("CREATE TABLE " + TABLE2 + "(BloodBag_Number TEXT PRIMARY KEY ,
BloodType TEXT, Description TEXT, Quantity TEXT)");
        sqLiteDatabase.execSQL("CREATE TABLE " + TABLE3 + "(Name TEXT , BloodType TEXT, PhoneNo
TEXT, ID INTEGER PRIMARY KEY AUTOINCREMENT, DonorID INTEGER)");
    }

    @Override
    public void onUpgrade(SQLiteDatabase sqLiteDatabase, int i, int i1) {
        sqLiteDatabase.execSQL("DROP TABLE IF EXISTS " + TABLE );
        sqLiteDatabase.execSQL("DROP TABLE IF EXISTS " + TABLE1 );
        sqLiteDatabase.execSQL("DROP TABLE IF EXISTS " + TABLE2 );
        sqLiteDatabase.execSQL("DROP TABLE IF EXISTS " + TABLE3);
        onCreate(sqLiteDatabase);
    }

    public boolean insertData3(String name, String bloodGroup, String phno){
        SQLiteDatabase sqLiteDatabase = this.getWritableDatabase();
        ContentValues contentValues = new ContentValues();
        contentValues.put(COL_15, name);
        contentValues.put(COL_18, bloodGroup);
        contentValues.put(COL_16, phno);
        long result = sqLiteDatabase.insert(TABLE3, COL_17, contentValues);
        if (result == -1){
            return false;
        }
        return true;
    }

    public boolean addDonorToRec(int donorID, int recID){
        SQLiteDatabase sqLiteDatabase = this.getWritableDatabase();
        ContentValues contentValues = new ContentValues();
        contentValues.put(COL_17, donorID);
        long result = sqLiteDatabase.update(TABLE3, contentValues,  "ID = ?", new
String[]{String.valueOf(recID)});
        if (result == -1){
            Log.d("add", "addDonorToRec: ");
            return false;
        }
        Log.d("add", "addDonorToRec: " + donorID + "  " + recID);
        return true;
    }

    public boolean insertData2(String bloodid,String bloodname,String blooddesc, String quantity)
    {
        SQLiteDatabase sqLiteDatabase = this.getWritableDatabase();
        ContentValues contentValues = new ContentValues();
        contentValues.put(COL_11, bloodid);
        contentValues.put(COL_12, bloodname);
        contentValues.put(COL_13, blooddesc);
```

```
    contentValues.put(COL_14, quantity);
    long result = sqLiteDatabase.insert(TABLE2, null, contentValues);
    if (result == -1){
        return false;
    }
    return true;
 }
public boolean insertData1(Integer eid,String name,String email, String phone, String salary )
{
   SQLiteDatabase sqLiteDatabase = this.getWritableDatabase();
   ContentValues contentValues = new ContentValues();
   contentValues.put(COL_6, eid);
   contentValues.put(COL_7, name);
   contentValues.put(COL_8, email);
   contentValues.put(COL_9, phone);
   contentValues.put(COL_10, salary);
   long result = sqLiteDatabase.insert(TABLE1, null, contentValues);
   if (result == -1){
      return false;
   }
   return true;
}
 public boolean insertData(String name, String email, String phone, String blood){
    SQLiteDatabase sqLiteDatabase = this.getWritableDatabase();
    ContentValues contentValues = new ContentValues();
    contentValues.put(COL_2, name);
    contentValues.put(COL_3, email);
    contentValues.put(COL_4, phone);
    contentValues.put(COL_5, blood);
    long result = sqLiteDatabase.insert(TABLE, null, contentValues);

    if (result == -1){
        return false;
    }
    return true;
 }
 public Cursor getAllData1(){
    SQLiteDatabase sqLiteDatabase = this.getWritableDatabase();
    Cursor res1 = sqLiteDatabase.rawQuery("SELECT * FROM " + TABLE1, null);
    return res1;
 }
 public Cursor getAllData2(){
    SQLiteDatabase sqLiteDatabase = this.getWritableDatabase();
    Cursor res2 = sqLiteDatabase.rawQuery("SELECT * FROM " + TABLE2, null);
    return res2;
 }

 public Cursor getAllData3(){
    SQLiteDatabase sqLiteDatabase = this.getWritableDatabase();
    Cursor res3 = sqLiteDatabase.rawQuery("SELECT * FROM " + TABLE3 + " ORDER BY " + COL_17 + "
```

```
ASC   ", null);
        return res3;
    }

    public Cursor getAllData(){
        SQLiteDatabase sqLiteDatabase = this.getWritableDatabase();
        Cursor res = sqLiteDatabase.rawQuery("SELECT * FROM " + TABLE, null);
        return res;
    }

    public Cursor getPossibleDonors(String blood){
        SQLiteDatabase sqLiteDatabase = this.getWritableDatabase();
        Cursor res = sqLiteDatabase.rawQuery("SELECT * FROM " + TABLE + " WHERE blood = ?", new
String[]{blood});
        return res;
    }

    public boolean updateData(String id, String name, String email, String phone, String blood){
        SQLiteDatabase sqLiteDatabase = this.getWritableDatabase();
        ContentValues contentValues = new ContentValues();
        contentValues.put(COL_2, name);
        contentValues.put(COL_3, email);
        contentValues.put(COL_4, phone);
        contentValues.put(COL_5, blood);
        sqLiteDatabase.update(TABLE, contentValues, "ID = ?", new String[] {id} );
        return true;
    }

    public boolean deleteData(String id){
        SQLiteDatabase sqLiteDatabase = this.getWritableDatabase();
        sqLiteDatabase.delete(TABLE, "ID = ?", new String[] {id});
        return true;
    }
}
```

Android Manifest

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.dbms.blooddonation">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
```

```
        android:supportsRtl="true"
        android:theme="@style/AppTheme"
        tools:ignore="GoogleAppIndexingWarning">
        <activity android:name=".RecipientResultsActivity"></activity>
        <activity android:name=".RecipientActivity" />
        <activity android:name=".InventoryResults" />
        <activity android:name=".BloodInventory" />
        <activity android:name=".SplashActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".NeedResults" />
        <activity android:name=".StaffResult" />
        <activity android:name=".StaffActivity" />
        <activity android:name=".EnrollActivity" />
        <activity android:name=".Frontactivity" />
        <activity android:name=".BloodResults" />
        <activity android:name=".MainActivity" />
        <activity android:name=".BloodActivity" />
    </application>

</manifest>
```

