Android

UI DESIGN-SESSION 2

SYAMKRISHNA BG
NATIONAL INFORMATICS CENTRE

CENTRE FOR COMPETENCE FOR MOBILE APP DEVELOPMENT , NIC KERALA

Context

Context allows access to application-specific resources and classes, as well as calls for application-level operations such as launching activities, broadcasting and receiving intents, etc. ... This is an abstract class whose implementation is provided by the **Android** system

Things that involve context are:

- Loading a resource.
- Launching a new activity.
- Creating views.
- obtaining system service.

Context

Different methods by which you can get context

getApplicationContext()
getContext()
getBaseContext()
or **this** (when in the activity class)

Example cases
getAssets()
getResources()
getPackageManager()
getString()
getSharedPrefsFile()

Adapters

In Android, Adapter is a bridge between UI component and data source that helps us to fill data in UI component. It holds the data and send the data to an Adapter view then view can takes the data from the adapter view and shows the data on different views like
as ListView, GridView, Spinner etc. For more customization in Views we uses the base adapter or custom adapters.

There are the some commonly used Adapter in Android used to fill the data in the UI components.

BaseAdapter – It is parent adapter for all other adapters

ArrayAdapter – It is used whenever we have a list of single items which is backed by an array

Custom ArrayAdapter – It is used whenever we need to display a custom list

SimpleAdapter – It is an easy adapter to map static data to views defined in your XML file

Custom SimpleAdapter – It is used whenever we need to display a customized list and needed to access the child items of the list or grid

NATIONAL INFORMATICS CENTRE

Android Fragments

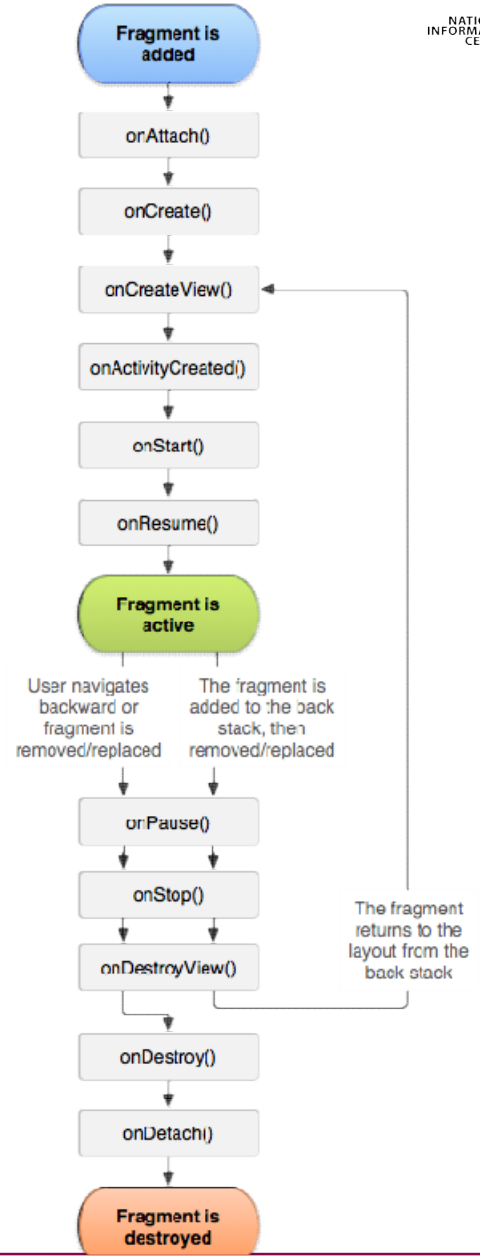
Fragments

A Fragment represents a behavior or **a portion of user interface** in a FragmentActivity.
You can combine multiple fragments in a single activity to build a multi-pane UI and **reuse** a fragment in multiple activities.

You can **think of a fragment as a modular section of an activity**, which has its **own lifecycle**, receives its own input events, and which you can add or remove while the activity is running (sort of like a "sub activity" that you can reuse in different activities).

