

Thadomal Shahani Engineering College

Bandra (W.), Mumbai- 400 050.

© CERTIFICATE ©

Certify that Mr./Miss Aman Malvia
of Computer Department, Semester VI with
Roll No. 2103109 has completed a course of the necessary
experiments in the subject Mobile Computing under my
supervision in the **Thadomal Shahani Engineering College**
Laboratory in the year 2023 - 2024

~~Teacher In-Charge~~

Head of the Department

Date 18/03/2024

Principal

CONTENTS

SR. NO.	EXPERIMENTS	PAGE NO.	DATE	TEACHERS SIGN.
1.	WAA to draw basic graphical 2D.	3	15/1/24	
2.	WAA to draw basic graphical 3D.	10	24/1/24	
3.	WAA to design a form with GUI components.	17	29/1/24	
4.	WAA to design GUI components using database.	27	5/2/24	
5.	WAA to develop EMI calculation.	42	12/2/24	
6.	WAA that create alert on receiving messages.	49	26/2/24	
7.	WAA to implement basic calculator.	55	4/3/24	
8.	Write a program to demonstrate cellular frequency reuse.	67	11/3/24	
9.	Write a program to demonstrate concept of DSSS.	77	18/3/24	
10.	Write a program to implement A3/A5/A8 GSM <u>Security Algorithm</u> .	83	26/3/24	
11.	Assignment 1	91	5/2/24	
12.	Assignment 2	98	18/3/24	

Page No - 8/03/2024

Experiment :- 1

Aim:- Write an Android Application (WAA) to draw basic graphical 2D primitives.

Android studio is the official Integrated Development Environment (IDE) for android application development. Android studio provides more features that enhance our productivity while building android apps. Features of android studio are it has a flexible Gradle-based build system, it has a fast and feature rich emulator for app testing, android studio has a consolidated environment where we can develop for all android devices, apply changes to the resource code of our running app without restarting the app and android studio also provides extensive testing tools and frameworks.

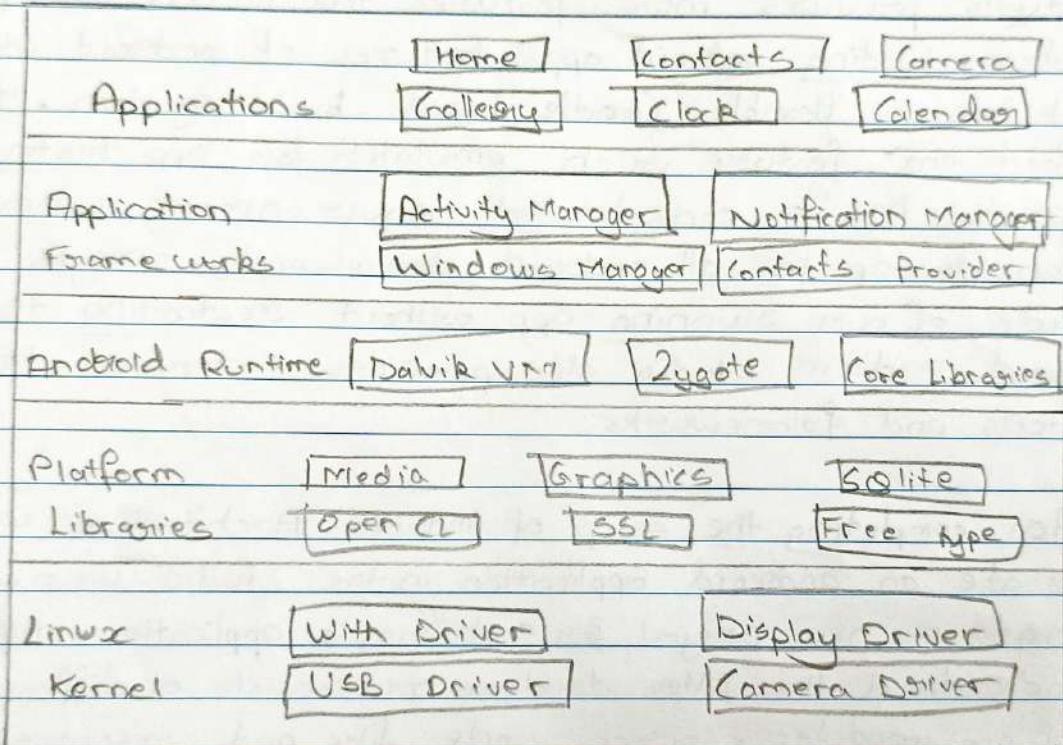
After completing the setup of Android Architecture we can create an android application in the studio. We need to create a new project for each sample application and we should understand the folder structure. It consists of different types of app modules, source code files and resource files. We will explore all folders and files in the android app.

- 1) Manifests folder 2) Java Folder 3) res (Resource) Folder
 - Drawable Folder
 - Layout Folder
 - Values Folder
 - Mipmap Folder
 - Gradient Scripts

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 provide platform for running an android application. the main component of android application architecture are following:-

- Applications • Application Frameworks • Android Runtime
- Platform Libraries • Linux Kernel



Steps to create a project in android studio:-

- 1) Create a project - File > New > New Project
- 2) Choose your project type - Select the type and click next.
- 3) Configure your project - Select the name of your project, specify package name, select language. Click Finish.

The code goes in the following way:-

- 1) package and imports.
- 2) Class declaration :- public class Main activity extends AppCompactActivity
- 3) Member Variables :- Bitmap bg;
Image View img;
- 4) On create Methods:- It is called when the activity is created.
- 5) Bitmap and ImageView Initialization : It creates a bitmap of size 720×1280 pixels and associates it with an 'Image View' from the layout.
- 6) Canvas creation:- It allows the drawing operations to work.
- 7) Drawing a circle
- 8) Drawing a rectangle
- 9) Drawing a triangle
- 10) Drawing the text
- 11) Image View Update:- It is updated with the modified bitmap containing the drawn shapes and text.

Conclusion:- Hence I have understood the creation of implemented 2D shapes in android studio.

AJW (A)

NAME: Arnav Malvia

ROLL NO.: 2103109

BATCH: C23

EXP 1 – Write Android Application to draw basic graphical 2D primitives.

CODE:

MainActivity.java:

```
//Name- Arnav Malvia Batch- C23 Roll No.- 2103109

package com.firstapp.app;

import android.app.Activity;
import android.graphics.Bitmap;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.graphics.drawable.BitmapDrawable;
import android.os.Bundle;
import android.widget.ImageView;

public class MainActivity extends Activity {

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

        // Creating a Bitmap
        Bitmap bg = Bitmap.createBitmap(720, 1280, Bitmap.Config.ARGB_8888);

        // Setting the Bitmap as background for the ImageView
        ImageView imageView = findViewById(R.id.imageView);
        imageView.setBackground(new BitmapDrawable(getResources(), bg));

        // Creating the Canvas Object
    }
}
```

```

Canvas canvas = new Canvas(bg);

// Creating the Paint Object and set its color & TextSize
Paint paint = new Paint();
paint.setColor(Color.BLUE);
paint.setTextSize(50);

// To draw a Rectangle
canvas.drawText("Rectangle", 420, 150, paint);
canvas.drawRect(400, 200, 650, 700, paint);

// To draw a Circle
canvas.drawText("Circle", 120, 150, paint);
canvas.drawCircle(200, 350, 150, paint);

// To draw a Square
canvas.drawText("Square", 120, 800, paint);
canvas.drawRect(50, 850, 350, 1150, paint);

// To draw a Line
canvas.drawText("Line", 480, 800, paint);
canvas.drawLine(520, 850, 520, 1150, paint);
}

}

```

activity_main.xml:

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

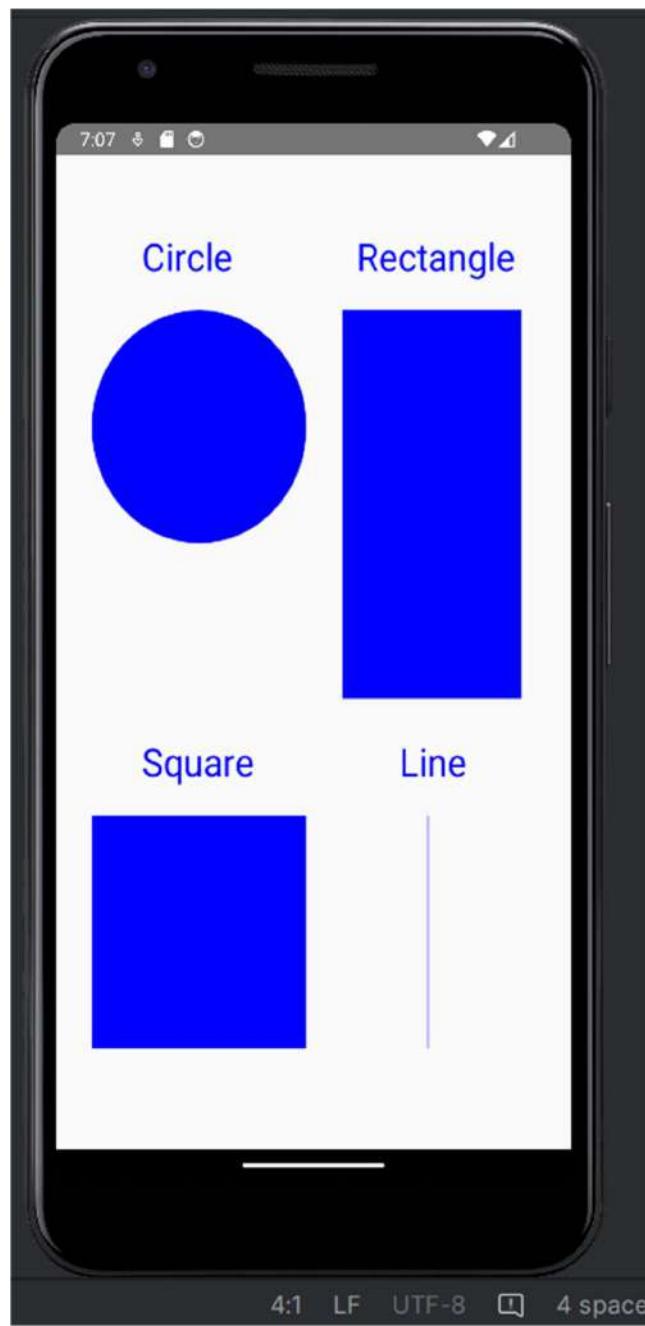
</RelativeLayout>

```

strings.xml:

```
<resources>
    <string name="app_name">MyFirstApp</string>
</resources>
```

OUTPUT:



Experiment :- 2

Aim:- WAA to draw Basic graphical 3D primitives.

The provided android application code, written in Java, creates a basic graphical representation of 3D primitives using a canvas and bitmap. The main activity initialises a bitmap with specific dimensions and sets it as the background for an ImageView in the app's layout. The canvas is used to draw lines and shapes, simulating a 3D cube and cone with the aid of paint objects. The cube and cone are represented through various line segments, forming a visual approximation of 3D geometry.

Understanding the code:-

- package com.example.a3dprimitives;

This line specifies the package name for the Java class. It provides a unique namespace for the class to avoid naming conflicts with other classes.

- import statements

These lines import necessary classes from the Android SDK. The classes include graphics-related classes for drawing shapes and text, classes for handling the application lifestyle and UI components like ImageView.

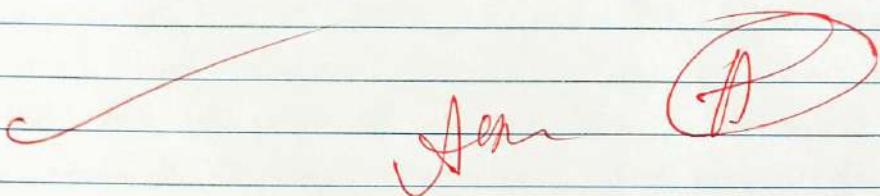
- public class MainActivity extends AppCompatActivity {

This line declares the 'MainActivity' class, which extends 'AppCompatActivity'. In Android the main activity is the entry point of the application.

- `protected void onCreate ()`
The 'onCreate' method is called when the activity is first created. It initializes the activity, sets the content view to the layout defined in `activity_main.xml`.
- `Bitmap bg = Bitmap CreateBitmap (920, 1380, Bitmap.config.ARGB_8888);`
This line creates a new bitmap ('bg') with a width of 920 pixels and height of 1380 pixels, using the ARGB-8888 configuration.
- `ImageView i = (ImageView) findViewById (R.id.imageView);`
`i.setbackgroundDrawable(new BitmapDrawable (bg));`
It finds the ImageView with the ID ~~Image~~ 'ImageView' from the layout and sets the background of the imageView to the created Bitmap.
- `Canvas canvas = new Canvas (bg);`
This line creates a canvas object associated with the bitmap, allowing drawing operations on the bitmap.
- `Paint paint = new Paint();`
`paint.setcolor (color. Black Black)`
`paint.setTextSize (56);`
A paint object is created. This object is set to ~~blue~~ draw in black with a text size of 56.
- `paint p = new new Paint();`
`p.setcolor (color Blue);`
`p.setTextSize (60);`
A p object is created. This object is set to draw in Blue with a text size of 60.

- RectF rectF = new RectF(200, 1100, 400, 1300);
canvas.drawArc(rectF, 0, 180, true, paint);
This part draws a semi circle on the canvas,
simulating a cone.

Conclusion :- Hence I ~~understood~~ implemented the implementation of
Basic graphical 3D primitives in Android Studio using Java.



NAME: Arnav Malvia

ROLL NO.: 2103109

BATCH: C23

EXP 2 – Write Android Application to draw basic graphical 3D primitives

CODE:

MainActivity.java:

```
//Name- Arnav Malvia Batch- C23 Roll No.- 2103109
```

```
package com.firstapp.secondapp;

import android.app.Activity;
import android.graphics.Bitmap;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.os.Bundle;
import android.widget.ImageView;

public class MainActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Bitmap bg = Bitmap.createBitmap(720, 1280, Bitmap.Config.ARGB_8888);
        ImageView img = findViewById(R.id.imageView);
        img.setImageBitmap(bg);

        Canvas canvas = new Canvas(bg);

        // paint object creation
        Paint paint1 = new Paint();
        paint1.setColor(Color.RED);
        paint1.setTextSize(50);
```

```
Paint paint2 = new Paint();
paint2.setColor(Color.BLUE);
paint2.setTextSize(50);
Paint paint = new Paint();
paint.setColor(Color.BLACK);
paint.setTextSize(50);

canvas.drawLine(0, 200, 300, 200, paint);
canvas.drawLine(300, 200, 300, 600, paint);
canvas.drawLine(300, 600, 0, 600, paint);
canvas.drawLine(0, 600, 0, 200, paint);

canvas.drawLine(100, 50, 400, 50, paint);
canvas.drawLine(400, 50, 400, 450, paint);
canvas.drawLine(400, 450, 100, 450, paint);
canvas.drawLine(100, 450, 100, 50, paint);

canvas.drawLine(100, 50, 0, 200, paint);
canvas.drawLine(400, 50, 300, 200, paint);

canvas.drawLine(300, 600, 400, 450, paint);
canvas.drawLine(100, 450, 0, 600, paint);

