PENGEMBANGAN APLIKASI PERANGKAT BERGERAK (MOBILE)

Android User Interface

K Candra Brata

andra.course@gmail.com

Delivering Android UI/UX

Keep it simple !

“ The more users feel in control of the system, the more they will like it.“

- Jakob Nielson-

User Interface

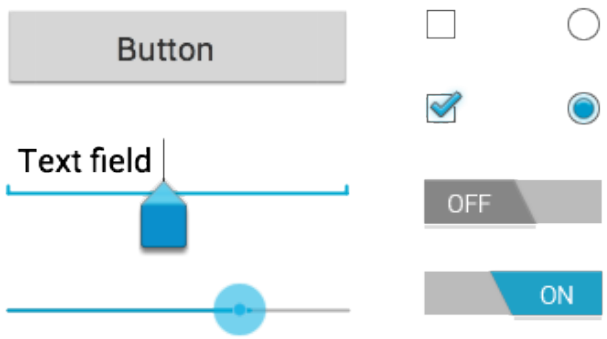
http://developer.android.com/guide/topics/ui/index.html

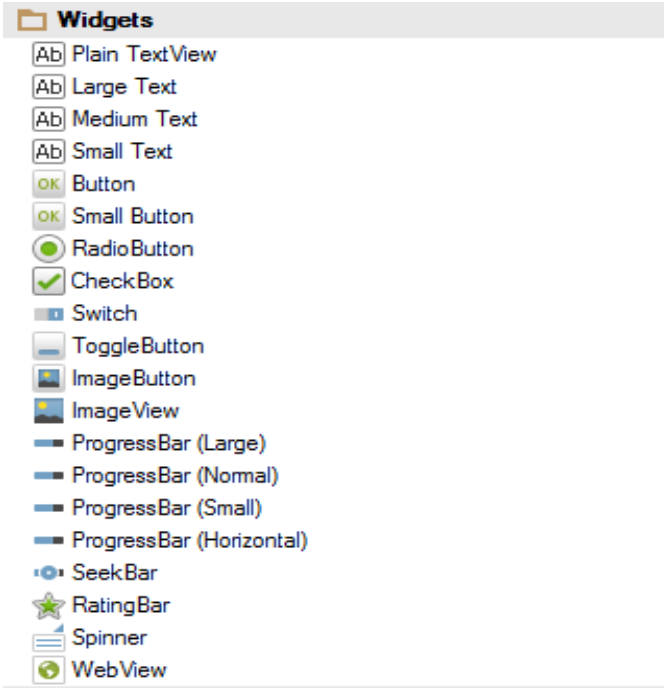
VIEWs

- Controls or widgets (**not App Widgets**).
- All UI controls, including layout classes, derived from views.

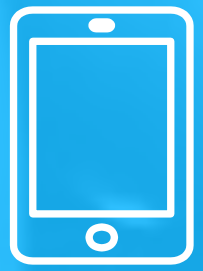
VIEW Groups

- Extension of View that can contain multiple child groups
- ViewGroup extended to provide layout managers that help you layout controls.






```
<Button android:id="@+id/button_send"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/button_send"
        android:onClick="sendMessage" />
```



ViewGroup

http://developer.android.com/guide/topics/ui/declaring-layout.html

Definition

- Merupakan layout atau jenis penyusunan komponen pada user interface, ada yang berjajar, saling menumpuk, dll.

- Layout yang paling banyak digunakan dalam pengembangan adalah **LinearLayout**, **RelativeLayout**, **FrameLayout**, dan **GridLayout**.

Layout

All layouts allow the developer to define attributes. Children can also define attributes which may be evaluated by their parent layout.

Children can specify their desired width and height via the following attributes.

Table 5. Width and height definition

Attribute	Description
`android:layout_width`	Defines the width of the widget.
`android:layout_height`	Defines the height of the widget.

Layout

wrap_content

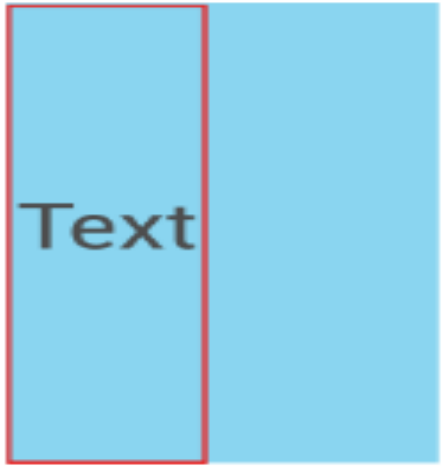
```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    ...
    />
```