FRONT ACTIVITY

```
package com.dbms.blooddonation;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.view.View;
import android.widget.Button;
import android.os.Bundle;

public class Frontactivity extends AppCompatActivity {

    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_frontactivity);
        Button button=findViewById(R.id.admin);
        Button b1=findViewById(R.id.enroll);
        Button b2=findViewById(R.id.need);
        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent adIntent = new Intent(Frontactivity.this, BloodActivity.class);
                startActivity(adIntent);
            }
        });
        b1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent enIntent = new Intent(Frontactivity.this, EnrollActivity.class);
                startActivity(enIntent);
            }
        });
        b2.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent needIntent = new Intent(Frontactivity.this, RecipientActivity.class);
                startActivity(needIntent);
            }
        });

    }

}
```

Blood Activity

```
package com.dbms.blooddonation;

import androidx.appcompat.app.AppCompatActivity;

import android.app.AlertDialog;
import android.content.Intent;
import android.database.Cursor;
import android.os.Bundle;
```

```
import android.text.TextUtils;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Spinner;
import android.widget.Toast;

import maes.tech.intentanim.CustomIntent;

public class BloodActivity extends AppCompatActivity implements
AdapterView.OnItemSelectedListener {

    DatabaseHelper databaseHelper;
    EditText name, email, phone, id;
    Button insert, update, delete, view,er,dr;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_blood);

        databaseHelper = new DatabaseHelper(this);

        id = findViewById(R.id.donor_id);
        name = findViewById(R.id.donor_name);
        email = findViewById(R.id.donor_email);
        phone = findViewById(R.id.donor_phone);
        er=findViewById(R.id.ins);
       dr=findViewById(R.id.ins1);
        insert = findViewById(R.id.insert);
        view = findViewById(R.id.view);
        update = findViewById(R.id.update);
        delete = findViewById(R.id.delete);

        final Spinner blood_group_spinner = findViewById(R.id.spinner_bg);
        ArrayAdapter<CharSequence> adapter = ArrayAdapter.createFromResource(this,
R.array.blood_groups,R.layout.spinner_item);

adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
        blood_group_spinner.setAdapter(adapter);
```

```
blood_group_spinner.setOnItemSelectedListener(this);
 er.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v)
    {
        Intent bIntent = new Intent(BloodActivity.this, StaffActivity.class);
        startActivity(bIntent);
    }
 });
 dr.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent dIntent = new Intent(BloodActivity.this, BloodInventory.class);
        startActivity(dIntent);
    }
 });
insert.setOnClickListener(new View.OnClickListener() {
   @Override
   public void onClick(View view) {
      String d_name = name.getText().toString();
      String d_email = email.getText().toString();
      String d_phone = phone.getText().toString();
      String d_blood = blood_group_spinner.getSelectedItem().toString();

      if( !TextUtils.isEmpty(d_blood) && !TextUtils.isEmpty(d_email) &&
!TextUtils.isEmpty(d_name) && !TextUtils.isEmpty(d_phone)){
         boolean isInserted =
databaseHelper.insertData(d_name,d_email,d_phone,d_blood);
         if (isInserted)
            Toast.makeText(BloodActivity.this, "Data Inserted Successfully",
Toast.LENGTH_LONG).show();
         else
            Toast.makeText(BloodActivity.this, "Insertion Error",
Toast.LENGTH_LONG).show();
         name.setText("");
         email.setText("");
         phone.setText("");
         Intent bloodIntent = new Intent(BloodActivity.this, BloodResults.class);
         startActivity(bloodIntent);
         CustomIntent.customType(BloodActivity.this, "right-to-left");
      }else{
         Toast.makeText(BloodActivity.this, "Required Fields Cannot Be Blank",
```

```
Toast.LENGTH_LONG).show();
            }
        }
    });

    view.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Cursor result = databaseHelper.getAllData();
            if(result.getCount() == 0){
                showMessage("Error", "No Data");
                return;
            }else{
                Intent bloodIntent = new Intent(BloodActivity.this, BloodResults.class);
                startActivity(bloodIntent);
                CustomIntent.customType(BloodActivity.this, "right-to-left");
            }

        }
    });

    update.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            String uID = id.getText().toString();
            if (!TextUtils.isEmpty(uID)){
                String u_name = name.getText().toString();
                String u_email = email.getText().toString();
                String u_phone = phone.getText().toString();
                String u_blood = blood_group_spinner.getSelectedItem().toString();

                boolean isUpdate = databaseHelper.updateData(uID, u_name, u_email,
u_phone, u_blood);

                if(isUpdate){
                    Toast.makeText(BloodActivity.this, "Data Updated Successfully",
Toast.LENGTH_LONG).show();
                }else{
                    Toast.makeText(BloodActivity.this, "Update Error!",
Toast.LENGTH_LONG).show();
                }
                name.setText("");
```

```
                email.setText("");
                phone.setText("");
                id.setText("");
            }else{
                Toast.makeText(BloodActivity.this, "Required Fields Cannot Be Blank",
Toast.LENGTH_LONG).show();
            }
        }
    });
}

@Override
public void onItemSelected(AdapterView<?> adapterView, View view, int i, long l) {
}

@Override
public void onNothingSelected(AdapterView<?> adapterView) {

}

public void showMessage(String title, String message){
    AlertDialog.Builder builder = new AlertDialog.Builder(this);
    builder.setCancelable(true);
    builder.setTitle(title);
    builder.setMessage(message);
    builder.show();
}

}
```

