Week 1

1) Aim: To write the step-by-step procedure to Install Android Studio.

Steps:

1. Head over to the Android Developers webpage.
2. Click on the link Download Android Studio and SDK tools.
3. Then choose the option to Download Android Studio based on your Operating System
4. For GNU/Linux systems such as Ubuntu, install the software from the relevant software store.

2) Aim: To write the step-by-step procedure to create AVD in Studio

Steps:

1. Open Android Studio.
2. Open Android Virtual Device Manager
3. Click on + Create Virtual Device.
4. Choose a device model and click next.
5. Choose an android software and if you don’t have it you can download it.
6. In this device, you can name the device and click on finish.
7. Now you can use this device to execute the Android Programs.

3) Aim: To write about Android Architecture with neat diagram

Answer:

Android architecture contains different number of components to support any android device needs. Android software contains an open-source Linux Kernel having collection of number of C/C++ libraries which are exposed through an application framework services.

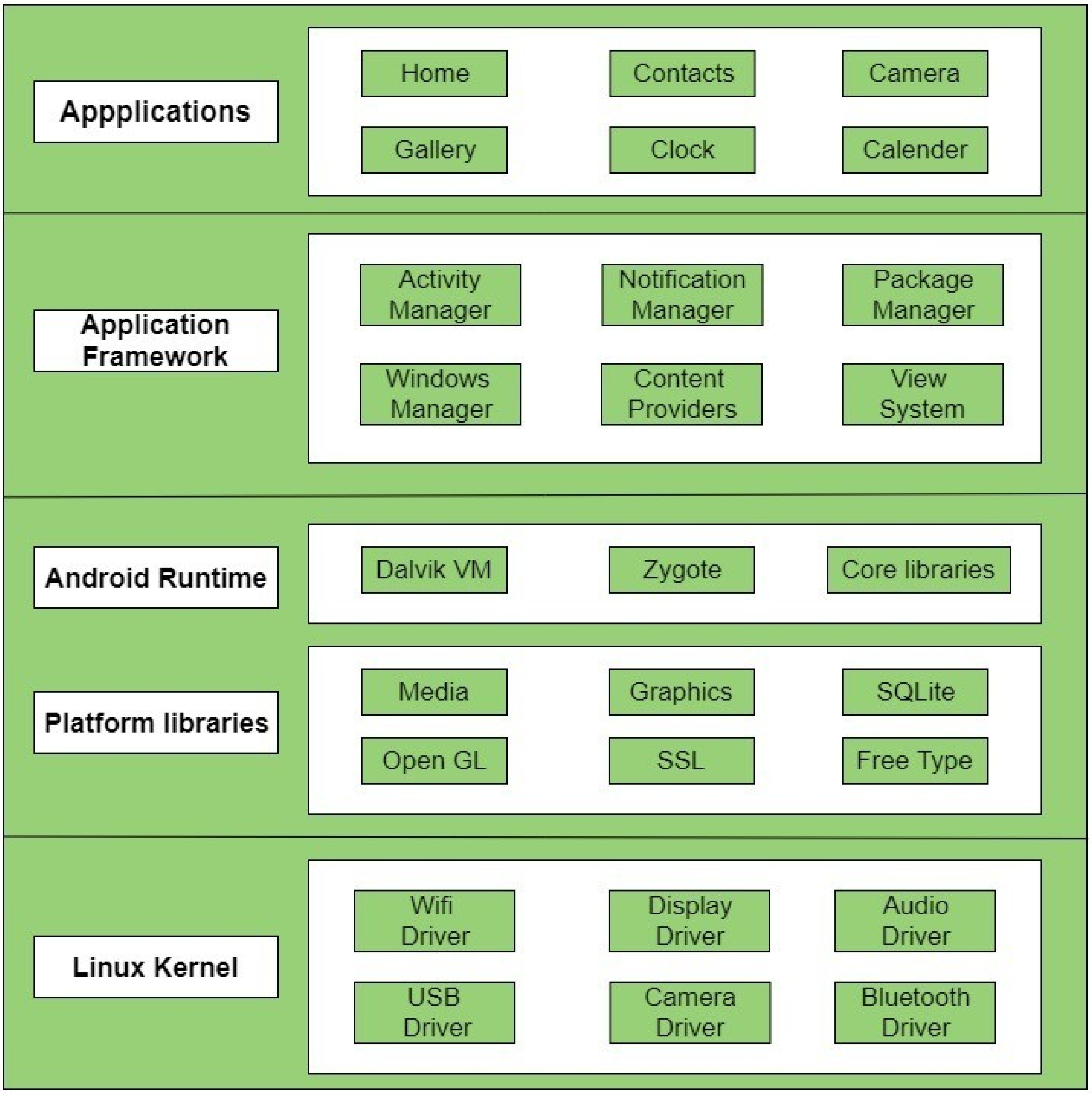
Among all the components Linux Kernel provides main functionality of operating system functions to smartphones and Dalvik Virtual Machine (DVM) provide platform for running an android application.

The main components of android architecture are following:-

Applications
Application Framework
Android Runtime
Platform Libraries

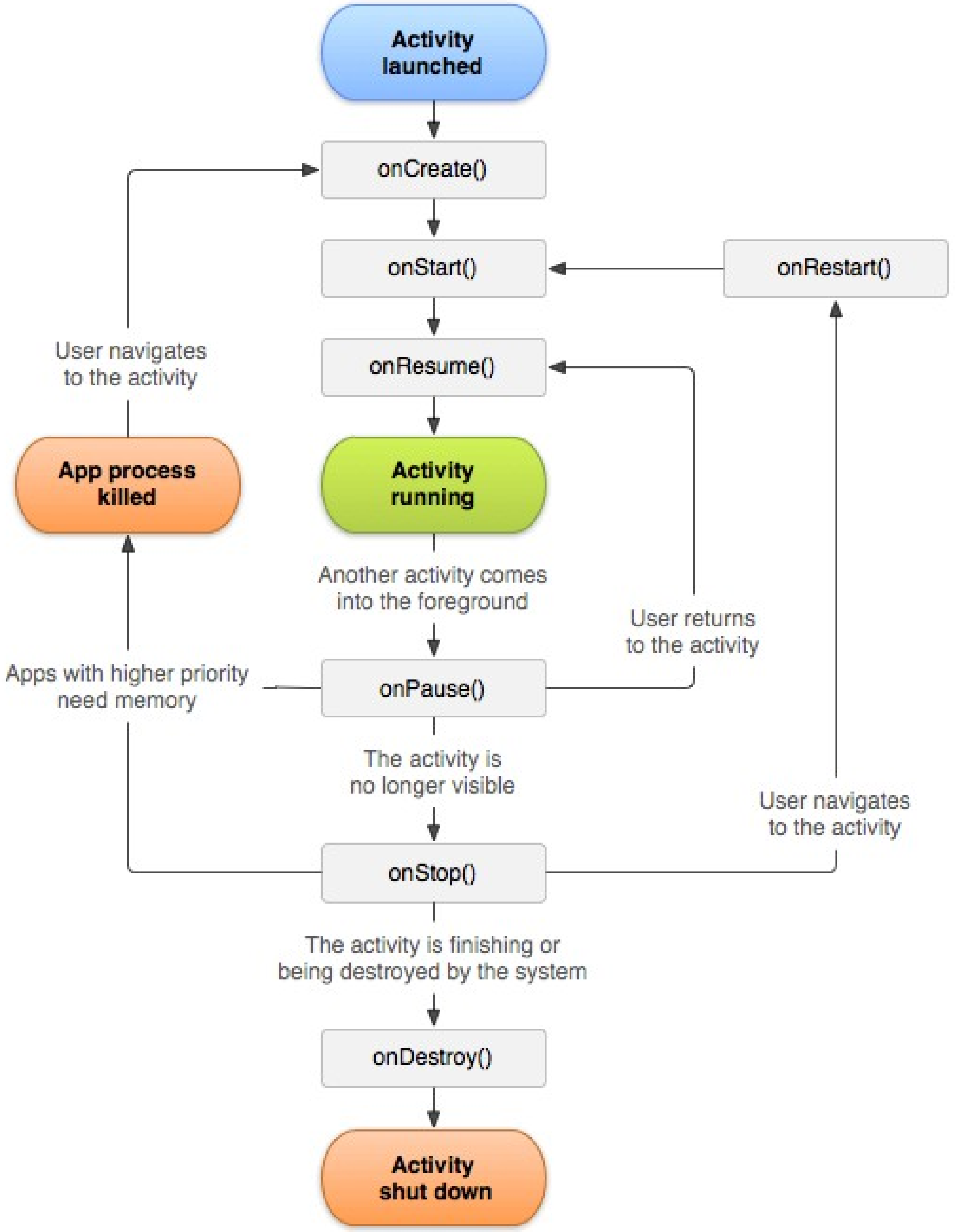
Linux Kernel

Pictorial representation of android architecture with several main components and their sub components –



4) Aim: To draw the Activity Life Cycle Diagram

Diagram:


Activity launched
onCreate()
onStart()
onRestart()
User navigates to the activity
onResume()
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

5) Aim: Demonstrate the use of `Toast` and `Log.d`.

Description:

Toast:

A Toast is a small pill shaped textbox container usually situated at the bottom of the device view.

It is used to display prompts to the user.

Syntax: `Toast.makeText(<Context>, <Message>, <Duration>).show();`

Log:

`Log.d` is a method that prints given message to the system log.

It is very useful in diagnosing issues with the program by having the program print logs during each of its execution steps.

Syntax: `Log.d(<Tag>, <Message>);`

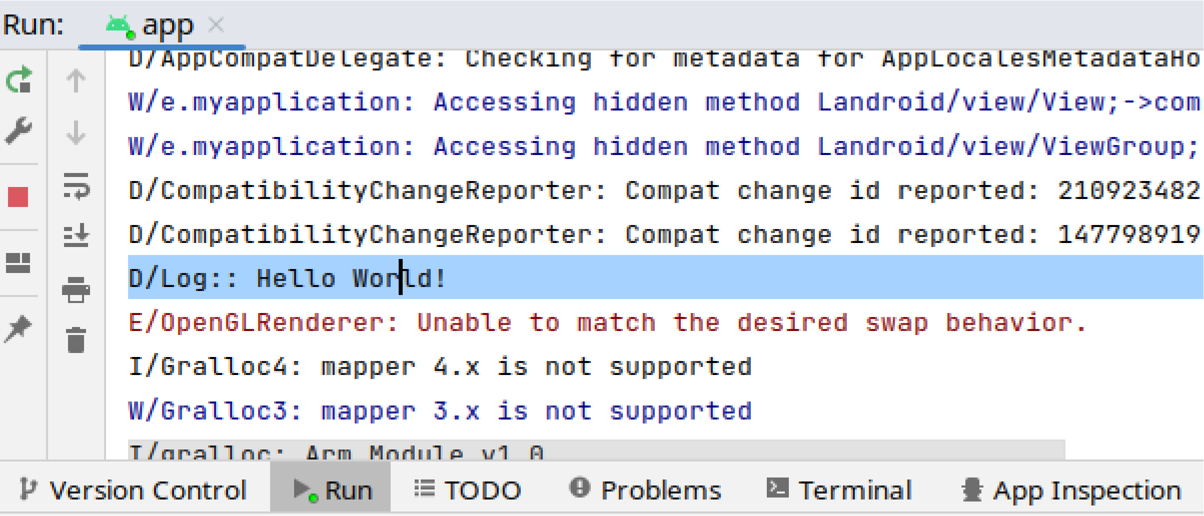
Where, `<Tag>` represents a string that acts as a category for the `<Message>`.

