NAME: Ananya Prasad

REG NO: 20BCE10093

COURSE: CSE3012

FACULTY: Dr Sandip Mal

DATE: 11-Oct-2022

SEMESTER: Fall 2022-23

SLOT: B11+B12

CLASS NO: 1045

1(b) To display selected radio option in a toast from four radio options.

XML File (activity_main.xml)

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"

  <LinearLayout
      android:layout_width="match_parent"
      android:layout_height="match_parent"
      android:orientation="vertical"

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"></TextView>

    <RadioGroup
        android:id="@+id/Radio"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

      <RadioButton
          android:id="@+id/rb1"
          android:text="Option1" />

      <RadioButton
          android:id="@+id/rb2"
          android:text="Option2" />

      <RadioButton
          android:id="@+id/rb3"
          android:text="Option3" />
```

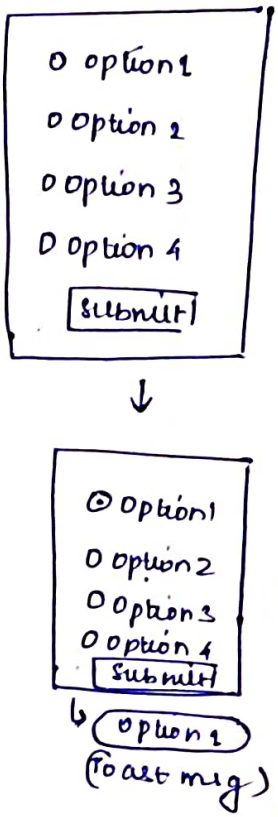
```
    < RadioButton
        android:id="@+id/rb4"
        android:text="Option 4" />
    </RadioGroup>
</LinearLayout>
```

```
</LinearLayout>
```

```
<Button
    android:id="@+id/button"
    android:text="Submit" />
```

ActivityMain.Java

```
package com.example.radiobuttonassignment
import ...
public class MainActivity extends AppCompatActivity {
    Button submit;
    RadioGroup radioGroup;
    RadioButton radioButton;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.layout1);
        radioGroup = findViewById(R.id.Radio);
        submit = findViewById(R.id.button);
        submit.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                int checked = radioGroup.getCheckedRadioButtonId();
                radioButton = findViewById(checked);
                Toast.makeText(getApplicationContext(), radioButton
                .getText().toString(), Toast.LENGTH_LONG).show();
            }
        });
    }
}
```

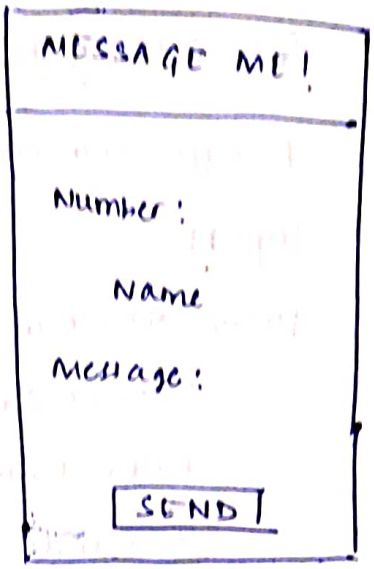


2 (a) Main Activity. Java

```
package com.example.messagetoanumber;
import ...
public class MainActivity extends AppCompatActivity {
    EditText sms;
    Button button
    @Override
    protected void onCreate (Bundle savedInstanceState) {
        super.onCreate (savedInstanceState);
        setContentView (R.layout.activity_main);
        sms = findViewById (R.id.edittext);
        button = findViewById (R.id.button);
        button.setOnClickListener (new View.OnClickListener (){
            @Override
            public void onClick (View view) {
                try { SmsManager smsManager = SmsManager.getDefault();
                      smsManger.sendTextMessage (phone.getText().to
                      toString (), null, null);
                      Toast.makeText (getApplicationContext (), "SMS sent,"
                      Toast.LENGTH_SHORT).show();
                } catch (Exception e) {
                      Toast.makeText (getApplicationContext (), "Failure",
                      Toast.LENGTH_SHORT).show();
                }
            }
        });
    }
}
```

Android Manifest.xml (for permissions)