Blood Inventory Activity

```
package com.dbms.blooddonation;

import androidx.appcompat.app.AppCompatActivity;

import android.app.AlertDialog;
import android.content.Intent;
import android.database.Cursor;
import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
```

```
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import maes.tech.intentanim.CustomIntent;

public class BloodInventory extends AppCompatActivity implements
AdapterView.OnItemSelectedListener {

    DatabaseHelper databaseHelper;
    EditText name, email, phone, id;
    Button insert, update, delete, view,er;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_blood_inventory);

        databaseHelper = new DatabaseHelper(this);

        id = findViewById(R.id.rec_name);
        name = findViewById(R.id.rec_blood_name);
        email = findViewById(R.id.blood_desc);
        phone = findViewById(R.id.blood_quantity);
        insert = findViewById(R.id.blood_insert);
        view = findViewById(R.id.blood_view);

        insert.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v)
            {
                Intent bIntent = new Intent(BloodInventory.this, StaffActivity.class);
                startActivity(bIntent);
            }
        });
        insert.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String b_id = id.getText().toString();
                String b_name = name.getText().toString();
                String b_desc = email.getText().toString();
```

```
            String b_quantity = phone.getText().toString();


            if(!TextUtils.isEmpty(b_desc) && !TextUtils.isEmpty(b_name) &&
!TextUtils.isEmpty(b_quantity)){
                boolean isInserted =
databaseHelper.insertData2(b_id,b_name,b_desc,b_quantity);
                if (isInserted)
                    Toast.makeText(BloodInventory.this, "Data Inserted Successfully",
Toast.LENGTH_LONG).show();
                else
                    Toast.makeText(BloodInventory.this, "Insertion Error",
Toast.LENGTH_LONG).show();
                name.setText("");
                email.setText("");
                phone.setText("");
            }else{
                Toast.makeText(BloodInventory.this, "Required Fields Cannot Be Blank",
Toast.LENGTH_LONG).show();
            }
        }
    });

    view.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Cursor result = databaseHelper.getAllData2();
            if(result.getCount() == 0){
                showMessage("Error", "No Data");
                return;
            }else{
                Intent bloodIntent = new Intent(BloodInventory.this, InventoryResults.class);
                startActivity(bloodIntent);
                CustomIntent.customType(BloodInventory.this, "right-to-left");
            }

        }
    });


}
```

```
    @Override
    public void onItemSelected(AdapterView<?> adapterView, View view, int i, long l) {
    }

    @Override
    public void onNothingSelected(AdapterView<?> adapterView) {

    }

    public void showMessage(String title, String message){
        AlertDialog.Builder builder = new AlertDialog.Builder(this);
        builder.setCancelable(true);
        builder.setTitle(title);
        builder.setMessage(message);
        builder.show();
    }

}
```

Blood Result Activity

```
package com.dbms.blooddonation;
import androidx.appcompat.app.AppCompatActivity;
import android.database.Cursor;
import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ListAdapter;
import android.widget.ListView;
import android.widget.Toast;
```

```
import java.util.ArrayList;

import maes.tech.intentanim.CustomIntent;

public class BloodResults extends AppCompatActivity {

    DatabaseHelper databaseHelper;
    EditText del_id_text;
    Button delete;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_blood_results);

        ListView listView = findViewById(R.id.listview);
        databaseHelper = new DatabaseHelper(this);

        del_id_text = findViewById(R.id.del_id);
        delete = findViewById(R.id.delete);

        ArrayList<String> arrayList = new ArrayList<>();
        Cursor data = databaseHelper.getAllData();

        if (data.getCount() == 0){
            Toast.makeText(BloodResults.this, "Database Empty", Toast.LENGTH_LONG).show();
        } else{
            StringBuffer stringBuffer = new StringBuffer();
            while(data.moveToNext()){
```

```
            arrayList.add("\n" + "ID: " + data.getString(0) + "\n" + "Name: " + data.getString(1) +
"\n" + "Email: " + data.getString(2) + "\n" + "Phone: " + data.getString(3) + "\n" + "Blood
Group: " + data.getString(4) + "\n");
        }
        ListAdapter listAdapter = new
ArrayAdapter<>(this,android.R.layout.simple_list_item_1, arrayList);
        listView.setAdapter(listAdapter);
    }


    delete.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            String dID = del_id_text.getText().toString();
            if (!TextUtils.isEmpty(dID)){
                boolean isDeleted = databaseHelper.deleteData(dID);
                if(isDeleted){
                    Toast.makeText(BloodResults.this, "Data Deleted Successfully",
Toast.LENGTH_LONG).show();
                }else{
                    Toast.makeText(BloodResults.this, "Delete Error!",
Toast.LENGTH_LONG).show();
                }
                del_id_text.setText("");
            }else{
                Toast.makeText(BloodResults.this, "Required Fields Cannot Be Blank",
Toast.LENGTH_LONG).show();
            }
        }
    });
  }
```

```
    @Override
    public void finish() {
        super.finish();
        CustomIntent.customType(BloodResults.this, "left-to-right");
    }
}
```