Program:

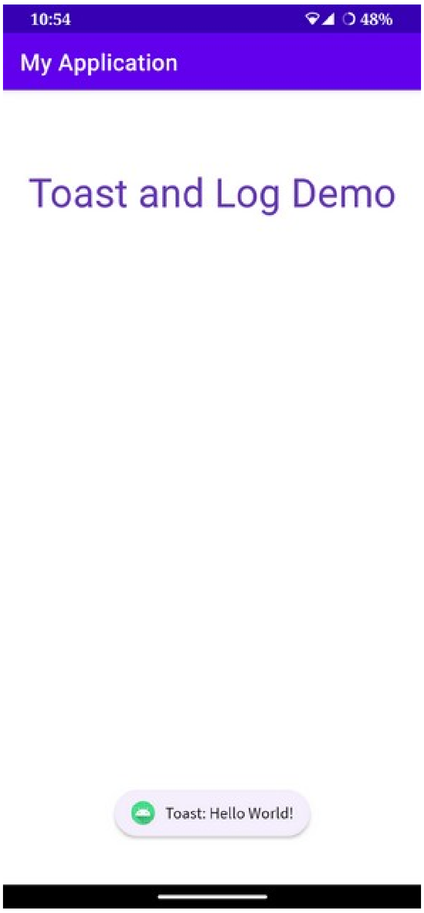
File: MainActivity.java

```java
package com.example.myapplication;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Toast.makeText(this, "Toast: Hello World!",
Toast.LENGTH_SHORT).show();
        Log.d("Log:", "Hello World!");
    }
}
```

Outputs:



```
D/AppCompatDelegate: Checking for metadata for AppLocalesMetadataHo
W/e.myapplication: Accessing hidden method Landroid/view/View;->com
W/e.myapplication: Accessing hidden method Landroid/view/ViewGroup;
D/CompatibilityChangeReporter: Compat change id reported: 210923482
D/CompatibilityChangeReporter: Compat change id reported: 147798919
D/Log:: Hello World!
E/OpenGLRenderer: Unable to match the desired swap behavior.
I/Gralloc4: mapper 4.x is not supported
W/Gralloc3: mapper 3.x is not supported
I/gralloc: Arm Module v1.0
```



Toast and Log Demo

Toast: Hello World!

Week 2

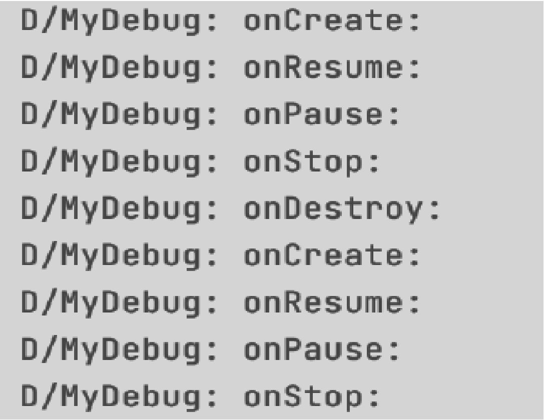
2) **Aim:** To demonstrate life cycle of an activity

Program:

MainActivity.java

```
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
Log.d("Debug", "onCreate: ");
}
@Override
protected void onPause() {
Log.d("Debug", "onPause: ");
super.onPause();
}
@Override
protected void onRestart() {
Log.d("Debug", "onRestart: ");
super.onRestart();
 }
@Override
protected void onResume() {
super.onResume();
Log.d("Debug", "onResume: ");
}
@Override
protected void onStop() {
super.onStop();
Log.d("Debug", "onStop: ");
}
@Override
protected void onDestroy() {
super.onDestroy(); Log.d("Debug2", "onDestroy: ");
}
```

Output:



```
D/MyDebug: onCreate:
D/MyDebug: onResume:
D/MyDebug: onPause:
D/MyDebug: onStop:
D/MyDebug: onDestroy:
D/MyDebug: onCreate:
D/MyDebug: onResume:
D/MyDebug: onPause:
D/MyDebug: onStop:
```

3) **Aim:** Write a program to demonstrate following Layouts

i) LinearLayout

ii) RelativeLayout

iii) GridLayout

iv) AbsoluteLayout

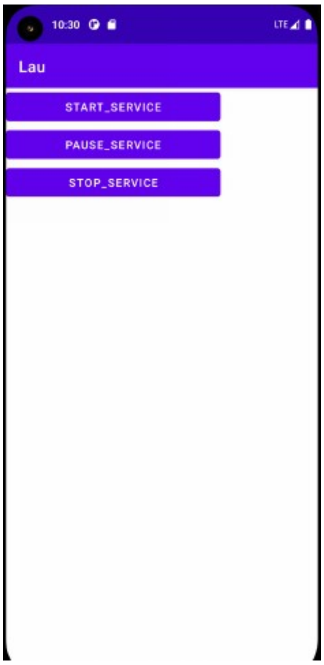
Description:

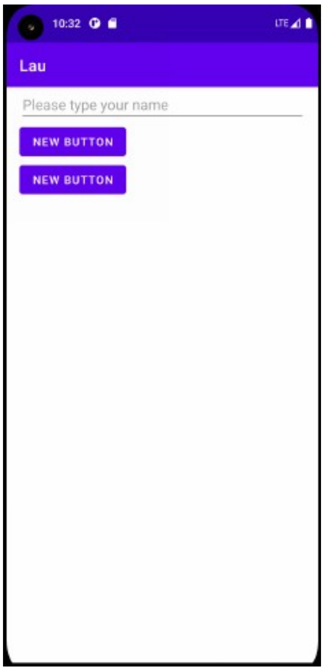
● **Linear layout:** Android LinearLayout is a view group that aligns all children either vertically or horizontally.

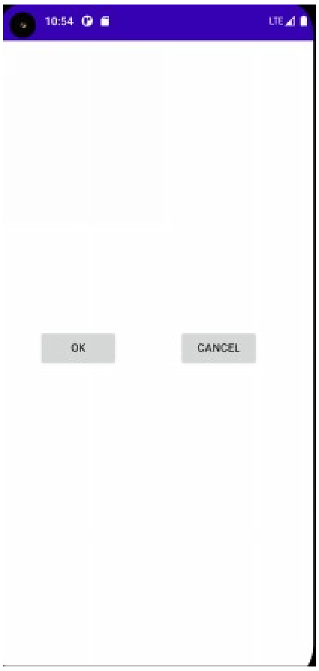
● **Relative layout:** Android RelativeLayout enables you to specify how child views are positioned relative to each other. The position of each view can be specified as relative to sibling elements or relative to the parent.

● **Grid Layout:** Android GridView shows items in a two-dimensional scrolling grid (rows & columns) and the grid items are not necessarily predetermined but they are automatically inserted to the layout using a ListAdapter.

● **Absolute Layout:** An Absolute Layout lets you specify exact locations (x/y coordinates) of its children. Absolute layouts are less flexible and harder to maintain than other types of layouts without absolute positioning.







Week 3

1, 3) Aim: Develop an Android Application that consists of a Login Page with the following:

- Support for multiple languages (Utilizes Strings.xml).

- Utilizes Colors.xml

Description:

Android has support for the separation of content with the code and design. This allows for easy translation into other languages along with a more modular and simplified codebase.

Strings.xml

This XML file consists of string values to be used in the application. These values can be referenced using the “`@string/NAME`” keyword in design part or “`R.string.NAME`” in code.

Colors.xml

This XML file consists of color values to be used in the application. These values can be referenced using the “`@color/NAME`” keyword in design part or “`R.color.NAME`” in code.

This program utilizes the following components:

- One TextView to display the title.

- Two EditText fields to obtain the Username and Password fields respectively.

- Four buttons, one for submit and the other three for setting the language.

Program:

Strings.xml

```
<resources>
    <string name="app_name">Assignment1</string>
    <string name="label_title_en">Enter your Details</string>
    <string name="label_email_en">Email</string>
    <string name="label_password_en">Password</string>
    <string name="label_submit_en"> Submit </string>

    <string name="label_title_hi"> विवरण दर्ज करें </string>
    <string name="label_submit_hi"> प्रस्तत </string>
    <string name="label_password_hi"> पासवर्ड </string>

    <string name="label_title_te">వివరాలను నమోదు చేయండి</string>
    <string name="label_submit_te">సమర్పించు</string>
    <string name="label_password_te">పాస్‌వర్డ్</string>
</resources>
```

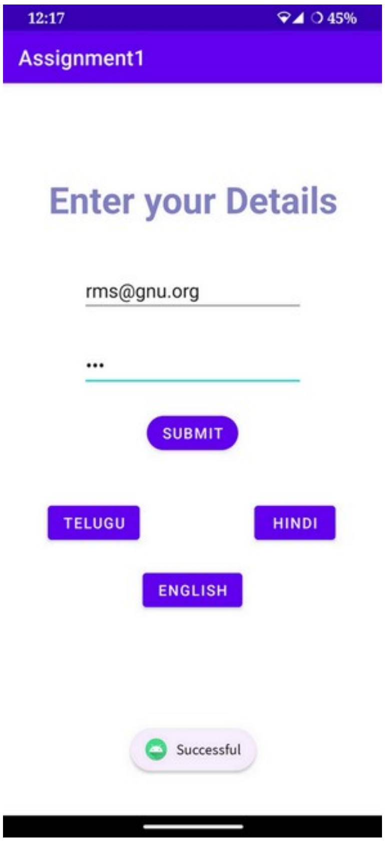
Colors.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="purple_200">#FFBB86FC</color>
    <color name="purple_700">#FF3700B3</color>
    <color name="red">#FFFF0000</color>
    <color name="teal_200">#FF03DAC5</color>
    <color name="teal_700">#FF018786</color>
    <color name="black">#FF000000</color>
</resources>
```

Design Layout:

```
package com.example.lab2;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;     import android.view.View;
import android.widget.Button; import android.widget.TextView;
import android.widget.Toast;  import org.w3c.dom.Text;

public class MainActivity extends AppCompatActivity {
    TextView title, passwd;
    Button login;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        title = findViewById(R.id.Label_Login);
        passwd = findViewById(R.id.Text_Password);
        TextView email = findViewById(R.id.Text_Email);
        int[] buttons = {R.id.button, R.id.button1, R.id.button2};
        login = findViewById(R.id.button_login);
        login.setOnClickListener( v -> {
            String msg;
            if (email.getText().toString().equals("rms@gnu.org") &&
passwd.getText().toString().equals("gnu"))
                msg = "Successful";
            else
                msg = "Invalid Details";
            Toast.makeText(this, msg, Toast.LENGTH_SHORT).show();
        });
        for(int i=0; i<buttons.length; i++) {
            int iVal = i;
            findViewById(buttons[iVal]).setOnClickListener(v -> {
                changeLang(v, iVal);
            });
        }}
    void changeLang(View v, int i){
        int[][] strArr = {
                {R.string.label_title_te, R.string.label_password_te,
R.string.label_submit_te},
                {R.string.label_title_hi, R.string.label_password_hi,
R.string.label_submit_hi},
                {R.string.label_title_en, R.string.label_password_en,
R.string.label_submit_en}
        };
        int[] colors = { R.color.teal_200, R.color.black, R.color.red};
        title.setText(strArr[i][0]);
        title.setTextColor(colors[i]);
        passwd.setHint(strArr[i][1]);
        login.setText(strArr[i][2]);
    }}
```

Outputs:

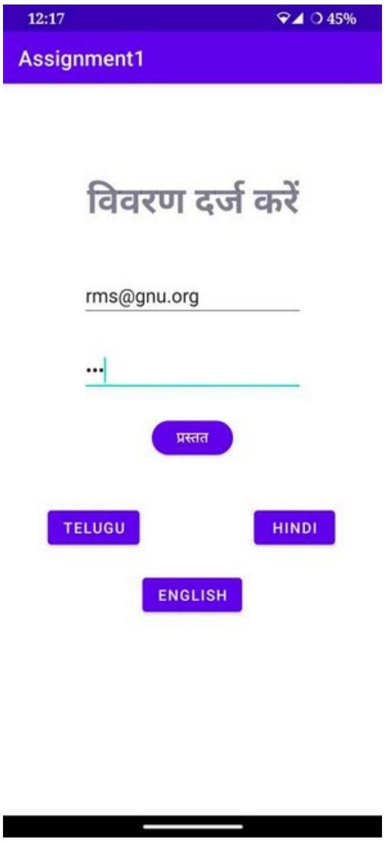


Enter your Details

rms@gnu.org

•••

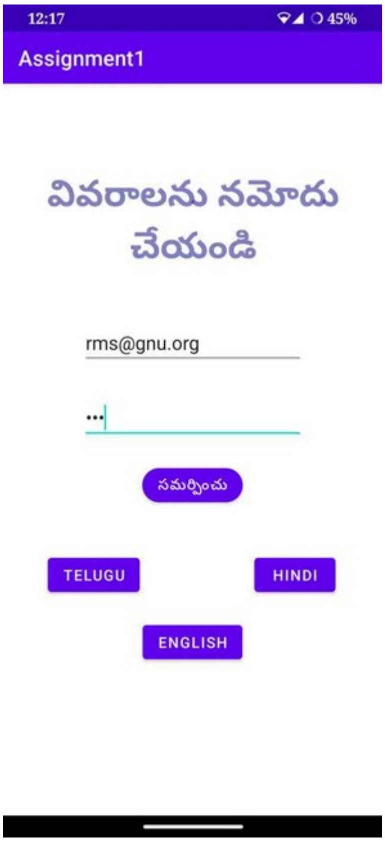
Successful



विवरण दर्ज करें

rms@gnu.org

•••



వివరాలను నమోదు చేయండి

rms@gnu.org

•••

2) Aim: To demonstrate Validation of Input fields along with a dynamically populated Spinner.

Description:

Emails can be validated using Java’s built-in `Pattern` class.

This class has several built-in methods for the validation of various fields such as phone numbers, emails, IP Addresses and so on.

Methods:

```
Patterns.EMAIL_ADDRESS.matcher(<String>.matches())
Patterns.PHONE.matcher(<String>.matches())
Patterns.IP_ADDRESS.matcher(<String>.matches())
```

The above methods check for given pattern in given text.

A Spinner is a drop down menu implementation element in android.

Methods:

```
setAdapter(ArrayAdapter<>);
```

Sets the provided ArrayAdapter to be used as the data to be displayed within the spinner.

```
setOnItemSelectedListener(View.OnSelectedListener{ });
```

Executes given function upon the selection of an item in the spinner.

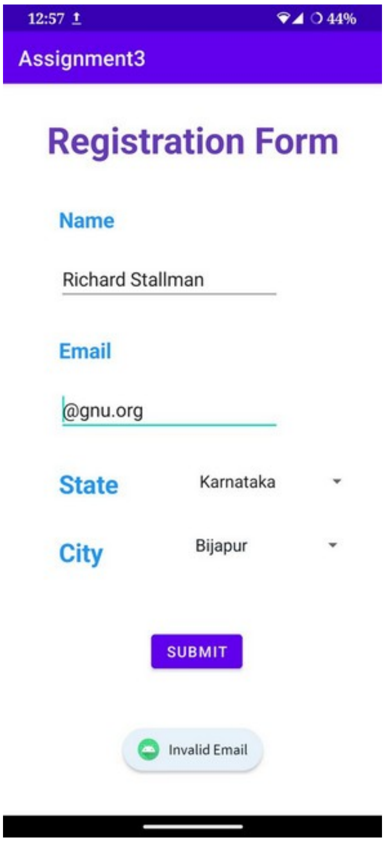
Program:

```
package com.example.assignment3;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;     import android.util.Patterns;
import android.view.View;     import android.widget.AdapterView;
import android.widget.ArrayAdapter; import android.widget.Button;
import android.widget.EditText;     import android.widget.Spinner;
import android.widget.TextView;     import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
```

```
       // initializing the spinners and the string arrays they're about to
hold
        final Spinner spinnerState = findViewById(R.id.spinner_state);
        final Spinner spinnerCity = findViewById(R.id.spinner_city);
        String[] statesArr = {"Telangana", "Andhra Pradesh", "Tamil Nadu",
"Karnataka"};
        String[][] citiesArr = {
                {"Hyderabad", "Warangal"},
                {"Amaravati", "Vishakapatnam"},
                {"Chennai", "Coimbatore"},
                {"Karnataka", "Bijapur"}
        };

        ArrayAdapter<String> states = new ArrayAdapter<String>(this,
android.R.layout.simple_spinner_item, statesArr);
        spinnerState.setAdapter(states);
        spinnerState.setOnItemSelectedListener(new
AdapterView.OnItemSelectedListener() {
            ArrayAdapter<String> cities;
            @Override
            public void onItemSelected(AdapterView<?> parent, View view, int
position, long id) {
                cities = new ArrayAdapter<String>(getApplicationContext(),
android.R.layout.simple_spinner_item, citiesArr[position]);
                spinnerCity.setAdapter(cities);
            }
            @Override
            public void onNothingSelected(AdapterView<?> parent) {
            }
        });
        findViewById(R.id.button).setOnClickListener(v -> {
            validate();
        });
    }

    void validate(){
        EditText name = findViewById(R.id.name), email =
findViewById(R.id.email);
        int[] stars = {R.id.star, R.id.star1};
        if (name.getText().toString().isEmpty() ||
email.getText().toString().isEmpty()) {
            Toast.makeText(getApplicationContext(), "Fill in the mandatory
fields!", Toast.LENGTH_SHORT).show();
            for (int i : stars) {
                TextView t = findViewById(i);
                t.setText("*");
            }
        }
        else if(!Patterns.EMAIL_ADDRESS.matcher(email.getText()).matches())
            Toast.makeText(getApplicationContext(), "Invalid Email",
Toast.LENGTH_SHORT).show();
        else
            Toast.makeText(getApplicationContext(), "All Correct!",
Toast.LENGTH_SHORT).show();
```

```
    }
}
```

Outputs:



Registration Form

Name

Richard Stallman

Email

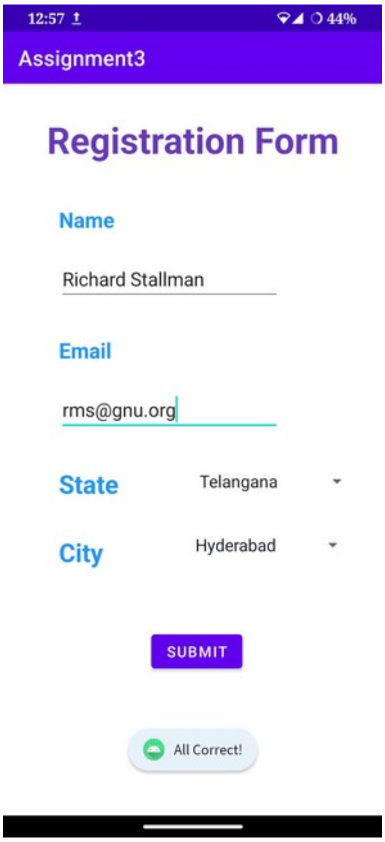
@gnu.org

State Karnataka

City Bijapur

SUBMIT

Invalid Email



Registration Form

Name

Richard Stallman

Email

rms@gnu.org

State Telangana

City Hyderabad

SUBMIT

All Correct!

4) **Aim:** Write an android program to turn on and off the Wi-Fi using on and off button.

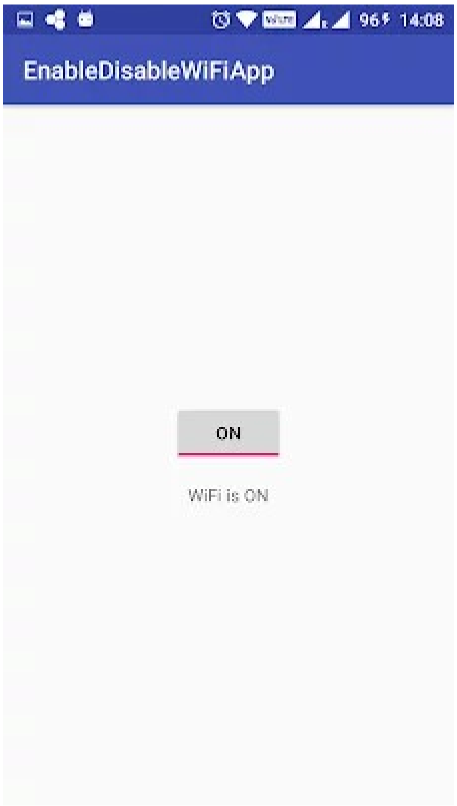
Procedure and Code:

Add the following lines to the manifest.xml file:

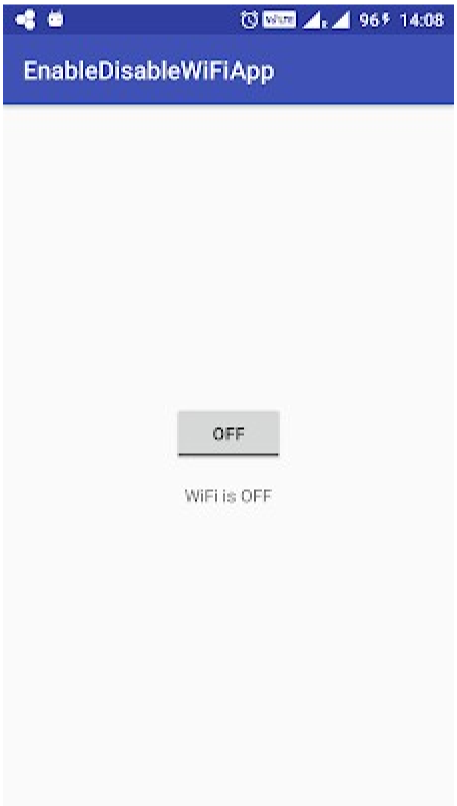
```
    <uses-permission
android:name="android.permission.ACCESS_WIFI_STATE"/>
    <uses-permission
android:name="android.permission.CHANGE_WIFI_STATE"/>
```

```
public class MainActivity extends AppCompatActivity {

    ToggleButton toggleButton;
    TextView textView;

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

        // Getting toggle button and textView from activity_main
        toggleButton = (ToggleButton) findViewById(R.id.toggleButton);
        textView = (TextView) findViewById(R.id.textView);

        // Put listener on toggle button
        toggleButton.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
            @Override
            public void onCheckedChanged(CompoundButton compoundButton, boolean checked) {
                if (checked) {
                    textView.setText("WiFi is ON");
                    WifiManager wifi = (WifiManager) getSystemService(Context.WIFI_SERVICE);
                    wifi.setWifiEnabled(true);
                } else {
                    textView.setText("WiFi is OFF");
                    WifiManager wifi = (WifiManager) getSystemService(Context.WIFI_SERVICE);
                    wifi.setWifiEnabled(false);
                }
            }
        });
        // For initial setting
        if (toggleButton.isChecked()) {
            textView.setText("WiFi is ON");
            WifiManager wifi = (WifiManager) getSystemService(Context.WIFI_SERVICE);
            wifi.setWifiEnabled(true);
        } else {
            textView.setText("WiFi is OFF");
```