match_parent

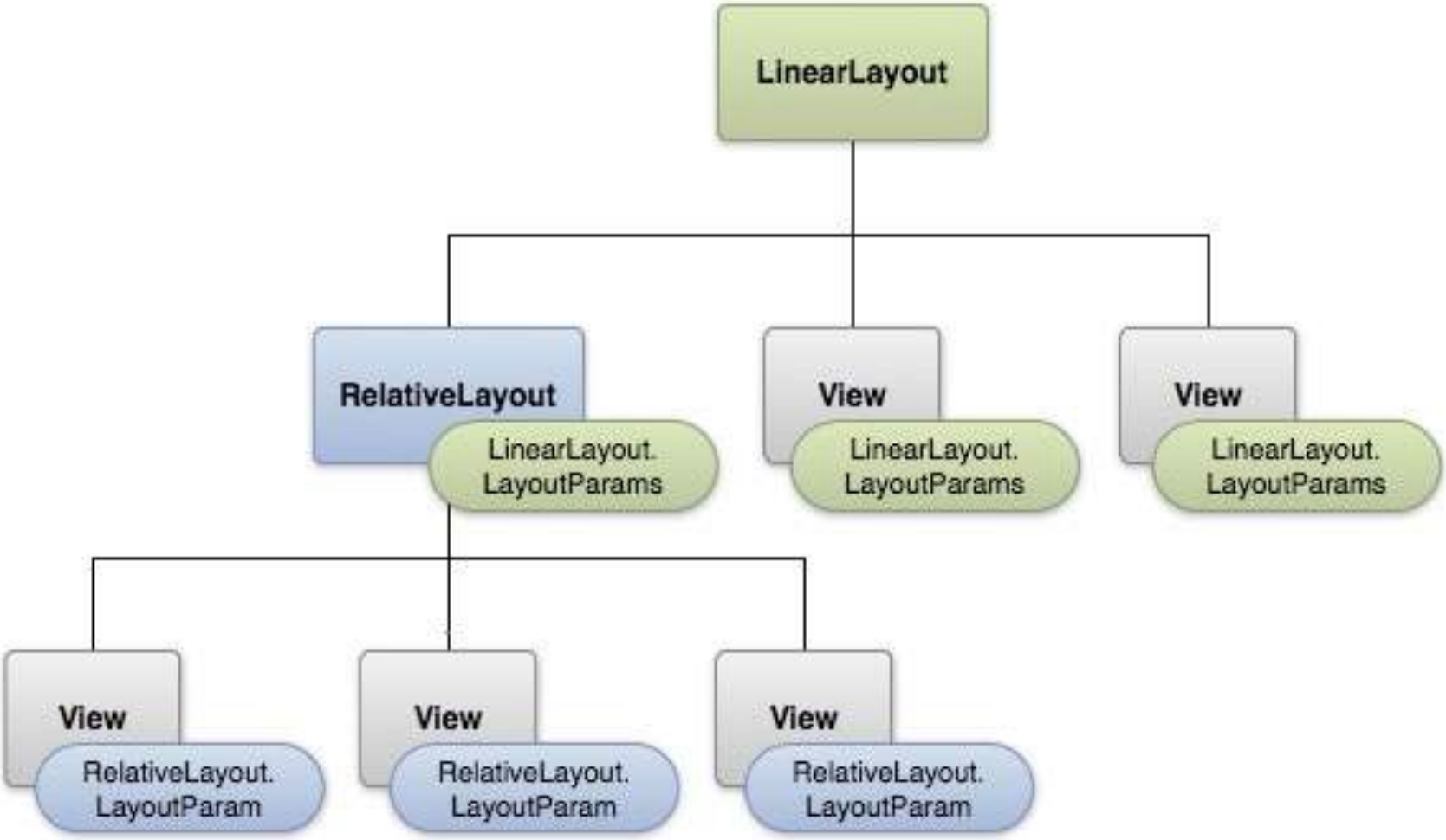
```
android:layout_width="wrap_content"
android:layout_height="match_parent"
```

```
android:layout_width="match_parent"
android:layout_height="wrap_content"
```


```
android:layout_width="match_parent"
android:layout_height="match_parent"
```



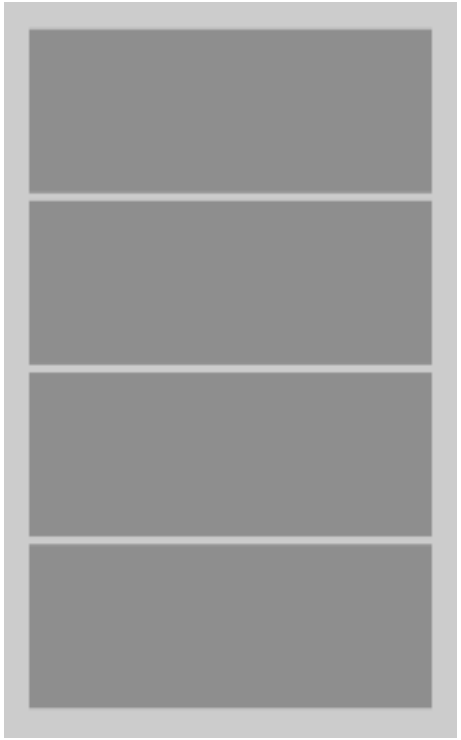
Layout Hierarchy Example



Linear Layout

Linear layout adalah susunan tata letak yang paling sederhana. Layout ini hanya memberikan susunan tata letak komponen secara garis lurus (linear). Bisa secara *Horizontal* maupun *Vertikal*.