Enroll Activity

```
package com.dbms.blooddonation;
import androidx.appcompat.app.AppCompatActivity;
import android.app.AlertDialog;
import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Spinner;
import android.widget.Toast;
import java.util.Random;

public class EnrollActivity extends AppCompatActivity implements
AdapterView.OnItemSelectedListener {

    DatabaseHelper databaseHelper;
    EditText name, email, phone, id;
    Button insert, update, delete, view;
```

```
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_enroll);

        databaseHelper = new DatabaseHelper(this);

        id = findViewById(R.id.donor_id);
        name = findViewById(R.id.donor_name);
        email = findViewById(R.id.donor_email);
        phone = findViewById(R.id.donor_phone);

        insert = findViewById(R.id.insert);

        final Spinner blood_group_spinner = findViewById(R.id.spinner_bg);
        ArrayAdapter<CharSequence> adapter = ArrayAdapter.createFromResource(this,
R.array.blood_groups,R.layout.spinner_item);

adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
        blood_group_spinner.setAdapter(adapter);
        blood_group_spinner.setOnItemSelectedListener(this);

        insert.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Random r = new Random();
                int i1 = r.nextInt(45 - 28) + 28;
                String d_name = name.getText().toString();
                String d_email = email.getText().toString();
                String d_phone = phone.getText().toString();
```

```
            String d_blood = blood_group_spinner.getSelectedItem().toString();


            if( !TextUtils.isEmpty(d_blood) && !TextUtils.isEmpty(d_email) &&
!TextUtils.isEmpty(d_name) && !TextUtils.isEmpty(d_phone)){
                boolean isInserted =
databaseHelper.insertData(d_name,d_email,d_phone,d_blood);
                if (isInserted)
                    Toast.makeText(EnrollActivity.this, "Data Inserted Successfully",
Toast.LENGTH_LONG).show();
                else
                    Toast.makeText(EnrollActivity.this, "Insertion Error",
Toast.LENGTH_LONG).show();
                name.setText("");
                email.setText("");
                phone.setText("");
            }else{
                Toast.makeText(EnrollActivity.this, "Required Fields Cannot Be Blank",
Toast.LENGTH_LONG).show();
            }
        }
    });



}


@Override
public void onItemSelected(AdapterView<?> adapterView, View view, int i, long l) {
}


@Override
public void onNothingSelected(AdapterView<?> adapterView) {
```

```
    }

    public void showMessage(String title, String message){
        AlertDialog.Builder builder = new AlertDialog.Builder(this);
        builder.setCancelable(true);
        builder.setTitle(title);
        builder.setMessage(message);
        builder.show();
    }

}
```

Inventory Result Activity

```
package com.dbms.blooddonation;
import androidx.appcompat.app.AppCompatActivity;
import android.database.Cursor;
import android.os.Bundle;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ListAdapter;
import android.widget.ListView;
import android.widget.Toast;

import java.util.ArrayList;

import maes.tech.intentanim.CustomIntent;
```

```
public class InventoryResults extends AppCompatActivity {

    DatabaseHelper databaseHelper;
    EditText del_id_text;
    Button delete;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_inventory_results);

        ListView listView = findViewById(R.id.listview3);
        databaseHelper = new DatabaseHelper(this);

        ArrayList<String> arrayList = new ArrayList<>();
        Cursor data = databaseHelper.getAllData2();

        if (data.getCount() == 0){
            Toast.makeText(InventoryResults.this, "Database Empty",
Toast.LENGTH_LONG).show();
        } else{
            StringBuffer stringBuffer = new StringBuffer();
            while(data.moveToNext()){
                arrayList.add("\n" + "BloodBag Number: " + data.getString(0) + "\n" + "Blood Type:
" + data.getString(1) + "\n" + "Description: " + data.getString(2) + "\n" + "Quantity: " +
data.getString(3) + "\n");
            }
            ListAdapter listAdapter2 = new
ArrayAdapter<>(this,android.R.layout.simple_list_item_1, arrayList);
```

```
            listView.setAdapter(listAdapter2);
        }

    }

    @Override
    public void finish() {
        super.finish();
        CustomIntent.customType(InventoryResults.this, "left-to-right");
    }
}
```

Main Activity

```
package com.dbms.blooddonation;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;

import maes.tech.intentanim.CustomIntent;

public class MainActivity extends AppCompatActivity {
    private static int SPLASH_TIME_OUT = 3000;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
```

```
        new Handler().postDelayed(new Runnable() {
            @Override
            public void run() {
                Intent bloodIntent = new Intent(MainActivity.this, BloodActivity.class);
                startActivity(bloodIntent);
                CustomIntent.customType(MainActivity.this, "right-to-left");
                finish();
            }
        },SPLASH_TIME_OUT);
    }
}
```

Need Result Activity

```
package com.dbms.blooddonation;

import androidx.appcompat.app.AppCompatActivity;

import android.database.Cursor;
import android.os.Bundle;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ListAdapter;
import android.widget.ListView;
import android.widget.Toast;

import java.util.ArrayList;

import maes.tech.intentanim.CustomIntent;
```

```
public class NeedResults extends AppCompatActivity {

    DatabaseHelper databaseHelper;
    EditText del_id_text;
    Button delete;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_need_results);

        ListView listView = findViewById(R.id.listviewneed);
        databaseHelper = new DatabaseHelper(this);

        del_id_text = findViewById(R.id.need_id);
        ArrayList<String> arrayList = new ArrayList<>();
        Cursor data = databaseHelper.getAllData();

        if (data.getCount() == 0){
            Toast.makeText(NeedResults.this, "Database Empty", Toast.LENGTH_LONG).show();
        } else{
            StringBuffer stringBuffer = new StringBuffer();
            while(data.moveToNext()){
                arrayList.add("\n" + "ID: " + data.getString(0) + "\n" + "Name: " + data.getString(1) +
"\n" + "Email: " + data.getString(2) + "\n" + "Phone: " + data.getString(3) + "\n" + "Blood
Group: " + data.getString(4) + "\n");
            }
            ListAdapter listAdapter = new
ArrayAdapter<>(this,android.R.layout.simple_list_item_1, arrayList);
```

```
        listView.setAdapter(listAdapter);
    }

  }

  @Override
  public void finish() {
    super.finish();
    CustomIntent.customType(NeedResults.this, "left-to-right");
  }
}
```