```
                WifiManager wifi = (WifiManager)
getSystemService(Context.WIFI_SERVICE);
                wifi.setWifiEnabled(false);
            }
        }
    }
```



WiFi is ON



WiFi is OFF

Week 4

Aim: 1) Write an Android Program to call following Build in application from user defined application using intents

a) Gallery

b) Map

c) Browser

d) Contact Details

e) Phone Call

2. Write an android program to call an activity from another activity.
3. Write an android program to demonstrate intent filter collision.

Description:

An Intent is a passive data structure that holds an abstract description of an operation to be performed.

Intents are classified into two types:

- **Explicit intents** specify which application will satisfy the intent, by supplying either the target app's package name or a fully-qualified component class name. You'll typically use an explicit intent to start a component in your own app, because you know the class name of the activity or service you want to start. For example, you might start a new activity within your app in response to a user action, or start a service to download a file in the background.
- **Implicit intents** do not name a specific component, but instead declare a general action to perform, which allows a component from another app to handle it. For example, if you want to show the user a location on a map, you can use an implicit intent to request that another capable app show a specified location on a map.

  Methods:

  `Intent(Action, Uri):` Creates an Implicit Intent that serves given Activity and URI.

  `Intent(FromActivity, ToActivity):` Creates an Explicit Intent that traverses from given `FromActivity` to given `ToActivity.`

Program:

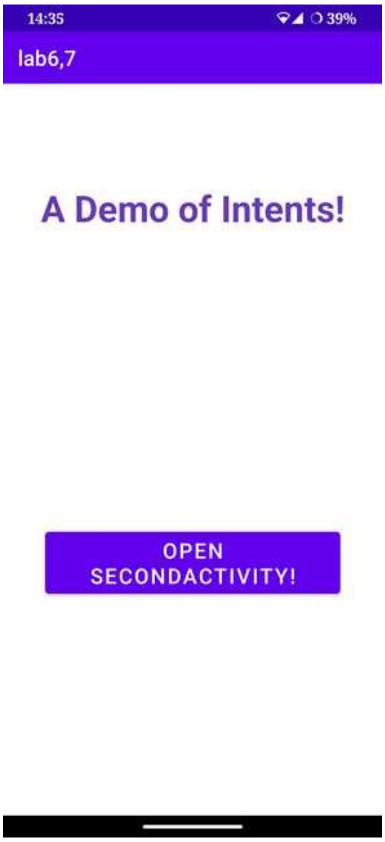
MainActivity.java

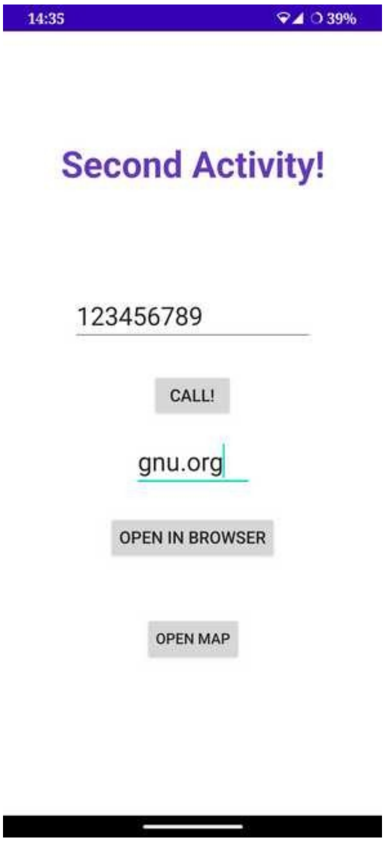
```java
package com.example.lab67;

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

public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        findViewById(R.id.button).setOnClickListener(v -> {
            startActivity(new Intent(this, SecondActivity.class));
        });
    }
}
```

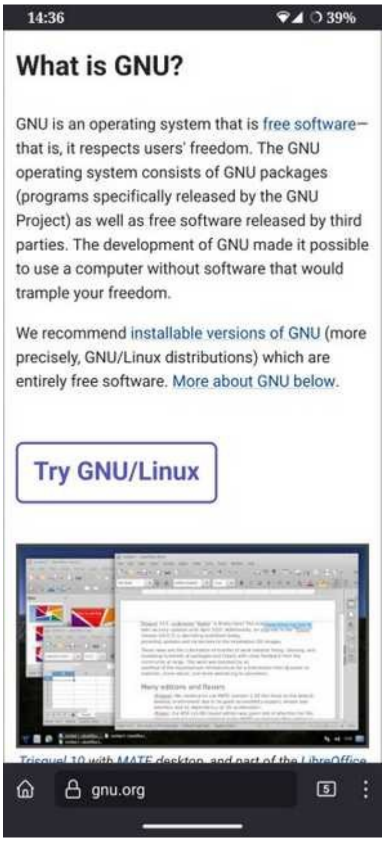
SecondActivity.java

```java
package com.example.lab67;

import android.app.Activity;       import android.content.Intent;
import android.net.Uri;            import android.os.Bundle;
import android.widget.EditText;    import android.widget.TextView;
import androidx.annotation.Nullable;

public class SecondActivity extends Activity {
    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_second);
        findViewById(R.id.button3).setOnClickListener(v -> {
            EditText phone = findViewById(R.id.textView3);
            startActivity(new Intent(Intent.ACTION_DIAL,
Uri.parse("tel:"+phone.getText().toString())));
        });
        findViewById(R.id.button4).setOnClickListener(v -> {
            EditText url = findViewById(R.id.textView4);
            startActivity(new Intent(Intent.ACTION_VIEW,
Uri.parse("https://"+url.getText().toString())));
        });
        findViewById(R.id.button5).setOnClickListener(v -> {
            startActivity(new
Intent(Intent.ACTION_VIEW,Uri.parse("geo:17.401206, 78.481396")));
        });
    }
}
```

Outputs:



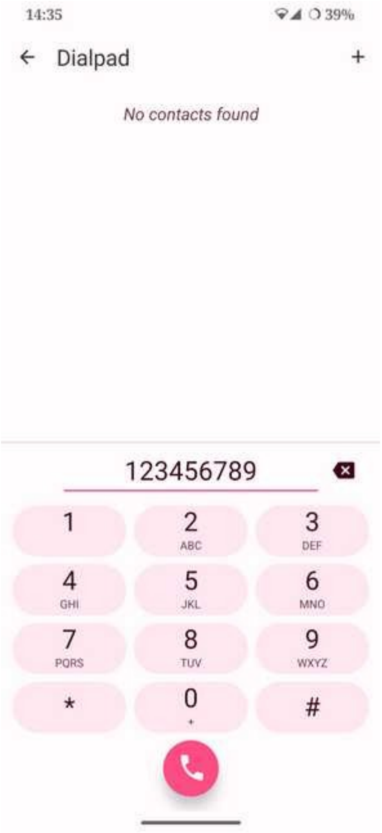


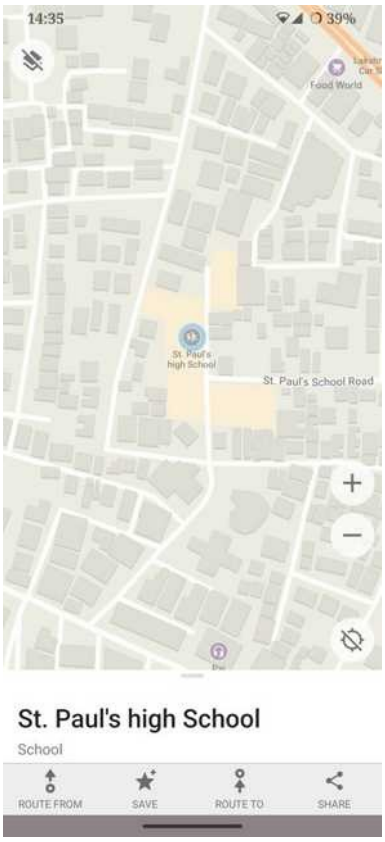


What is GNU?

GNU is an operating system that is free software—that is, it respects users' freedom. The GNU operating system consists of GNU packages (programs specifically released by the GNU Project) as well as free software released by third parties. The development of GNU made it possible to use a computer without software that would trample your freedom.

We recommend installable versions of GNU (more precisely, GNU/Linux distributions) which are entirely free software. More about GNU below.





Aim: 4) Demonstration of Progress Dialog.

Description:

A Progress Dialog is a design element that is used to represent the progress or the extent of completion of a task such as a file download, a loading screen and so on.

In this demonstration, we will be representing two forms of Progress Dialog:

Horizontal: Progress is represented using a horizontal bar that fills with the same rate as the progress of the task that's indicated.

Spin: Progress is shown by a spinning animation. There is no representation of progress in this form.

To simulate progress, a thread with a sleep operation is used.

Methods:

`setMessage(String)`

`setTitle(String)`

`setProgressStyle(String):` Used to set the style of the progress dialog. Such as `ProgressDialog.STYLE_HORIZONTAL` or `ProgressDialog.STYLE_SPINNER`.

`show():` Used to bring the created Progress Dialog into view.

`setMax(int):` Used to set the Maximum value for a Horizontal Progress Dialog.

`setProgress(int):` Used to set progress against value of `setMax()`

`setCancelable(bool):` To set whether the dialog can be cancelled.

`dismiss():` To dismiss and hide the Progress Dialog from view.

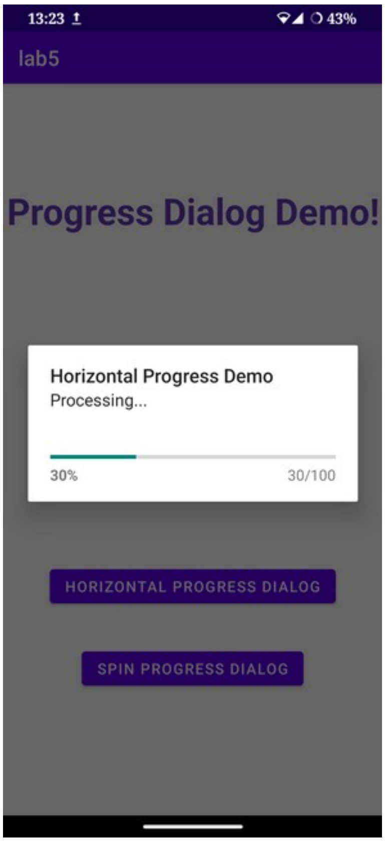
Program:

```
package com.example.lab5;

import androidx.appcompat.app.AppCompatActivity;

import android.app.ProgressDialog;
import android.os.Bundle;
import android.widget.Button;

public class MainActivity extends AppCompatActivity {

    @Override
```

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    Button b1 = findViewById(R.id.button);
    Button b2 = findViewById(R.id.button2);
    b1.setOnClickListener(v -> {
        horizontalProgress();
    });
    b2.setOnClickListener(v -> {
        spinProgress();
    });

}

void spinProgress() {
    ProgressDialog proDia = new ProgressDialog(this);
    proDia.setMax(100);
    proDia.setMessage("Processing...");
    proDia.setTitle("Spin Progress Demo");
    proDia.setProgressStyle(ProgressDialog.STYLE_SPINNER);
    proDia.show();
    proDia.setCancelable(false);
    new Thread(new Runnable() {
        @Override
        public void run() {
            try {
                Thread.sleep(10000);
            } catch (Exception e) {
                e.printStackTrace();
            }
            proDia.dismiss();
        }
    }).start();
}

void horizontalProgress() {
    ProgressDialog proDia = new ProgressDialog(this);
    proDia.setMax(100);
    proDia.setMessage("Processing...");
    proDia.setTitle("Horizontal Progress Demo");
    proDia.setProgressStyle(ProgressDialog.STYLE_HORIZONTAL);
    proDia.show();
    final int totProg = 100;
    new Thread() {
        @Override
        public void run() {
            int jump = 0;
            while (jump < totProg) {
                try {
                    sleep(200);
                    jump += 2;
                    proDia.setProgress(jump);
                } catch (InterruptedException e) {
                    e.printStackTrace();
```

```
                        }
                    }
                    proDia.dismiss();
                }
            }.start();
        }
}
```

Outputs:



13:23 43%

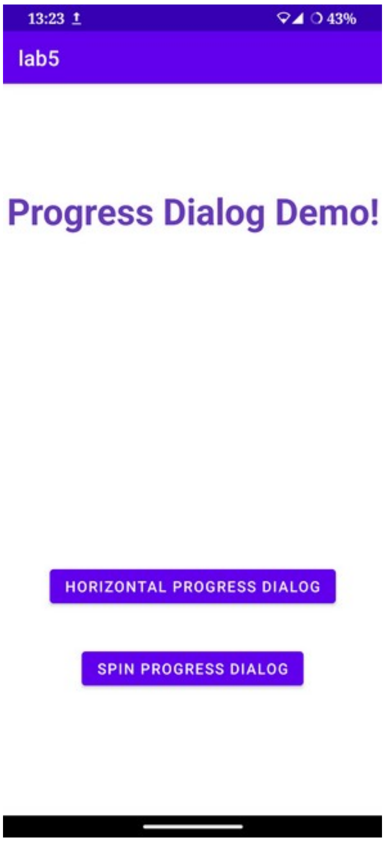
lab5

Progress Dialog Demo!