Lifecycle

https://developer.android.com/guide/components/fragments

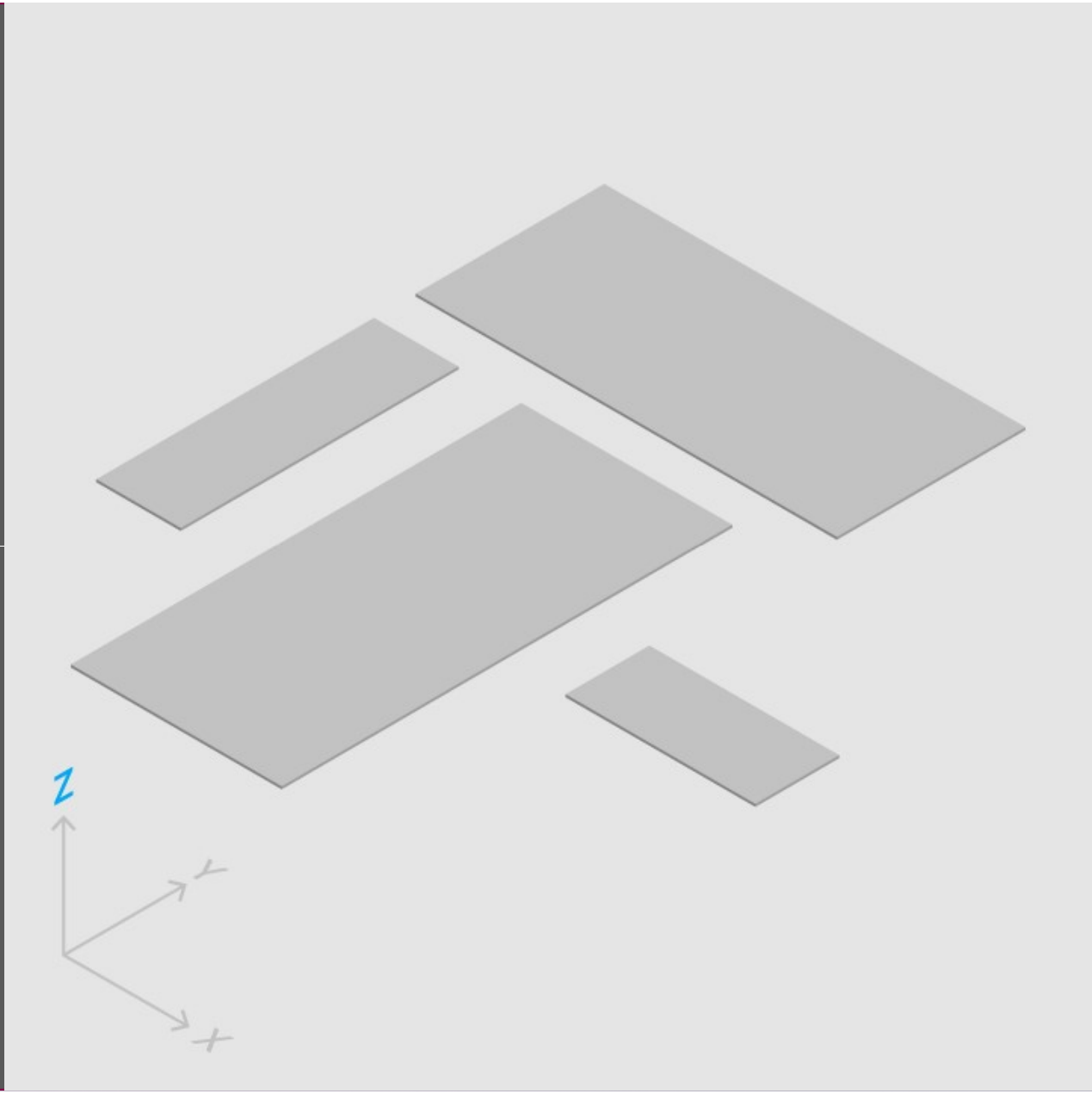


Material Design

Material design is a comprehensive guide for visual, motion, and interaction design across platforms and devices.

To use material design in your Android apps, follow the guidelines defined in the material design specification

Use the new components and styles available in the material design support library.