Tergantung pada **android:orientation**



cobalinearlayout.XML

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

    <EditText
        android:id="@+id/editText1"
        android:layout_width="200dp"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:layout_margin="10dip"
        android:hint="write on me!"/>

    <SeekBar
        android:id="@+id/seekBar1"
        android:layout_width="200dp"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:layout_marginTop="10dp"
        android:progress="50" />

    <Button
        android:id="@+id/btnClickMe"
        android:layout_width="270dp"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginTop="10dp"
        android:text="Click me !!" />

</LinearLayout>
```

Tambahkan kode berikut di dalam method **onCreate(Bundle)** di dalam class Activity.

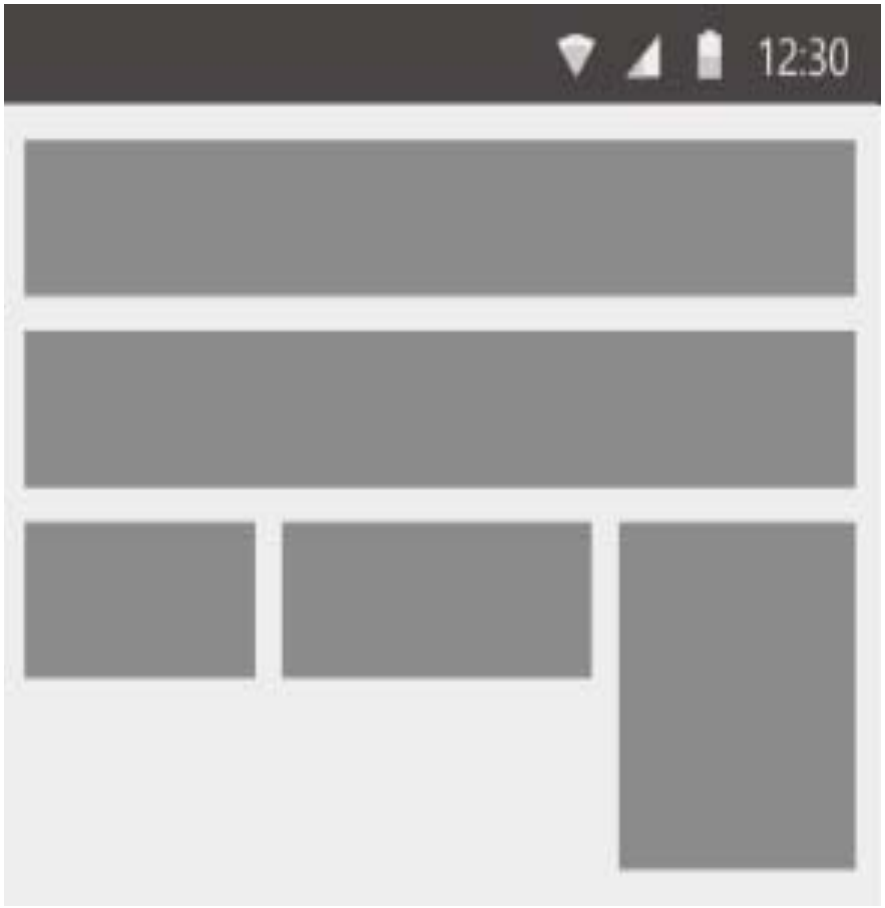
HardCoding Layout

```
LinearLayout ll = new LinearLayout(this);
ll.setOrientation(LinearLayout.VERTICAL);
ll.setGravity(Gravity.CENTER);

EditText myEditText = new EditText(this);
SeekBar mySeekBar = new SeekBar(this);
Button myBtn = new Button(this);

myEditText.setText("Text Goes Here!");
mySeekBar.setProgress(50);
myBtn.setText("Click Here !!");

int lHeight = LinearLayout.LayoutParams.WRAP_CONTENT;
int lWidth = 500;

LinearLayout.LayoutParams params= new LinearLayout.LayoutParams(lWidth,lHeight);
params.setMargins(10, 10, 10, 10);