//prism
canvas.drawLine(90, 900, 400, 800, paint);

canvas.drawLine(90, 900, 50, 1100, paint);
canvas.drawLine(90, 900, 140, 1100, paint);
canvas.drawLine(140, 1100, 50, 1100, paint);

canvas.drawLine(140, 1100, 450, 1000, paint);

canvas.drawLine(450, 1000, 400, 800, paint);

canvas.drawLine(400, 800, 360, 1000, paint);
canvas.drawLine(360, 1000, 450, 1000, paint);

canvas.drawLine(360, 1000, 300, 1100, paint);
}

}
```

activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

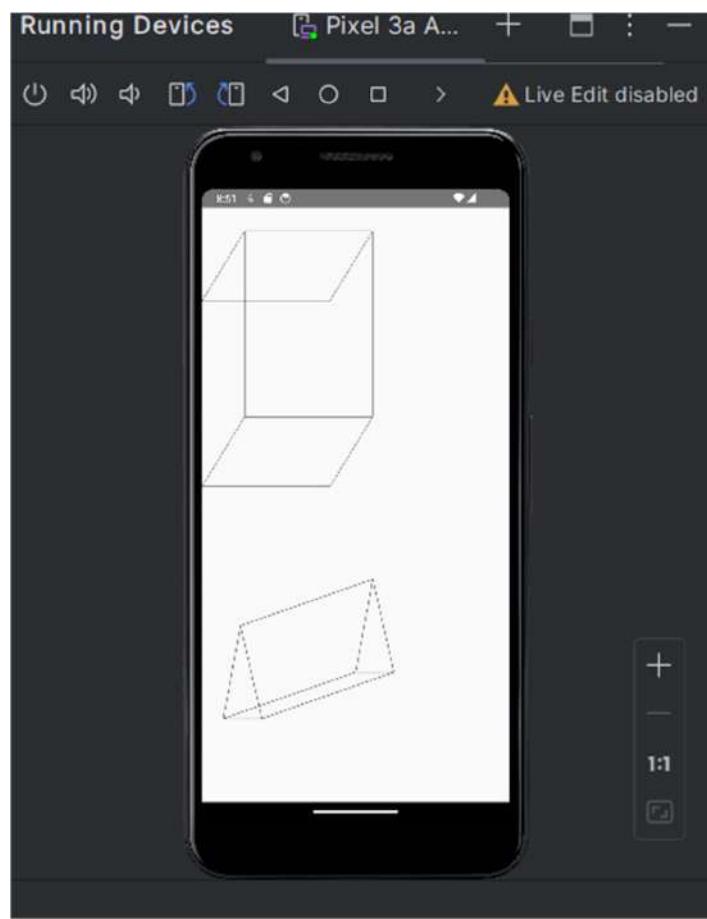
    <ImageView
        android:id="@+id/imageView"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:scaleType="fitXY" />

</RelativeLayout>
```

strings.xml:

```
<resources>
    <string name="app_name">My 3d APP</string>
</resources>
```

OUTPUT:



Experiment :- 3

Aim:- Write an Android application to design a form with GUI components.

Graphical User Interface

Theory:-

Android studio is the official integrated development environment for android application development. Android studio provides more features that enhance our productivity while building android apps.

Imports and built-in functions used:

1) android.widget.* ;

Imports under android.widget.* cover a range of UI components. In this code, I am importing classes such as editText, CheckBox, RadioButton and Radio Group. These classes are part of android.widget framework.

2) android.widget.Toast :

This imports as necessary for displaying short duration messages to the user. It provides a simple feedback mechanism often used for notifying users of actions or displaying brief information.

3) FindViewById :

It is used to locate a view in the current layout by its ID. In this code it's employed to initialise various UI elements like EditText, CheckBox, Radio Group and Button by associating them with their

corresponding XML layout IDs.

4) SetOn Click Listener :-

It is ~~not~~ used to set an event listener for a view. It takes an instance of the view. On Click Listener interface, defining the behaviour, that should occur when the associated view is clicked. ~~@~~ This code, is used to capture the click event on the submit button.

Conclusion :

Thus I ~~have~~ designed a form with GUI components in android studio.

Amritpal Singh

NAME: Arnav Malvia

ROLL NO.: 2103109

BATCH: C23

EXP 3 – Write android application to design a form with GUI components

CODE:

MainActivity.java:

```
//Name- Arnav Malvia Batch- C23 Roll No.- 2103109
```

```
package com.firstapp.thirdapp;

import android.app.Activity;
import android.app.DatePickerDialog;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.DatePicker;
import android.widget.EditText;
import android.widget.Spinner;
import android.widget.Toast;

import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Locale;

public class MainActivity extends Activity {
    private EditText editTextFirstName, editTextLastName, editTextEmail,
    editTextPhoneNumber, editTextPassword;
    private Spinner spinnerGender;
    private Button buttonSelectDate;
    private Calendar calendar;

    @Override
```

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

    editTextFirstName = findViewById(R.id.editTextFirstName);
    editTextLastName = findViewById(R.id.editTextLastName);
    editTextEmail = findViewById(R.id.editTextEmail);
    editTextPhoneNumber = findViewById(R.id.editTextPhoneNumber);
    editTextPassword = findViewById(R.id.editTextPassword);
    spinnerGender = findViewById(R.id.spinnerGender);
    buttonSelectDate = findViewById(R.id.buttonSelectDate);

    ArrayAdapter<CharSequence> adapter =
    ArrayAdapter.createFromResource(this,
        R.array.gender_array, android.R.layout.simple_spinner_item);

    adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_ite
m);
    spinnerGender.setAdapter(adapter);

    calendar = Calendar.getInstance();

    buttonSelectDate.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            new DatePickerDialog(MainActivity.this, dateSetListener,
                calendar.get(Calendar.YEAR),
                calendar.get(Calendar.MONTH),
                calendar.get(Calendar.DAY_OF_MONTH)).show();
        }
    });

    spinnerGender.setOnItemSelectedListener(new
    AdapterView.OnItemSelectedListener() {
        @Override
        public void onItemSelected(AdapterView<?> parent, View view, int position,
        long id) {
            // Handle gender selection
        }

        @Override
        public void onNothingSelected(AdapterView<?> parent) {
            // Do nothing
        }
    });
}
```

```

        }
    });

Button buttonProceed = findViewById(R.id.buttonProceed);
buttonProceed.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // Proceed button click action
        // Gather user information and process
    }
});

Button buttonReset = findViewById(R.id.buttonReset);
buttonReset.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // Reset button click action
        // Clear all fields
    }
});
}

private DatePickerDialog.OnDateSetListener dateSetListener = new
DatePickerDialog.OnDateSetListener() {
    @Override
    public void onDateSet(DatePicker view, int year, int monthOfYear, int
dayOfMonth) {
        calendar.set(Calendar.YEAR, year);
        calendar.set(Calendar.MONTH, monthOfYear);
        calendar.set(Calendar.DAY_OF_MONTH, dayOfMonth);
        updateLabel();
    }
};

private void updateLabel() {
    String myFormat = "MM/dd/yyyy"; //In which you need put here
    SimpleDateFormat sdf = new SimpleDateFormat(myFormat, Locale.US);

    buttonSelectDate.setText(sdf.format(calendar.getTime()));
}
}

```

activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp"
    tools:context=".MainActivity">

    <EditText
        android:id="@+id/editTextFirstName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="First Name" />

    <EditText
        android:id="@+id/editTextLastName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Last Name" />

    <EditText
        android:id="@+id/editTextEmail"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Email" />

    <EditText
        android:id="@+id/editTextPhoneNumber"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Phone Number" />

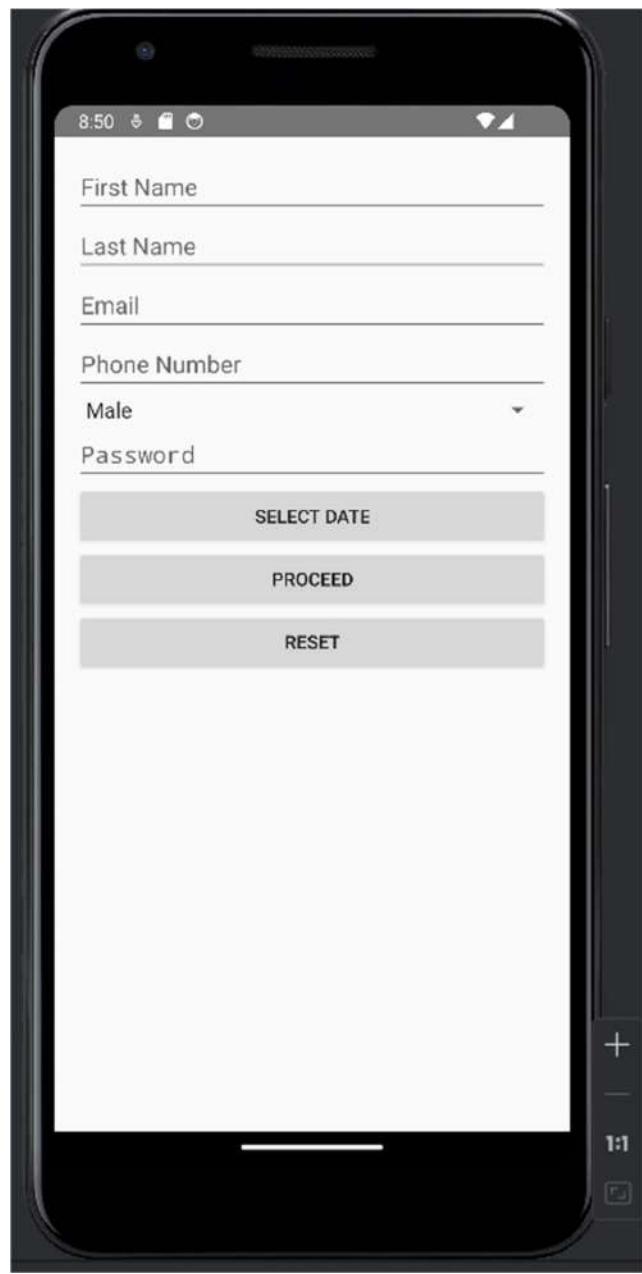
    <Spinner
        android:id="@+id/spinnerGender"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />
```

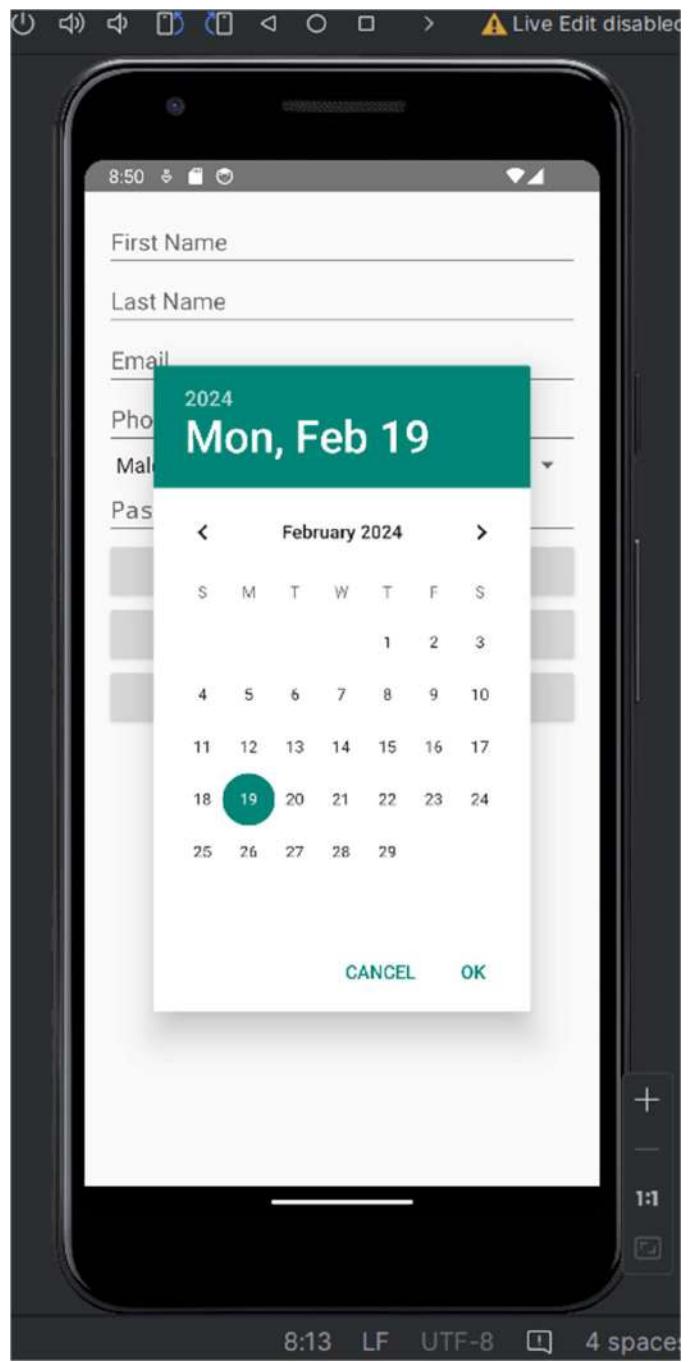
```
<EditText  
    android:id="@+id/editTextPassword"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:hint="Password"  
    android:inputType="textPassword" />  
  
<Button  
    android:id="@+id/buttonSelectDate"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="Select Date" />  
  
<Button  
    android:id="@+id/buttonProceed"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="Proceed" />  
  
<Button  
    android:id="@+id/buttonReset"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="Reset" />  
  
</LinearLayout>
```

strings.xml:

```
<resources>  
    <string name="app_name">EXP3_102</string>  
    <string-array name="gender_array">  
        <item>Male</item>  
        <item>Female</item>  
        <item>Other</item>  
    </string-array>  
</resources>
```

OUTPUT:





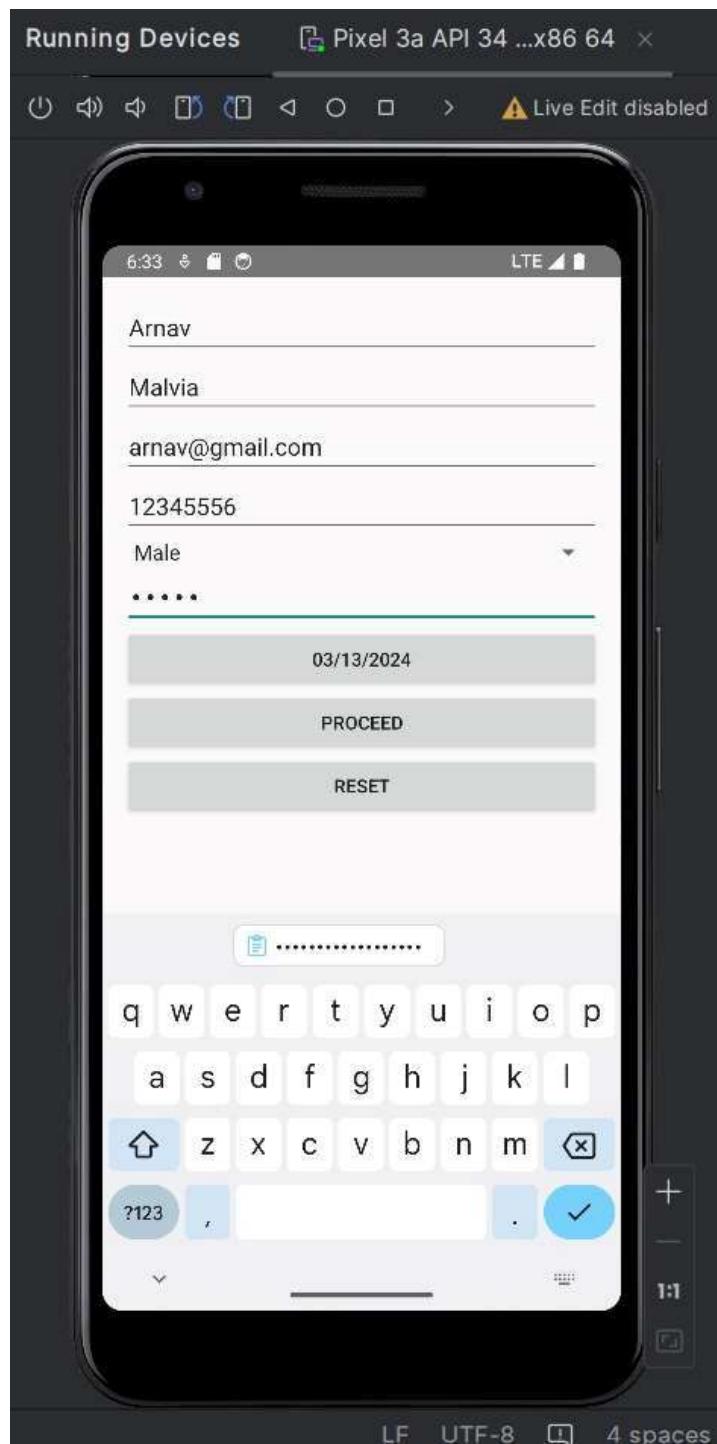
8:13

LF

UTF-8

!