```
<user-permission android:name="android.permission.
        RECEIVE_SMS" tools:ignore=
"PermissionImpliesUnsupportedChromeOSHardware" />
<user-permission android:name="android.permission.
        SEND_SMS"
    tools:ignore="PermissionImpliesUnsupportedChromeOS
    Hardware" />
```



3(b) Relational Database Management System and Firebase are both database but they both are very different.

	RDBMS	FIREBASE
*	RDBMS classifies data into various tables based on related data types. It is based on domain-specific programming language, often called structured query language. This can be on cloud or on-premise.	* Firebase Realtime Database (Firebase) is a cloud based NOSQL database that syncs and stores data in real-time. It is used to create serverless apps. (Non relational)
*	It has predefined schemas and is table based.	* Firebase has dynamic schema and is not necessarily table based as updation is easy without compromising with design.
*	It is mostly used for relational data and transactions.	* Suitable for realtime applications
*	It is on the system or several clouds provide support managed version.	* Support is given by Google Cloud Platform.
*	Based on concepts of tables	* Based on the concept of collections and documents.
*	Tough to scale up as all are related.	* Easily scalable.
*	Rigid in design	* Very flexible design.

Code to insert data to firebase

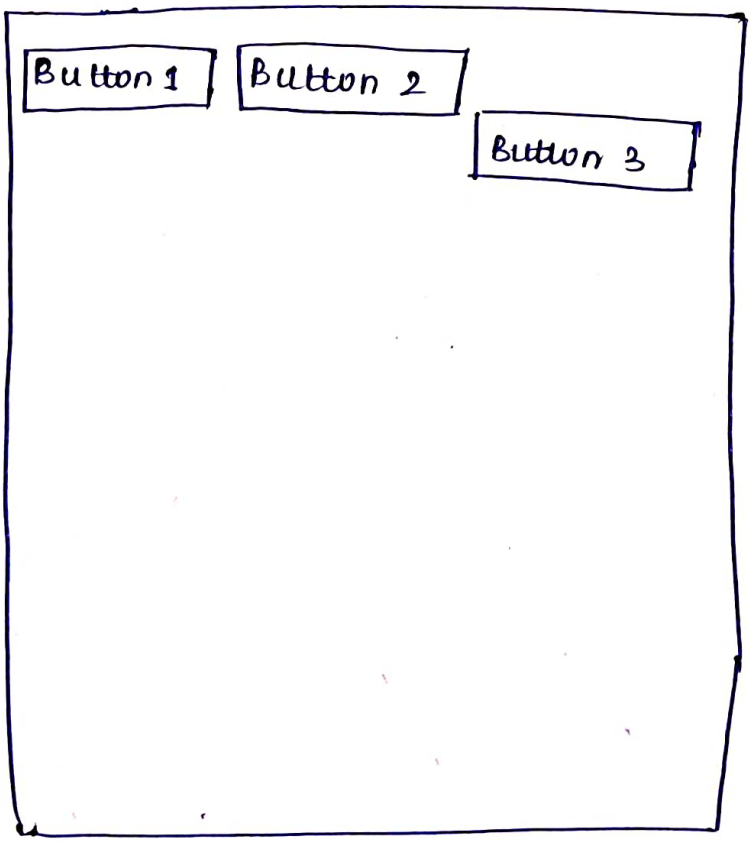
```
send = (Button) findViewById (R.id.sendButton);
sendmessage = (EditText) findViewById (R.id.messageArea);
Firebase.setAndroidContext (this);
final Firebase ref = new Firebase (" firebase chat link ");
Send.setOnclick listener (new View.OnClickListener () {
    @Override
    public void onClick (View v) {
        message message = new message ("hi", send.message.
        getText().toString (), "123");
        ref.push ().setValue (message);
        sendmessage.setText (" ");
    }
}
```

4 xml code example

```
<linearlayout
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical">
    <Button
        android:id="@+id/button1";
        android:text="Button 1"
        android:layout_weight="1">
    </Button>
    <Button
        android:id="@+id/button2";
        android:layout_width="wrap_content";
        android:layout_height="wrap_content";
        android:text="Button 2";
        android:layout_weight="1";
        android:layout_gravity="center";
    </Button>
```

```
<Button
        android:id="@+id/button3";
        android:layout_width="wrap_content";
        android:layout_height="wrap_content";
        android:text="Button 3";
        android:layout_weight="1";
        android:layout_gravity='top right';
        android:layout_margintop=10dp;
    </Button>
</LinearLayout>
```

Output :



(a) layout_gravity : It is used for alignment. It is used for demonstrating alignment settings.

eg top right, top left, center, bottom right etc.

(b) layout_height/width : It is used to specify absolute values for the component height and width respectively

units include ⇒ dp = density independent pixels

sp = scale-independent pixels

pt = 1/72 of inch

px = pixels

mm and in

other constants are match_parent = element occupies all the available width / height and wrap_content = width / height is determined by the content of the element.

(c) layout_orientation : It helps to configure the root layout with they type of orientation to be set for the application, horizontal or vertical.

(d) layout_weight : Free space in an app is distributed proportionally to element weight values using this feature. ~~If we do~~

eg If we want equal space for, use weight = 1. (50-50)

(e) layout margin : Margin can be on all sides at once or only on the sides

eg margin = 50 dp. ⇒ This makes a margin of 50dp around the element.

eg margin top = 10dp ⇒ Makes a margin of 10dp above the element.

5 Program to use SetOnChangeListener function (

```
public void onChange (final EditText Ed, final Runnable FRun) {
    Ed.setOnChangeListener (new OnChangeListener () {
    @ Override
    Public void onFocusChange (View v, boolian hasFocus) {
        if (hasFocusChange (dar.s = " " + Ed.Te.. getText (); }
        else if (! coms (dar. s, " "+ EdTe.. getText ())) {(new Hander(
        post (FRun) ; ] }
    } ) ;
```