ll.addView(myEditText, params);
ll.addView(mySeekBar, params);
ll.addView(myBtn, params);
setContentView(ll);
```

Relative Layout

Relative layout digunakan untuk menempatkan elemen yang bergantung pada posisi elemen sebelumnya.

- **Android:layout_above**
- **Android:layout_alignBaseline**
- **Android:layout_alignbottom**
- **Android:layout_alignLeft**
- **Android:layout_alignParentBottom**
- **Android:layout_alignParentLeft**
- **Android:layout_alignParentRight**
- **Android:layout_alignParentTop**
- **Android:layout_alignRight**
- **Android:layout_alignTop**
- **Android:layout_alignWithParentIfMissing**
- **Android:layout_below**
- **Android:layout_centerHorizontal**
- **Android:layout_centerInParent**
- **Android:layout_centerVertical**
- **Android:layout_toLeftOf**

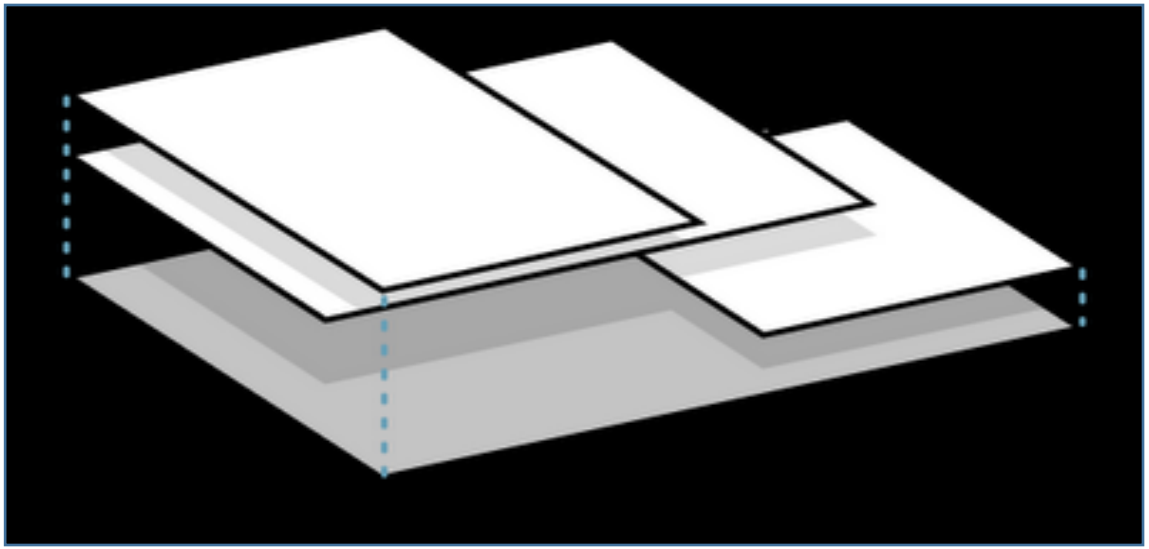


cobarelativelayout.XML

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

    <EditText
        android:id="@+id/txt1"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:hint="@string/intro" />

    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp"
        android:layout_below="@+id/txt1"
        android:orientation="vertical">

        <Button
            android:id="@+id/button1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Button 1" />

        <Button
            android:id="@+id/button2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Button 2" />
    </LinearLayout>
</RelativeLayout>
```

Frame Layout

FrameLayout is a layout manager which draws all child elements on top of each other (overlapping). This allows to create nice visual effects.



cobaframelayout.XML

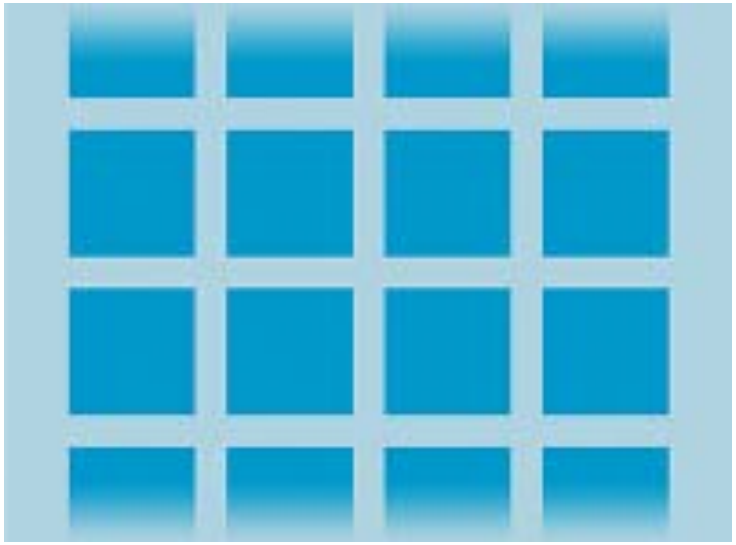
```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <ImageView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:scaleType="fitCenter"
        android:src="@mipmap/ic_launcher" />

    <TextView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:gravity="center"
        android:text="Frame Demo"
        android:textSize="100px"
        android:textStyle="bold"
        android:textColor="#ffff"/>

</FrameLayout>
```

Grid Layout

Require min API 14 (android 4.0).

This layout allows you to organize a view into a Grid. GridLayout separates its drawing area into: rows, columns, and cells.

You can specify how many columns you want to define for each View, in which row and column it should be placed as well as how many columns and rows it should use.