4 spaces



Experiment :- 4

Aim:- WAA to design GUI components with database.

Theory:-

For this experiment we use the same GUI and form that was designed in experiment 3, and we add SQL database connectivity to the form.

This will lead to:

When the user fills in their name, date-of-birth, etc and presses submit, the data will be stored in SQL database.

To achieve this we follow the following steps:-

1. Create a SQLite Database Helper class.

import android.content.Context;

import android.database.sqlite.SQLiteDatabase;

import android.database.sqlite.SQLiteOpenHelper;

public class MyDBHelper extends SQLiteOpenHelper {

~~public void onCreate(SQLiteDatabase db) {~~

~~String CREATE_TABLE = "Create Table" + TABLE_Name +
"(" + column_Name + " TEXT"~~

~~+ column_DOB + " TEXT"~~

~~+ column_Gender " TEXT"~~

~~db.execSQL(CREATE_TABLE)~~

3

~~public void insertData (String name, String dob, String gender)~~

```
SQLite Database db = this.getWritableDatabase();  
String INSERT_DATA = "INSERT INTO TABLE_NAME {"  
    + "column_name",  
    + "column_DOB",  
    + "column_Gender"  
    + "values ('" + name + "', '" + dob + "', '" + gender + "');  
db.execSQL(INSERT_DATA);  
db.close();  
}  
}
```

In ~~the~~ activity main Java, in onSubmit Button click function add the code:

```
dbHelper.insertData(name, dob, gender);
```

Conclusion :-

I ~~have~~ have connected the front end to SQL which will insert data upon pressing 'Submit' button.

~~Step~~ ✓ D

NAME: Arnav Malvia

ROLL NO.: 2103109

BATCH: C23

EXP 4 – Write android application to design GUI components using Database

CODE:

MainActivity.java:

```
//Name- Arnav Malvia Batch- C23 Roll No.- 2103109
```

```
package com.firstapp.seventhappdb;

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

import androidx.appcompat.app.AppCompatActivity;

import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;

public class MainActivity extends AppCompatActivity {

    FirebaseAuth auth;
    Button button;
    TextView textView;
    FirebaseUser user;

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

        auth = FirebaseAuth.getInstance();
```

```

button = findViewById(R.id.logout);
textView = findViewById(R.id.user_details);
user = auth.getCurrentUser();

if (user == null) {
    // Redirect to LoginActivity if user is not logged in
    startActivity(new Intent(MainActivity.this, Login.class));
    finish();
} else {
    // Display user details
    textView.setText(user.getEmail());
}

button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        // Log out user
        FirebaseAuth.getInstance().signOut();
        // Redirect to LoginActivity after logging out
        startActivity(new Intent(MainActivity.this, Login.class));
        finish();
    }
});
}
}

```

activity_main.xml:

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical"
    android:gravity="center"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
    <TextView
        android:id="@+id/user_details"

```

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Hellow World!"/>

<Button
    android:text="@string/logout"
    android:id="@+id/logout"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">

</Button>
</LinearLayout>
```

Login.java:

```
package com.firstapp.seventhappdb;

import android.content.Intent;
import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.ProgressBar;
import android.widget.TextView;
import android.widget.Toast;
import com.google.android.material.textfield.TextInputEditText;

import androidx.appcompat.app.AppCompatActivity;

import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;

public class Login extends AppCompatActivity {
    TextInputEditText editTextEmail, editTextPassword;
    Button buttonLogin;
    FirebaseAuth mAuth;
    ProgressBar progressBar;
    TextView textView;

    @Override
```

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_login);
    mAuth = FirebaseAuth.getInstance();
    editTextEmail = findViewById(R.id.email);
    editTextPassword = findViewById(R.id.password);
    buttonLogin = findViewById(R.id.btn_login);
    progressBar = findViewById(R.id.progressBar);
    textView = findViewById(R.id.registerNow);

    textView.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent intent = new Intent(getApplicationContext(), Register.class);
            startActivity(intent);
            finish();
        }
    });

    buttonLogin.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            String email = editTextEmail.getText().toString().trim();
            String password = editTextPassword.getText().toString().trim();

            if (TextUtils.isEmpty(email)) {
                Toast.makeText(Login.this, "Enter email",
                        Toast.LENGTH_SHORT).show();
                return;
            }
            if (TextUtils.isEmpty(password)) {
                Toast.makeText(Login.this, "Enter password",
                        Toast.LENGTH_SHORT).show();
                return;
            }

            progressBar.setVisibility(View.VISIBLE);

            mAuth.signInWithEmailAndPassword(email, password)
                    .addOnCompleteListener(Login.this, task -> {
                        progressBar.setVisibility(View.GONE);
                        if (task.isSuccessful()) {
                            // Sign in success, update UI with the signed-in user's information

```

```
        Toast.makeText(Login.this, "Login Successful",
Toast.LENGTH_SHORT).show();
        Intent intent = new Intent(Login.this, MainActivity.class);
        startActivity(intent);
        finish();
    } else {
        // If sign in fails, display a message to the user.
        Toast.makeText(Login.this, "Authentication failed.",
Toast.LENGTH_SHORT).show();
    }
});  
});  
}  
}  
}
```

activity_login.java:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:orientation="vertical"
    android:gravity="center"
    android:layout_height="match_parent"
    android:padding="15dp"
    tools:context=".Login">
    <TextView
        android:text="@string/login"
        android:textSize="25sp"
        android:textStyle="bold"
        android:layout_marginBottom="20dp"
        android:gravity="center"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />
    <com.google.android.material.textfield.TextInputLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <com.google.android.material.textfield.TextInputEditText
```

```
    android:id="@+id/email"
    android:hint="@string/email"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

</com.google.android.material.textfield.TextInputEditText>
</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.textfield.TextInputLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <com.google.android.material.textfield.TextInputEditText
        android:id="@+id/password"
        android:hint="@string/password"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        </com.google.android.material.textfield.TextInputEditText>
    </com.google.android.material.textfield.TextInputLayout>
    <ProgressBar
        android:id="@+id/progressBar"
        android:visibility="gone"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"/>
    <Button
        android:id="@+id/btn_login"
        android:text="@string/login"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">

</Button>
<TextView
    android:textStyle="bold"
    android:textSize="20sp"
    android:gravity="center"
    android:layout_marginTop="20dp"
    android:id="@+id/registerNow"
    android:text="@string/click_to_register"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />
</LinearLayout>
```

Register.java:

```
package com.firstapp.seventhappdb;

import android.content.Intent;
import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.ProgressBar;
import android.widget.TextView;
import android.widget.Toast;
import com.google.android.material.textfield.TextInputEditText;

import androidx.appcompat.app.AppCompatActivity;

import com.google.firebase.auth.FirebaseAuth;

public class Register extends AppCompatActivity {

    TextInputEditText editTextEmail, editTextPassword;
    Button buttonReg;
    FirebaseAuth mAuth;
    ProgressBar progressBar;
    TextView textView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_register);
        mAuth = FirebaseAuth.getInstance();
        editTextEmail = findViewById(R.id.email);
        editTextPassword = findViewById(R.id.password);
        buttonReg = findViewById(R.id.btn_register);
        progressBar = findViewById(R.id.progressBar);
        textView = findViewById(R.id.loginNow);

        textView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
```

```

        Intent intent = new Intent(getApplicationContext(), Login.class);
        startActivity(intent);
        finish();
    }
});

buttonReg.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        String email = editTextEmail.getText().toString().trim();
        String password = editTextPassword.getText().toString().trim();

        if (TextUtils.isEmpty(email)) {
            Toast.makeText(Register.this, "Enter email",
Toast.LENGTH_SHORT).show();
            return;
        }
        if (TextUtils.isEmpty(password)) {
            Toast.makeText(Register.this, "Enter password",
Toast.LENGTH_SHORT).show();
            return;
        }

        progressBar.setVisibility(View.VISIBLE);

        mAuth.createUserWithEmailAndPassword(email, password)
            .addOnCompleteListener(Register.this, task -> {
                progressBar.setVisibility(View.GONE);
                if (task.isSuccessful()) {
                    Toast.makeText(Register.this, "Registration successful.",
Toast.LENGTH_SHORT).show();
                    Intent intent = new Intent(getApplicationContext(),
MainActivity.class);
                    startActivity(intent);
                    finish();
                } else {
                    Toast.makeText(Register.this, "Authentication failed.",
Toast.LENGTH_SHORT).show();
                }
            });
    }
});

```

```
    }  
}
```

activity_register.java:

```
<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:app="http://schemas.android.com/apk/res-auto"  
    xmlns:tools="http://schemas.android.com/tools"  
    android:layout_width="match_parent"  
    android:orientation="vertical"  
    android:gravity="center"  
    android:layout_height="match_parent"  
    android:padding="15dp"  
    tools:context=".Register">  
    <TextView  
        android:text="@string/register"  
        android:textSize="25sp"  
        android:textStyle="bold"  
        android:layout_marginBottom="20dp"  
        android:gravity="center"  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content" />  
    <com.google.android.material.textfield.TextInputLayout  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content">  
        <com.google.android.material.textfield.TextInputEditText  
            android:id="@+id/email"  
            android:hint="@string/email"  
            android:layout_width="match_parent"  
            android:layout_height="wrap_content">  
        </com.google.android.material.textfield.TextInputEditText>  
    </com.google.android.material.textfield.TextInputLayout>  
    <com.google.android.material.textfield.TextInputLayout  
        android:layout_width="match_parent"
```

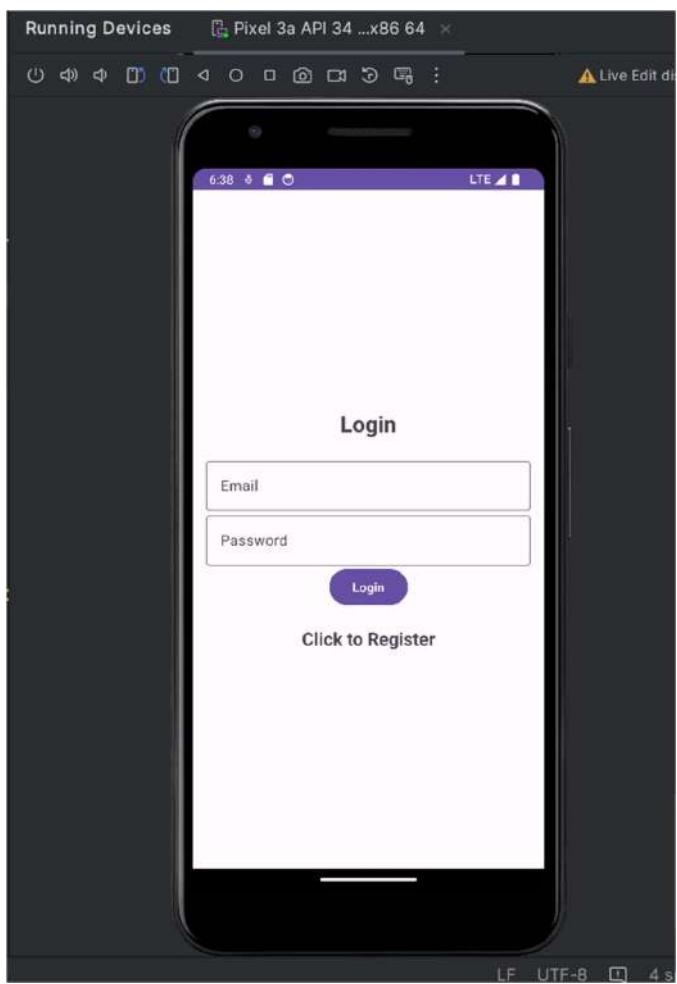
```
        android:layout_height="wrap_content">

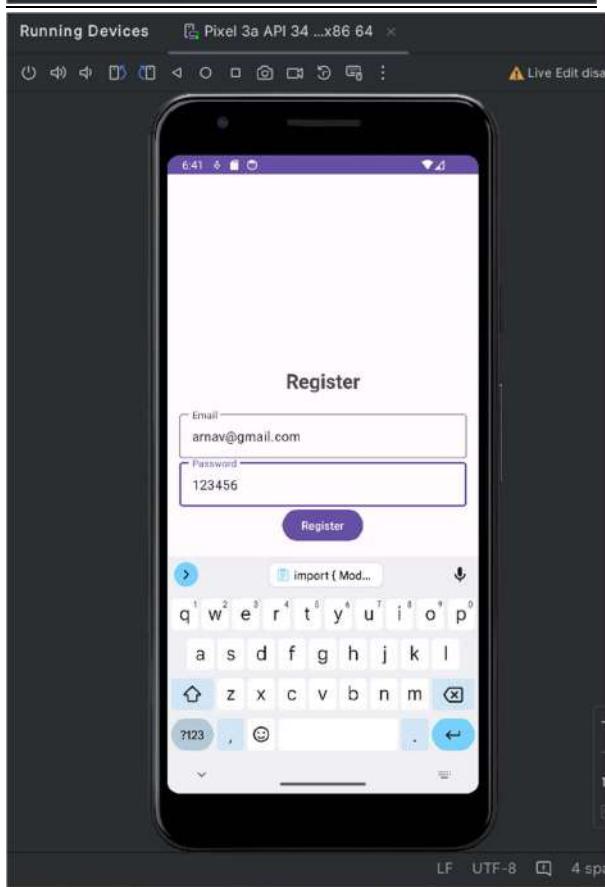
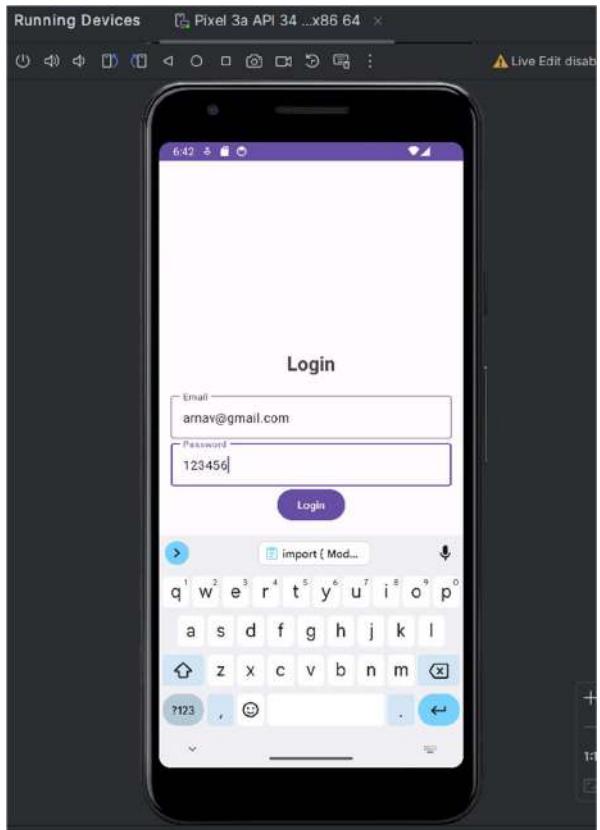
        <com.google.android.material.textfield.TextInputEditText
            android:id="@+id/password"
            android:hint="@string/password"
            android:layout_width="match_parent"
            android:layout_height="wrap_content">

        </com.google.android.material.textfield.TextInputEditText>
    </com.google.android.material.textfield.TextInputLayout>
<ProgressBar
    android:id="@+id/progressBar"
    android:visibility="gone"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"/>
<Button
    android:id="@+id	btn_register"
    android:text="@string/register"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">

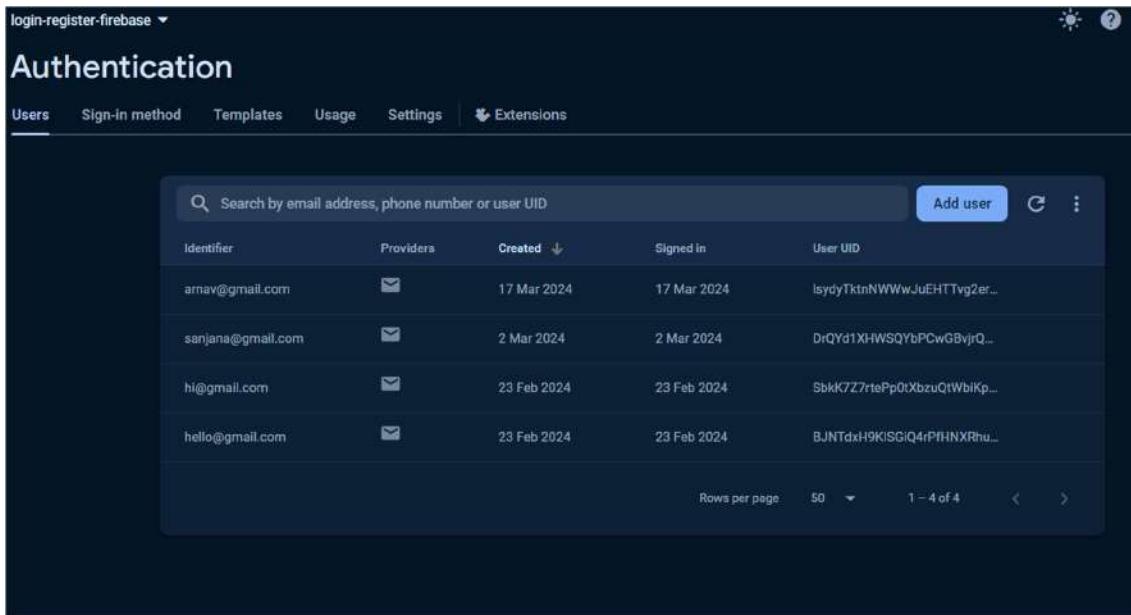
</Button>
<TextView
    android:textStyle="bold"
    android:textSize="20sp"
    android:gravity="center"
    android:layout_marginTop="20dp"
    android:id="@+id/loginNow"
    android:text="@string/click_to_login"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />
</LinearLayout>
```