Elevation

NATIONAL INFORMATICS CENTRE

Navigation

Layout

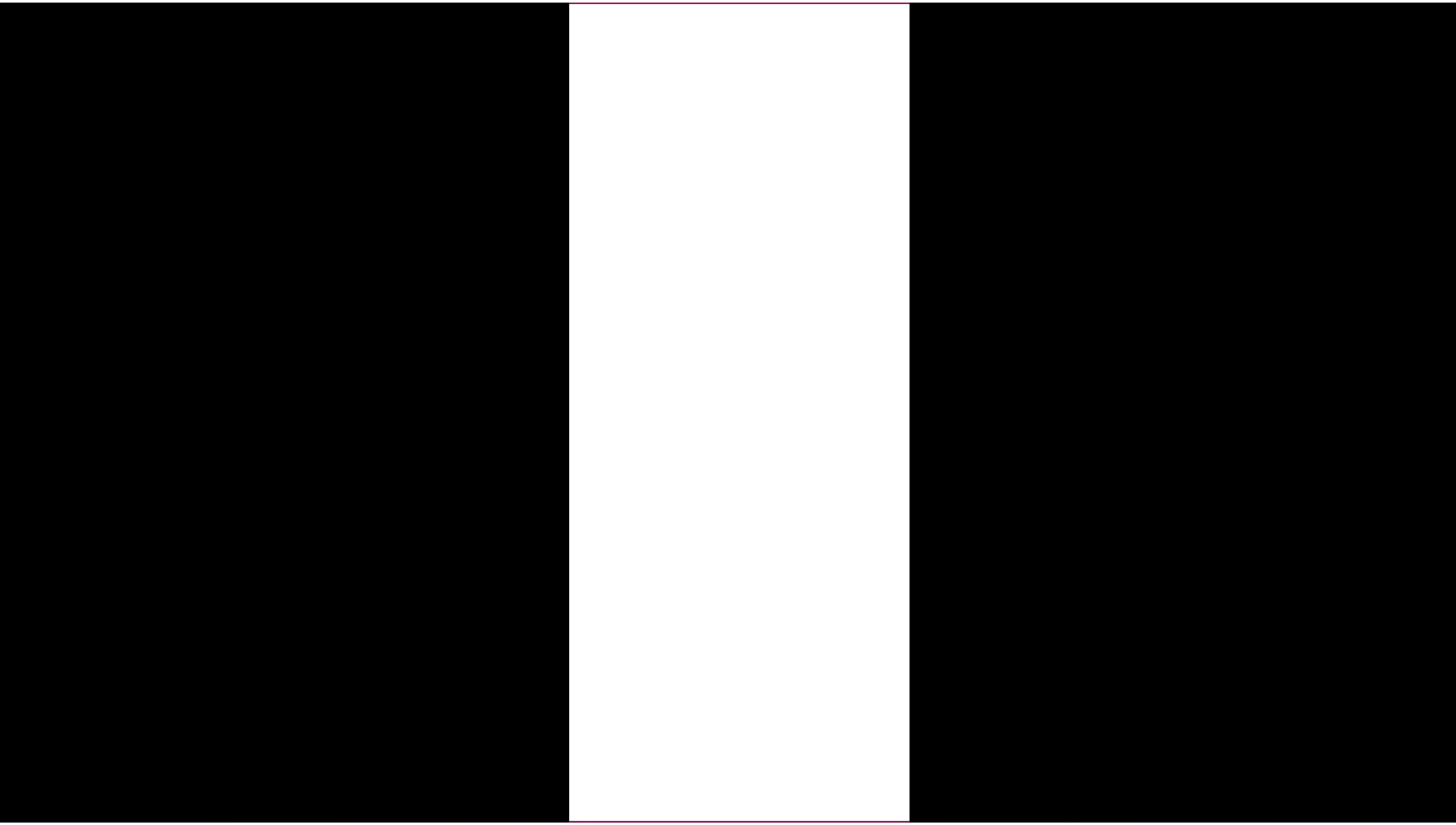
Colour

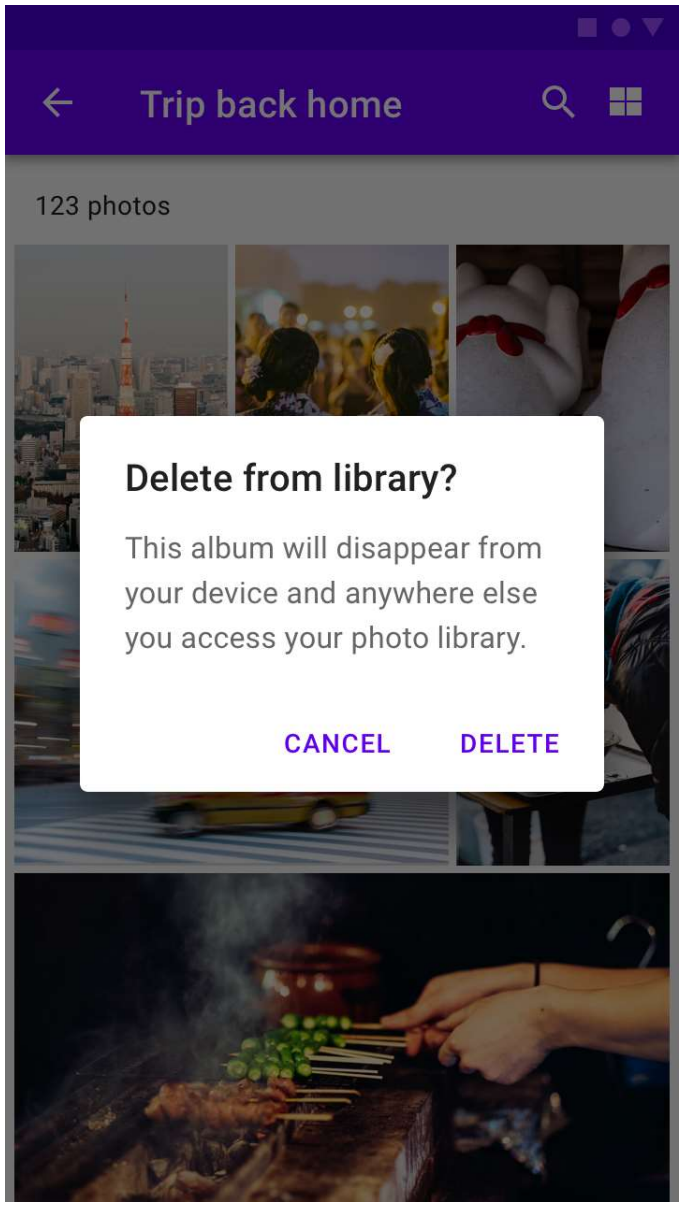
Typography

Shape

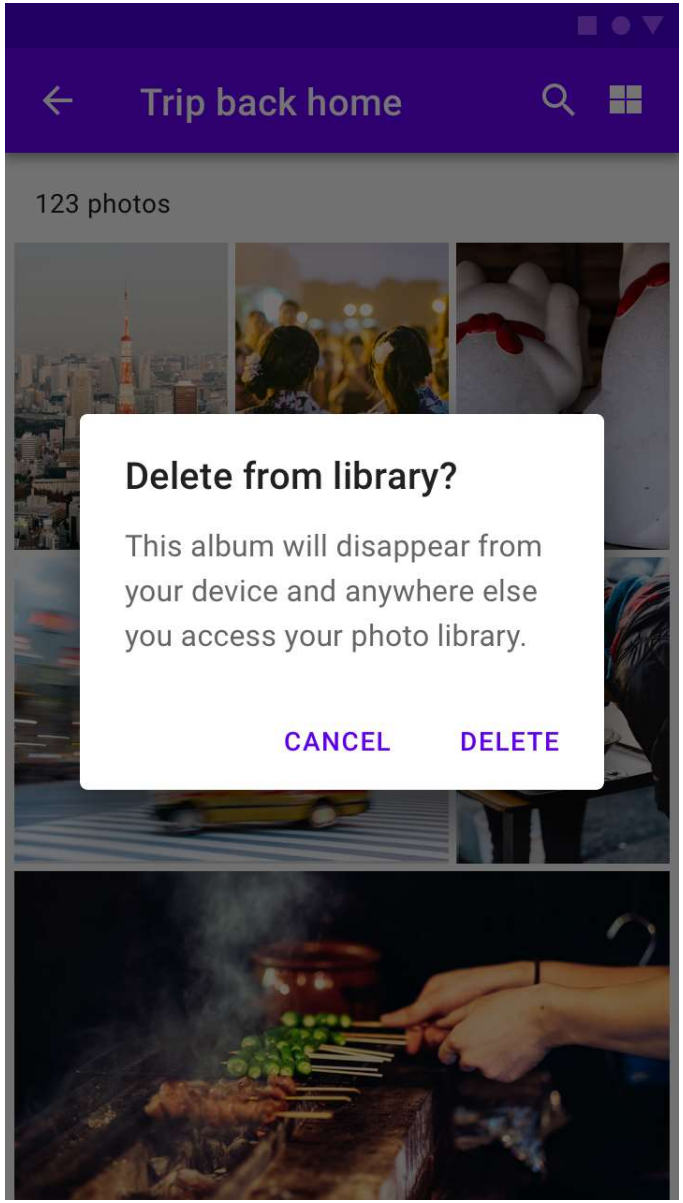
Motion

Interaction

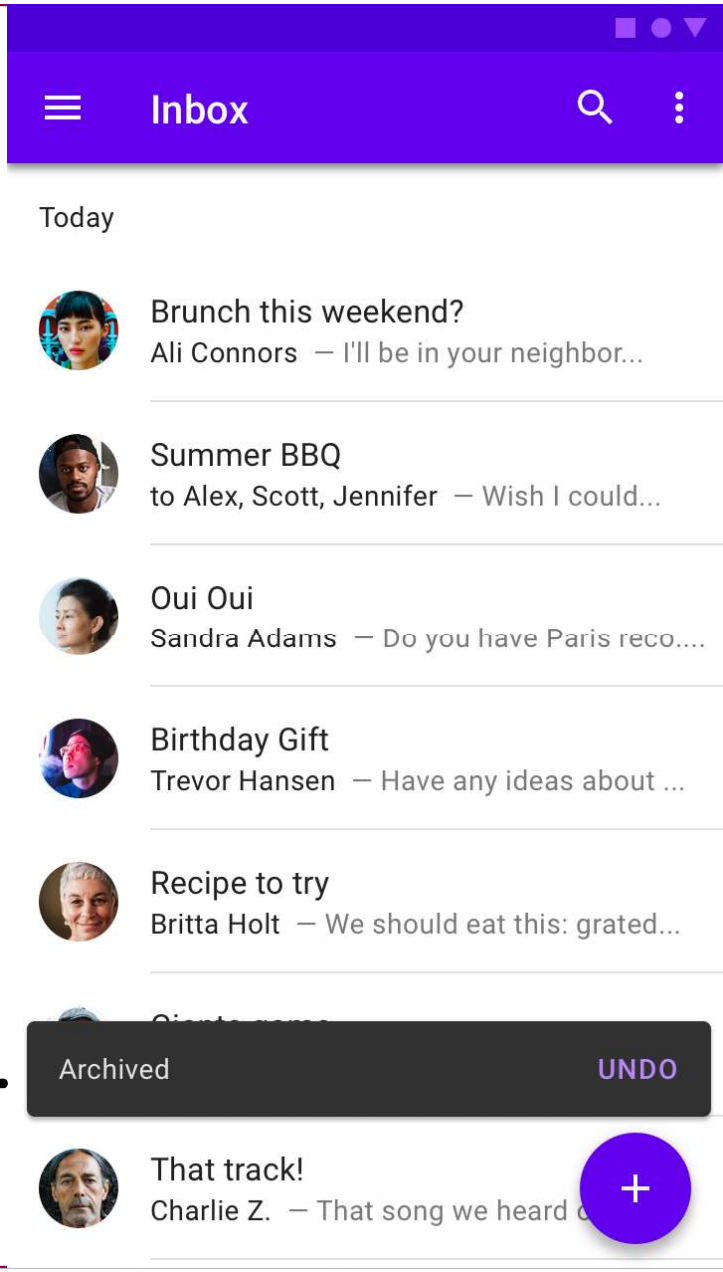




AlertDialogue



