INTRODUCTION

* 1. [History 1](#_bookmark3)
  2. [Android Versions 1](#_bookmark4)
  3. Architecture of Android 2
  4. [Android Studio Installation 3](#_bookmark5)

1. INTRODUCTION TO THE PROJECT
   1. [Overview of the project 5](#_bookmark6)
   2. [Aim of the project 6](#_bookmark7)
2. SYSTEM DESIGN 7
3. IMPLEMENTATION 17
4. TESTING 23
5. RESULTS 24
6. CONCLUSION AND FUTURE ENHANCEMENT 27
7. REFERENCES 28

|  |  |  |
| --- | --- | --- |
| **Fig No.** | **Figure Name** | **Page No.** |
| Fig 3.1 | System Flowchart | 7 |
| Fig 3.2 | App Initial UI | 9 |
| Fig 3.3 | Dashboard Initial UI | 11 |
| Fig 3.4 | Admin Initial UI | 13 |
| Fig 3.5 | Signup Initial UI | 14 |
| Fig 3.6 | Login Initial UI | 16 |
| Fig 6.1 | App Screen | 24 |
| Fig 6.2 | Dashboard | 24 |
| Fig 6.3 | Valid Donor Insertion | 24 |
| Fig 6.4 | Valid Updation | 25 |
| Fig 6.5 | Admin Views User Entries | 25 |
| Fig 6.6 | User Deletion | 25 |
| Fig 6.7 | On Successful Signup | 25 |
| Fig 6.8 | On Successful Signin | 26 |
| Fig 6.9 | View User Entries | 26 |
| Fig 6.10 | View Updated User Entries | 26 |

**Chapter 1**

# INTRODUCTION TO ANDROID

### History

Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It allows developers to write managed code in java language, controlling the device via google developed java libraries.Android is a linux based operatin system.It is available for download on Windows, macOS and Linux based operating systems or as a subscription-based service in 2020.It is a replacement for the Eclipse Android Development Tools (E-ADT) as the primary IDE for native Android application development. Android Studio was announced on May 16, 2013 at the Google I/O conference. It was in early access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from version 0.8 which was released in June 2014.The first stable build was released in December 2014, starting from version 1.0.On May 7, 2019, Kotlin replaced Java as Google's preferred language for Android app development. Java is still supported, as is C++.

### Android Versions

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Release date** | **Version** | **Release date** |
| Arctic Fox (2020.3.1) | July 2021 | 3.0 | October 2017 |
| 4.2 | May 2021 | 2.3 | March 2017 |
| 4.0 | May 2020 | 2.1 | April 2016 |
| 3.6 | February 2020 | 2.0 | April 2016 |
| 3.5 | August 2019 | 1.5 | November 2015 |
| 3.3 | January 2019 | 1.3 | July 2015 |
| 3.2 | September 2018 | 1.2 | April 2015 |
| 3.1 | March 2018 | 1.1 | February 2015 |
| 3.0 | October2017 | 1.0 | December 2014 |

* 1. **Android Architecture**

Android architecture contains different number of components to support any android device needs. Android software contains an open-source Linux Kernel having collection of number of C/C++ libraries which are exposed through an application framework services. Among all the components Linux Kernel provides main functionality of operating system functions to smartphones and Dalvik Virtual Machine (DVM) provide platform for running an android application.

The main components of android architecture are following: -

* + - Applications
    - Application Framework
    - Android Runtime
    - Platform Libraries
    - Linux Kernel

**Applications** – Applications is the top layer of android architecture. The pre-installed applications like home, contacts, camera, gallery etc and third-party applications downloaded from the play store like chat applications, games etc. will be installed on this layer only.

**Application framework** – Application Framework provides several important classes which are used to create an Android application. It provides a generic abstraction for hardware access and also helps in managing the user interface with application resources. Generally, it provides the services with the help of which we can create a particular class and make that class helpful for the Applications creation. It includes different types of services activity manager, notification manager, view system, package manager etc.

**Application runtime** – Android Runtime environment is one of the most important part of Android. It contains components like core libraries and the Dalvik virtual machine(DVM). Mainly, it provides the base for the application framework and powers our application with the help of the core libraries. The core libraries enable us to implement android applications using the standard JAVA or Kotlin programming languages.

**Platform libraries** – The Platform Libraries includes various C/C++ core libraries and Java based libraries such as Media, Graphics, Surface Manager, OpenGL etc. to provide a support forandroid development.

**Linux Kernel** – Linux Kernel is heart of the android architecture. It manages all the available drivers such as display drivers, camera drivers, Bluetooth drivers, audio drivers, memory drivers,etc. which are required during the runtime. The Linux Kernel will provide an abstraction layer between the device hardware and the other components of android architecture. It is responsiblefor management of memory, power, devices etc.

**1.3 Android Studio Installation**

Before downloading and installing Android Studio, the following requirements are essential.Operating System Version - Microsoft Windows 7/8/10 (32-bit or 64-bit).

Random Access Memory (RAM) - Minimum 4 GB RAM and 8 GB RAM recommended. Free Disk Space - Minimum 2 GB and 4 GB recommended.

Minimum Required JDK Version - Java Development Kit (JDK) 8.Minimum Screen Resolution - 1280 \* 800.resolution

**Step 1**: To download the Android Studio, visit the official Android Studio website in your web browser.

**Step 2**: Click on the "Download Android Studio" option.

**Step 3**: Double click on the downloaded "Android Studio-ide.exe" file.

**Step 4**: "Android Studio Setup" will appear on the screen and click "Next" to proceed.

**Step 5**: Select the components that you want to install and click on the "Next" button.

**Step 6**: Now, browse the location where you want to install the Android Studio and click "Next" to proceed.

**Step 7**: Choose a start menu folder for the "Android Studio" shortcut and click the "Install"button to proceed.

**Step 8**: After the successful completion of the installation, click on the "Next" button. **Step 9**: Click on the "Finish" button to proceed. Android studio welcome screen will appear onthe screen.

**Step 10**: "Android Studio Setup Wizard" will appear on the screen with the welcome wizard. Click on the "Next" button.