OUTPUT:





FIREBASE:



The screenshot shows the Firebase Authentication console under the 'login-register-firebase' project. The 'Users' tab is selected. A search bar at the top allows searching by email address, phone number, or user UID. Below it is a table displaying four user entries. The columns are 'Identifier', 'Providers', 'Created', 'Signed in', and 'User UID'. Each row shows an email address, a mail icon indicating email as the provider, and the date of creation and sign-in. The User UIDs are partially visible.

Identifier	Providers	Created	Signed in	User UID
amav@gmail.com	✉️	17 Mar 2024	17 Mar 2024	lsydyTktnNWWwJuEHTTvg2er...
sanjana@gmail.com	✉️	2 Mar 2024	2 Mar 2024	DrQYd1XHWSQYbPCwGBvjrQ...
hi@gmail.com	✉️	23 Feb 2024	23 Feb 2024	SbkK7Z7rtePp0XbzuQtWbIKp...
hello@gmail.com	✉️	23 Feb 2024	23 Feb 2024	BJNTdxH9KISGjQ4rPfHNXRhu...

Rows per page: 50 | 1 – 4 of 4

Experiment:- 5

Aim:- WAP to develop an EMI calculator application.

Theory:-

Android studio is the official integrated development environment for android application development. Android studio provides more features that enhance our productivity while building android app,

Imports and built in functions used:

1) android.widget.*;

Imports under android.widget.* cover a range of UI components. In this code, I am importing classes such as EditText, CheckBox, RadioButton and Radio Group.

These classes are part of android.widget package.

2) Find View By Id:

It is used to locate a view in the current layout by its ID. In this code it's employed to initialise various UI elements like, editText, CheckBox, Radio Group and Button by associating them with their corresponding XML layout IDs.

3) set OnClickListener:-

It is used to set an event listener for a view. It takes an instance of the view - OnClickListener interface, defining the behaviour that should occur when the associated view is clicked. In this code it is used to capture

the click event on the submit button.

4) Calculate EMI:

calculate EMI method retrieves the input values from the Edit text fields, converts them to double and then performs the EMI calculation using the formula for calculation EMI for a loan. The result is displayed in the Text View.

Conclusion:-

Thus I ~~have~~ designed an emi calculator with GUI components in android studio.

Jaya

NAME: Arnav Malvia

ROLL NO.: 2103109

BATCH: C23

EXP 5 – Write android application to develop EMI calculator application

CODE:

MainActivity.java:

```
//Name- Arnav Malvia Batch- C23 Roll No.- 2103109
```

```
package com.firstapp.fourthapp;
```

```
import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

public class MainActivity extends Activity {
    private EditText editTextLoanAmount, editTextInterestRate, editTextLoanTenure;
    private Button buttonCalculate;
    private TextView textViewResult;

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

        editTextLoanAmount = findViewById(R.id.editTextLoanAmount);
        editTextInterestRate = findViewById(R.id.editTextInterestRate);
        editTextLoanTenure = findViewById(R.id.editTextLoanTenure);
        buttonCalculate = findViewById(R.id.buttonCalculate);
        textViewResult = findViewById(R.id.textViewResult);
```

```

buttonCalculate.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        calculateEMI();
    }
});

private void calculateEMI() {
    String loanAmountStr = editTextLoanAmount.getText().toString();
    String interestRateStr = editTextInterestRate.getText().toString();
    String loanTenureStr = editTextLoanTenure.getText().toString();

    if (loanAmountStr.isEmpty() || interestRateStr.isEmpty() ||
    loanTenureStr.isEmpty()) {
        Toast.makeText(this, "Please enter all fields",
        Toast.LENGTH_SHORT).show();
        return;
    }

    double loanAmount = Double.parseDouble(loanAmountStr);
    double annualInterestRate = Double.parseDouble(interestRateStr);
    double monthlyInterestRate = annualInterestRate / 12 / 100;
    int loanTenureMonths = Integer.parseInt(loanTenureStr);

    double emi = calculateEMI(loanAmount, monthlyInterestRate,
    loanTenureMonths);
    String result = String.format(getString(R.string.result_label) + " %.2f", emi);
    textViewResult.setText(result);
}

private double calculateEMI(double loanAmount, double monthlyInterestRate, int
loanTenureMonths) {
    double emi;
    emi = loanAmount * monthlyInterestRate * Math.pow(1 + monthlyInterestRate,
    loanTenureMonths) /
        (Math.pow(1 + monthlyInterestRate, loanTenureMonths) - 1);
    return emi;
}
}

```

activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp"
    tools:context=".MainActivity">

    <EditText
        android:id="@+id/editTextLoanAmount"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="@string/loan_amount_hint"
        android:inputType="numberDecimal" />

    <EditText
        android:id="@+id/editTextInterestRate"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/editTextLoanAmount"
        android:hint="@string/interest_rate_hint"
        android:inputType="numberDecimal" />

    <EditText
        android:id="@+id/editTextLoanTenure"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/editTextInterestRate"
        android:hint="@string/loan_tenure_hint"
        android:inputType="number" />

    <Button
        android:id="@+id/buttonCalculate"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/editTextLoanTenure"
        android:text="@string/calculate_button_label" />

    <TextView
        android:id="@+id/textViewResult"
```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/buttonCalculate"
        android:text="@string/result_label"
        android:textStyle="bold" />

    </RelativeLayout>
```

strings.xml:

```
<resources>
    <string name="app_name">EMI Calculator</string>
    <string name="loan_amount_hint">Loan Amount</string>
    <string name="interest_rate_hint">Interest Rate (%)</string>
    <string name="loan_tenure_hint">Loan Tenure (months)</string>
    <string name="calculate_button_label">Calculate EMI</string>
    <string name="result_label">EMI: </string>
</resources>
```

OUTPUT:



Experiment :- 6

Aim:- WAA that creates an alert on receiving a message.

Theory:-

The provided code is for an Android application named "Message Alertapp". It consists of a single activity, 'MainActivity' and an associated XML Layout file, 'activity_main.xml'. Additionally it includes string resources defined in strings.xml for localisation.

MainActivity.java:

1. imports : Import necessary packages and classes
2. Class Declaration : Define the Main Activity class which extends 'Activity'.
3. onCreate() : Override the onCreate method, which sets the context view to the layout defined in activity_main.xml. It also sets up a click listener for the "Receive Message" button.
4. showMessageAlert() : Method to display a alert dialog when the Receive Message button is clicked. It creates an 'Alert Dialog.Builder', sets the title, message and button text using styling resources and shows the dialog.

activity_main.xml

1. Layout : Defines a 'RelativeLayout' with padding.
2. Button (Receive Message) : Displays a button with Text "Receive Message" at the center of the layout. It has an ID referred in 'MainActivity' and its text is defined in 'strings.xml'.

Strings.xml:

1. String Resources: Contains various String resources used in the application, such as app name, button label, alert dialog title, message and button text.

The applications functionality is straight forward: when the "Receive Message" button is clicked, it triggers an alert dialog displaying a message indicating that the user has received a new message. The user can then dismiss the dialog by clicking the "OK" button. The use of string resources facilitates easy localisation and management of text context ensuring better user experience across different languages and regions.

Conclusion:- Implemented an alert box on receiving a message in android studio.

Aman A

NAME: Arnav Malvia

ROLL NO.: 2103109

BATCH: C23

EXP 6 – Write android application that creates an alert on receiving a message

CODE:

MainActivity.java:

```
//Name- Arnav Malvia Batch- C23 Roll No.- 2103109
```

```
package com.firstapp.fifthapp;

import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class MainActivity extends Activity {

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

        Button buttonReceiveMessage = findViewById(R.id.buttonReceiveMessage);
        buttonReceiveMessage.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                showMessageAlert();
            }
        });
    }
}
```

```

private void showMessageAlert() {
    AlertDialog.Builder builder = new AlertDialog.Builder(this);
    builder.setTitle(getString(R.string.alert_dialog_title))
        .setMessage(getString(R.string.alert_dialog_message))
        .setPositiveButton(getString(R.string.alert_dialog_button_ok), new
DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            // Do something when OK button is clicked
        }
    })
    .show();
}
}

```

activity_main.xml:

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/buttonReceiveMessage"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true"
        android:text="@string/receive_message_button_label" />

</RelativeLayout>

```

strings.xml:

```

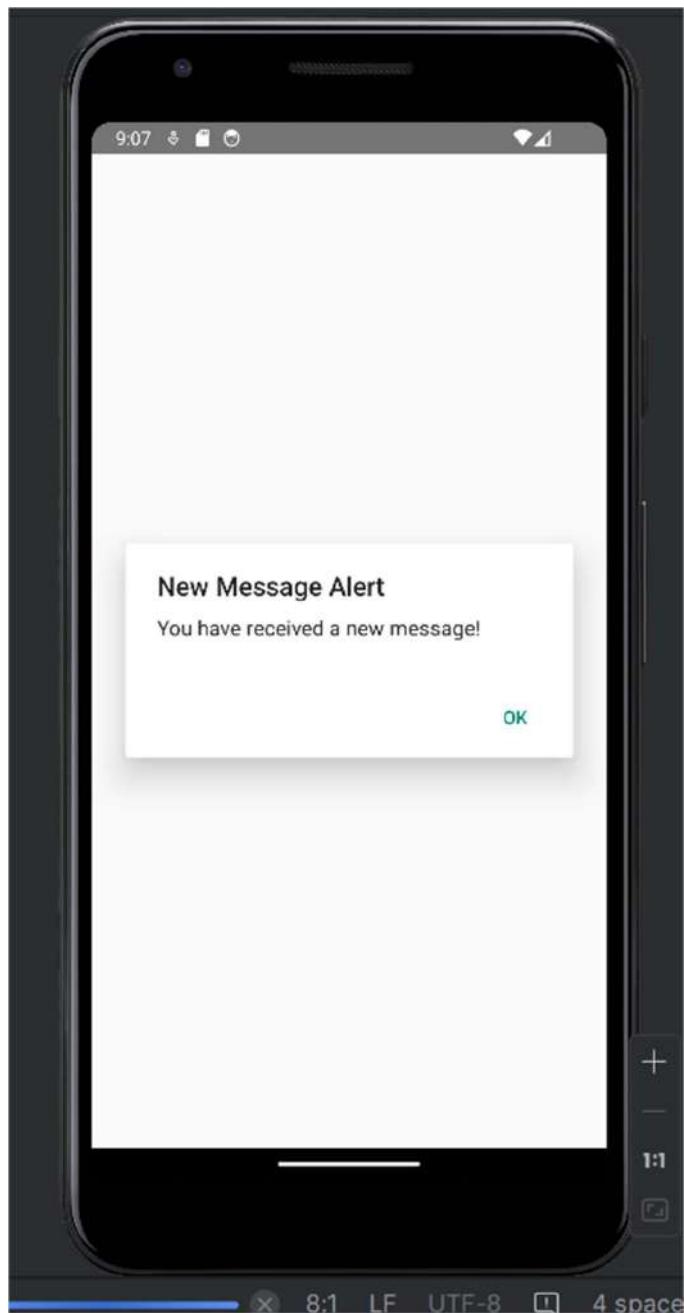
<resources>
    <string name="app_name">MessageAlertApp</string>
    <string name="receive_message_button_label">Receive Message</string>

```

