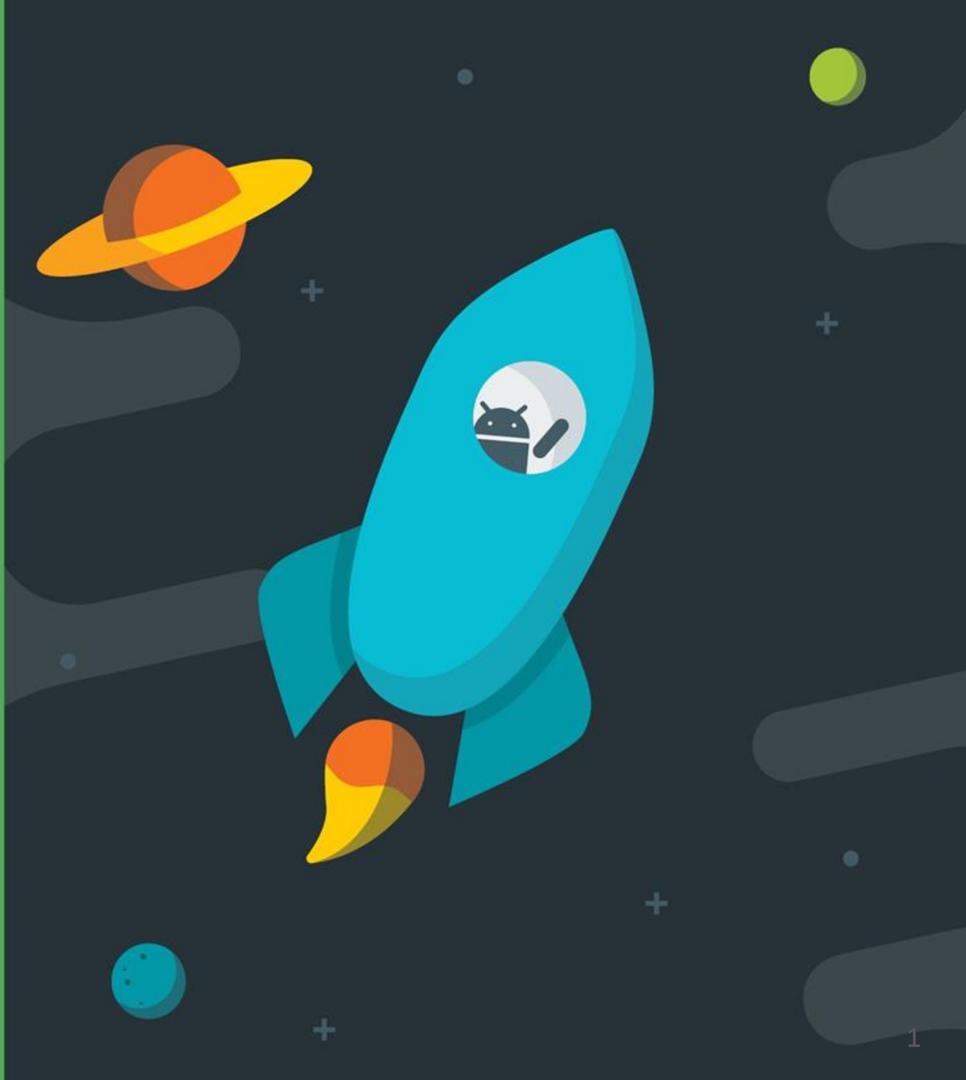
Android Application Development

Phone Dialler Application



Problem Statement

Develop an phone dialer application. On pressing the **Call** button, the app must make a call to the entered number.

Approach to problems (Procedure)

1. Analyse the problem statement and build the logic
2. Design the app using tools like Figma, etc
3. Implement the design in your Android Studio
4. Implement the logic in your Android Studio
5. Run the project(app)

1.1 Analysing the problem statement

Problem statement: A **call** to be made on press of a **button**

Requirements for the UI:

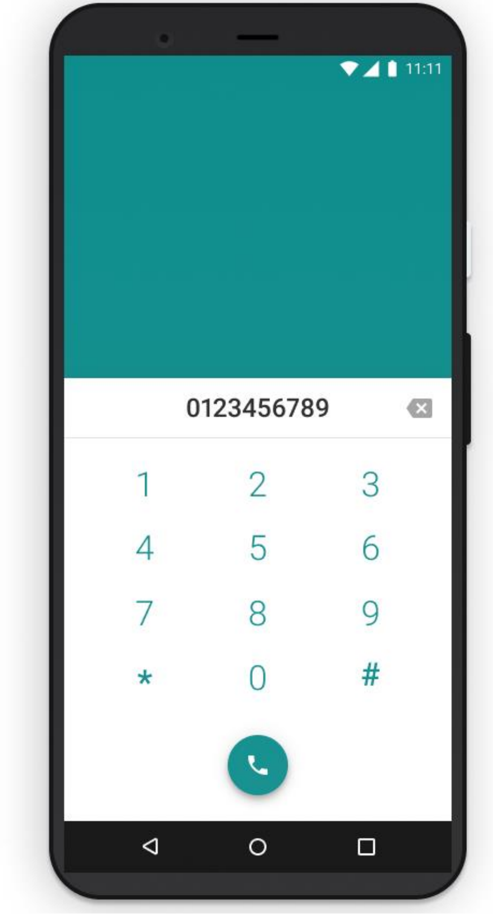
1. a Call button
2. a button Layout for numbers(0-9, *, #)
3. a TextView to view the entered number
4. a backspace/clear button

1.2 Building the logic

1. Number key -> append that number to the TextView.
2. Backspace -> remove last char, remove all char
3. Call -> send the number as Uri object to an Intent, which will handle Call.