Horizontal Progress Demo

Processing...

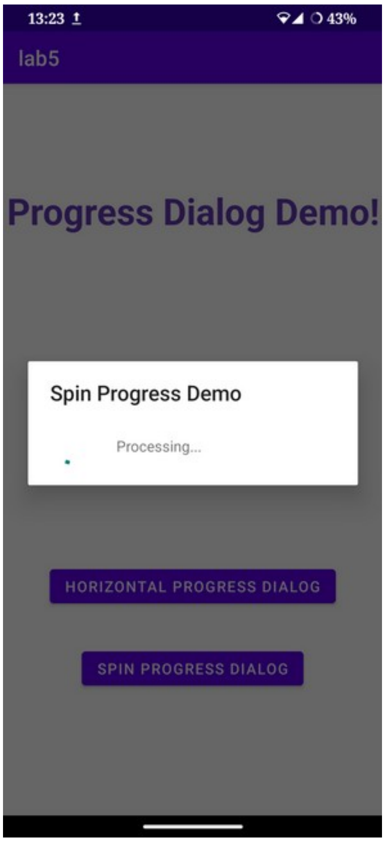
30% 30/100



13:23 43%

lab5

Progress Dialog Demo!



13:23 43%

lab5

Progress Dialog Demo!

Spin Progress Demo

Processing...

Week 5 and 6

Aim: 1) To create an Android Application using Fragments.

Description:

Fragments are used to form reusable portions of an app's user interface.

They can define their own layout, have their own lifecycle and can handle their own input events. But they must be *hosted* by an Activity or another Fragment.

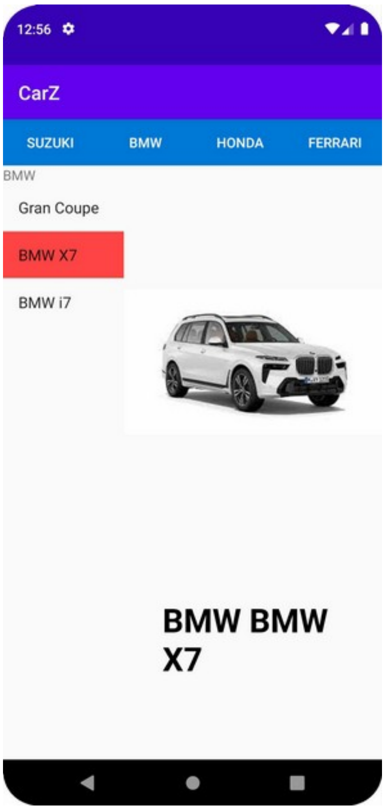
Methods:

MainActivity.java:

```
public class MainActivity extends AppCompatActivity {
    FragmentManager fm;
    FragmentTransaction ft;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        Bundle bundle = new Bundle();
        bundle.putString("company", "Suzuki");
        Fragment listFrag = new ListFragment();
        listFrag.setArguments(bundle);
        fm = getSupportFragmentManager();
        ft = fm.beginTransaction();
        ft.add(R.id.frame1, listFrag);
        ft.commit();

        bundle.putString("name", "Ciaz");
        bundle.putString("company", "Suzuki");
        Fragment detailsFrag = new DetailsFragment();
        detailsFrag.setArguments(bundle);
        fm = getSupportFragmentManager();
        ft = fm.beginTransaction();
        ft.add(R.id.frame2, detailsFrag);
        ft.commit();
    }

    public void companyClick(View v) {
        Button b = (Button)v;
        String company = b.getText().toString();
        Toast.makeText(this, company, Toast.LENGTH_SHORT).show();

        Bundle bundle = new Bundle();
        bundle.putString("company", company);
        Fragment listFrag = new ListFragment();
        listFrag.setArguments(bundle);
        fm = getSupportFragmentManager();
        ft = fm.beginTransaction();
        ft.replace(R.id.frame1, listFrag);
        ft.commit();

    }

}
```

ListFragment.java:

```
public class ListFragment extends Fragment {
    String company;
    ListView list;
    TextView tv;
    HashMap<String, String[]> carModels = new HashMap<>();
    FragmentManager fm;
    FragmentTransaction ft;
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
viewGroup, Bundle savedInstanceState) {
        View v = inflater.inflate(R.layout.listfragment, viewGroup,
false);
        carModels.put("Suzuki", new String[] {"Ciaz", "Dzire",
"Baleno", "Alto"});
        carModels.put("BMW", new String[] {"Gran Coupe", "BMW X7",
"BMW i7"});
        carModels.put("Honda", new String[] {"Jazz", "City",
"Amaze"});
        carModels.put("Ferrari", new String[] {"Ferrari F8", "Roma",
"Portofino"});
        carModels.put("Default", new String[] {"Modle1", "Model2"});
        return v;
    }

    @Override
    public void onViewCreated(View view, Bundle savedInstaceState) {
        Bundle data = getArguments();
        list = view.findViewById(R.id.listView);
        tv = view.findViewById(R.id.textView2);
        if(data != null){
            company = data.getString("company");
        }
        else {
            company = "Default";
        }
        super.onViewCreated(view, savedInstaceState);

        ArrayAdapter<String> ad = new
ArrayAdapter<String>(getActivity(),
android.R.layout.simple_list_item_1, carModels.get(company));
        list.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> adapterView, View
view, int position, long id) {
                String[] models = carModels.get(company);
                Log.d("Success", company+"++++++++
+"+models[position]);
                Bundle bundle = new Bundle();
                bundle.putString("name", models[position]);
                bundle.putString("company", company);
                Fragment detailsFrag = new DetailsFragment();
                detailsFrag.setArguments(bundle);
                fm = getActivity().getSupportFragmentManager();
                ft = fm.beginTransaction();
                ft.replace(R.id.frame2, detailsFrag);
                ft.commit();
            }
        });
        list.setAdapter(ad);
        tv.setText(company);
        list.setSelector(android.R.color.holo_red_light);

        //Setting First Car Details for every company button click
        String[] models = carModels.get(company);
        Bundle bundle = new Bundle();
        bundle.putString("name",models[0]);
        bundle.putString("company", company);
        Fragment detailsFrag = new DetailsFragment();
        detailsFrag.setArguments(bundle);
        fm = getActivity().getSupportFragmentManager();
        ft = fm.beginTransaction();
        ft.replace(R.id.frame2, detailsFrag);
        ft.commit();
    }
}
```


12:56

CarZ

SUZUKI BMW HONDA FERRARI

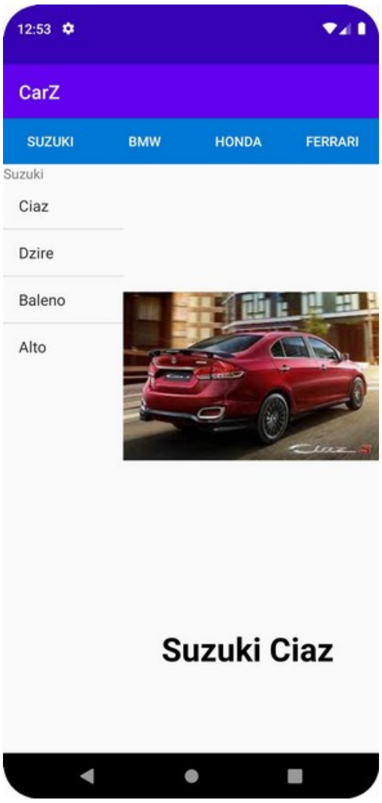
BMW

Gran Coupe

BMW X7

BMW i7

BMW BMW X7


12:53

CarZ

SUZUKI BMW HONDA FERRARI

Suzuki

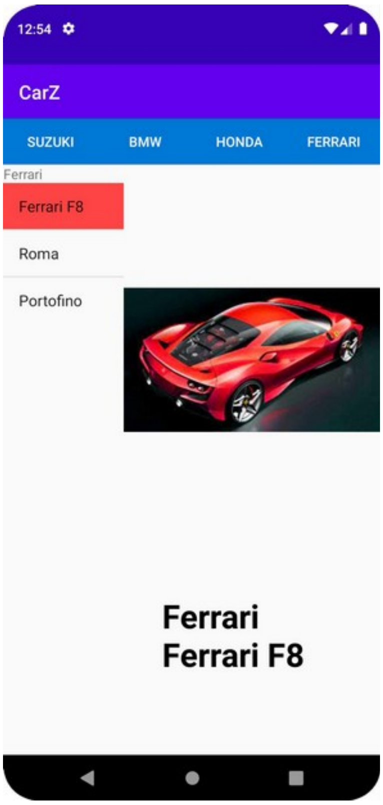
Ciaz

Dzire

Baleno

Alto

Suzuki Ciaz


12:54

CarZ

SUZUKI BMW HONDA FERRARI

Ferrari

Ferrari F8

Roma

Portofino

Ferrari Ferrari F8

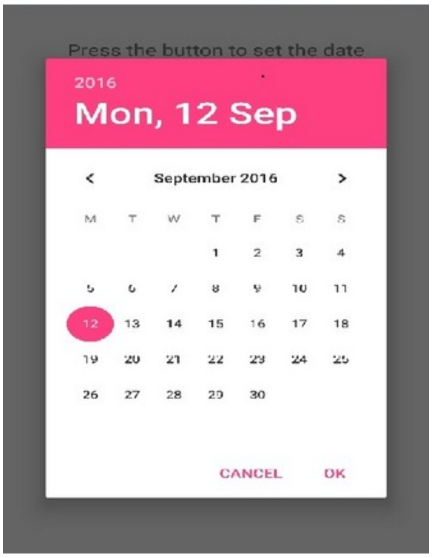
Aim: To demonstrate DatePickerDialog and TimePickerDialog.

Program:

MainActivity.java

```
public class MainActivity extends Activity {
   private DatePicker datePicker;
   private Calendar calendar;
   private TextView dateView;
   private int year, month, day;

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

      dateView = (TextView) findViewById(R.id.textView3);
      calendar = Calendar.getInstance();
      year = calendar.get(Calendar.YEAR);

      month = calendar.get(Calendar.MONTH);
```

```java
      day = calendar.get(Calendar.DAY_OF_MONTH);
      showDate(year, month+1, day);
   }

   @SuppressWarnings("deprecation")
   public void setDate(View view) {
      showDialog(999);
      Toast.makeText(getApplicationContext(), "ca",
         Toast.LENGTH_SHORT)
      .show();
   }

   @Override
   protected Dialog onCreateDialog(int id) {
      // TODO Auto-generated method stub
      if (id == 999) {
         return new DatePickerDialog(this,
            myDateListener, year, month, day);
      }
      return null;
   }

   private DatePickerDialog.OnDateSetListener myDateListener = new
      DatePickerDialog.OnDateSetListener() {
      @Override
      public void onDateSet(DatePicker arg0,
         int arg1, int arg2, int arg3) {
         // TODO Auto-generated method stub
         // arg1 = year
         // arg2 = month
         // arg3 = day
         showDate(arg1, arg2+1, arg3);
      }
   };

   private void showDate(int year, int month, int day) {
      dateView.setText(new StringBuilder().append(day).append("/")
      .append(month).append("/").append(year));
   }
}

```



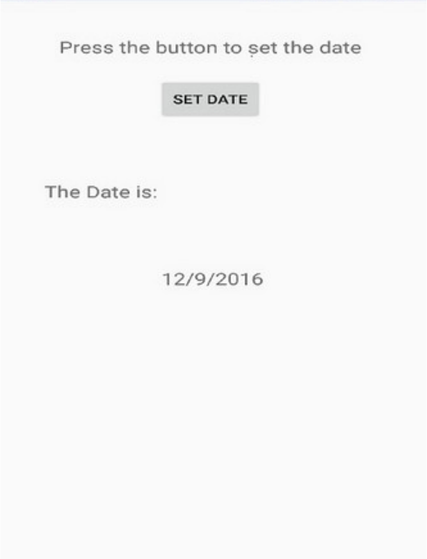
Press the button to set the date

2016

Mon, 12 Sep

September 2016

M	T	W	T	F	S	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		



Press the button to set the date

The Date is:

12/9/2016

Aim: To demonstrate the following components in Android.