**Step 11**: Select (check) the “Standard” option if you are a beginner and do not have any idea about Android Studio.It will install the most common settings and options for you. Click “Next” to proceed

**Step 12**: Now, select the user interface theme as you want. Then, click on the "Next" button.

**Step 13**: Now, click on the "Finish" button to download all the SDK components.

**Step 14**: After downloading all the necessary components, click on the "Finish" button. Android Studio will be successfully installed in the system.

**Chapter 2**

# INTRODUCTION TO PROJECT

### Overview of the project

It is known that India has performed the second largest transplants in the year 2019 (first being United States), but India still lags a lot. Only 0.01% of people donate their organs after death and the number of live donors is more in India. Cadaver donations are only about five percent of the total donations that take place. Some of the main causes are lack of awareness, religious and superstitious beliefs and strict laws. Some people are interested in this social cause but don't have enough knowledge as to what to do and how to donate. Another factor is, people are not literate enough to know the importance of this cause. So, we are providing our nation with an Organ Donation Android Application to spread awareness and give an easy facility to our people to save life.

Organ donation is the donation of biological tissue or an organ of the human body, from a living or dead person to a living recipient in need of a transplantation. Transplantable organs and tissues are removed in a surgical procedure following a determination, based on the donor’s medical and social history, of which are suitable for transplantation. This android application will help user to get organ donor. This system has two entities namely, Admin and User. Admin can login using credentials. Admin can view all Donor and they can view and delete Donors. Admin can view list of donations and user list. User can register and login using credentials. User can change their password in case of security, View and update their profile. User can manage request by adding and updating request. User can also check the assigned donor. User can manage donors.

Over 5 lakh people in our country face death due to lack of organs, this is due to lack of awareness and connection between the donors and seekers. In India the rate of organ donation per million is comparatively very low than other countries around the globe. People are not ready to donate due to misconceptions and there are very less donors as compared to seekers in our country. There is no such market opportunity in this application, it’s an application which is meant for the lifesaving purpose and will be a revolutionary step for saving the lives of our people.

18CSMP68 Mobile Application Development Mini Project Organ Donation App

### Aim of the project

Need of Organ Donation Websites:

* + - To promote organ donation for transplantation as a treatment for many life threatening diseases including heart disease, kidney disease, liver disease, diabetes and cystic ﬁbrosis
    - To educate and inform the community, patients and their families and health professionals about organ and tissue donation to markedly improve rates of donation.
    - To work in partnership with Department of Health (DOH), clinicians, and hospitals to promote best practice professional training and ensure that the family of every potential donor is oﬀered the option of donation in a caring and respectful manner.
    - To provide support, care, information and advocacy for people and with end stage organ failure, donor families, living donors transplant recipients and their families.
    - To provide policy advice to DOH, clinicians and hospitals.
    - Assuring compliance with all external regulatory bodies, including but not limited to: the Organ Procurement and Transplantation Network (OPTN), the United Network for Organ Sharing (UNOS),Centers for Medicare and Medicaid Services (CMS) Conditions of Participation (COP), the Missouri State Department of Health (DOH), The Joint Commission (TJC) Standards
    - Ensuring the programs accreditation
    - Identifying opportunities for improvement
    - Prioritizing performance improvement and patient safety projects within organ transplantation
    - Continuously audit compliance and regulatory standards related to organ transplantation
    - Ensuring policies and procedures applicable to organ transplantation are evidence based, regularly reviewed and audited for compliance.

**Chapter 3**

# SYSTEM DESIGN

### System Flow Chart

**Admin**: This module fixates on the donors and Recipient.

**Hospital:** It manages the track of hospitals, stores data of hospital to database as hospital registers verifies the details of hospital etc.

**Donor**: Each member register as a Donor has a utilizer-id and password, which identifies them uniquely. Then member is given a authenticate form. They enter the authenticate details utilizer- id and password.

**Recipient**: Recipient are the one who requires organ for someone cognate with them or themselves.

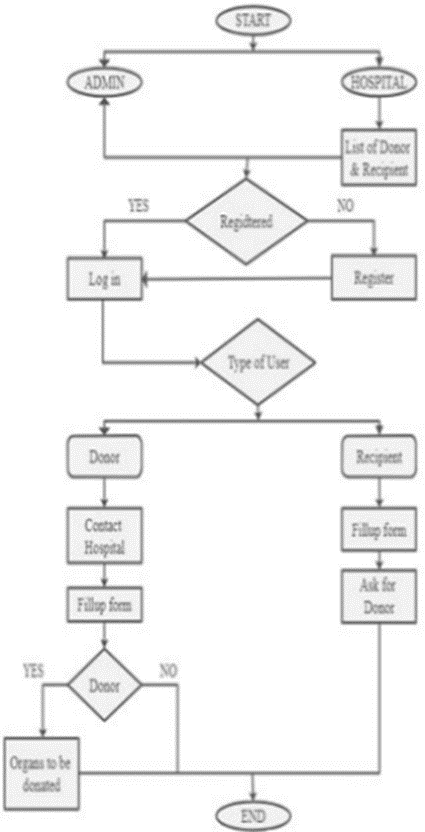


Figure 3.1: System Flowchart

### activity\_main.xml

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout xmlns:andr[oid="htt](http://schemas.android.com/apk/res/android)p:/[/sc](http://schemas.android.com/apk/res/android)he[mas.android.com/apk/res/android"](http://schemas.android.com/apk/res/android)

xmlns:app=["http://sc](http://schemas.android.com/apk/res-auto)h[emas.android.com/apk/res-auto"](http://schemas.android.com/apk/res-auto) xmlns:t[ools="ht](http://schemas.android.com/tools)tp:/[/sc](http://schemas.android.com/tools)he[mas.android.com/tools"](http://schemas.android.com/tools) android:layout\_width="match\_parent" android:layout\_height="match\_parent" tools:context=".MainActivity">