Recipient Activity

```
package com.dbms.blooddonation;
import androidx.appcompat.app.AppCompatActivity;
import android.app.AlertDialog;
import android.content.Intent;
import android.database.Cursor;
import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;
```

```
import maes.tech.intentanim.CustomIntent;

public class RecipientActivity extends AppCompatActivity implements
AdapterView.OnItemSelectedListener {

    DatabaseHelper databaseHelper;
    EditText name, phone, id;
    TextView group;
    Button insert, update, delete, view,er;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_recipient);

        databaseHelper = new DatabaseHelper(this);

        name = findViewById(R.id.rec_name);
        group = findViewById(R.id.group);
        phone = findViewById(R.id.rec_phno);
        insert = findViewById(R.id.blood_insert);
        view = findViewById(R.id.blood_view);

        final Spinner blood_group_spinner = findViewById(R.id.spinner_bg);
        ArrayAdapter<CharSequence> adapter = ArrayAdapter.createFromResource(this,
R.array.blood_groups,R.layout.spinner_item);

adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
        blood_group_spinner.setAdapter(adapter);
```

```
    blood_group_spinner.setOnItemSelectedListener(this);

    insert.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            String recName = name.getText().toString();
            String recGroup = group.getText().toString();
            String recPhone = phone.getText().toString();



            if(!TextUtils.isEmpty(recGroup) && !TextUtils.isEmpty(recName) &&
!TextUtils.isEmpty(recPhone)){
                boolean isInserted = databaseHelper.insertData3(recName, recGroup,
recPhone);
                if (isInserted)
                    Toast.makeText(RecipientActivity.this, "Data Inserted Successfully",
Toast.LENGTH_LONG).show();
                else
                    Toast.makeText(RecipientActivity.this, "Insertion Error",
Toast.LENGTH_LONG).show();
                name.setText("");
                group.setText("");
                phone.setText("");
            }else{
                Toast.makeText(RecipientActivity.this, "Required Fields Cannot Be Blank",
Toast.LENGTH_LONG).show();
            }
        }
    });

    view.setOnClickListener(new View.OnClickListener() {
```

```
        @Override
        public void onClick(View view) {
            Cursor result = databaseHelper.getAllData3();
            if(result.getCount() == 0){
                showMessage("Error", "No Data");
                return;
            }else{
                Intent bloodIntent = new Intent(RecipientActivity.this,
RecipientResultsActivity.class);
                startActivity(bloodIntent);
                CustomIntent.customType(RecipientActivity.this, "right-to-left");
            }

        }
    });
}

@Override
public void onItemSelected(AdapterView<?> adapterView, View view, int i, long l) {
}

@Override
public void onNothingSelected(AdapterView<?> adapterView) {
}

public void showMessage(String title, String message){
    AlertDialog.Builder builder = new AlertDialog.Builder(this);
    builder.setCancelable(true);
    builder.setTitle(title);
    builder.setMessage(message);
```

```
        builder.show();
    }

}
```

Recipient Result Activity

```
package com.dbms.blooddonation;

import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;

import android.content.DialogInterface;
import android.database.Cursor;
import android.os.Bundle;
import android.text.Html;
import android.text.Spanned;
import android.util.Log;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.BaseAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ListAdapter;
import android.widget.ListView;
import android.widget.Toast;

import java.util.ArrayList;
```

```
import java.util.Arrays;

import maes.tech.intentanim.CustomIntent;

public class RecipientResultsActivity extends AppCompatActivity  {

    DatabaseHelper databaseHelper;
    EditText del_id_text;
    Button delete;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_recipient_results);

        final ListView listView = findViewById(R.id.listviewneed);
        databaseHelper = new DatabaseHelper(this);

        final ArrayList<Spanned> arrayList = new ArrayList<>();
        final Cursor dataOuter = databaseHelper.getAllData3();

        if (dataOuter.getCount() == 0){
            Toast.makeText(RecipientResultsActivity.this, "Database Empty",
Toast.LENGTH_LONG).show();
        } else{
            StringBuffer stringBuffer = new StringBuffer();
            while(dataOuter.moveToNext()){
                arrayList.add(parse(dataOuter));
            }
```

```
        final ArrayAdapter listAdapter2 = new
ArrayAdapter<>(this,android.R.layout.simple_list_item_1, arrayList);
        listView.setAdapter(listAdapter2);
        listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
          @Override
          public void onItemClick(AdapterView<?> parent, View view, final int position, long
id) {
            dataOuter.moveToPosition(position);
            final Cursor data = databaseHelper.getPossibleDonors(dataOuter.getString(1));
            AlertDialog.Builder builder = new
AlertDialog.Builder(RecipientResultsActivity.this);
            builder.setTitle("Choose a donor");
            ArrayList<String> donors = new ArrayList<>();


            while(data.moveToNext()){
              donors.add("\n" + "ID: " + data.getString(0) + "\n" + "Name: " +
data.getString(1) + "\n" + "Email: " + data.getString(2) + "\n" + "Phone: " + data.getString(3) +
"\n" + "Blood Group: " + data.getString(4) + "\n");
            }
            String[] donorsArray = donors.toArray(new String[0]);
            Log.d("donors", "onItemClick: " + Arrays.toString(donorsArray));
            builder.setItems(donorsArray, new DialogInterface.OnClickListener() {
              @Override
              public void onClick(DialogInterface dialog, int which) {
                data.moveToPosition(which);
                Log.d("gg", "onClick: " + dataOuter.moveToPosition(position));
                Log.d("gg", "onClick: " + position);
                databaseHelper.addDonorToRec(data.getInt(0), dataOuter.getInt(3));
                arrayList.set(position, parse(dataOuter));
                listAdapter2.notifyDataSetChanged();
              }
```

```
                });

                AlertDialog dialog = builder.create();
                dialog.show();
            }
        });
    }

}
private Spanned parse(Cursor dataOuter){
    String donorId = String.valueOf(dataOuter.getInt(4));
    if(dataOuter.getInt(4) == 0){
        donorId = "Not Assigned";
    }
    String str = "<br>" + "ID:" + dataOuter.getInt(3) + "<br>" + "Name: " +
dataOuter.getString(0) + "<br>" + "Blood Type: " + dataOuter.getString(1) + "<br>" +
"PhoneNo: " + dataOuter.getString(2) + "<br>" + "DonorID: " + donorId + "<br>";
    if(dataOuter.getInt(4) != 0){
        str = "<b>" + str + "</b>";
    }
    return Html.fromHtml(str);
}

@Override
public void finish() {
    super.finish();
    CustomIntent.customType(RecipientResultsActivity.this, "left-to-right");
}
}
```