- AutoCompleteTextView
- Spinner
- ListView

Description:

AutoCompleteTextView: This is a special form of TextBox where as one provides input, a list of possible auto-completions are suggested.

ListView: A layout consisting of scrollable rows.

Spinner: A drop down menu implementation for android.

Methods:

`setAdapter(ArrayAdapter<>):` Sets the provided ArrayAdapter to be used as the data to be displayed within the spinner.

`setOnItemSelectedListener(View.OnSelectedListener{ }):`

Executes given function upon the selection of an item in the spinner.

`setThreshold(int):` Used to set character threshold for triggering auto completions.

`setAdapter(Adapter):` Used to set the associated Adapter.

`setDropDownViewResource(layout):` Used to set the spinner layout.

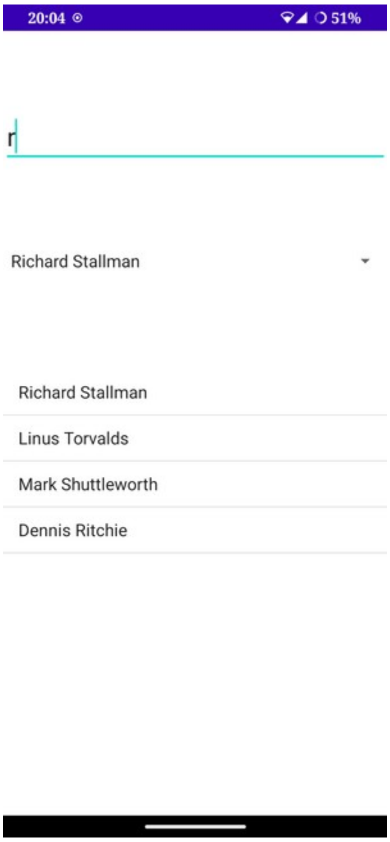
`OnItemSelectedListener():` Used to add a method that shall be invoked upon the selection of an item in the Spinner / List.

Program:

MainActivity.java

```
package com.example.lab10;
import android.app.Activity;                import android.os.Bundle;
import android.view.View;                   import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.AutoCompleteTextView;
import android.widget.ListView;             import android.widget.Spinner;
import android.widget.Toast;                import androidx.annotation.Nullable;

public class MainActivity extends Activity {
    AutoCompleteTextView act;
    Spinner sp;
    ListView l;
    String[] v={"Richard Stallman","Linus Torvalds","Mark
Shuttleworth","Dennis Ritchie"};
    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        act=findViewById(R.id.autoCompleteTextView);
        sp=findViewById(R.id.spinner);
        l=findViewById(R.id.list);
        ArrayAdapter<String> a=new ArrayAdapter<String>(this,
android.R.layout.simple_dropdown_item_1line,v);
        act.setThreshold(1);
        act.setAdapter(a);

        ArrayAdapter<String> b=new
ArrayAdapter<String>(this,android.R.layout.simple_spinner_item,v);

b.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
        sp.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener()
{
            @Override
            public void onItemSelected(AdapterView<?> adapterView, View view,
int i, long l) {
                Toast.makeText(getApplicationContext(),v[i]+"is
selected",Toast.LENGTH_LONG).show();
            }
            @Override
            public void onNothingSelected(AdapterView<?> adapterView) {

            }
        });
        sp.setAdapter(b);
        ArrayAdapter<String> c=new
ArrayAdapter<String>(this,android.R.layout.simple_list_item_1,v);
        l.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> adapterView, View view,
int i, long l) {
                Toast.makeText(getApplicationContext(),v[i]+"is
seleted",Toast.LENGTH_LONG).show();
            }
        });
        l.setAdapter(c);
    }
}
```

Outputs:



r

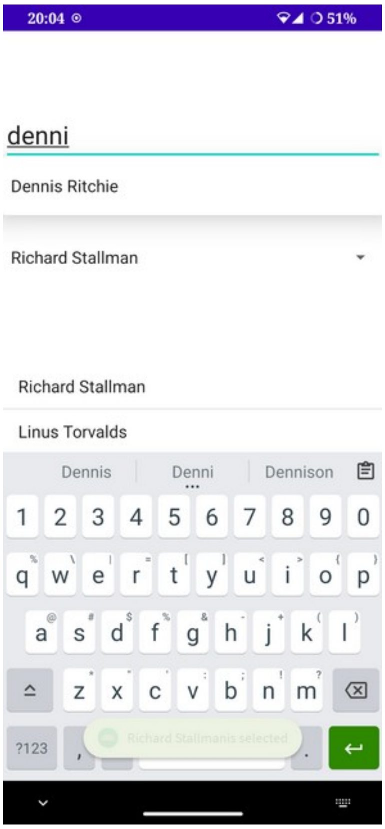
Richard Stallman

Richard Stallman

Linus Torvalds

Mark Shuttleworth

Dennis Ritchie



denni

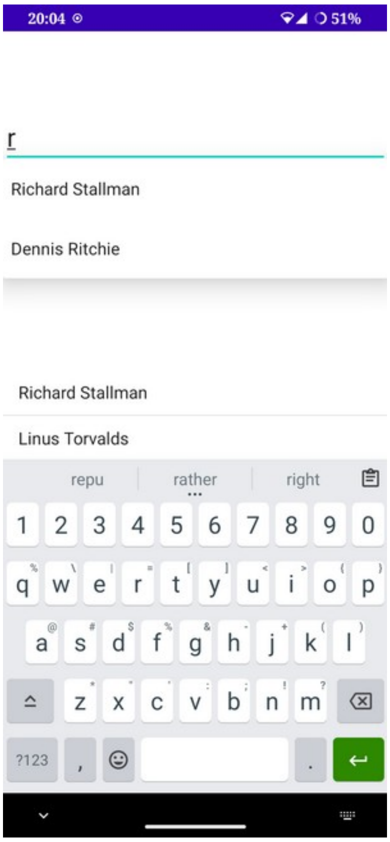
Dennis Ritchie

Richard Stallman

Richard Stallman

Linus Torvalds

Richard Stallmanis selected



r

Richard Stallman

Dennis Ritchie

Richard Stallman

Linus Torvalds

Aim: Demonstrate the use of Custom Adapters by building a Gallery Application.

Description:

Adapters:

An Adapter in Android acts as a bridge between a View / Design element and its underlying data.

There are several predefined Adapter types provided by Android.

In a few cases, an application requires a custom built adapter.

Hence, in such a situation, a Custom Adapter can be built by extending the `BaseAdapter` class.

Methods:

`getView():` Used to return a View to the calling design element after modifying it according to the data.

`getCount():` Used to return the number of data items used in the Adapter.

Gallery:

Gallery is a predefined View provided by Android to implement a Gallery i.e a scrollable list of images.

It has been deprecated since Android API version 16. `HorizontalScrollView` and `ViewPager` are recommended as replacements.

Methods:

`setAdapter():` Used to set the Adapter for the Gallery.

`SetSpacing():` Used to set the spacing between the images.

`SetImageResource():` Used to set the image dynamically.

Program:

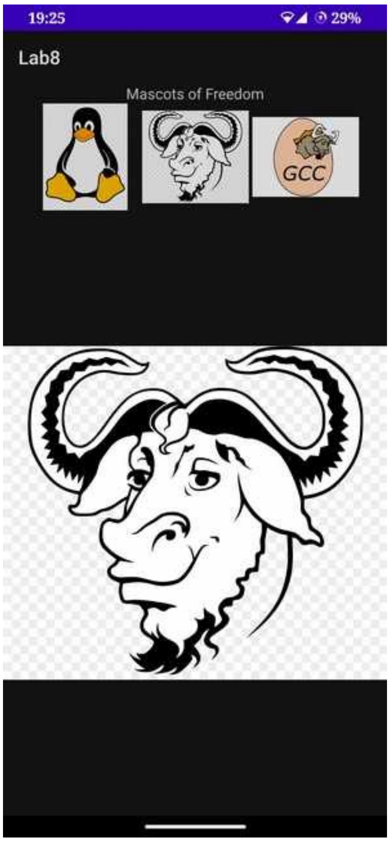
MainActivity.java

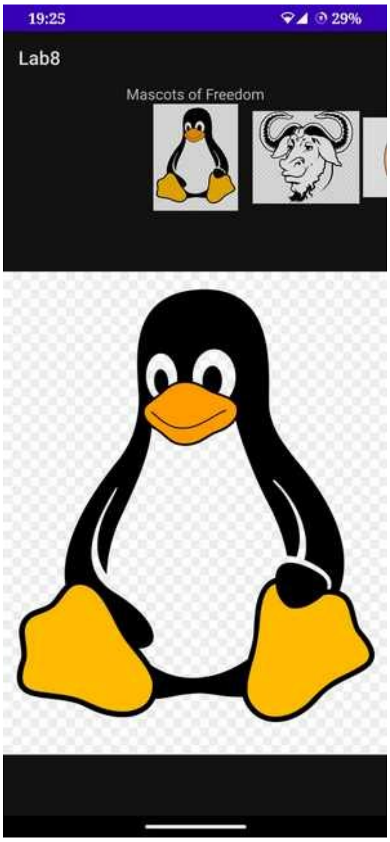
```java
package com.example.lab8;

import android.widget.Gallery;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ImageView;
import androidx.annotation.Nullable;

import com.example.lab8.R;

public class MainActivity extends AppCompatActivity {
    Gallery simpleGal;
    GalleryCustomAdapter galleryCustomAdapter;
    ImageView selectedImage;
    int[] imgs = {R.drawable.img,R.drawable.img_1,R.drawable.img_2};
    @Override
    protected void onCreate(@Nullable Bundle saved){
        super.onCreate(saved);
        setContentView(R.layout.gallery);
        simpleGal = findViewById(R.id.gal);
        selectedImage = findViewById(R.id.igmv);
        galleryCustomAdapter = new
GalleryCustomAdapter(getApplicationContext(),imgs);
        simpleGal.setAdapter(galleryCustomAdapter);
        simpleGal.setSpacing(10);
        simpleGal.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> adapterView, View view,
int i, long l)
            {
                selectedImage.setImageResource(imgs[i]);
            }
        });
    }
}
```

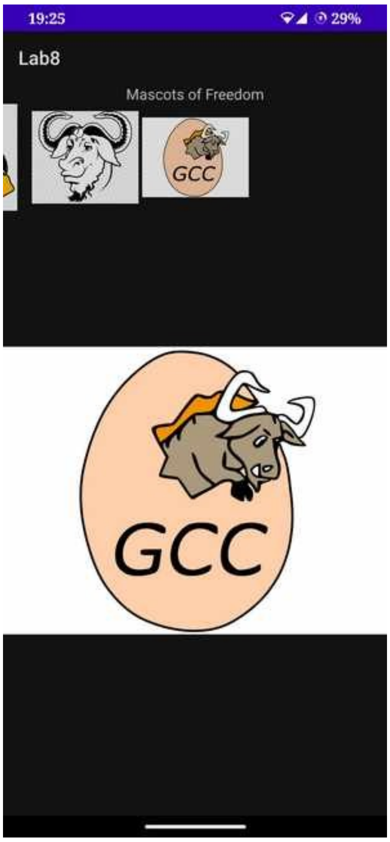
GalleryAdapter.java

```java
package com.example.lab8;
import android.content.Context;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.Gallery;
import android.widget.ImageView;

class GalleryCustomAdapter extends BaseAdapter {
    private final Context context;
    private final int[] imgs;
    public GalleryCustomAdapter(Context c, int[] imgs){
        context = c;
        this.imgs = imgs;
    }
    public int getCount(){
        return imgs.length;
    }
    public Object getItem(int pos){
        return imgs[pos];
    }
    public long getItemId(int pos){
        return pos;
    }
    public View getView(int position, View picView, ViewGroup parent){
        ImageView imageView = new ImageView(context);
        imageView.setImageResource(imgs[position]);
        imageView.setLayoutParams(new Gallery.LayoutParams(300,300));
        return imageView;
    }
}
```

Outputs:







Week 6

Aim: Write an Android Application to generate Notifications.