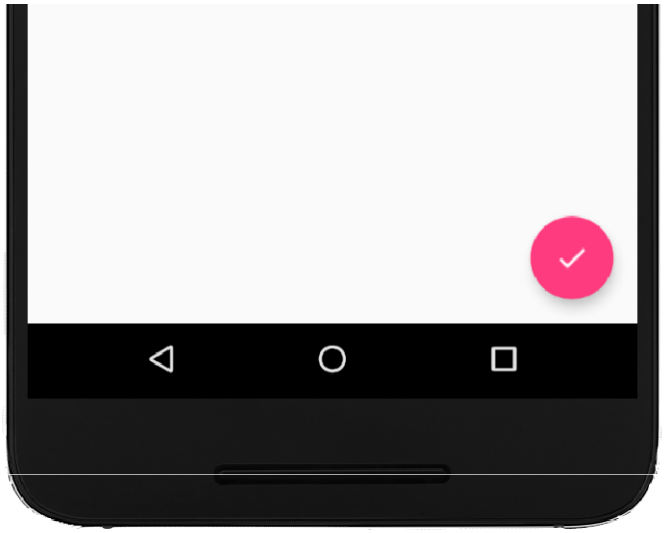
AlertDialogue



SnackBar

Cards

A floating action button



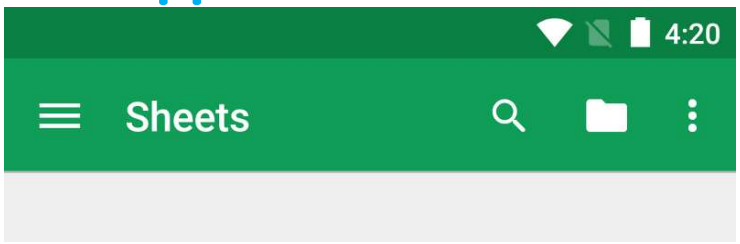
View

```
<android.support.design.widget.FloatingActionButton
    android:id="@+id/fab"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="end|bottom"