Splash Activity

```
package com.dbms.blooddonation;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;

import java.util.Timer;
import java.util.TimerTask;

public class SplashActivity extends AppCompatActivity {
    private static final long DELAY = 2000;
    private boolean scheduled = false;
    private Timer splashTimer;



    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_splash);
        splashTimer = new Timer();
        splashTimer.schedule(new TimerTask()
        {
            @Override
            public void run()
            {
                SplashActivity.this.finish();
                startActivity(new Intent(SplashActivity.this, Frontactivity.class));
```

```
            }
        }, DELAY);
        scheduled = true;
    }

    @Override
    protected void onDestroy()
    {
        super.onDestroy();
        if (scheduled)
            splashTimer.cancel();
        splashTimer.purge();
    }
}
```

Staff Activity

```
package com.dbms.blooddonation;

import androidx.appcompat.app.AppCompatActivity;

import android.app.AlertDialog;
import android.content.Intent;
import android.database.Cursor;
import android.os.Bundle;
import android.text.TextUtils;
```

```
import android.view.View;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import maes.tech.intentanim.CustomIntent;

public class StaffActivity extends AppCompatActivity implements
AdapterView.OnItemSelectedListener {

    DatabaseHelper databaseHelper;
    EditText empname, empemail, empphone, empid,empsalary;
    Button empinsert, update, delete, empview;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_staff);

        databaseHelper = new DatabaseHelper(this);

        empid = findViewById(R.id.emp_id);
        empname = findViewById(R.id.emp_name);
        empemail = findViewById(R.id.emp_email);
        empphone = findViewById(R.id.emp_phone);
        empsalary = findViewById(R.id.emp_salary);
        empinsert = findViewById(R.id.emp_insert);
        empview = findViewById(R.id.emp_view);
```

```
empinsert.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Integer e_id= Integer.valueOf(empid.getText().toString());
        String e_name = empname.getText().toString();
        String e_email = empemail.getText().toString();
        String e_phone = empphone.getText().toString();
        String e_salary=empsalary.getText().toString();

        if( !TextUtils.isEmpty(e_salary) && !TextUtils.isEmpty(e_email) &&
!TextUtils.isEmpty(e_name) && !TextUtils.isEmpty(e_phone)){
            boolean isInserted =
databaseHelper.insertData1(e_id,e_name,e_email,e_phone,e_salary);
            if (isInserted)
                Toast.makeText(StaffActivity.this, "Data Inserted Successfully",
Toast.LENGTH_LONG).show();
            else
                Toast.makeText(StaffActivity.this, "Insertion Error",
Toast.LENGTH_LONG).show();
            empid.setText("");
            empname.setText("");
            empemail.setText("");
            empphone.setText("");
            empsalary.setText("");

        }else{
            Toast.makeText(StaffActivity.this, "Required Fields Cannot Be Blank",
Toast.LENGTH_LONG).show();
        }
    }
});
```

```
    empview.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Cursor result = databaseHelper.getAllData1();
            if(result.getCount() == 0){
                showMessage("Error", "No Data");
                return;
            }else{
                Intent bloodIntent = new Intent(StaffActivity.this, StaffResult.class);
                startActivity(bloodIntent);
                CustomIntent.customType(StaffActivity.this, "right-to-left");
            }

        }
    });
}

@Override
public void onItemSelected(AdapterView<?> adapterView, View view, int i, long l) {
}

@Override
public void onNothingSelected(AdapterView<?> adapterView) {

}

public void showMessage(String title, String message){
    AlertDialog.Builder builder = new AlertDialog.Builder(this);
```

```
        builder.setCancelable(true);
        builder.setTitle(title);
        builder.setMessage(message);
        builder.show();
    }

}
```

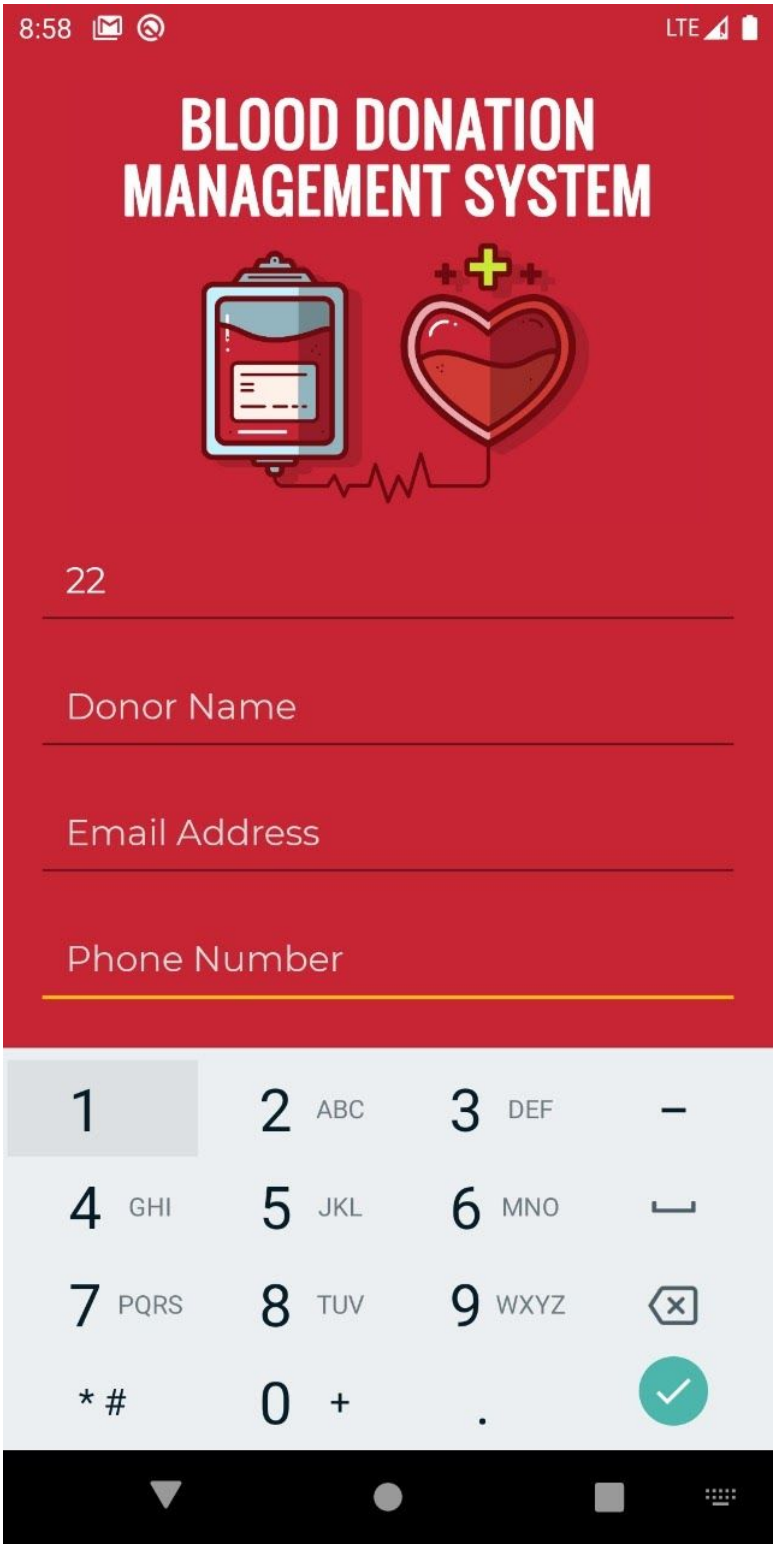
Staff Result Activity

```
package com.dbms.blooddonation;

import androidx.appcompat.app.AppCompatActivity;

import android.database.Cursor;
import android.os.Bundle;
import android.widget.ArrayAdapter;
import android.widget.ListAdapter;
import android.widget.ListView;
import android.widget.Toast;

import java.util.ArrayList;

import maes.tech.intentanim.CustomIntent;

public class StaffResult extends AppCompatActivity {

    DatabaseHelper databaseHelper;
```