- Simple Notification
- Missed call Notification
- Reminder Notification

Description:

Notifications on Android are managed by the `NotificatonManager` class.

The general procedure is as follows:

- Create a `NotificationManager` object.

- For Android versions 8 and above, create a `NotificationChannel`.

- Use the `NotificationManager.builder` class to set Title, Context, Message text, Message SubText, Icon and Intents, if any.

Methods:

- `getSystemService(Context.NOTIFICATION_SERVICE):` Used to create the `NotificationManager` Object.
- `NotificationChannel("channel_id","channel_name", NotificationManager.IMPORTANCE_DEFAULT):` Used to create Notification Channel.
- `NotificationCompat.Builder(getApplicationContext(), "channel_id"):` Creation of builder class.
- `builder.setContentTitle();`
- `builder.setSmallIcon();`
- `builder.setContentText();`
- `builder.setSubText();`

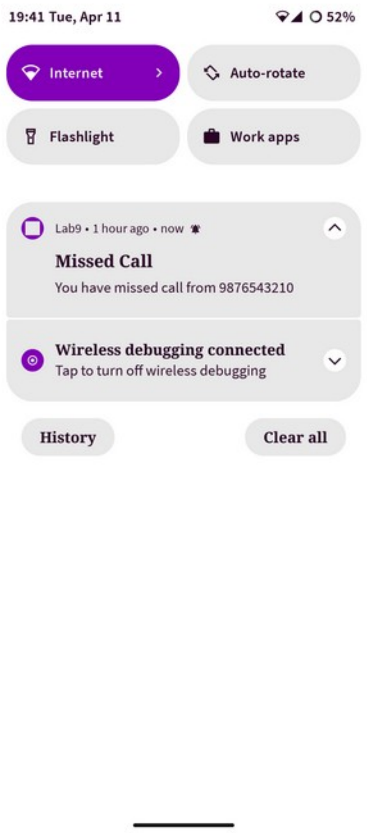
Program:

```
package com.example.lab9;
import android.app.NotificationChannel;   import android.app.PendingIntent;
import android.app.NotificationManager;   import android.content.Context;
import android.content.ComponentName;     import android.content.Intent;
import android.net.Uri;                   import android.os.Build;
import android.os.Bundle;                 import android.view.View;
import android.widget.Button;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.NotificationCompat;

public class MainActivity extends AppCompatActivity {
    Button b, missedCall, sameApp;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.notification);
        NotificationManager notificationMan =
(NotificationManager)getSystemService(Context.NOTIFICATION_SERVICE);
        if(Build.VERSION.SDK_INT ≥ Build.VERSION_CODES.O) {
            NotificationChannel ntch = new
NotificationChannel("channel_id","channel_name",
NotificationManager.IMPORTANCE_DEFAULT);
            notificationMan.createNotificationChannel(ntch);
        }

        b = findViewById(R.id.button);
        missedCall = findViewById(R.id.missedCall);
        sameApp = findViewById(R.id.sameApp);
        b.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                NotificationCompat.Builder builder = new
NotificationCompat.Builder(getApplicationContext(), "channel_id");
                builder.setContentTitle("Submit MAD Record");
                builder.setSmallIcon(R.drawable.freedo);
                builder.setContentText("MAD Record Submission date
Upcoming!!");
                builder.setSubText("Alert");
                notificationMan.notify(1, builder.build());
            }
        });

        missedCall.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                NotificationCompat.Builder builder = new
NotificationCompat.Builder(MainActivity.this, "channel_id");
                builder.setContentTitle("Missed Call");
                builder.setSmallIcon(R.drawable.freedo);
                builder.setContentText("You have missed call from
9876543210");
                builder.setSubText("1 hour ago");
                Intent intent = new Intent();
                intent.setAction(Intent.ACTION_DIAL);
                intent.setData(Uri.parse("tel:+919876543210"));
                PendingIntent pendingIntent =
PendingIntent.getActivity(getApplicationContext(), 0, intent,
PendingIntent.FLAG_IMMUTABLE);
                builder.setContentIntent(pendingIntent);
                notificationMan.notify(1, builder.build());
            }
        });

        sameApp.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                NotificationCompat.Builder builder = new
NotificationCompat.Builder(MainActivity.this, "channel_id");
                builder.setContentTitle("Open App");
                builder.setSmallIcon(R.drawable.freedo);
                builder.setContentText("You can click here to find
happiness");
                builder.setSubText("happiness key");
                Intent intent = new Intent(MainActivity.this,
MainActivity.class);
                PendingIntent pendingIntent =
PendingIntent.getActivity(getApplicationContext(), 0, intent,
PendingIntent.FLAG_IMMUTABLE);
                builder.setContentIntent(pendingIntent);
                notificationMan.notify(1, builder.build());
            }
        });
    }
}
```

Notification.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/notify"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <Button
        android:id="@+id/button"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Button" />

    <Button
        android:id="@+id/missedCall"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Missed Call Notification" />

    <Button
        android:id="@+id/sameApp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Open same app" />
</LinearLayout>
```

Outputs:



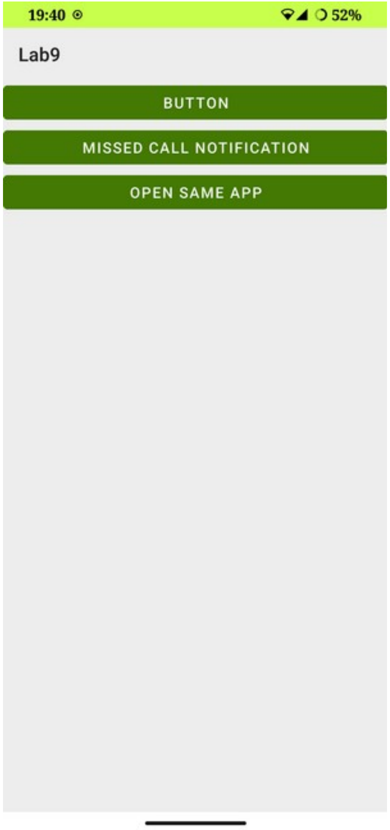
Lab9 • 1 hour ago • now

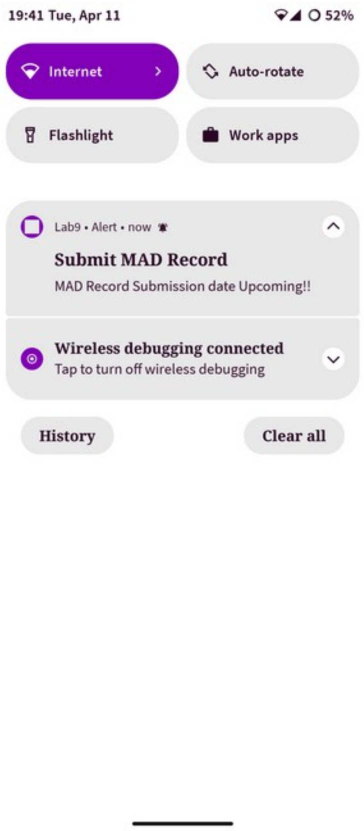
Missed Call

You have missed call from 9876543210

Wireless debugging connected

Tap to turn off wireless debugging





Lab9 • Alert • now

Submit MAD Record

MAD Record Submission date Upcoming!!

Wireless debugging connected

Tap to turn off wireless debugging

Week 7 and 8

Aim: To demonstrate SQLite on Android.

Program:

MainActivity.java

```
public class MainActivity extends AppCompatActivity {
    EditText et_id;
    EditText et_name;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        et_id = findViewById(R.id.stud_id);
        et_name = findViewById(R.id.stud_name);

        findViewById(R.id.btn_store).setOnClickListener(v -> {
            // Insertion
            String name = et_name.getText().toString();
            int id = Integer.parseInt(et_id.getText().toString());
            StudentContract.insertRecord(this, id, name);
            Toast.makeText(this, "Inserted Successfully!",
Toast.LENGTH_SHORT).show();
        });

        findViewById(R.id.btn_fetch).setOnClickListener(v -> {
            // Fetching
            startActivity(new Intent(this, DisplayRecords.class));
        });

        findViewById(R.id.btn_update).setOnClickListener(v -> {
            // Updation
            String id = et_id.getText().toString();
            String new_name = et_name.getText().toString();
            StudentContract.updateRecord(this, id, new_name);
        });

        findViewById(R.id.btn_delete).setOnClickListener(v -> {
            // Deletion
            String id = et_id.getText().toString();
            String name = et_name.getText().toString();
            StudentContract.deleteRecord(this, id, name);
        });

        findViewById(R.id.btn_drop).setOnClickListener(v -> {
            StudentContract.truncateTable();
        });
    }
    @Override
    protected void onDestroy() {
        StudentContract.closeDbHelper();
        super.onDestroy();
    }
}
```

SQLHelper.java

```
import static android.provider.UserDictionary.Words._ID;
//import static com.example.sqlitedemo.StudentContract.Entry;

import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import com.example.sqlitedemo.StudentContract.Entry;

public class SQLHelper extends SQLiteOpenHelper {

    public static final int DB_VERSION = 1;
    public static final String DB_NAME = "Student DB";

    public SQLHelper(Context context) {
        super(context, DB_NAME, null, DB_VERSION);
    }
    private static final String QUERY_CREATE = "CREATE TABLE
"+Entry.TABLE_NAME+
            " ("+Entry._ID+" INTEGER PRIMARY KEY,"+Entry.COL_1+"
INTEGER,"+Entry.COL_2+" TEXT)";
    private static final String QUERY_DROP = "DROP TABLE IF EXISTS
"+Entry.TABLE_NAME;

    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL(QUERY_CREATE);
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion)
{
        db.execSQL(QUERY_DROP);
        onCreate(db);
    }
    public void truncateTable(SQLiteDatabase db) {
        db.execSQL(QUERY_DROP);
        onCreate(db);
    }
}
```

RecyclerViewAdapter.java

```java
public class RecyclerViewAdapter extends
RecyclerView.Adapter<RecyclerViewAdapter.ViewHolder> {

    private final LinkedList<LinkedList<String>> student_data;

    public RecyclerViewAdapter(LinkedList<LinkedList<String>> data) {
        student_data = data;
    }

    @NonNull
    @Override
    public ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int
viewType) {
        View view = LayoutInflater.from(parent.getContext())
                .inflate(R.layout.recyclerview_row, parent, false);
        return new ViewHolder(view);
    }

    @Override
    public void onBindViewHolder(@NonNull RecyclerViewAdapter.ViewHolder
holder, int position) {
        holder.text_view_id.setText(student_data.get(position).get(0));
        holder.text_view_name.setText(student_data.get(position).get(1));
    }

    @Override
    public int getItemCount() {
        return student_data.size();
    }

    public static class ViewHolder extends RecyclerView.ViewHolder {
        public final TextView text_view_id, text_view_name;
        public ViewHolder(View view) {
            super(view);
            text_view_id = (TextView) view.findViewById(R.id.row_item_id);
            text_view_name = (TextView)
view.findViewById(R.id.row_item_name);
        }
/*        public TextView getTextView()
        {
            return text_view_id;
        }*/
    }
}
```

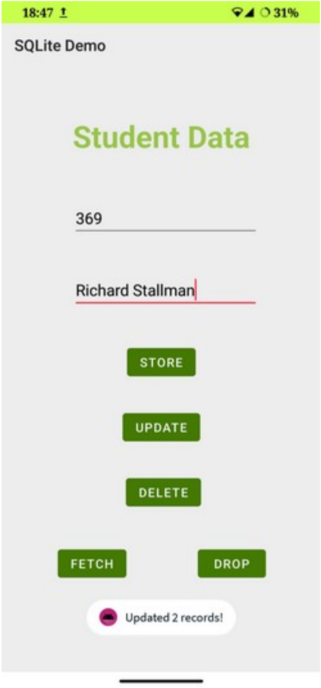
DisplayRecords.java