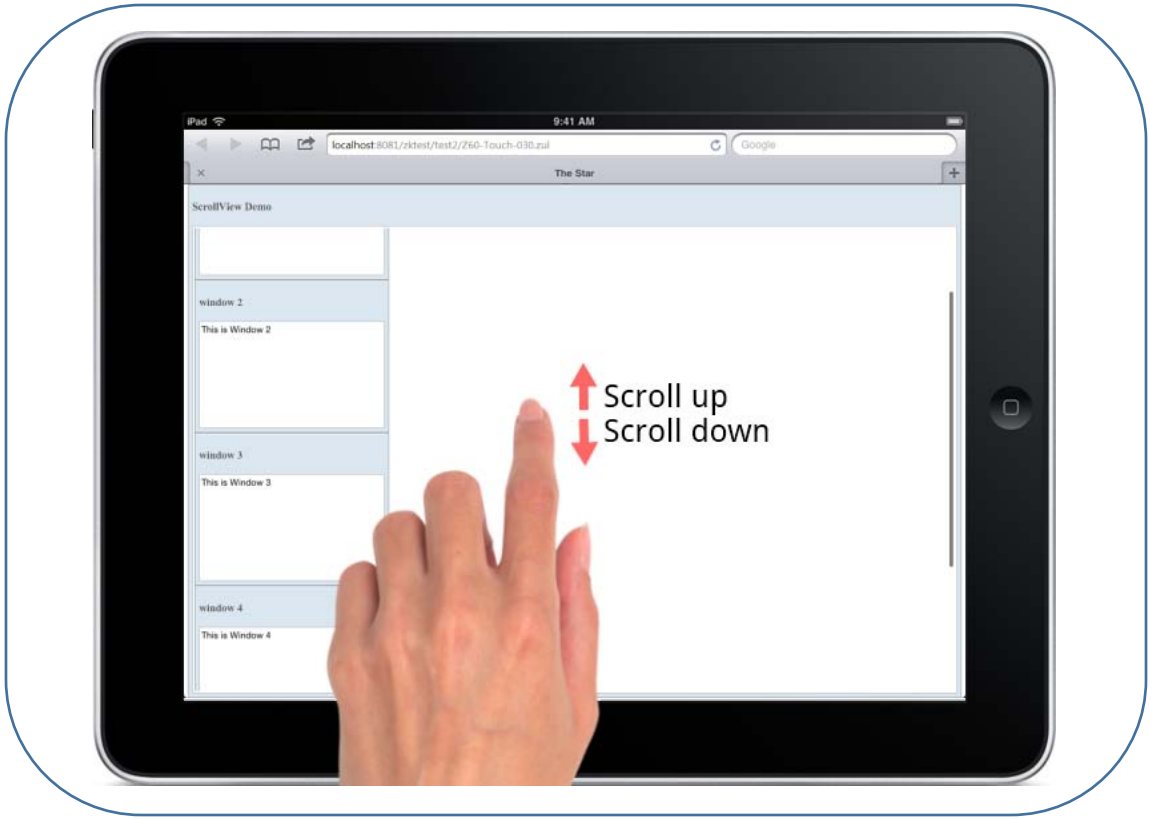


cobagridlayout.XML

```
<?xml version="1.0" encoding="utf-8"?>
<GridLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/GridLayout1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:columnCount="4"
    android:useDefaultMargins="true" >
    <TextView
        android:layout_column="0"
        android:layout_columnSpan="3"
        android:layout_gravity="center_horizontal"
        android:layout_marginTop="40dp"
        android:layout_row="0"
        android:text="User Credentials"
        android:textSize="32dip" />
    <TextView
        android:layout_column="0"
        android:layout_gravity="right"
        android:layout_row="1"
        android:text="User Name: " >
    </TextView>
    <EditText
        android:id="@+id/input1"
        android:layout_column="1"
        android:layout_columnSpan="2"
        android:layout_row="1"
        android:ems="10" />

    <TextView
        android:layout_column="0"
        android:layout_gravity="right"
        android:layout_row="2"
        android:text="Password: " >
    </TextView>

    <EditText
        android:id="@+id/input2"
        android:layout_column="1"
        android:layout_columnSpan="2"
        android:layout_row="2"
        android:inputType="textPassword"
        android:ems="8" />

    <Button
        android:id="@+id/button1"
        android:layout_column="2"
        android:layout_row="3"
        android:text="Login" />

</GridLayout>
```