<Button

android:id="@+id/button" android:layout\_width="191dp" android:layout\_height="48dp" android:layout\_marginTop="12dp" android:background="#30D5C8" android:backgroundTint="#30D5C8" android:foregroundTint="#DA0707" android:text="ADMIN" android:textColor="#161717" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/button5" app:strokeColor="#A12D2D" />

<Button

android:id="@+id/button4" android:layout\_width="191dp" android:layout\_height="48dp" android:layout\_marginTop="20dp" android:background="#30D5C8" android:backgroundTint="#30D5C8" android:foregroundTint="#DA0707" android:text="HOSPITAL" android:textColor="#1B1C1C" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toBottomOf="@+id/button" app:strokeColor="#A12D2D" />

<Button

android:id="@+id/button5" android:layout\_width="191dp" android:layout\_height="49dp" android:background="#EEF3F2" android:backgroundTint="#EEF4F4" android:foregroundTint="#DA0707" android:text="DONAR" android:textColor="#101010"

app:layout\_constraintBottom\_toBottomOf="parent" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toTopOf="parent" app:layout\_constraintVertical\_bias="0.718" app:strokeColor="#A12D2D" />

<ImageView android:id="@+id/imageView" android:layout\_width="300dp" android:layout\_height="300dp"

app:layout\_constraintBottom\_toBottomOf="parent" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintHorizontal\_bias="0.495" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toTopOf="parent" app:layout\_constraintVertical\_bias="0.366" app:srcCompat="@drawable/img" />

<TextView android:id="@+id/textView2" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:background="#FA6B9B"

android:text="ORGAN DONATING APP" android:textColor="#673AB7" android:textSize="34sp" app:layout\_constraintBottom\_toBottomOf="parent" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintHorizontal\_bias="0.497" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toTopOf="parent" app:layout\_constraintVertical\_bias="0.109" />

</androidx.constraintlayout.widget.ConstraintLayout>

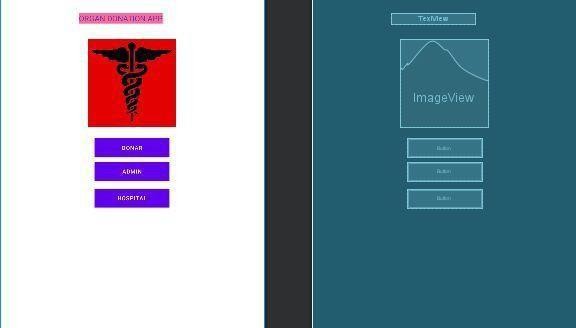


Fig 3.2 App Initial UI

### activity\_main2.xml

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android=["http://schemas.android.com/](http://schemas.android.com/apk/res/android)a[pk/res/android"](http://schemas.android.com/apk/res/android)

xmlns:app=["http://sc](http://schemas.android.com/apk/res-auto)h[emas.android.com/apk/res-auto"](http://schemas.android.com/apk/res-auto) xmlns:t[ools="ht](http://schemas.android.com/tools)tp:/[/sc](http://schemas.android.com/tools)he[mas.android.com/tools"](http://schemas.android.com/tools) android:layout\_width="match\_parent" android:layout\_height="match\_parent" android:background="#FA6B9B" tools:context=".MainActivity2">

<Button

android:id="@+id/b1" android:layout\_width="166dp" android:layout\_height="50dp" android:layout\_marginTop="60dp" android:text="Kidney" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintHorizontal\_bias="0.454" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toTopOf="parent" />

<Button

android:id="@+id/b2" android:layout\_width="166dp" android:layout\_height="50dp" android:layout\_marginTop="36dp" android:text="Eyes" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintHorizontal\_bias="0.453" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toBottomOf="@+id/b1" />

<Button

android:id="@+id/b3" android:layout\_width="166dp" android:layout\_height="50dp" android:layout\_marginTop="36dp" android:text="Liver" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintHorizontal\_bias="0.453" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toBottomOf="@+id/b2" />

<Button

android:id="@+id/b4" android:layout\_width="166dp" android:layout\_height="50dp" android:layout\_marginTop="36dp" android:text="Heart"

app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintHorizontal\_bias="0.453" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toBottomOf="@+id/b3" />

<Button

android:id="@+id/b5" android:layout\_width="166dp" android:layout\_height="50dp" android:layout\_marginTop="36dp" android:text="Lung" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintHorizontal\_bias="0.453" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toBottomOf="@+id/b4" />

<Button

android:id="@+id/b6" android:layout\_width="166dp" android:layout\_height="50dp" android:layout\_marginTop="36dp" android:text="Bone marrow" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintHorizontal\_bias="0.453" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toBottomOf="@+id/b5" />

</androidx.constraintlayout.widget.ConstraintLayout>

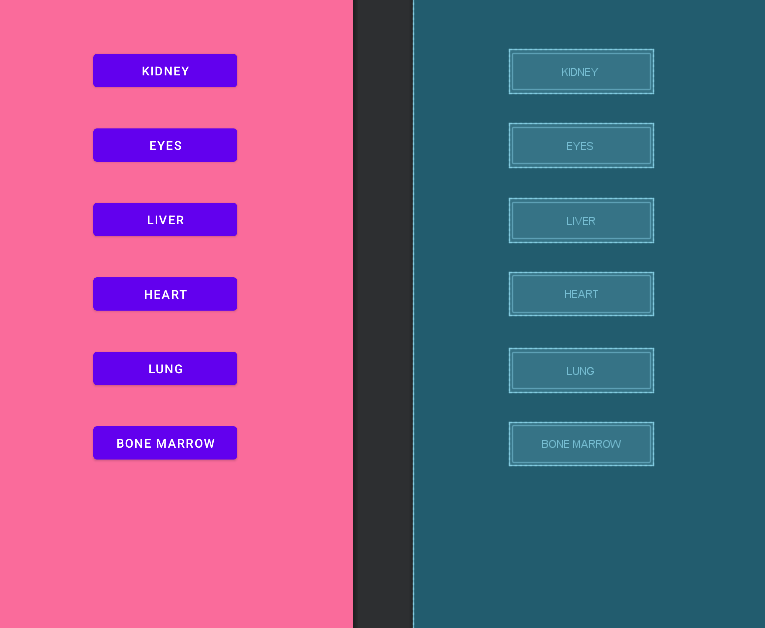


Fig 3.3 Dashboard Initial UI

### activity\_main5.xml

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android=["http://schemas.android.com/](http://schemas.android.com/apk/res/android)a[pk/res/android"](http://schemas.android.com/apk/res/android) xmlns:app=["ht](http://schemas.android.com/apk/res-auto)t[p://schemas.android.com/apk/res-auto"](http://schemas.android.com/apk/res-auto) xmlns:t[ools="ht](http://schemas.android.com/tools)tp:/[/sc](http://schemas.android.com/tools)he[mas.android.com/tools"](http://schemas.android.com/tools) android:layout\_width="match\_parent" android:layout\_height="match\_parent"

android:background="#FA6B9B" android:padding="10dp" tools:context=".MainActivity5">