```
<string name="alert_dialog_title">New Message Alert</string>
<string name="alert_dialog_message">You have received a new message!</string>
<string name="alert_dialog_button_ok">OK</string>
</resources>
```

OUTPUT:





Experiment :- 7

Aim:- WAA to implement Basic Calculator using Android.

Theory :-

The provided code is for an Android calculator application. It consists of a single activity 'MainActivity', and an associated XML layout file, 'activity_main.xml', which defines the user interface.

MainActivity.java:

1. imports : Import necessary packages and classes.
2. Class Declaration: Define the 'MainActivity' class which extends 'activity'.
3. Variable Declaration: Declare private variables for a 'Text View' to display input and results; and a 'StringBuilder' to store user input.
4. OnCreate(): Override the 'OnCreate()' method, which sets the context view to the layout defined in 'activity_main.xml' and initialises the 'TextView' and 'String Builder'.

5. Button Click methods:

- a) OnDigitClick(view view): This method handles the click event for digit buttons (0-9).
- b) OnOperatorClick(view view): This method handles the click even for operator buttons (+, -, ×, ÷).
- c) OnEqualsClick(view view): This method is invoked when the equals button (=) is clicked.
- d) OnClearClick(view view): This method clears the input stored and updates the display accordingly.

- e) On Backspace Click (View view): Handles the backspace button click by deleting the last character.
- f) On Toggle Sign Click (view view): Toggles the sign of the current input by appending a negative sign (-) if its positive, or removing it if its negative.
- 6. updateDisplay(): Update the text displayed in the 'Text View' with the current input stored in the 'String Builder'.
- 7. EvaluateExpression(): Perform arithmetic calculations based on the input string.

activity_main.xml:

- 1) Layout: Defines a RelativeLayout with padding
- 2) TextView(Display): Displays the current input and results
- 3) Buttons: Layout with a 'GridLayout' containing buttons for digits, arithmetic operators, decimal point, clear and equals. Each button has an ID and an 'onClick' attribute to call the corresponding method.

The application functions as a basic calculator, allowing users to perform basic arithmetic operations by entering digits and operators. It provides features such as entering a digit, clearing input, deleting characters and toggling the sign of the input.

Conclusion:- Implemented a basic calculator in android studio.

Yash (A)

NAME: Arnav Malvia

ROLL NO.: 2103109

BATCH: C23

EXP 7 – Write android application to implement basic calculator using android

CODE:

MainActivity.java:

```
//Name- Arnav Malvia Batch- C23 Roll No.- 2103109
```

```
package com.firstapp.sixthapp;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

public class MainActivity extends Activity {
    private TextView textViewDisplay;
    private StringBuilder inputStringBuilder;

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

        textViewDisplay = findViewById(R.id.textViewDisplay);
        inputStringBuilder = new StringBuilder();
    }

    public void onDigitClick(View view) {
        String digit = ((Button) view).getText().toString();
        inputStringBuilder.append(digit);
        updateDisplay();
    }
}
```

```
public void onOperatorClick(View view) {
    String operator = ((Button) view).getText().toString();
    inputStringBuilder.append(" ").append(operator).append(" ");
    updateDisplay();
}

public void onEqualsClick(View view) {
    String input = inputStringBuilder.toString();
    // Perform calculation and update display
    try {
        double result = evaluateExpression(input);
        textViewDisplay.setText(String.valueOf(result));
    } catch (IllegalArgumentException e) {
        textViewDisplay.setText("Error");
    }
}

public void onClearClick(View view) {
    inputStringBuilder.setLength(0);
    updateDisplay();
}

public void onBackspaceClick(View view) {
    if (inputStringBuilder.length() > 0) {
        inputStringBuilder.deleteCharAt(inputStringBuilder.length() - 1);
        updateDisplay();
    }
}

public void onToggleSignClick(View view) {
    if (inputStringBuilder.length() > 0) {
        char lastChar = inputStringBuilder.charAt(inputStringBuilder.length() - 1);
        if (lastChar == '-') {
            inputStringBuilder.deleteCharAt(inputStringBuilder.length() - 1);
        } else {
            inputStringBuilder.append("-");
        }
        updateDisplay();
    }
}

private void updateDisplay() {
```

```

        textViewDisplay.setText(inputStringBuilder.toString());
    }

private double evaluateExpression(String expression) {
    // Split the expression into operands and operator
    String[] tokens = expression.split(" ");
    double operand1 = Double.parseDouble(tokens[0]);
    String operator = tokens[1];
    double operand2 = Double.parseDouble(tokens[2]);

    // Perform the operation
    switch (operator) {
        case "+":
            return operand1 + operand2;
        case "-":
            return operand1 - operand2;
        case "×":
            return operand1 * operand2;
        case "÷":
            if (operand2 == 0) {
                throw new IllegalArgumentException("Division by zero");
            }
            return operand1 / operand2;
        default:
            throw new IllegalArgumentException("Invalid operator");
    }
}

```

activity_main.java:

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp"
    tools:context=".MainActivity">

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

```

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="16dp"
    android:background="#CCCCCC"
    android:padding="16dp"
    android:textSize="24sp"
    android:textColor="#000000"
    android:text="0"
    android:gravity="right"/>

<GridLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/textViewDisplay"
    android:rowCount="5"
    android:columnCount="4"
    android:layout_marginTop="16dp">

    <!-- Row 1 -->
    <Button
        android:id="@+id/buttonClear"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="C"
        android:layout_row="0"
        android:layout_column="0"
        android:layout_columnSpan="2"
        android:onClick="onClearClick" />

    <Button
        android:id="@+id/buttonDivide"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="÷"
        android:layout_row="0"
        android:layout_column="2"
        android:onClick="onOperatorClick" />

    <!-- Row 2 -->
    <Button
        android:id="@+id/button7"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```

```
    android:text="7"
    android:layout_row="1"
    android:layout_column="0"
    android:onClick="onDigitClick" />

<Button
    android:id="@+id/button8"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="8"
    android:layout_row="1"
    android:layout_column="1"
    android:onClick="onDigitClick" />

<Button
    android:id="@+id/button9"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="9"
    android:layout_row="1"
    android:layout_column="2"
    android:onClick="onDigitClick" />

<Button
    android:id="@+id/buttonMultiply"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="×"
    android:layout_row="1"
    android:layout_column="3"
    android:onClick="onOperatorClick" />

<!-- Row 3 -->
<Button
    android:id="@+id/button4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="4"
    android:layout_row="2"
    android:layout_column="0"
    android:onClick="onDigitClick" />

<Button
```

```
        android:id="@+id/button5"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="5"
        android:layout_row="2"
        android:layout_column="1"
        android:onClick="onDigitClick" />

<Button
        android:id="@+id/button6"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="6"
        android:layout_row="2"
        android:layout_column="2"
        android:onClick="onDigitClick" />

<Button
        android:id="@+id/buttonMinus"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="-"
        android:layout_row="2"
        android:layout_column="3"
        android:onClick="onOperatorClick" />

<!-- Row 4 -->
<Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="1"
        android:layout_row="3"
        android:layout_column="0"
        android:onClick="onDigitClick" />

<Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="2"
        android:layout_row="3"
        android:layout_column="1"
```

```
        android:onClick="onDigitClick" />

    <Button
        android:id="@+id/button3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="3"
        android:layout_row="3"
        android:layout_column="2"
        android:onClick="onDigitClick" />

    <Button
        android:id="@+id/buttonPlus"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="+"
        android:layout_row="3"
        android:layout_column="3"
        android:onClick="onOperatorClick" />

<!-- Row 5 -->
    <Button
        android:id="@+id/button0"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="0"
        android:layout_row="4"
        android:layout_column="0"
        android:layout_columnSpan="2"
        android:onClick="onDigitClick" />

    <Button
        android:id="@+id/buttonDot"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text "."
        android:layout_row="4"
        android:layout_column="2"
        android:onClick="onDigitClick" />

    <Button
        android:id="@+id/buttonEquals"
        android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
        android:text="="
        android:layout_row="4"
        android:layout_column="3"
        android:onClick="onEqualsClick" />

    </GridLayout>

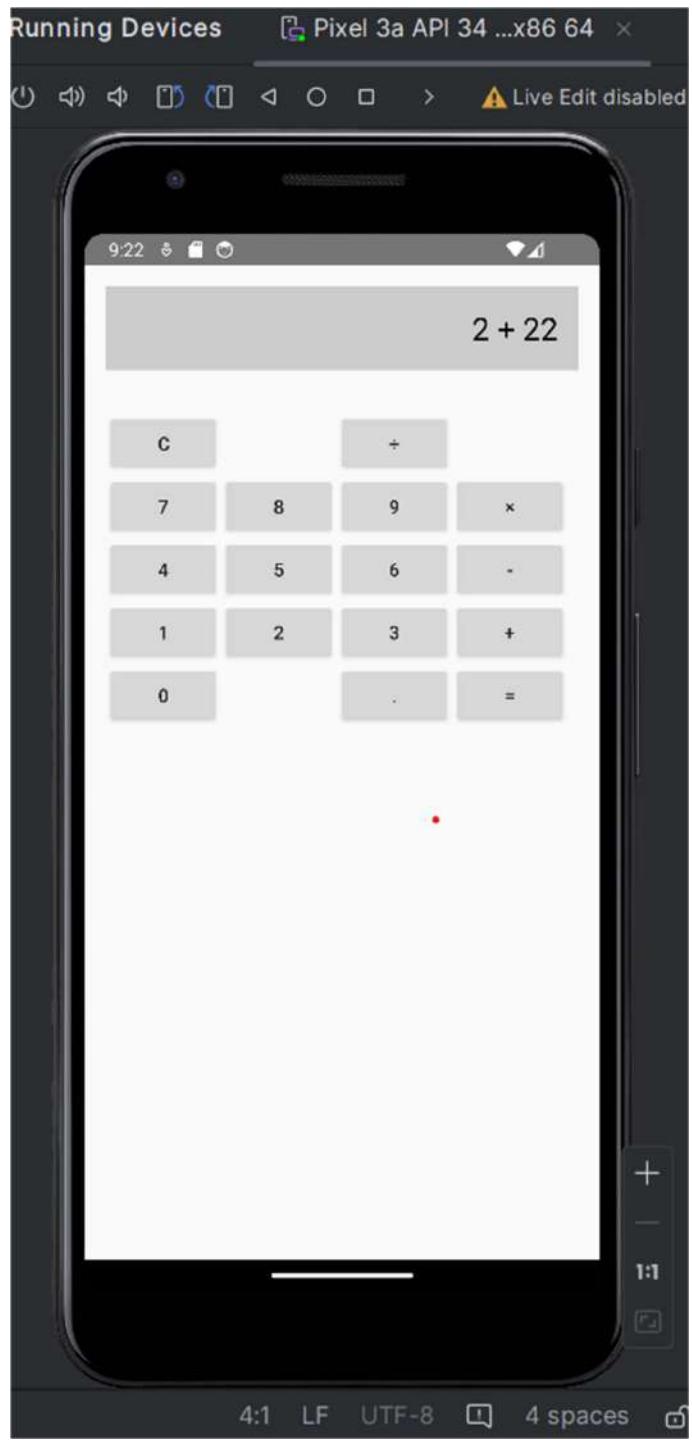
</RelativeLayout>
```

strings.xml:

```
<resources>
    <string name="app_name">Calculator</string>
</resources>
```

OUTPUT:





Experiment :- 8

Aim:- Write a program to demonstrate cellular frequency reuse.

Theory:-

Frequency reuse is the scheme in which allocation and reuse of channels throughout a coverage region is done.

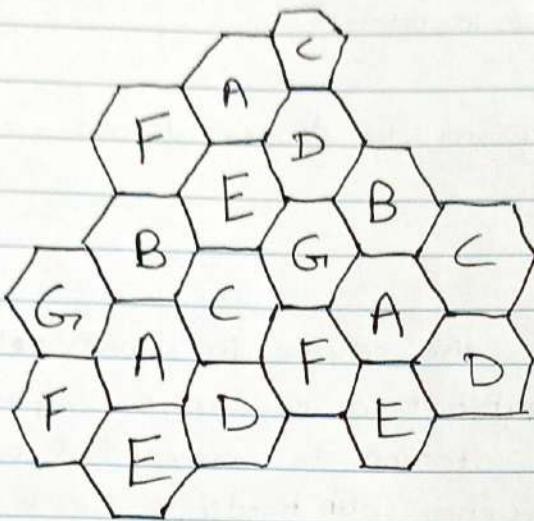
Each cellular base station is allocated a group of radio channels or frequency sub bands to be used ~~with~~ within a small geographic area known as cell.

The shape of the cell is hexagonal.

The process of selecting and allocating the frequency sub-bands for all of the cellular base station within a system is called Frequency reuse or Frequency planning.

Advantages of frequency reuse:-

- 1) It improves Quality of Service (QoS)
- 2) In frequency reuse scheme total bandwidth is divided into different sub-bands that are used by the cells.
- 3) Frequency reuse scheme allows operators to reuse the same frequencies at different sites of the the cell.
- 4) It offers increased network capacity, improved signal quality, enhanced coverage and efficient resource allocation.



Cell with the same letter use same set of channel groups or frequencies sub-band.

To find total number of channels allocated to a cell

S = Total number of duplex channels available to use

k = Channels allocated to each cell ($k \leq S$)

N = Total number of cells or cluster size.

$$\therefore S = kN$$

$$\text{Frequency Reuse Factor} = \frac{1}{N}$$

The value of N

$$N = I^2 + I \times J + J^2$$

I, J = Positive integers indicating position of cell

N = Total number of cells / size of cluster.

If a cluster is replicated or repeated M time then capacity $C = MKN = MS$

(~~$\because S = kN$~~)
 Ans AF

Conclusion :- Implemented a program to demonstrate cellular frequency reuse.