ScrollView

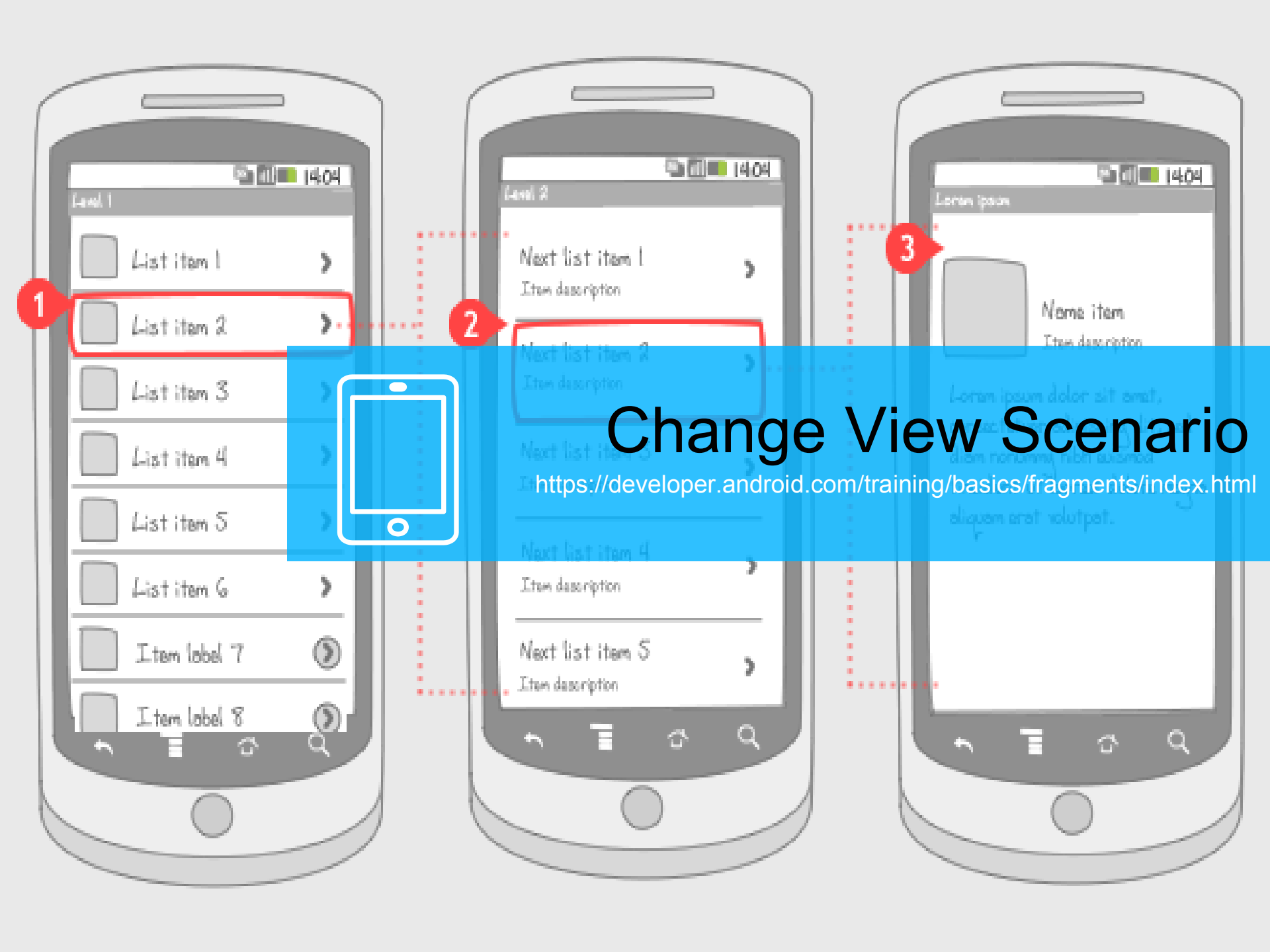
The ScrollView class can be used to contain one View that might be to big to fit on one screen. In this case ScrollView will display a scroll bar to scroll the context.



cobascrollview.XML

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fillViewport="true"
    android:orientation="vertical" >

    <TextView
        android:id="@+id/TextView01"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:paddingLeft="8dip"
        android:paddingRight="8dip"
        android:paddingTop="8dip"
        android:text="This is a header"
        android:textAppearance="?android:attr/textAppearanceLarge" >
    </TextView>