<TextView android:id="@+id/texttitle"

android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:layout\_marginTop="20dp" android:text="Please enter the details below" android:textSize="24dp" />

<EditText android:id="@+id/name"

android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:layout\_below="@+id/texttitle" android:layout\_marginTop="3dp" android:hint="Name" android:inputType="textPersonName" android:textSize="24dp" />

<Button

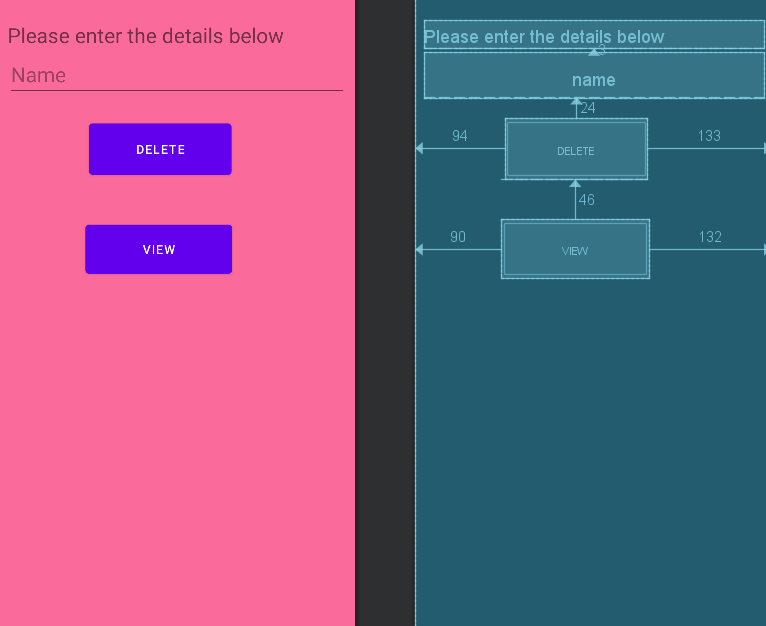
android:id="@+id/button2" android:layout\_width="174dp" android:layout\_height="71dp" android:layout\_below="@+id/name" android:layout\_alignParentStart="true" android:layout\_alignParentEnd="true" android:layout\_marginStart="94dp" android:layout\_marginTop="24dp" android:layout\_marginEnd="133dp" android:text="DELETE" />

<Button

android:id="@+id/button67" android:layout\_width="180dp" android:layout\_height="68dp" android:layout\_below="@+id/button2" android:layout\_alignParentStart="true" android:layout\_alignParentEnd="true" android:layout\_marginStart="90dp" android:layout\_marginTop="46dp"

android:layout\_marginEnd="132dp" android:text="View" />

</RelativeLayout>



### activity\_signup.xml

Fig 3.4 Admin Initial UI

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android=["http://schemas.android.com/](http://schemas.android.com/apk/res/android)a[pk/res/android"](http://schemas.android.com/apk/res/android) xmlns:app=["http://sc](http://schemas.android.com/apk/res-auto)h[emas.android.com/apk/res-auto"](http://schemas.android.com/apk/res-auto) xmlns:t[ools="ht](http://schemas.android.com/tools)tp:/[/sc](http://schemas.android.com/tools)he[mas.android.com/tools"](http://schemas.android.com/tools) android:layout\_width="match\_parent" android:layout\_height="match\_parent"

android:padding="10dp" tools:context=".Signup">

<EditText android:id="@+id/username"

android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:hint="User Name" android:layout\_marginTop="50dp"/>

<EditText android:id="@+id/password"

android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:hint="Password" android:layout\_marginTop="50dp" android:layout\_below="@+id/username"/>

<EditText android:id="@+id/password" android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:hint="Password" android:layout\_marginTop="50dp"

android:layout\_below="@+id/username"/>

<Button

android:id="@+id/btnsignup" android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:text="Register" android:layout\_marginTop="50dp" android:layout\_below="@+id/repassword"/>

<Button

android:id="@+id/btnsignin" android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:text="Existing user! Go to Sign in page" android:layout\_marginTop="50dp" android:layout\_below="@+id/btnsignup"/>