NAME: Arnav Malvia

ROLL NO.: 2103109

BATCH: C23

EXP 8 – Write a program to demonstrate Cellular Frequency Reuse

CODE:

```
#!/usr/bin/python
# Name- Arnav Malvia Batch- C23 Roll No.-2103109

from math import *
from tkinter import *

# Base class for Hexagon shape
class Hexagon(object):
    def __init__(self, parent, x, y, length, color, tags):
        self.parent = parent
        self.x = x
        self.y = y
        self.length = length
        self.color = color
        self.size = None
        self.tags = tags
        self.draw_hex()

    # draw one hexagon
    def draw_hex(self):
        start_x = self.x
        start_y = self.y
        angle = 60
        coords = []
        for i in range(6):
            end_x = start_x + self.length * cos(radians(angle * i))
            end_y = start_y + self.length * sin(radians(angle * i))
            coords.append([start_x, start_y])
            start_x = end_x
            start_y = end_y
```

```

    start_x = end_x
    start_y = end_y
    self.parent.create_polygon(coords[0][0],
        coords[0][1],
        coords[1][0],
        coords[1][1],
        coords[2][0],
        coords[2][1],
        coords[3][0],
        coords[3][1],
        coords[4][0],
        coords[4][1],
        coords[5][0],
        coords[5][1],
        fill=self.color,
        outline="black",
        tags=self.tags)

# class holds frequency reuse logic and related methods
class FrequencyReuse(Tk):
    CANVAS_WIDTH = 800
    CANVAS_HEIGHT = 650
    TOP_LEFT = (20, 20)
    BOTTOM_LEFT = (790, 560)
    TOP_RIGHT = (780, 20)
    BOTTOM_RIGHT = (780, 560)

    def __init__(self, cluster_size, columns=16, rows=10, edge_len=30):
        Tk.__init__(self)
        self.textbox = None
        self.curr_angle = 330
        self.first_click = True
        self.reset = False
        self.edge_len = edge_len
        self.cluster_size = cluster_size
        self.reuse_list = []
        self.all_selected = False
        self.curr_count = 0
        self.hexagons = []
        self.co_cell_endp = []
        self.reuse_xy = []
        self.canvas = Canvas(self,
            width=self.CANVAS_WIDTH,

```

```

        height=self.CANVAS_HEIGHT,
        bg="#4dd0e1")
self.canvas.bind("<Button-1>", self.call_back)
self.canvas.focus_set()
self.canvas.bind('<Shift-R>', self.resets)
self.canvas.pack()
self.title("Frequency reuse and co-channel selection")
self.create_grid(16, 10)
self.create_textbox()
self.cluster_reuse_calc()

# show lines joining all co-channel cells
def show_lines(self):
    # center(x,y) of first hexagon
    approx_center = self.co_cell_endp[0]
    self.line_ids = []
    for k in range(1, len(self.co_cell_endp)):

        end_xx = (self.co_cell_endp[k])[0]
        end_yy = (self.co_cell_endp[k])[1]

        # move i^th steps
        l_id = self.canvas.create_line(approx_center[0], approx_center[1],
                                       end_xx, end_yy)
        if j == 0:
            self.line_ids.append(l_id)
            dist = 0
        elif i >= j and j != 0:
            self.line_ids.append(l_id)
            dist = j
        # rotate counter-clockwise and move j^th step
        l_id = self.canvas.create_line(
            end_xx, end_yy, end_xx + self.center_dist * dist *
            cos(radians(self.curr_angle - 60)),
            end_yy + self.center_dist * dist *
            sin(radians(self.curr_angle - 60)))
        self.line_ids.append(l_id)
        self.curr_angle -= 60

def create_textbox(self):
    txt = Text(self.canvas,
               width=80,
               height=1,

```

```

        font=("Helvetica", 12),
        padx=10,
        pady=10)
txt.tag_configure("center", justify="center")
txt.insert("1.0", "Select a Hexagon")
txt.tag_add("center", "1.0", "end")
self.canvas.create_window((0, 600), anchor='w', window=txt)
txt.config(state=DISABLED)
self.textbox = txt

def resets(self, event):
    if event.char == 'R':
        self.reset_grid()

# clear hexagonal grid for new i/p
def reset_grid(self, button_reset=False):
    self.first_click = True
    self.curr_angle = 330
    self.curr_count = 0
    self.co_cell_endp = []
    self.reuse_list = []
    for i in self.hexagons:
        self.canvas.itemconfigure(i.tags, fill=i.color)

    try:
        self.line_ids
    except AttributeError:
        pass
    else:
        for i in self.line_ids:
            self.canvas.after(0, self.canvas.delete, i)
        self.line_ids = []

if button_reset:
    self.write_text("Select a Hexagon")

# create a grid of Hexagons
def create_grid(self, cols, rows):
    size = self.edge_len
    for c in range(cols):
        if c % 2 == 0:
            offset = 0
        else:

```

```

        offset = size * sqrt(3) / 2
    for r in range(rows):
        x = c * (self.edge_len * 1.5) + 50
        y = (r * (self.edge_len * sqrt(3))) + offset + 15
        hx = Hexagon(self.canvas, x, y, self.edge_len, "#fafafa",
                      "{}{},{}".format(r, c))
        self.hexagons.append(hx)

# calculate reuse distance, center distance and radius of the hexagon
def cluster_reuse_calc(self):
    self.hex_radius = sqrt(3) / 2 * self.edge_len
    self.center_dist = sqrt(3) * self.hex_radius
    self.reuse_dist = self.hex_radius * sqrt(3 * self.cluster_size)

def write_text(self, text):
    self.textbox.config(state=NORMAL)
    self.textbox.delete('1.0', END)
    self.textbox.insert('1.0', text, "center")
    self.textbox.config(state=DISABLED)

#check if the co-channels are within visible canvas
def is_within_bound(self, coords):
    if self.TOP_LEFT[0] < coords[0] < self.BOTTOM_RIGHT[0] \
    and self.TOP_RIGHT[1] < coords[1] < self.BOTTOM_RIGHT[1]:
        return True
    return False

#gets called when user selects a hexagon
#This function applies frequency reuse logic in order to
#figure out the positions of the co-channels
def call_back(self, evt):

    selected_hex_id = self.canvas.find_closest(evt.x, evt.y)[0]
    hexagon = self.hexagons[int(selected_hex_id - 1)]
    s_x, s_y = hexagon.x, hexagon.y
    approx_center = (s_x + 15, s_y + 25)

    if self.first_click:
        self.first_click = False
        self.write_text(
            """Now, select another hexagon such
            that it should be a co-cell of
            the original hexagon."""

```

```

)
self.co_cell_endp.append(approx_center)
self.canvas.itemconfigure(hexagon.tags, fill="green")

for _ in range(6):

    end_xx = approx_center[0] + self.center_dist * i * cos(
        radians(self.curr_angle))
    end_yy = approx_center[1] + self.center_dist * i * sin(
        radians(self.curr_angle))

    reuse_x = end_xx + (self.center_dist * j) * cos(
        radians(self.curr_angle - 60))
    reuse_y = end_yy + (self.center_dist * j) * sin(
        radians(self.curr_angle - 60))

    if not self.is_within_bound((reuse_x, reuse_y)):
        self.write_text(
            """co-cells are exceeding canvas boundary.
Select cell in the center"""
        )
        self.reset_grid()
        break

    if j == 0:
        self.reuse_list.append(
            self.canvas.find_closest(end_xx, end_yy)[0])
    elif i >= j and j != 0:
        self.reuse_list.append(
            self.canvas.find_closest(reuse_x, reuse_y)[0])

    self.co_cell_endp.append((end_xx, end_yy))
    self.curr_angle -= 60

else:
    curr = self.canvas.find_closest(s_x, s_y)[0]
    if curr in self.reuse_list:
        self.canvas.itemconfigure(hexagon.tags, fill="green")
        self.write_text("Correct! Cell {} is a co-cell.".format(
            hexagon.tags))
    if self.curr_count == len(self.reuse_list) - 1:
        self.write_text("Great! Press Shift-R to restart")
        self.show_lines()

```

```

        self.curr_count += 1

    else:
        self.write_text("Incorrect! Cell {} is not a co-cell.".format(
            hexagon.tags))
        self.canvas.itemconfigure(hexagon.tags, fill="red")

if __name__ == '__main__':
    print(
        """Enter i & j values. common (i,j) values are:
        (1,0), (1,1), (2,0), (2,1), (3,0), (2,2)"""
    )
    i = int(input("Enter i: "))
    j = int(input("Enter j: "))
    if i == 0 and j == 0:
        raise ValueError("i & j both cannot be zero")
    elif j > i:
        raise ValueError("value of j cannot be greater than i")
    else:
        N = (i**2 + i * j + j**2)
        print("N is {}".format(N))
freqreuse = FrequencyReuse(cluster_size=N)
freqreuse.mainloop()

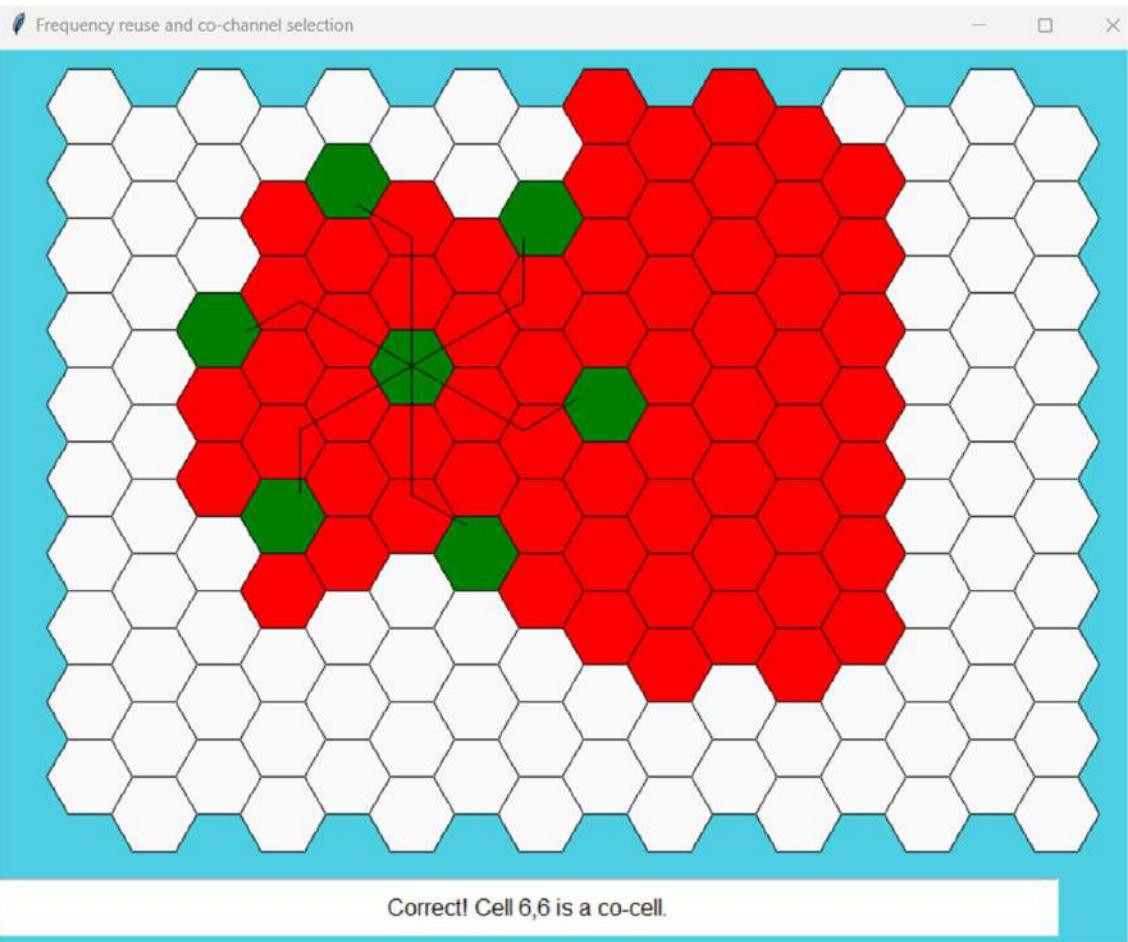
```

OUTPUT:

```

C:\Users\sanja\AppData\Local\Programs\Python\Python311\python.exe C:\Users\sanja\PycharmProjects\pythonProject\CellularFrequencyReuse.py
Enter i & j values. common (i,j) values are:
        (1,0), (1,1), (2,0), (2,1), (3,0), (2,2)
Enter i: 2
Enter j: 1
N is 7

```



Experiment :- 9

Aim:- Write a program to explain concept of DSSS.

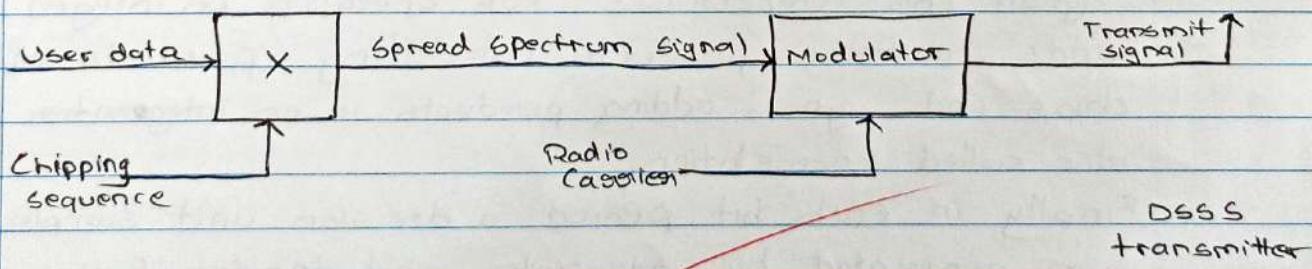
Theory:-

Spread Spectrum includes techniques involving spreading bandwidth needed to transmit data, reducing narrowband interference.

Direct Sequence Spread Spectrum (DSSS) systems take a user bit stream and perform an (XOR) with a so called chipping sequence.

Wireless systems use the sequence 10110111000, called as Barker's code. Barker's code exhibits a good robustness against interference and insensitivity to multipath propagation.

The first step in a DSSS transmitter is the spreading of user data with chipping sequence (digital modulation). The spread signal is then modulated with a radio carrier (radio modulation).

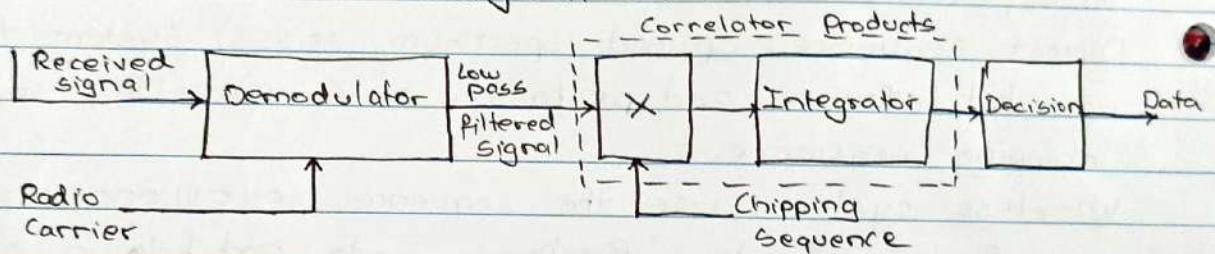


Assuming a user signal with a bandwidth of 1 MHz, spreading it with 11-chip Barker code would result in a signal with 11 MHz bandwidth. The radio carrier then

shifts it to carrier frequency eg. 2.4 GHz. This signal is then transmitted.

DSSS Receiver:

The receiver has to perform inverse functions of two transmitter modulation steps. However noise and multi path propagation require additional mechanisms to reconstruct the original data.



First step in the receiver involves demodulating the received signal. This results in a signal with approximately same bandwidth as original spread spectrum signal.

The receiver has to know original chipping sequence, sequences have to be precisely synchronized since receiver calculates product of chip with incoming signal. This comprises XOR operation. An integrator adds all these products. Calculating products of chips and signal, adding products in an integrator is also called correlation.

Finally in each bit period a decision unit samples sum generated by integrator and decides if this sum represents a binary 1 or a 0.

Eg: Transmission of user data 01

User Data

0

1

XOR with Barker code

10110111000

10110111000

Spread Spectrum

10110111000

01001000111

They are concatenated to 22 digits and sent. At receiver, perform XOR operations on received signal with same Barker's code.

Received Signal

10110111000

01001000111

XOR with Barker Code

10110111000

10110111000

Result

00000000000

~~11111111111~~

Now result is given to integrator which performs sum of products. Sum of products for first part is 0 and next is 11. The decision unit maps sums less than 4 to binary 0, and sums larger than 7 to binary 1.

This constitutes the original user data i.e. 01.