</ScrollView>
```

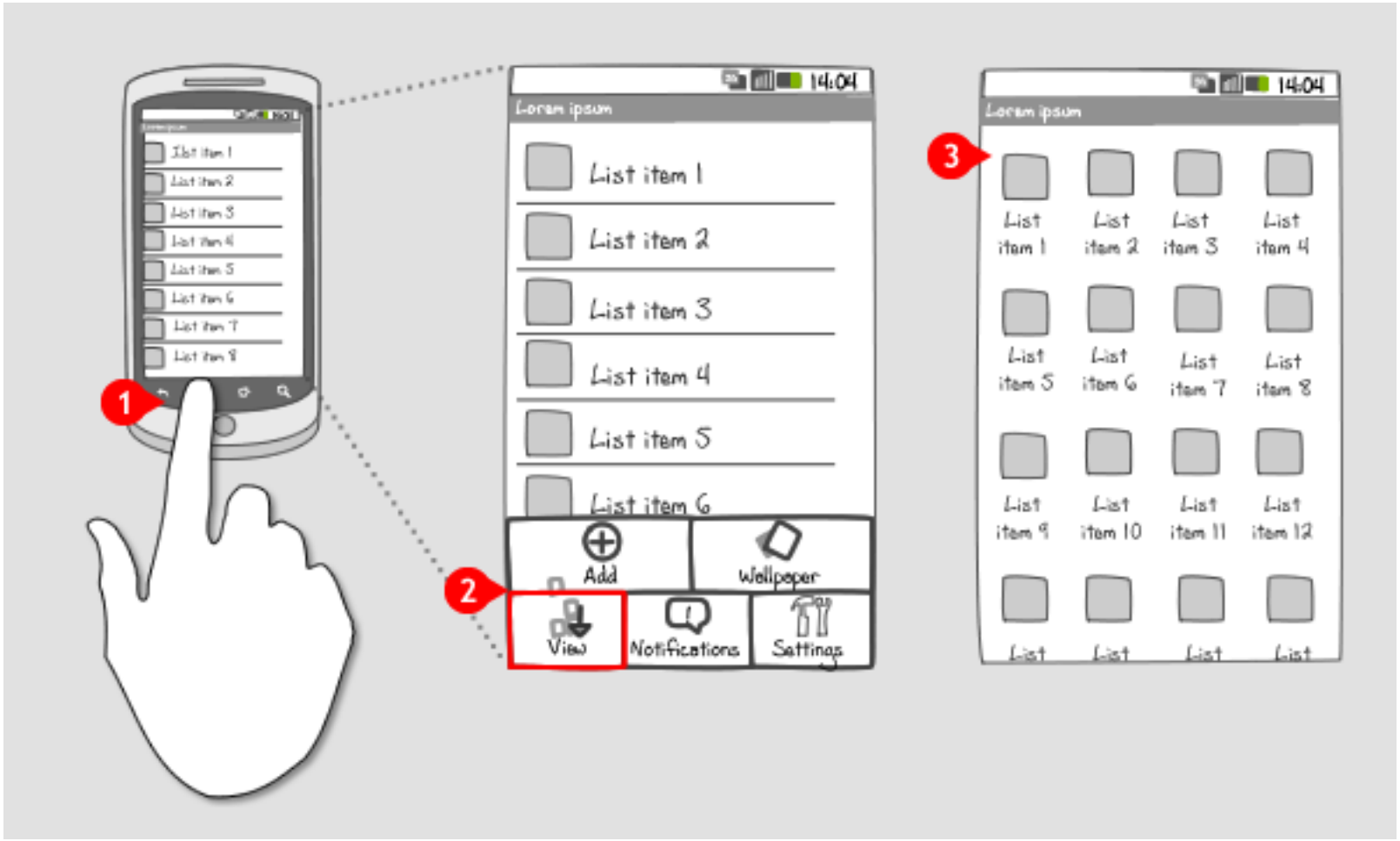
The **android:fillViewport="true"** attribute ensures that the scrollview is set to the full screen even if the elements are smaller than one screen.



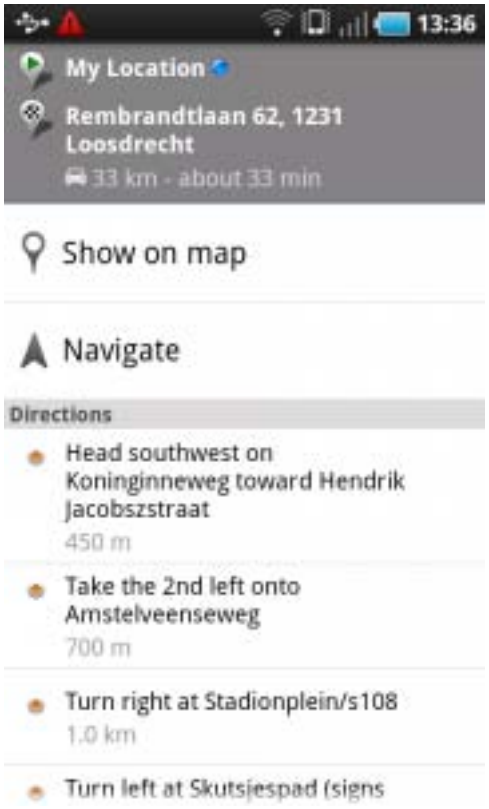
Change View Scenario