    android:src="@drawable/ic_my_icon"
    android:layout_margin="16dp" />
```

Activity

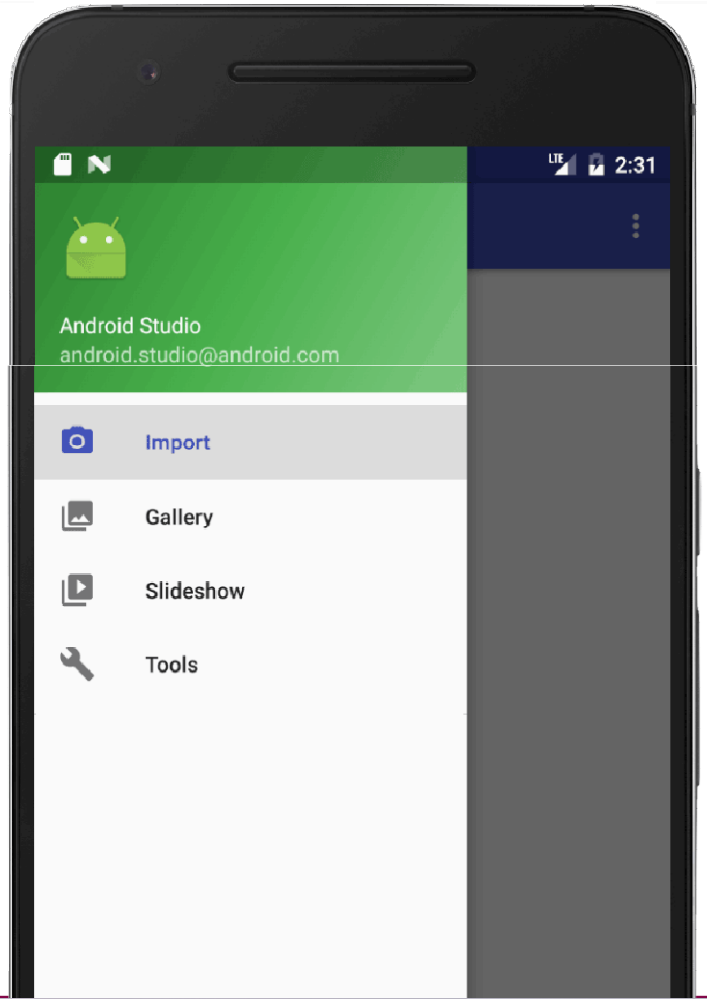
```
FloatingActionButton fab = findViewById(R.id.fab);
fab.setOnClickListener(new View.OnClickListener() {
  @Override
  public void onClick(View view) {
    Snackbar.make(view, "Here's a Snackbar", Snackbar.LENGTH_LONG)
        .setAction("Action", null).show();
  }
});
```

The app bar




```
dependencies
{
  implementation 'com.android.support:appcompat-v7:27.1.1'
  implementation 'com.android.support:design:27.1.1'
}
```

A navigation drawer



NATIONAL INFORMATICS CENTRE

Thank you