Conclusion :- Implemented a program to demonstrate concept of DSSS.

✓ *Dear*

(*At*)

NAME: Arnav Malvia

ROLL NO.: 2103109

BATCH: C23

EXP 9 – Write a program to explain concept of DSSS

CODE:

```
// Name- Arnav Malvia Batch-C23 Roll No.- 2103109

import java.util.Scanner;

public class DSSS {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        // Get input string from the user
        System.out.print("Enter input string: ");
        String inputString = scanner.nextLine();

        // Generate a spreading code (pseudo-random sequence)
        int[] spreadingCode = generateSpreadingCode(inputString.length());

        // Encrypt the input string using DSSS
        int[] encryptedMessage = encrypt(inputString, spreadingCode);

        // Decrypt the encrypted message
        String decryptedMessage = decrypt(encryptedMessage, spreadingCode);

        // Display input string, encrypted message, and decrypted message
        System.out.println("\nInput String: " + inputString);
        System.out.print("Encrypted Message (Binary): ");
        displayBinary(encryptedMessage);
        System.out.println("Decrypted Message: " + decryptedMessage);

        scanner.close();
    }

    // Generate a pseudo-random spreading code based on the length of the input string
```

```

public static int[] generateSpreadingCode(int length) {
    int[] code = new int[length];
    // For simplicity, using a simple repetitive pattern for the spreading code
    for (int i = 0; i < length; i++) {
        code[i] = i % 2; // Alternating 0s and 1s
    }
    return code;
}

// Encrypt the input string using DSSS
public static int[] encrypt(String input, int[] spreadingCode) {
    int[] encryptedMessage = new int[input.length()];
    for (int i = 0; i < input.length(); i++) {
        encryptedMessage[i] = input.charAt(i) ^ spreadingCode[i]; // XOR operation
    }
    return encryptedMessage;
}

// Decrypt the encrypted message using the spreading code
public static String decrypt(int[] encryptedMessage, int[] spreadingCode) {
    StringBuilder decryptedMessage = new StringBuilder();
    for (int i = 0; i < encryptedMessage.length; i++) {
        decryptedMessage.append((char) (encryptedMessage[i] ^ spreadingCode[i]));
    }
    // XOR operation
    return decryptedMessage.toString();
}

// Display an array in binary format
public static void displayBinary(int[] array) {
    for (int i = 0; i < array.length; i++) {
        System.out.print(Integer.toBinaryString(array[i]));
    }
    System.out.println();
}
}

```

OUTPUT:

```
"C:\Program Files\Java\jdk-18.0.2\bin\java.exe" "-javaagent:C:\Program  
Enter input string: 1101  
  
Input String: 1101  
Encrypted Message (Binary): 110001110000110000110000  
Decrypted Message: 1101  
  
Process finished with exit code 0
```

Experiment :- 10

Aim :- Write a program to implement A3/A5/A8 GSM security algorithms.

Theory :-

GSM offers precise security measures some of which maintains privacy and confidentiality of users identity and data while others ensure only registered users access the network.

Principles of GSM security:

- 1) Access control to SIM card - This is done by use of personal Information Number (PIN) to get access to SIM card.
- 2) Anonymity - Hiding the identity and location of user. This is done by using a TMSI number.
- 3) Authentication - of subscriber so as to ensure only registered and authorised users have access to network.
- 4) Encryption - of data and signal to protect them against interception.

GSM security issues:

- 1) Security is not implemented in fixed part.
- 2) Encryption is only between base station and mobile station length of K_c is 64 bit which is not sufficient enough.
- 3) Authentication is from mobile station to network ~~and vice versa~~ is not possible.
- 4) No measures to maintain integrity is provided.
- 5) Ciphering algorithms are not available for public.

The way the whole thing is carried out is shown as follows according to the following steps:

- At the time of service provisioning the IMSI
- individual subscriber authentication key (k_i)
- authentication algorithm (A3) performed
- cipher key generation algorithm (A8) performed.
- encryption algorithm (A5) performed.

In practice A3 and A8 are generally implemented together (known as A3/A8).

Algorithm of A3:

The first procedure of authentication is carried out by this algorithm. It is used to authenticate the identity of both the subscribers connected in the network. The keys are 128 bit and we use DES algorithm of cryptography here to send the authentication information to each other.

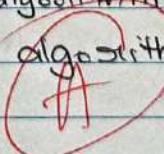
Algorithm of A8:

A8 is a key generation algorithm needed to calculate the session key (k_e). The calculation of k_e depends on k_i and RAND. They are standardised and can be chosen independently by each operator.

Algorithm of A5:

It is a stream cipher used to encrypt over the air transmissions. The ciphering is based on k_e and the frame number. They are specified at international level to enable roaming.

~~Conclusion:- Implemented A3/A5/A8 algorithm and understood the core concepts behind security algorithm in GSM.~~

Ahu 

NAME: Arnav Malvia

ROLL NO.: 2103109

BATCH: C23

EXP 10 – Write a program to implement A3/A5/A8 GSM security algorithm

CODE FOR A3 GSM ALGORITHM:

```
// Name-Arnav Malvia Batch-C23 Roll No.-2103109
```

```
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.util.Random;

public class A3GSMAlgorithm {
    public static void main(String[] args) throws NoSuchAlgorithmException {
        // Generate random 128-bit secret key and message
        long k = generateRandomKey();
        long m = generateRandomMessage();

        // Apply A3 algorithm
        String sres = a3Algorithm(k, m);

        // Print results
        System.out.println("128-bit Secret Key (K in hexadecimal): " +
Long.toHexString(k));
        System.out.println("128-bit Random Message (M in hexadecimal): " +
Long.toHexString(m));
        System.out.println("RES/SRES (in hexadecimal): " + sres);
    }

    // Generate a random 128-bit secret key
    public static long generateRandomKey() {
        Random random = new Random();
        return random.nextLong();
    }

    // Generate a random 128-bit message
    public static long generateRandomMessage() {
```

```

        Random random = new Random();
        return random.nextLong();
    }

    // A3 algorithm implementation
    public static String a3Algorithm(long k, long m) throws
NoSuchAlgorithmException {
    // Convert key and message to byte arrays
    byte[] keyBytes = toByteArray(k);
    byte[] messageBytes = toByteArray(m);

    // Apply hashing function (MD5) to the concatenated key and message
    MessageDigest md = MessageDigest.getInstance("MD5");
    byte[] hash = md.digest(concatenateArrays(keyBytes, messageBytes));

    // Convert the hash to hexadecimal string (SRES)
    StringBuilder sres = new StringBuilder();
    for (byte b : hash) {
        sres.append(String.format("%02X", b));
    }
    return sres.toString();
}

// Convert long value to byte array (big-endian)
public static byte[] toByteArray(long value) {
    byte[] result = new byte[16];
    for (int i = 0; i < 8; i++) {
        result[i] = (byte) (value >> (56 - i * 8));
    }
    return result;
}

// Concatenate two byte arrays
public static byte[] concatenateArrays(byte[] a, byte[] b) {
    byte[] result = new byte[a.length + b.length];
    System.arraycopy(a, 0, result, 0, a.length);
    System.arraycopy(b, 0, result, a.length, b.length);
    return result;
}
}

```

OUTPUT FOR A3 GSM ALGORITHM:

```
"C:\Program Files\Java\jdk-18.0.2\bin\java.exe" "-javaagent:C:\Program F
128-bit Secret Key (K in hexadecimal): 8fdfef58180019fa6
128-bit Random Message (M in hexadecimal): 3abec974be6ef242
RES/SRES (in hexadecimal): 2E53E42500515CD3E2D7D72819077BAB

Process finished with exit code 0
```

CODE FOR A5 GSM ALGORITHM:

```
// Name-Arnav Malvia Batch-C23 Roll No.-2103109
```

```
public class A5GSMAlgorthm {

    private int[] register1 = new int[19];
    private int[] register2 = new int[22];
    private int[] register3 = new int[23];

    public A5GSMAlgorthm() {
        // Initialize registers with arbitrary values
        for (int i = 0; i < 19; i++) {
            register1[i] = 0;
        }
        for (int i = 0; i < 22; i++) {
            register2[i] = 0;
        }
        for (int i = 0; i < 23; i++) {
            register3[i] = 0;
        }
    }

    public void setKey(String key) {
        // Set the key for register 1
        for (int i = 0; i < 19; i++) {
            register1[i] = Character.getNumericValue(key.charAt(i % key.length()));
        }

        // Set the key for register 2
        for (int i = 0; i < 22; i++) {
            register2[i] = Character.getNumericValue(key.charAt((i + 19) %
key.length()));
        }
    }
}
```

```

// Set the key for register 3
for (int i = 0; i < 23; i++) {
    register3[i] = Character.getNumericValue(key.charAt((i + 41) %
key.length()));
}
}

public void generateKeyStream(int numBits) {
    for (int i = 0; i < numBits; i++) {
        int majority = (register1[8] & register2[10]) ^ (register1[8] & register3[10]) ^
(register2[10] & register3[10]);
        int newBit = majority ^ register1[18] ^ register2[21] ^ register3[22];

        shiftRegister(register1);
        shiftRegister(register2);
        shiftRegister(register3);

        System.out.print(newBit);
    }
    System.out.println();
}

private void shiftRegister(int[] register) {
    int feedback = (register[13] ^ register[16] ^ register[17] ^ register[18]) & 0x01;
    for (int i = register.length - 1; i > 0; i--) {
        register[i] = register[i - 1];
    }
    register[0] = feedback;
}

public static void main(String[] args) {
    A5GSMAgorithm a5 = new A5GSMAgorithm();
    String key = "01010101010101010"; // Example key
    String randomMessage = "10101010101010101"; // Example random message
    a5.setKey(key);
    System.out.println("Secret Key: " + key);
    System.out.println("Random Message: " + randomMessage);
    System.out.print("Generated Key Stream: ");
    a5.generateKeyStream(100); // Generate 100 key bits
}
}

```

OUTPUT FOR A5 GSM ALGORITHM:

```
A5GSMAlgorithm x
↑ "C:\Program Files\Java\jdk-18.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2022
Secret Key: 0101010101010101010
Random Message: 10101010101010101
Generated Key Stream: 00011110010101010100001100110000010101010111010111010101010111010101011101000011011000011100011110
Process finished with exit code 0
```

CODE FOR A8 GSM ALGORITHM:

// Name-Arnav Malvia Batch-C23 Roll No.-2103109

```
public class A8GSMAlgorithm {  
  
    // Constants  
    private static final int LFSR_LENGTH = 22;  
    private static final int[] INIT_STATE = {1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0};  
  
    // Variables  
    private int[] lfsr = new int[LFSR_LENGTH];  
  
    // Constructor  
    public A8GSMAlgorithm() {  
        // Initialize LFSR with initial state  
        System.arraycopy(INIT_STATE, 0, lfsr, 0, LFSR_LENGTH);  
    }  
  
    // Clocking function  
    private void clock() {  
        int feedback = lfsr[0] ^ lfsr[1] ^ lfsr[2] ^ lfsr[8];  
        for (int i = LFSR_LENGTH - 1; i > 0; i--) {  
            lfsr[i] = lfsr[i - 1];  
        }  
        lfsr[0] = feedback;  
    }  
  
    // Generate key stream  
    public int[] generateKeystream(int numBits) {
```

```

int[] keystream = new int[numBits];
for (int i = 0; i < numBits; i++) {
    keystream[i] = lfsr[LFSR_LENGTH - 1];
    clock();
}
return keystream;
}

public void printInitialState() {
    System.out.println("Initial State of LFSR:");
    for (int bit : lfsr) {
        System.out.print(bit);
    }
    System.out.println();
}

public static void main(String[] args) {
    A8GSMAgorithm a8 = new A8GSMAgorithm();
    a8.printInitialState();

    int[] keystream = a8.generateKeystream(100); // Generate 100 key bits
    System.out.println("Generated Key Stream:");
    for (int bit : keystream) {
        System.out.print(bit);
    }
}
}

```

OUTPUT FOR A8 GSM ALGORITHM:

```

"C:\Program Files\Java\jdk-18.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Com
Initial State of LFSR:
1010010010010010010010010
Generated Key Stream:
010010010010010010110000011100110001010000001111111101110110001011100111010001001001101011000
Process finished with exit code 0

```

Assignment :- I

Q1 Explain in detail with merits and demerits

- a) Snooping TCP
- b) Mobile TCP
- c) I-TCP (Indirect TCP)

Ans 1 a) Snooping TCP :- It is one of the classical TCP improvement approaches. This approach is designed to solve the end to end semantic loss in I-TCP. The basic concept is to buffer packets close to the mobile node and retransmit them locally if a packet is lost.

Merits: 1) The end to end TCP semantic is preserved - The packet is not acknowledged by the foreign agent. If the foreign agent or base station fails the solution reverts to standard TCP.

2) No modifications at Fixed host - The fixed computer TCP does not need any changes. The majority of the changes are made at the foreign agent.

3) No packet loss during handovers - In case of handover, if only data is not passed to the new foreign agent, there will be a time out at the fixed host and activating retransmission of the packet via mobile IP to a new COA.

Demerits:- 1) The behavior of the wireless link - Snooping TCP does not isolate the behavior of the wireless link or I-TCP. Transmission errors can spread to the correspondent nodes.

2) A mobile node needs additional mechanisms - The use of NACK between the foreign agent and the mobile node requires the mobile node to have additional mechanisms. For arbitrary

mobile nodes, this method is no longer transparent.

3) Encryption at end to end - If such encryption schemes are used end to end between the correspondent node and mobile node, snooping and buffering data can be considered worthless.

b) Mobile TCP :- M-TCP has the same approach goal as I-TCP and snooping TCP, to prevent the sender window from shrinking if bit errors occur or disconnection but not congestion cause current problems. M-TCP wants to improve overall throughput, to lower the delay, to maintain end to end semantics of TCP and to provide a more efficient handover. Additionally it is specially adapted to the problems arising from lengthy or frequent disconnections.

Merits = 1) It maintains TCP end to end semantics. The SH does not send any ACK itself but forwards the ACKs from the MH.

Demerits: 1) As the SH does not act as a proxy as in I-TCP, packet loss on the wireless link due to bit errors is propagated to the sender. M-TCP assumes low bit error rates, which is not valid always a valid assumption.

2) A modified TCP on a wireless link not only requires modifications to the MH protocol software but also new network elements like the bandwidth manager.

c) I-TCP (Indirect TCP) :- It is a split connection protocol proposed by Bakre and Badrinath in 1995. It splits an End to End TCP connection between fixed host and mobile host into two separate connections at an intermediate called MSR.

Merit: 1) No need to make changes in TCP protocol used by computers in fixed network or any other device which do not

need optimisation.

- 2) Transmission errors on wireless links do not propagate in fixed network.
- 3) The optimized TCP is used only on wireless link without affecting stability of internet.

Demerits: 1) Loss of end to end semantic packets already acknowledgements before being delivered.