```java
public class DisplayRecords extends Activity {

    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.display_entries);
        LinkedList<LinkedList<String>> student_array = new LinkedList<>();
        Cursor cursor =
StudentContract.fetchAllRecords(getApplicationContext());
        while(cursor.moveToNext()) {
            int id =
cursor.getInt(cursor.getColumnIndexOrThrow(StudentContract.Entry.COL_1));
            String name =
cursor.getString(cursor.getColumnIndexOrThrow(StudentContract.Entry.COL_2));
            LinkedList<String> row = new LinkedList<>();
            row.add(String.valueOf(id));
            row.add(name);
            student_array.add(row);
        }
        cursor.close();

        RecyclerView recycler_view = findViewById(R.id.recycler_view);
        recycler_view.setLayoutManager(new LinearLayoutManager(this));
        RecyclerViewAdapter adapter = new RecyclerViewAdapter(student_array);
        recycler_view.setAdapter(adapter);
    }
}
```

StudentContract.java

```
package com.example.sqlitedemo;

public final class StudentContract {
    private StudentContract() {}

    public static class Entry implements BaseColumns {
        public static final String TABLE_NAME = "Student_Data";
        public static final String COL_1 = "ID";
        public static final String COL_2 = "Name";
    }

    static SQLHelper db_helper;

    public static void insertRecord(Context context, int id, String name) {
        SQLiteDatabase db;
        ContentValues values;

        db_helper = new SQLHelper(context);
        db = db_helper.getWritableDatabase();
        values = new ContentValues();
        values.put(Entry.COL_1, id);
        values.put(Entry.COL_2, name);
        long new_row_id = db.insert(Entry.TABLE_NAME, null, values);
        if (new_row_id == -1)
            Log.d("DBERR","Error Inserting Data");
    }
    public static Cursor fetchAllRecords(Context context) {
        db_helper = new SQLHelper(context);
        SQLiteDatabase db = db_helper.getReadableDatabase();
        String[] proj = {Entry.COL_1, Entry.COL_2};

        return db.query(Entry.TABLE_NAME, proj,
                null, null, null, null, null);
    }

    public static void updateRecord(Context context, String id, String
new_name) {
        db_helper = new SQLHelper(context);
        SQLiteDatabase db = db_helper.getWritableDatabase();
        ContentValues values = new ContentValues();
        values.put(Entry.COL_2, new_name);

        String sel = Entry.COL_1 + " LIKE ?";
        String[] sel_args = {id};
        int count = db.update(Entry.TABLE_NAME, values, sel, sel_args);
        Toast.makeText(context, "Updated "+count+" records!",
Toast.LENGTH_SHORT).show();
    }

    public static void deleteRecord(Context context, String id, String name)
{
        if (id.isEmpty() && name.isEmpty()) {
            Toast.makeText(context, "Empty Fields!",
Toast.LENGTH_SHORT).show();
            return;
        }
        String sel = (id.isEmpty())? Entry.COL_2 + " LIKE ?": Entry.COL_1 + "
LIKE ?";
        String[] sel_args = (id.isEmpty())? new String[]{name} : new String[]
{id};

        db_helper = new SQLHelper(context);
        SQLiteDatabase db = db_helper.getWritableDatabase();
        int deleted_rows_count = db.delete(Entry.TABLE_NAME, sel, sel_args);
        Toast.makeText(context, "Deleted "+deleted_rows_count+" entries!",
Toast.LENGTH_SHORT).show();
    }

    public static void truncateTable() {
        SQLiteDatabase db = db_helper.getWritableDatabase();
        db_helper.truncateTable(db);
    }

    public static void closeDbHelper() {
        db_helper.close();
    }
}
```

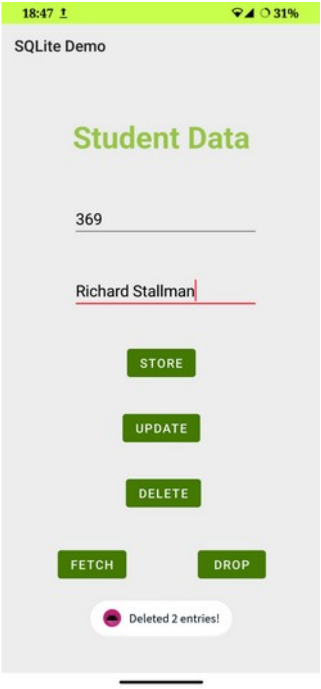
Outputs:



369

Richard Stallman

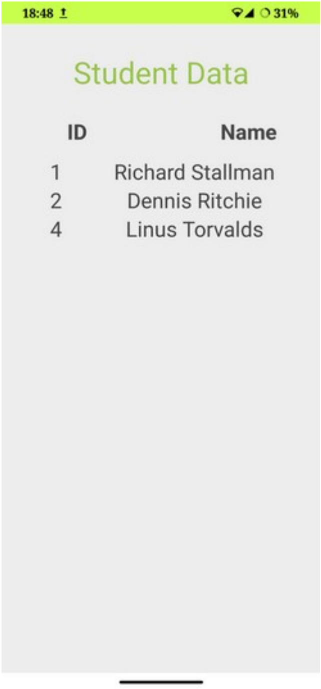
Updated 2 records!



369

Richard Stallman

Deleted 2 entries!



ID	Name
1	Richard Stallman
2	Dennis Ritchie
4	Linus Torvalds

Aim: To demonstrate Persistent storage using Files on Android.

Program:

```
public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        EditText stud_id = findViewById(R.id.stud_id);
        EditText stud_name = findViewById(R.id.stud_name);

        findViewById(R.id.btn_store).setOnClickListener(v -> {
            String id = stud_id.getText().toString();
            String name = stud_name.getText().toString();
            if (id.isEmpty() || name.isEmpty()) {
                Toast.makeText(this, "One of the fields is empty!",
Toast.LENGTH_SHORT).show();
                return;
            }
            String text = id + ',' + name + '\n';
            writeToFile(text);
        });

        findViewById(R.id.btn_fetch).setOnClickListener(v -> {
```

```java
            startActivity(new Intent(this, DisplayRecords.class));
        });
        findViewById(R.id.btn_update).setOnClickListener(v -> {
            String id = stud_id.getText().toString();
            updateData(id);
        });

    }


    void writeToFile(String text) {
        try (FileOutputStream file_out_stream =
openFileOutput(Student.FILE_NAME, Context.MODE_PRIVATE |
Context.MODE_APPEND)){
            file_out_stream.write(text.getBytes());
        } catch (IOException e) {
            Log.d("ERROR: ", e.getMessage());
        }
    }

    void updateData(String id) {
        File file = new File(getFilesDir(), Student.FILE_NAME);
        try {
            Scanner sc = new Scanner(file);
        } catch (IOException e) {
            Log.d ("ERROR: ", "updateData, IO Access error");
            e.printStackTrace();
        }

    }
```

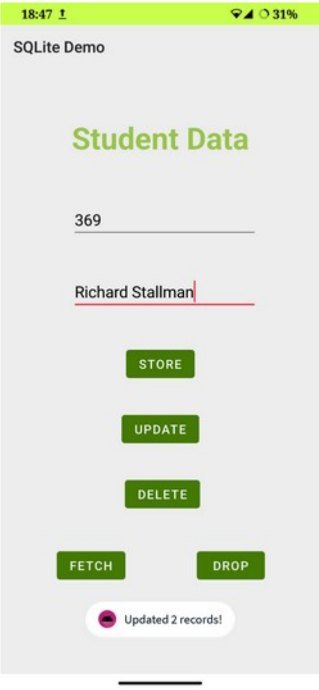
DisplayRecords.java

```java
package com.example.filesdemo;

public class DisplayRecords extends Activity {

    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.display_entries);
        LinkedList<LinkedList<String>> student_array = new LinkedList<>();

        try (FileInputStream file_in_stream =
openFileInput(Student.FILE_NAME)) {
            StringBuilder text = new StringBuilder();
            InputStreamReader in_stream_reader = new
InputStreamReader(file_in_stream);
            BufferedReader reader = new BufferedReader(in_stream_reader);
            String line = reader.readLine();
            while (line != null) {
                LinkedList<String> row = new LinkedList<>();
                String str[] = line.split(",");
                row.add(str[0]);
                row.add(str[1]);
                student_array.add(row);
                line = reader.readLine();
            }
            System.out.println("Your stuff is: "+text);
        } catch (IOException e) {
            Log.d("ERROR: ", e.getMessage());
        }

        RecyclerView recycler_view = findViewById(R.id.recycler_view);
        recycler_view.setLayoutManager(new LinearLayoutManager(this));
        RecyclerViewAdapter adapter = new RecyclerViewAdapter(student_array);
        recycler_view.setAdapter(adapter);
    }

    void fetchAllRecords() {

    }
}
```

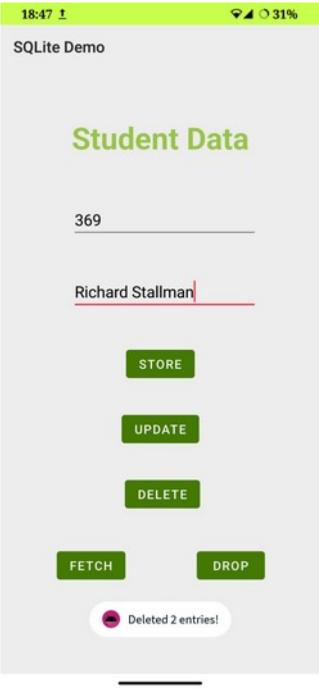
RecyclerViewAdapter.java

```
import java.util.LinkedList;

public class RecyclerViewAdapter extends
RecyclerView.Adapter<RecyclerViewAdapter.ViewHolder> {

    private final LinkedList<LinkedList<String>> student_data;

    public RecyclerViewAdapter(LinkedList<LinkedList<String>> data) {
        student_data = data;
    }

    @NonNull
    @Override
    public ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int
viewType) {
        View view = LayoutInflater.from(parent.getContext())
                .inflate(R.layout.recyclerview_row, parent, false);
        return new ViewHolder(view);
    }

    @Override
    public void onBindViewHolder(@NonNull ViewHolder holder, int position) {
        holder.text_view_id.setText(student_data.get(position).get(0));
        holder.text_view_name.setText(student_data.get(position).get(1));
    }

    @Override
    public int getItemCount() {
        return student_data.size();
    }
    public static class ViewHolder extends RecyclerView.ViewHolder {
        public final TextView text_view_id, text_view_name;
        public ViewHolder(View view) {
            super(view);
            text_view_id = (TextView) view.findViewById(R.id.row_item_id);
            text_view_name = (TextView)
view.findViewById(R.id.row_item_name);
        }
    }
}
```

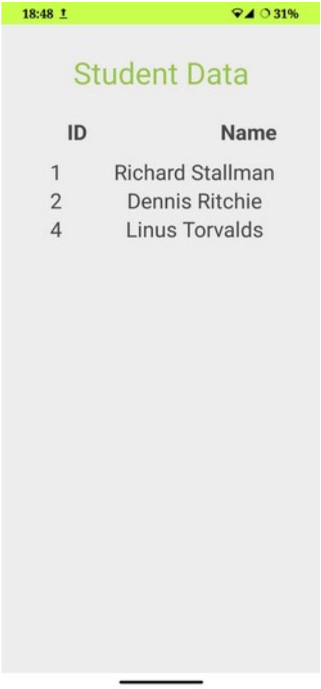
Output:



Updated 2 records!



Deleted 2 entries!



Student Data

ID	Name
1	Richard Stallman
2	Dennis Ritchie
4	Linus Torvalds