</RelativeLayout>

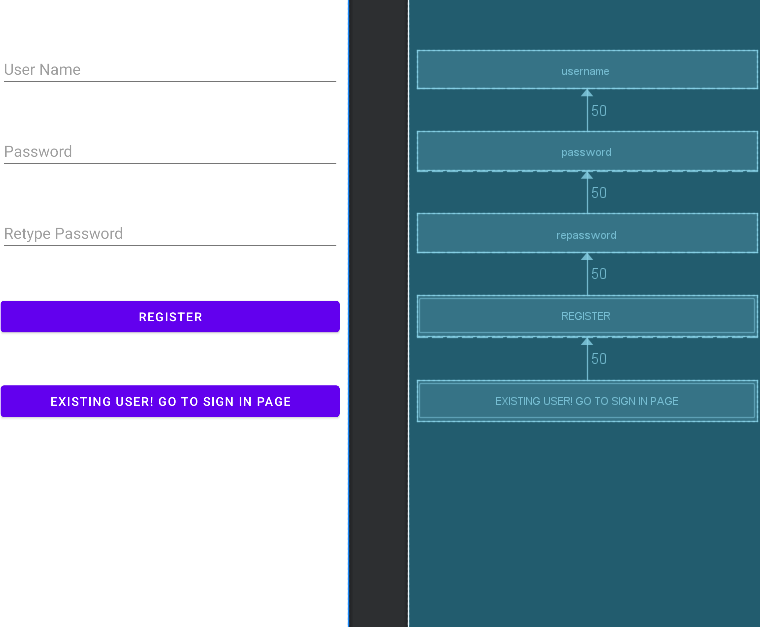


Fig 3.5 Signup Initial UI

### activity\_login.xml

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android=["http://schemas.android.com/](http://schemas.android.com/apk/res/android)a[pk/res/android"](http://schemas.android.com/apk/res/android) xmlns:app=["http://sc](http://schemas.android.com/apk/res-auto)h[emas.android.com/apk/res-auto"](http://schemas.android.com/apk/res-auto) xmlns:t[ools="ht](http://schemas.android.com/tools)tp:/[/sc](http://schemas.android.com/tools)he[mas.android.com/tools"](http://schemas.android.com/tools) android:layout\_width="match\_parent" android:layout\_height="match\_parent"

android:padding="10dp" tools:context=".LoginActivity">

<EditText android:id="@+id/username1" android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:hint="User Name" android:layout\_marginTop="50dp"/>

<EditText android:id="@+id/password1" android:layout\_width="match\_parent"

android:layout\_height="wrap\_content" android:hint="Password" android:layout\_marginTop="50dp" android:layout\_below="@+id/username1"/>

<Button

android:id="@+id/btnsignin1" android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:text="Sign in" android:layout\_marginTop="50dp" android:layout\_below="@+id/password1"/>

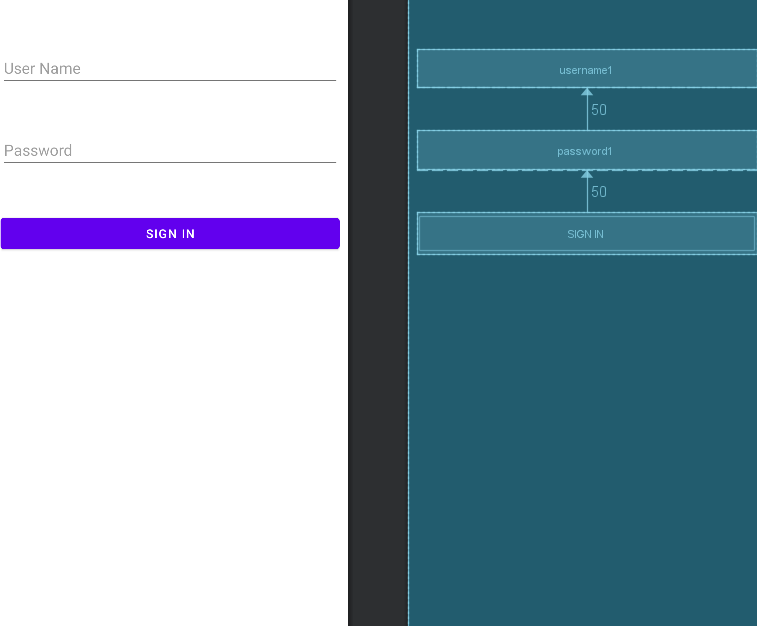
</RelativeLayout>

Figure 3.6 Login Initial UI

**Chapter 4**

# IMPLEMENTATION

The following java codes are used in the project for correct execution of the app.

### MainActivity.java

package com.example.madfinal;

import androidx.appcompat.app.AppCompatActivity;

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

public class MainActivity extends AppCompatActivity { Button button5;

Button button4; Button button;

@Override

protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity\_main); button5=findViewById(R.id.button5);

button4 =findViewById(R.id.button4); button =findViewById(R.id.button); button5.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

Intent intent=new Intent(MainActivity.this,MainActivity2.class); startActivity(intent);

}

});

button4.setOnClickListener(new View.OnClickListener() { @Override

public void onClick(View view) {

Intent intent=new Intent(MainActivity.this,Signup.class); startActivity(intent);

}

});

button.setOnClickListener(new View.OnClickListener() { @Override

public void onClick(View view) {

Intent intent=new Intent(MainActivity.this,MainActivity5.class); startActivity(intent);

}

});

}

}

### MainActivity2.java

package com.example.madfinal;

import androidx.appcompat.app.AppCompatActivity; import android.content.Intent;

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

public class MainActivity2 extends AppCompatActivity { Button b1;

Button b2; Button b3; Button b4; Button b5; Button b6;

@Override

protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity\_main2); b1=findViewById(R.id.b1);

b2 =findViewById(R.id.b2); b3=findViewById(R.id.b3); b4 =findViewById(R.id.b4); b5=findViewById(R.id.b5); b6 =findViewById(R.id.b6);

b1.setOnClickListener(new View.OnClickListener() { @Override

public void onClick(View view) {

Intent intent=new Intent(MainActivity2.this,MainActivity3.class); startActivity(intent);

}

});

b2.setOnClickListener(new View.OnClickListener() { @Override

public void onClick(View view) {

Intent intent=new Intent(MainActivity2.this,MainActivity3.class); startActivity(intent);

}

});

b3.setOnClickListener(new View.OnClickListener() { @Override

public void onClick(View view) {

Intent intent=new Intent(MainActivity2.this,MainActivity3.class); startActivity(intent);

}

});

b4.setOnClickListener(new View.OnClickListener() { @Override

public void onClick(View view) {

Intent intent=new Intent(MainActivity2.this,MainActivity3.class); startActivity(intent);

}

});

b5.setOnClickListener(new View.OnClickListener() { @Override

public void onClick(View view) {

Intent intent=new Intent(MainActivity2.this,MainActivity3.class); startActivity(intent);

}

});

b6.setOnClickListener(new View.OnClickListener() { @Override

public void onClick(View view) {

Intent intent=new Intent(MainActivity2.this,MainActivity3.class); startActivity(intent);