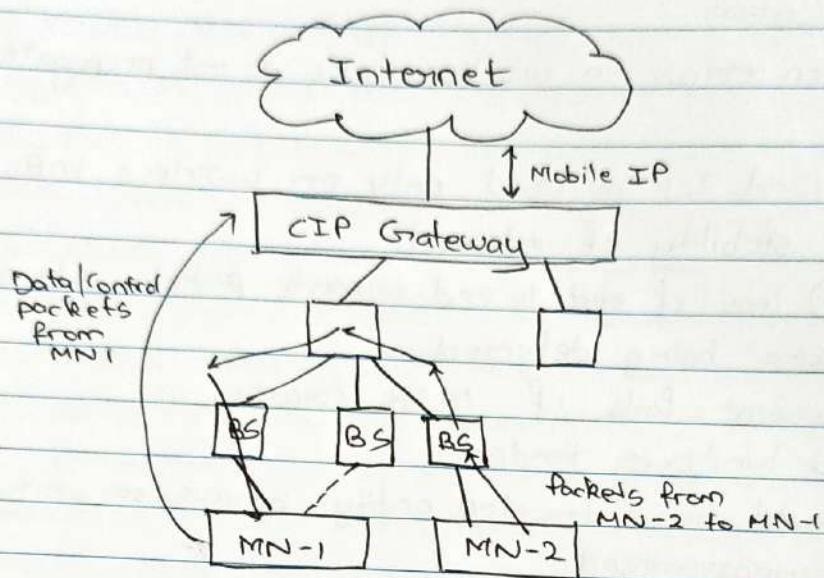
2) Whole scheme fails if MSR crashes.

3) Increased handover tendency.

4) MSR should be a trusted entity. Serious security threat if MSR is compromised.

Q2 Explain cellular (Mobile) IP in detail.

Ans 2 Cellular IP provides local handovers without renewed registration by installing a singular single cellular IP gateway for each domain, which acts to the foreign world as Foreign agent. IP mobility protocols can complement mobile IP by offering fast and consistent handover control in limited geographical areas. Inside this domain each node collects routing information for accessing MNs based on origin of packets sent by the MNs towards the CIPGW. Soft handover is achieved by simultaneous forwarding of packets destined for a mobile node along multiple paths. Thus because of a soft handover, mobile node moving between adjacent cells will temporarily be able to receive packets from the old as well as new BS. The architecture of cellular IP is simple, elegant and is self configuring in most conditions. Cellular IP requires changes to the basic mobile IP protocol and is not transparent to existing systems.



Advantages:-

1) Cellular IP is mostly self configuring and integration of CIPGW with a firewall would help facilitate better mobility administration.

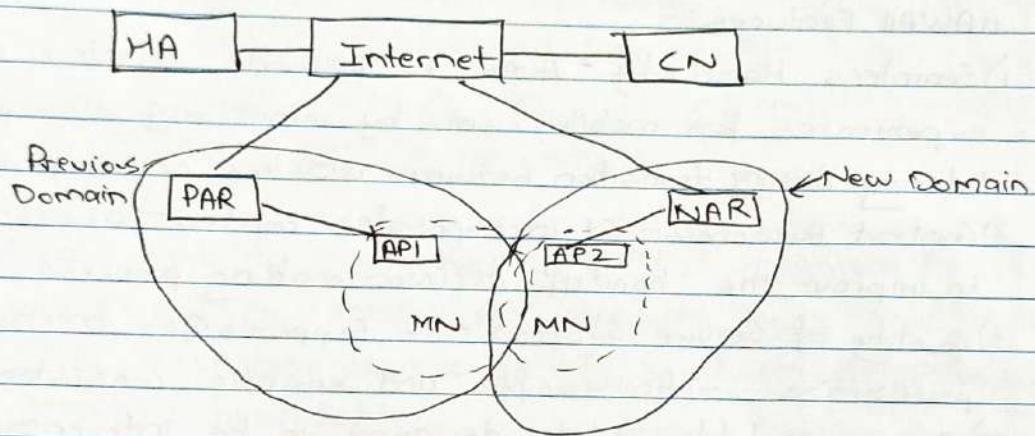
Disadvantages:-

- 1) Transparency - Not transparent to existing systems.
- 2) Efficiency - Additional load introduced by forwarding a packet in multiple path.
- 3) Security - System in network can obtain a copy of all packets with MN's source address to CIPGW.

Q3 Explain MIPv6.

~~Ans 3 Mobile IPv6 is a protocol developed to support mobile devices in IPv6 networks. IPv6 is the latest version of Internet Protocol which provides an expanded address space and other features compared to IPv4. MIPv6 allows mobile devices to move differently between different IPv6 networks while maintaining ongoing communications. This mobility support is crucial for devices such as smartphones,~~

tablets and laptops that frequently change their point of attachment to the internet.



Advantages:-

- 1) Seamless Mobility
- 2) Scalable
- 3) Secure
- 4) Transparent to Applications

Disadvantages:-

- 1) Complex
- 2) Battery Drain
- 3) Packet Overhead
- 4) Handover Latency

Q4 Write short note on HAWAII.

Ans 4 In mobile computing HAWAII (Handoff-Aware Wireless Access Infrastructure) is a framework designed to improve the handoff process for mobile devices as they move between different wireless access points or networks. Handoff also known as handover is a process by which a mobile device transitions its connection from one access point to another while maintaining ongoing communication sessions. The HAWAII framework focuses on enhancing the efficiency and reliability of handoff mechanisms, in wireless networks, particularly in scenarios where mobile devices move between

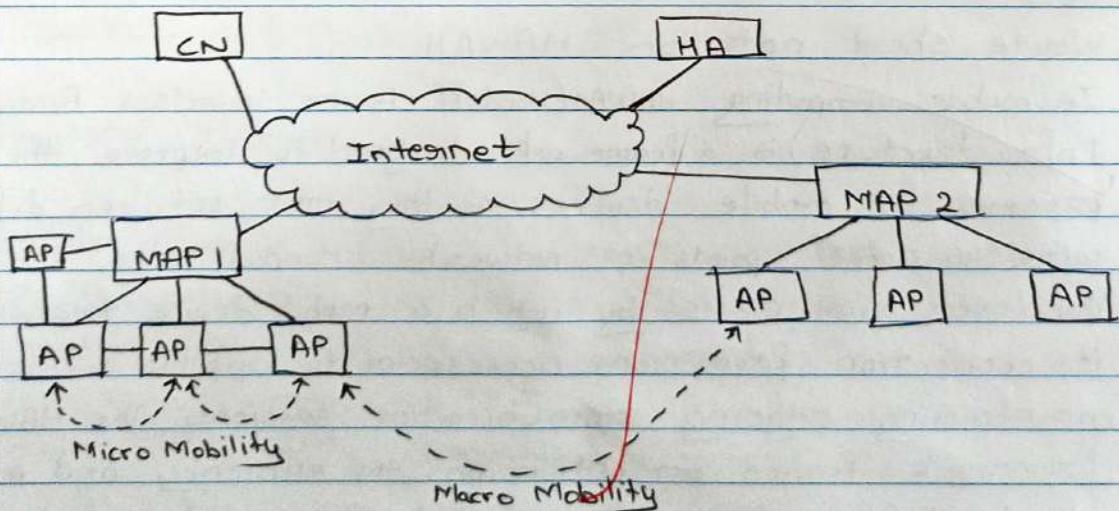
heterogeneous wireless technologies such as Wi-Fi and cellular networks.

HAWAII Features:-

- 1) Seamless Handoff - Aims to provide seamless handoff experiences for mobile users by minimising disruptions and latency during transition between wireless access points.
- 2) Context Awareness - It incorporates context awareness capabilities to improve the handoff decision making process.
- 3) Quality of Service Support - It supports QoS mechanisms to prioritize critical traffic and ensure consistent level of service.
- 4) Interoperability - It is designed to be interoperable with existing wireless networking standards and protocols, allowing it to seamlessly integrate into diverse network infrastructures.
- 5) Resource optimisation - HAWAII optimises the utilization of network resources by efficiently managing handoff related signaling and overhead.

Q5 Explain HMIPv6 in detail.

Ans 5



HMIPv6 Architecture

Hierarchical Mobile IPv6 is an extension of the mobile IPv6 protocol designed to improve handover performance and reduce signaling overhead in large-scale mobile networks. HMIPv6 introduces a hierarchical mobility management scheme that divides the network into multiple domains, allowing for localized mobility management and more efficient routing of packets.

Key components:

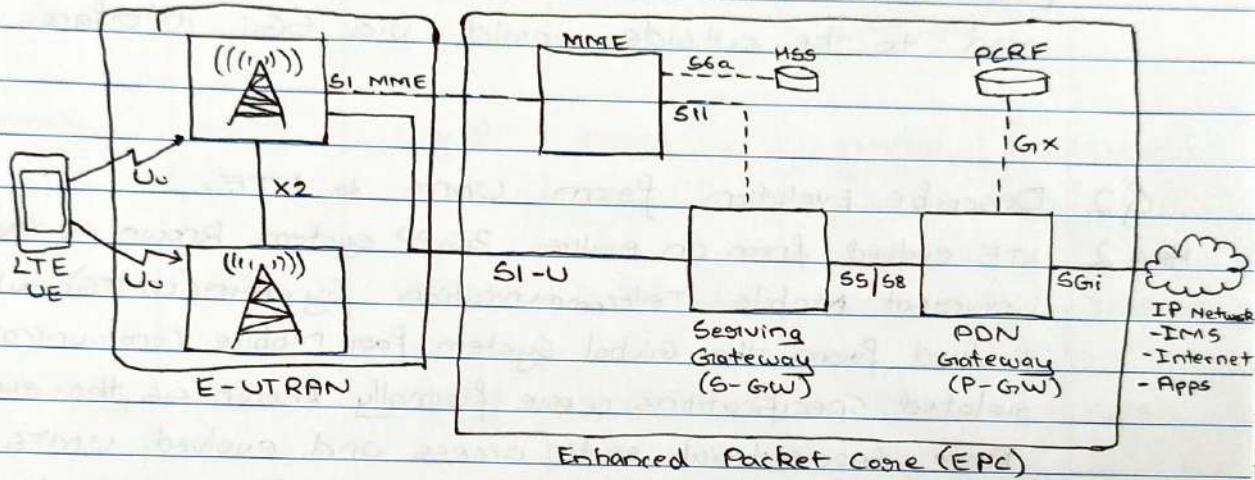
- 1) Hierarchical Mobility Management - HMIPv6 organises the network into multiple mobility domains each managed by a Mobility Anchor Point (MAP). These domains are typically based on geographic or administrative boundaries.
- 2) Mobility Anchor Point (MAP) - The MAP is a router located at the edge of mobility domain. It maintains a binding cache that stores mapping between mobile nodes home addresses and their current care of addresses within the domain.
- 3) Local Mobility Anchor (LMA) - In HMIPv6 the home agent functionality is divided into two components the LMA and the HA (Home Agent). LMA is responsible for managing the mobility of mobile nodes across multiple mobility domains.
- 4) Regional Mobility Agent (RMA) - RMA serves as the intermediary between the MAP's and the LMA providing localised mobility management within a region of mobility domains.
- 5) Handover Optimisation - HMIPv6 optimises handover performance by reducing the signaling overhead associated with MIPv6 handovers. Instead of updating the home agent with every handover event, mobile nodes only need to register their new care of address with the MAP within their current domain.

Also pt

Assignment :- 2

Q1 Draw and explain SAE architecture.
Ans 1

SAE Architecture -



SAE (System Architecture Evolution) is a new network architecture designed to simplify LTE networks. It establishes a flat architecture similar to other IP based communications networks.

SAE uses an eNB and Access Gateway (aGW) and removes the RNC and SGSN from the equivalent 3G network architecture. This allows the network to be built with an All-IP based network architecture.

SAE also includes entities to allow full inter-working with other related wireless technology (WCDMA, WiMAX, WLAN, etc.). These entities can specifically manage and permit the non-3GPP technologies to interface directly with the network.

The high level network architecture of LTE is compromised of following three main components:

- (i) The User Equipment (UE)
- (ii) The Evolved UMTS Terrestrial Radio Access Network (E-UTRAN)
- (iii) The Evolved Packet Core (EPC)

The EPC provides the means to communicate with packet data networks in the outside world such as internet, private corporate networks or the IP multimedia subsystem. Between the ENB and the E-UTRAN there is Uu interface. EPC is connected to the E-UTRAN via the S1 interface and to the outside world via SGi interface.

Q2 Describe Evolution from UMTS to LTE.

Ans 2 LTE evolved from an earlier 3GPP system known as the Universal Mobile Telecommunication System (UMTS), which in turn evolved from the Global System for Mobile Communications. Even related specifications were formally known as the evolved UMTS Terrestrial Radio Access and evolved UMTS terrestrial radio access network. First version of LTE was documented in Release 8 of the 3GPP specifications.

Reasons :

- 1) There was a need to ensure the continuity of competitiveness of the 3G system for the future.
- 2) Users growing demands for higher data rates and quality of service.
- 3) There was a need for Packet Switch optimised system.
- 4) Continued demand for cost reduction.
- 5) Low complexity.
- 6) For avoiding unnecessary fragmentation of technologies for paired and unpaired band operation.

Q3 Compare mobile generations (1G, 2G, 3G, 4G, 5G).

Ans 3 Technology

	1G	2 G	3 G	4 G	5G
Data Rate	2Kbps	64Kbps	Upto 2 Mbps	100Mbps moving 1 Gbps stationary	Higher than 1Gbps
Technology	Analog	Digital	CDMA	Wi-Max Wi-Fi	WWW
Core Network	PSTN	PSTN	Packet NW	Internet	Internet
Multiplexing	FDMA	TDMA/ CDMA	W-CDMA	CDMA	CDMA
Switching	Circuit	Circuit, Packet	Packet	All packet	All packet
Contrast	Mobility	Secure , Mass adoption	Better Internet Experience	Faster broadband internet	Better coverage no dropped calls.

Q4 What are self organising networks?

Ans 4 SON stands for Self Organising Network.

~~It means that just add eNB whenever you want to put and just connect power and switch on, it would configure all of its configuration by itself and makes itself ready for service.~~

SON is like plug and play facility functionality.

Normally when a system operator constructs a network, they go through the following steps.

- i) Network planning
- ii) Bringing the hardware to the locations determined by the Network Planning process.
- iii) Hardware installation.
- iv) Basic configuration
- v) Optimising parameters.

The main goal of SON is to automate those portions of human efforts involved in above mentioned process.

In a more general way SON frame work.

Q 5 Explain VOLTE in detail.

Ans 5 VOLTE stands for Voice Over Long Term Evolution.

It is a digital packet voice service that is delivered over IP via an LTE access network.

When 3GPP started designing the LTE system, prime focus was to create a system which can achieve high data throughput with low latency.

LTE is an IP network and the ability to carry a voice was not given much importance. Therefore for LTE networks to carry traditional circuit switched voice calls, a different solution was required.

This solution to carrying voice over IP in LTE networks is commonly known as "VOLTE". Basically VOLTE system converts voice into data stream, which is then transmitted using the data connection.

VOLTE is based on the the IMS (IP multimedia system)
IMS is an architectural framework for delivering multimedia

communications services such as voice, video and text messaging over IP networks.

Benefits:-

- 1) Provides a more efficient use of spectrum than traditional voice.
- 2) Meets the rising demand for richer, more reliable services.
- 3) Eliminates the need to have voice on one network and data on another.
- 4) Can be deployed in parallel with video calls over LTE and multimedia services including video share, multimedia messaging, chat and file transfer.
- 5) Ensures that video services are fully interoperable across the operator community, just as voice services are.
- 6) Increases handset battery life by 40%.
- 7) Provides rapid cell establishment time.

Ayan

(A+)