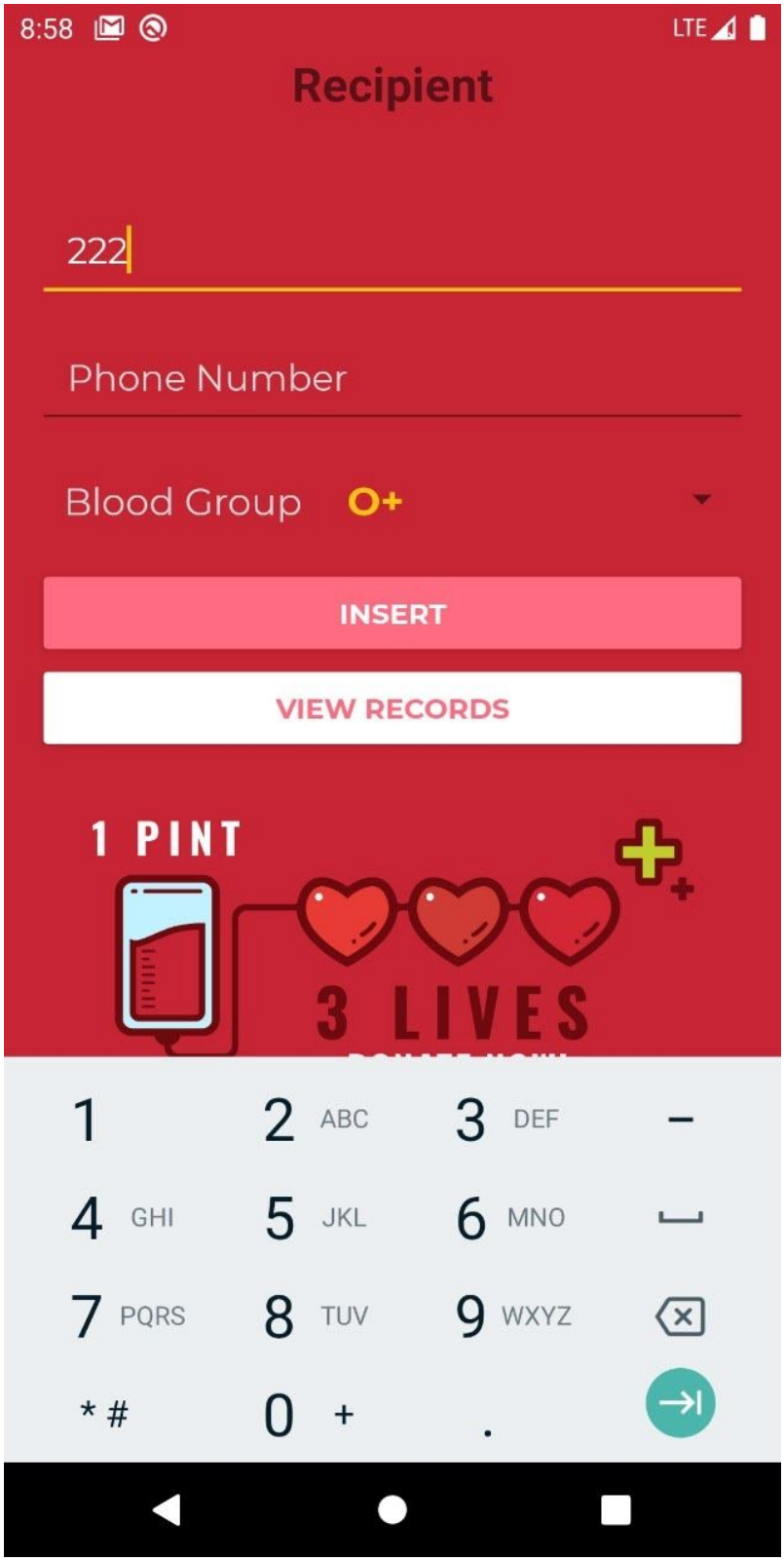
```
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_staff_result);

        ListView listView = findViewById(R.id.listview1);
        databaseHelper = new DatabaseHelper(this);

        ArrayList<String> arrayList1 = new ArrayList<>();
        Cursor data = databaseHelper.getAllData1();

        if (data.getCount() == 0){
            Toast.makeText(StaffResult.this, "Database Empty", Toast.LENGTH_LONG).show();
        } else{
            StringBuffer stringBuffer = new StringBuffer();
            while(data.moveToNext()){
                arrayList1.add("\n" + "ID: " + data.getString(0) + "\n" + "Name: " + data.getString(1)
+ "\n" + "Email: " + data.getString(2) + "\n" + "Phone: " + data.getString(3) + "\n" + "Salary: " +
data.getString(4) + "\n");
            }
            ListAdapter listAdapter1 = new
ArrayAdapter<>(this,android.R.layout.simple_list_item_1, arrayList1);
            listView.setAdapter(listAdapter1);
        }



    }

    @Override
```