}

});

}

}

### MainActivity5.java

package com.example.madfinal;

import androidx.appcompat.app.AlertDialog;

import androidx.appcompat.app.AppCompatActivity;

import android.database.Cursor; import android.os.Bundle; import android.view.View; import android.widget.Button; import android.widget.TextView; import android.widget.Toast;

public class MainActivity5 extends AppCompatActivity { TextView name;

Button view,delete; DBHelper DB;

@Override

protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity\_main5);

name = findViewById(R.id.name); view = findViewById(R.id.button67); delete = findViewById(R.id.button2); DB = new DBHelper(this);

view.setOnClickListener(new View.OnClickListener() { @Override

public void onClick(View view) { Cursor res = DB.getdata1(); if(res.getCount()==0){

Toast.makeText(MainActivity5.this, "No Entry Exists", Toast.LENGTH\_SHORT).show();

return;

}

StringBuffer buffer = new StringBuffer(); while(res.moveToNext()){

buffer.append("Name :"+res.getString(0)+"\n"); buffer.append("Contact :"+res.getString(1)+"\n"); buffer.append("Date of Birth :"+res.getString(2)+"\n\n");

}

AlertDialog.Builder builder = new AlertDialog.Builder(MainActivity5.this); builder.setCancelable(true);

builder.setTitle("User Entries"); builder.setMessage(buffer.toString()); builder.show();

} });

delete.setOnClickListener(new View.OnClickListener() { @Override

public void onClick(View view) {

String nameTXT = name.getText().toString();

Boolean checkudeletedata = DB.deletedata(nameTXT); if(checkudeletedata==true)

Toast.makeText(MainActivity5.this, "Entry Deleted", Toast.LENGTH\_SHORT).show();

else

Toast.makeText(MainActivity5.this, "Entry Not Deleted", Toast.LENGTH\_SHORT).show();

} });

}

}

### Signup.java

package com.example.madfinal;

import androidx.appcompat.app.AppCompatActivity;

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

public class Signup extends AppCompatActivity {

EditText username, password, repassword; Button signup, signin;

DBHelper1 DB; @Override

protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity\_signup);

username = (EditText) findViewById(R.id.username); password = (EditText) findViewById(R.id.password); repassword = (EditText) findViewById(R.id.repassword); signup = (Button) findViewById(R.id.btnsignup); signin = (Button) findViewById(R.id.btnsignin);

DB = new DBHelper1(this);

signup.setOnClickListener(new View.OnClickListener() { @Override

public void onClick(View view) {

String user = username.getText().toString(); String pass = password.getText().toString(); String repass = repassword.getText().toString();

if(user.equals("")||pass.equals("")||repass.equals("")) Toast.makeText(Signup.this, "Please enter all the fields",

Toast.LENGTH\_SHORT).show(); else{

if(pass.equals(repass)){

Boolean checkuser = DB.checkusername(user); if(checkuser==false){

Boolean insert = DB.insertData(user, pass); if(insert==true){

Toast.makeText(Signup.this, "Registered successfully", Toast.LENGTH\_SHORT).show();

Intent intent = new Intent(getApplicationContext(),LoginActivity.class); startActivity(intent);

}else{

Toast.makeText(Signup.this, "Registration failed", Toast.LENGTH\_SHORT).show();

}

}

else{

Toast.makeText(Signup.this, "User already exists! please sign in",

Toast.LENGTH\_SHORT).show();

}

}else{

Toast.makeText(Signup.this, "Passwords not matching", Toast.LENGTH\_SHORT).show();

}

} }

});

signin.setOnClickListener(new View.OnClickListener() { @Override

public void onClick(View view) {

Intent intent = new Intent(getApplicationContext(), LoginActivity.class); startActivity(intent);

}

});

}

}

### LoginActivity.java

package com.example.madfinal;

import androidx.appcompat.app.AppCompatActivity; import android.content.Intent;

import android.os.Bundle; import android.view.View; import android.widget.Button; import android.widget.EditText; import android.widget.Toast;

public class LoginActivity extends AppCompatActivity {

EditText username, password; Button btnlogin;

DBHelper1 DB; @Override

protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity\_login);

username = (EditText) findViewById(R.id.username1); password = (EditText) findViewById(R.id.password1); btnlogin = (Button) findViewById(R.id.btnsignin1); DB = new DBHelper1(this);

btnlogin.setOnClickListener(new View.OnClickListener() { @Override

public void onClick(View view) {

String user = username.getText().toString(); String pass = password.getText().toString();

if(user.equals("")||pass.equals("")) Toast.makeText(LoginActivity.this, "Please enter all the fields",

Toast.LENGTH\_SHORT).show(); else{

Boolean checkuserpass = DB.checkusernamepassword(user, pass); if(checkuserpass==true){

Toast.makeText(LoginActivity.this, "Sign in successfull", Toast.LENGTH\_SHORT).show();

GlobalVariable.message=username.getText().toString();

Intent intent = new Intent(getApplicationContext(), MainActivity4.class); startActivity(intent);

}else{

Toast.makeText(LoginActivity.this, "Invalid Credentials", Toast.LENGTH\_SHORT).show();

}

}

}

});

}

}

**Chapter 5**

# TESTING

Software testing in an essential phase in the development life cycle of an application. Testing ensures that the developed system meets its functional and non-functional requirements. Two important terms in software testing are Verification and Validation. Verification is the process of evaluating work-products like requirement specs, design specs and test cases etc. of different development phases to make sure that they meet the requirements for that phase. It ensures that the system is built in the right way. Whereas Validation is the process of evaluating the software at the end of the development phase to make sure that it meets the business requirements. It is used to make sure that the product fulfils its intended use and that the end product is built right. One of the most important tools to test and debug an Android app is the Dalvik debug monitor server (DDMS) that is part of the Android framework. DDMS helps you to debug your code as it prints errors, warning and other information from your code. It also provide stack traces for exceptions on the Logcat output. Various other testing strategies have been adopted to make sure the correctness of the Image Steganography app. They are discussed in this chapter.