2. Designing the app

Phone dialler App



Link for the design for the app.

Next steps

3. Implementing the UI
4. Implementing the logic
5. Running the project

Android
Studio

TextView example

1. Link the TextView object with the View

```
TextView textView = findViewById(R.id.text);
```

2. Retrieving the content of the TextView object

```
String str = textView.getText().toString();
```

2. Setting the value for the TextView object

```
textView.setText("hii, I'm a TextView");
```

4. Append string to the TextView object

```
textView.append("You are appending this string");
```

Intent types

Navigate to different Activity (Explicit Intent)

```
Intent it = new Intent(Activity1.class, Activity2.class);
startActivity(it);
```

Show a web page (Implicit Intent)

```
Uri uri = Uri.parse("http://www.google.com");
Intent it = new Intent(Intent.ACTION_VIEW,uri);
startActivity(it);
```

Sending an implicit Intent with data URI

1. Create an Intent for action

```
Intent intent = new Intent(Intent.ACTION_CALL);
```

2. Provide data as a URI

```
intent.setData(Uri.parse("tel:8005551234"));
```

3. Start the Activity

```
startActivity(intent);
```

AndroidManifest.xml

Pre-requirements before asking CALL permission:

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package=“course.aad.phone_dialer">

    <uses-permission android:name="android.permission.CALL_PHONE" />

    <application
        android:allowBackup="true“
        …..
        <activity android:name=".MainActivity">
            <intent-filter> … </intent-filter>
        </activity>
    </application>

</manifest>
```

Call and Save Contact

```
private void call(String number) {
    Intent intent = new Intent(Intent.ACTION_CALL);
    if (number.contains("#"))
        number = number.replaceAll("#", Uri.encode("#"));
    Uri uri = Uri.parse("tel:" + number);
    intent.setData(uri);
    startActivity(intent);
}


private void saveMethod(String inputPhoneNo) {
    Intent intent = new Intent(ContactsContract.Intents.Insert.ACTION);
    intent.setType(ContactsContract.RawContacts.CONTENT_TYPE);
    intent.putExtra(ContactsContract.Intents.Insert.PHONE, inputPhoneNo);
    startActivity(intent);
}
```

Request permission

```
private void requestPermissionFromUser() {
    String permissionString = Manifest.permission.CALL_PHONE;
    String[] permissionStringArray = new String[]{ permissionString };
    if (ContextCompat.checkSelfPermission(this, permissionString) != PackageManager.PERMISSION_GRANTED) {
        ActivityCompat.requestPermissions(MainActivity.this, permissionStringArray, REQUEST_CODE);
    }
}

@Override
public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[]
grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions, grantResults);
    if (requestCode == REQUEST_CODE) {
        if (grantResults[0] == PackageManager.PERMISSION_GRANTED) {
            makeToast("Permission Granted");
        } else {
            makeToast("Permission Denied");
        }
    } else
        super.onRequestPermissionsResult(requestCode, permissions, grantResults);
}
```

Retrieving contacts

```
 ArrayAdapter<String> adapter = new ArrayAdapter<>(this,
         android.R.layout.simple_list_item_1, android.R.id.text1, list);
 listView.setAdapter(adapter);


private void getContacts() {
    Uri uri = ContactsContract.CommonDataKinds.Phone.CONTENT_URI;
    // to pass the all contacts to the cursor
    String ascendingOrder = ContactsContract.CommonDataKinds.Phone.DISPLAY_NAME + " ASC";
    Cursor cursor = getContentResolver().query(uri, null, null, null, ascendingOrder);
    // to fetch all the contacts
    while (cursor.moveToNext()) {
        String nameIndex = ContactsContract.CommonDataKinds.Phone.DISPLAY_NAME;
        String mobIndex = ContactsContract.CommonDataKinds.Phone.NUMBER;
        String name = cursor.getString(cursor.getColumnIndex(nameIndex));
        String mob = cursor.getString(cursor.getColumnIndex(mobIndex));
        list.add(name + "\n" + mob);
    }
    cursor.close();
}
```

Any questions?

For future references

1. For the source code of the Phone Dialler App.
2. Google Developer Course: https://developer.android.com/courses
3. For more resources: awesome-android-learning-resources
4. YouTube

Contact me for any queries,

Raghavendra K M
ISE 6th sem
BMSIT

GitHub: **Raghuvorkady**

Email: **raghavendrakm300@gmail.com**

LinkedIn: **raghavendra-k-m-2214b0194/**

Twitter: **Raghavendra_K_M**

Thank you