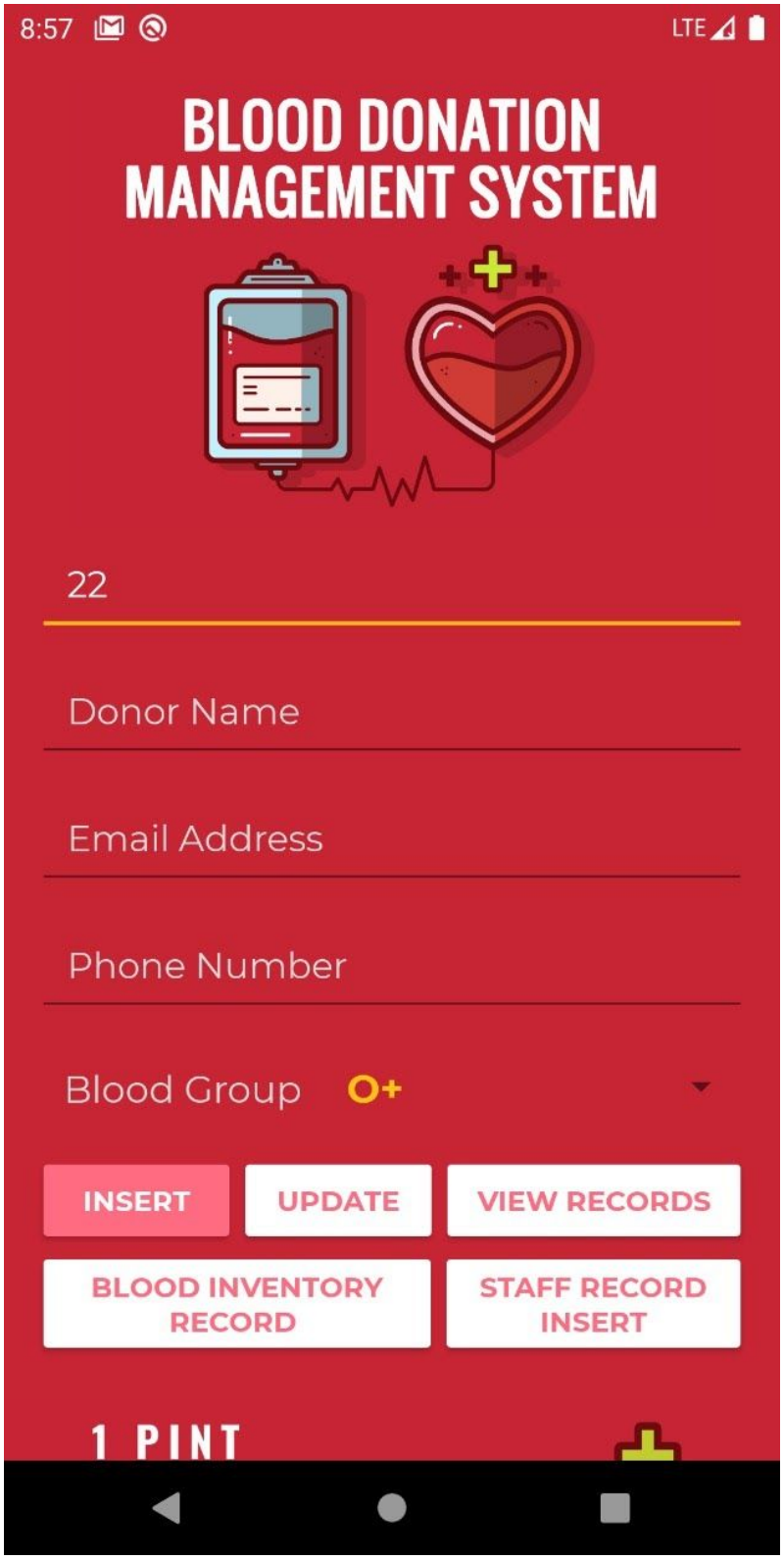
```
    public void finish() {
        super.finish();
        CustomIntent.customType(StaffResult.this, "left-to-right");
    }
}
```

SCREENSHOTS OF MANIPULATION AT FRONT END




8:59
LTE
Blood Records
ID
DELETE
ID: 1
Name: Amit
Email: amit.ananthkumar@gmail.com
Phone: 8921816808
Blood Group: O+


8:58
LTE
Recipient
222
Phone Number
Blood Group
O+
INSERT
VIEW RECORDS
1 PINT
3 LIVES


8:57
LTE
BLOOD DONATION MANAGEMENT SYSTEM
22
Donor Name
Email Address
Phone Number
Blood Group
O+
INSERT
UPDATE
VIEW RECORDS
BLOOD INVENTORY RECORD
STAFF RECORD INSERT
1 PINT

Project GitHub Link with Contributors

https://github.com/imaryandokania/Blood-Donation-Management.git

APK Unsigned Link

http://www.mediafire.com/file/3c5ri3w137ogh0c/Blood_Donation_App/file

REFERENCES

1. *Problems related to blood donation in India* https://www.atmph.org/article.asp?issn=1755-6783;year=2012;volume=5;issue=1;spage=50;epage=52;aulast=Aggarwal#:~:text=Many%20blood%20banks%20in%20India,in%20with%20whole%20blood%20transfusions.

2. https://beginnersbook.com/2015/05/normalization-in-dbms/

3. https://www.researchgate.net/publication/339032343_Blood_bank_and_Donor_Management_system