### Test Cases

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl.**  **No.** | **Test Case** | **Action** | **Result** |
| 1 | On tapping Donor | It will goes to Organ Selection Page | Pass |
| 2 | Once user select the Organ | It will goes to Donor details  Page | Pass |
| 3 | On pressing Admin button | It gives Admin login page | Pass |
| 4 | On pressing view button | Admin can view the Donpr  details | Pass |
| 5 | On pressing delete button | Admin can delete the Donor details | Pass |
| 6 | On clicking Hospital button | Hospital member can login or signup. | Pass |
| 7 | On pressing view button | Hospital member can view the donor details | Pass |

Table 5.2.1: test cases

**Chapter 6**

# RESULTS

The following screens appear in sequence to the user on clicking of respective buttons and texts.



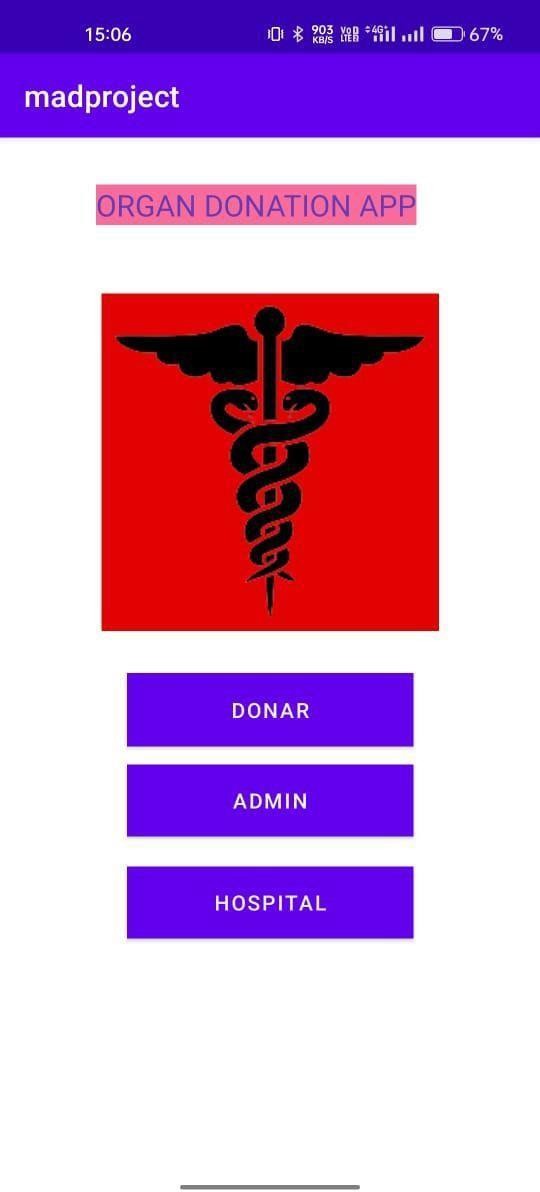


Fig 6.1 App Screen Fig 6.2 Dashboard Fig 6.3 Valid Donor Insertion

Fig 6.1 represents the front screen of the app; Fig 6.2 represents the dashboard with various options for the donor to choose from and Fig 6.3 shows the valid message when the new user record is inserted by clicking on Insert Button.

Fig 6.4 shows the valid information is updated by clicking on Update Button and Fig 6.5 represents the Admin viewing user details of all the existing users and Fig 6.6 shows the valid message when the admin enters an existing username clicks on Delete Button.

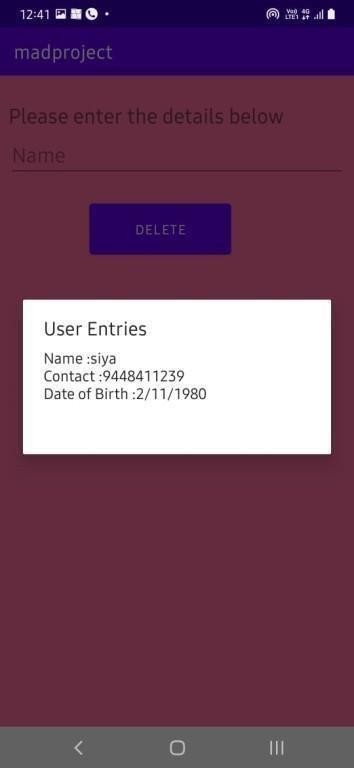
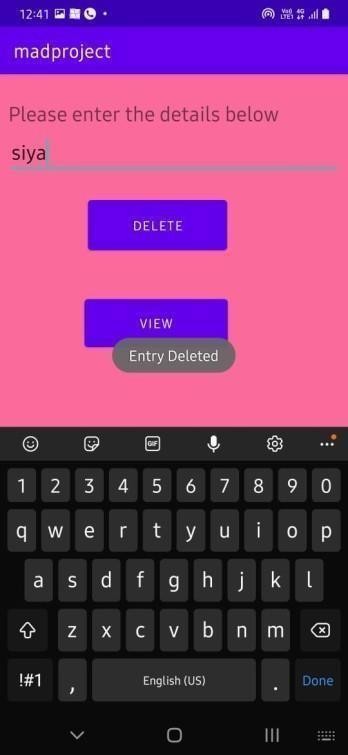
  

Fig 6.4 Valid Updation Fig 6.5 Admin Views User Entries Fig 6.6 User Deletion

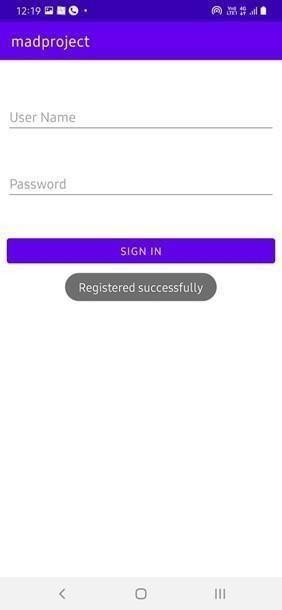
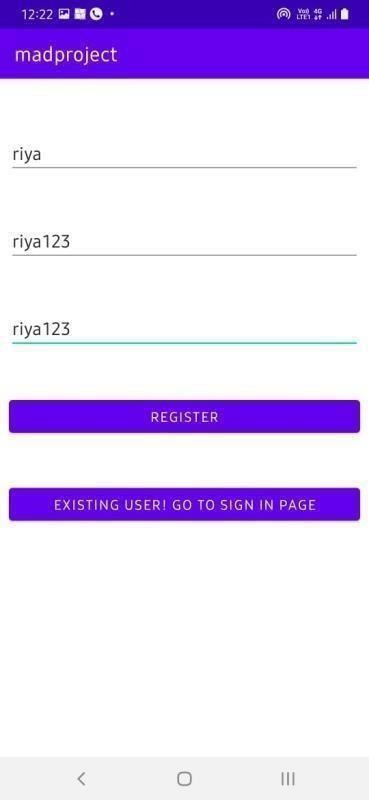


Fig 6.7 On Successful Signup

Fig 6.7 shows the valid information entered by the user and clicks on Register Button.

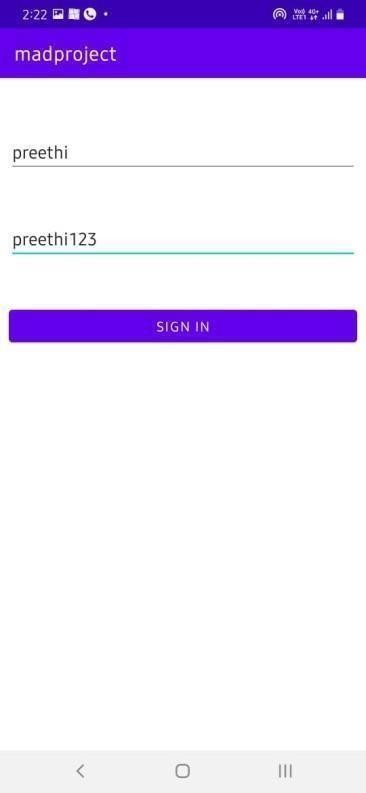
 

Fig 6.8 On Successful Signin

Fig 6.8 represents the screen with correct Signin information where the username and password entered by user while creating matches.

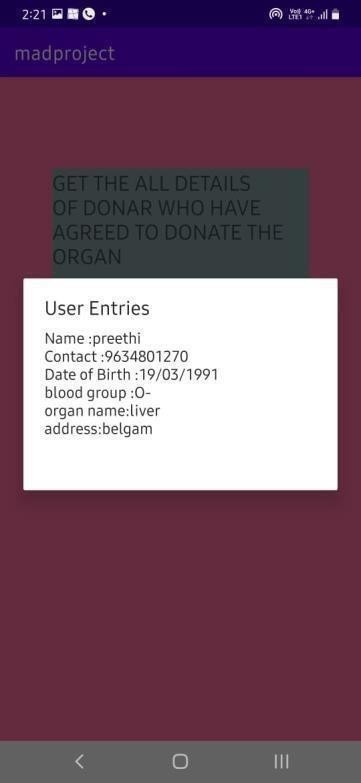
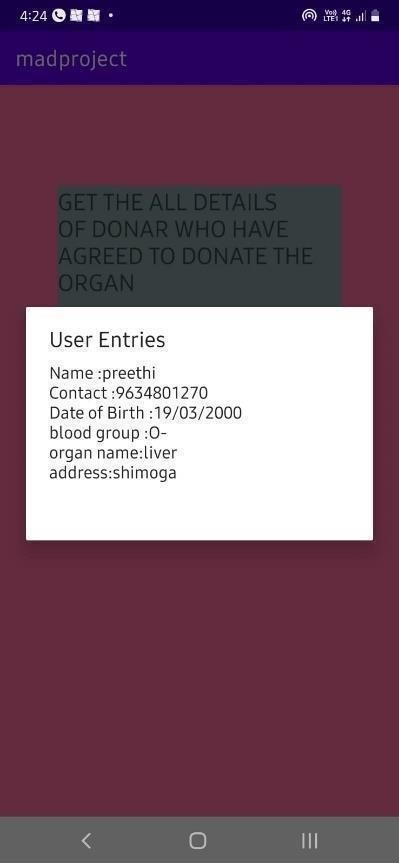
 

Fig 6.9 View User Entries Fig 6.10 View Updated User Entries

Fig 6.9 displays the user details like name, contact, DOB, blood group, organ name, address etc on successful Signin and Fig 6.10 displays the updated user details.

**Chapter 7**

# CONCLUSION AND FUTURE ENHANCEMENT

Organ donation is the donation of biological tissue or an organ of the human body, from a living or dead person to a living recipient in need of a transplantation. Transplantable organs and tissues are removed in a surgical procedure following a determination, based on the donor's medical and social history, of which are suitable for transplantation.This application will be the most user-friendly platform to enroll people in such good activity and will save many lives of the people who are dying due to lack of organ donation. It will create a long- term lifesaving role for every citizen in our country. Our prime target is to provide organs to the seeker when they are in need and make it a life saver platform for those who are dying due to the lack of organs and also targets the human race and spreads the concept of saving life after one's death.

### FUTURE ENHANCEMENT

* interactive chatbox to communicate efficiently with the donor.
* To help the recipient to find correct match of donor and doctors.
* the system can be fully automated with doctor’s diagnosis.
* the system can be made to read the medical reports and can find matches by specially taking time of requirement as a major consideration.
* the system can incorporate video calling for online conversation between donor and recepient.

**Chapter 8**

# REFERENCES

* + 1. <https://android-developers.googleblog.com/>
    2. <https://en.wikipedia.org/wiki/Android_version_history>
    3. <https://developer.android.com/>
    4. <https://developer.android.com/reference/android/widget/TextView>
    5. <https://developer.android.com/training/basics/firstapp>
    6. <https://firebase.google.com/>
    7. <https://www.sitepoint.com/starting-android-development-creating-todo-app/>
    8. <https://www.geeksforgeeks.org/>