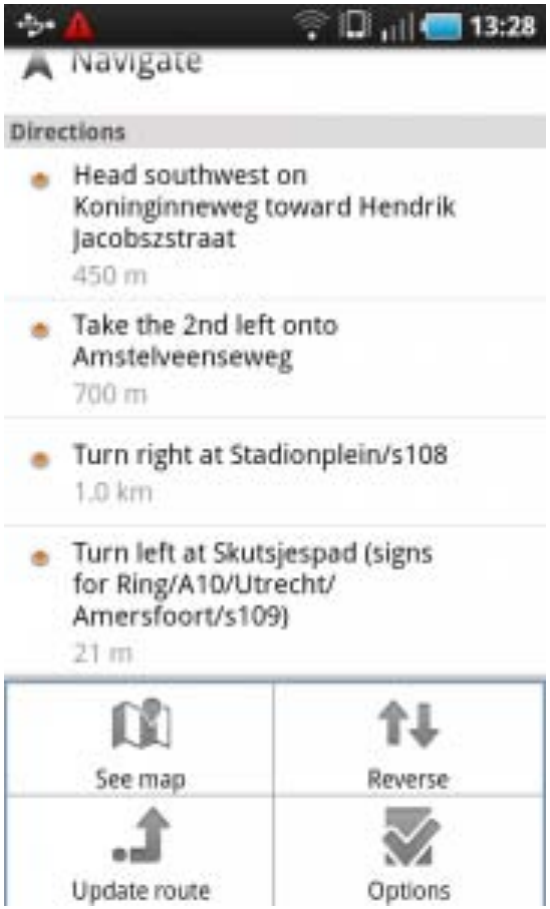
https://developer.android.com/training/basics/fragments/index.html

Options menu



Options menu

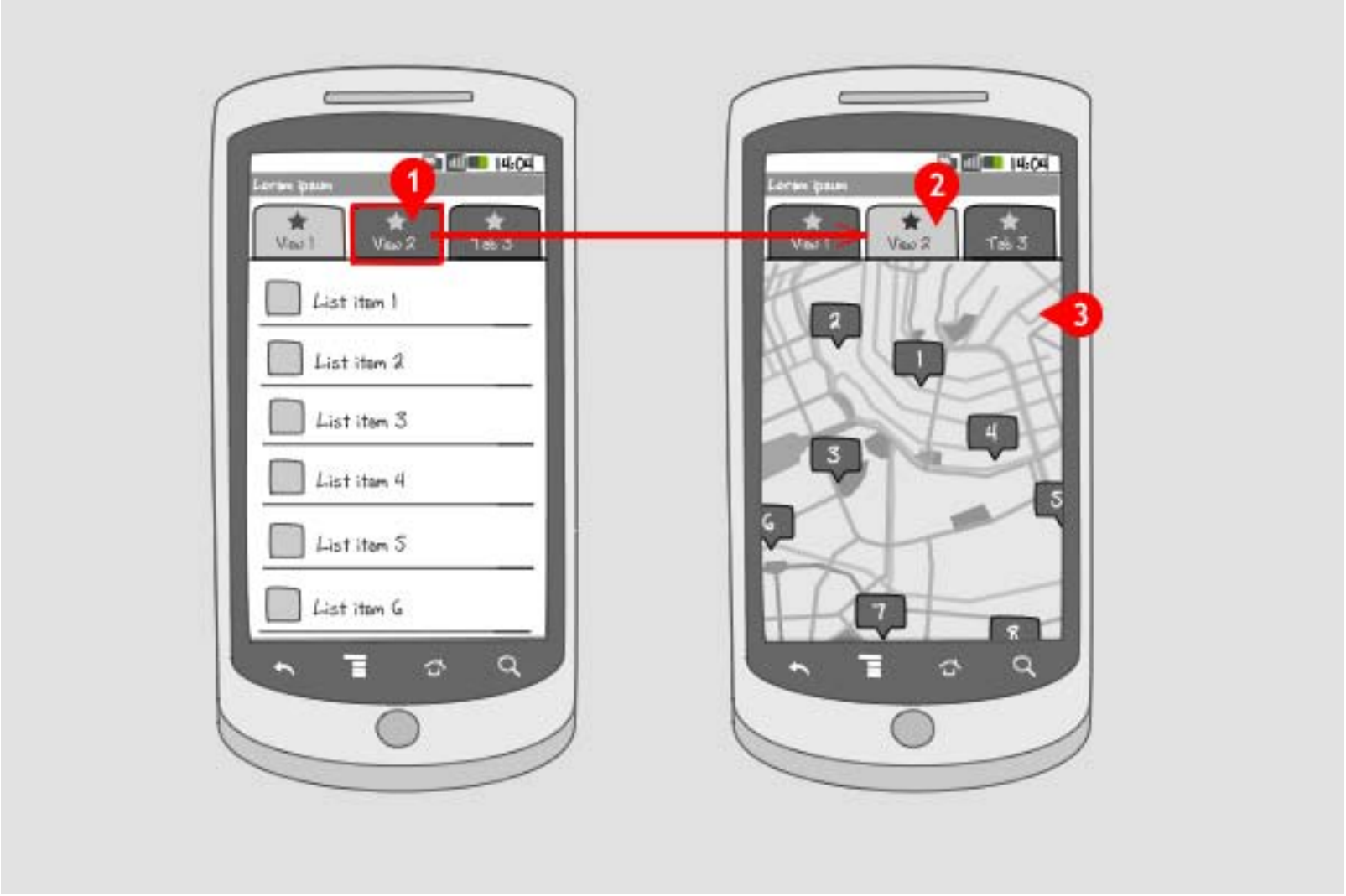




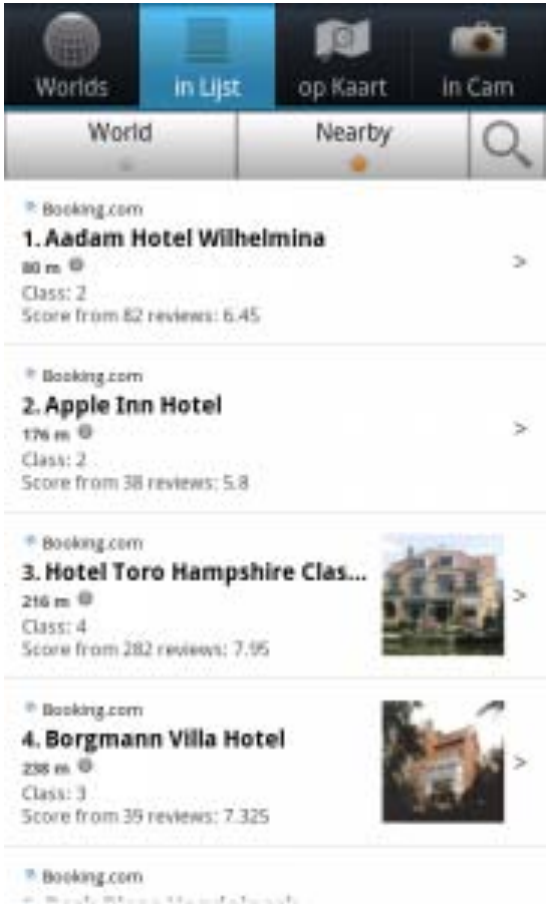
Dedicated Button

Dedicated Button

Tab bar

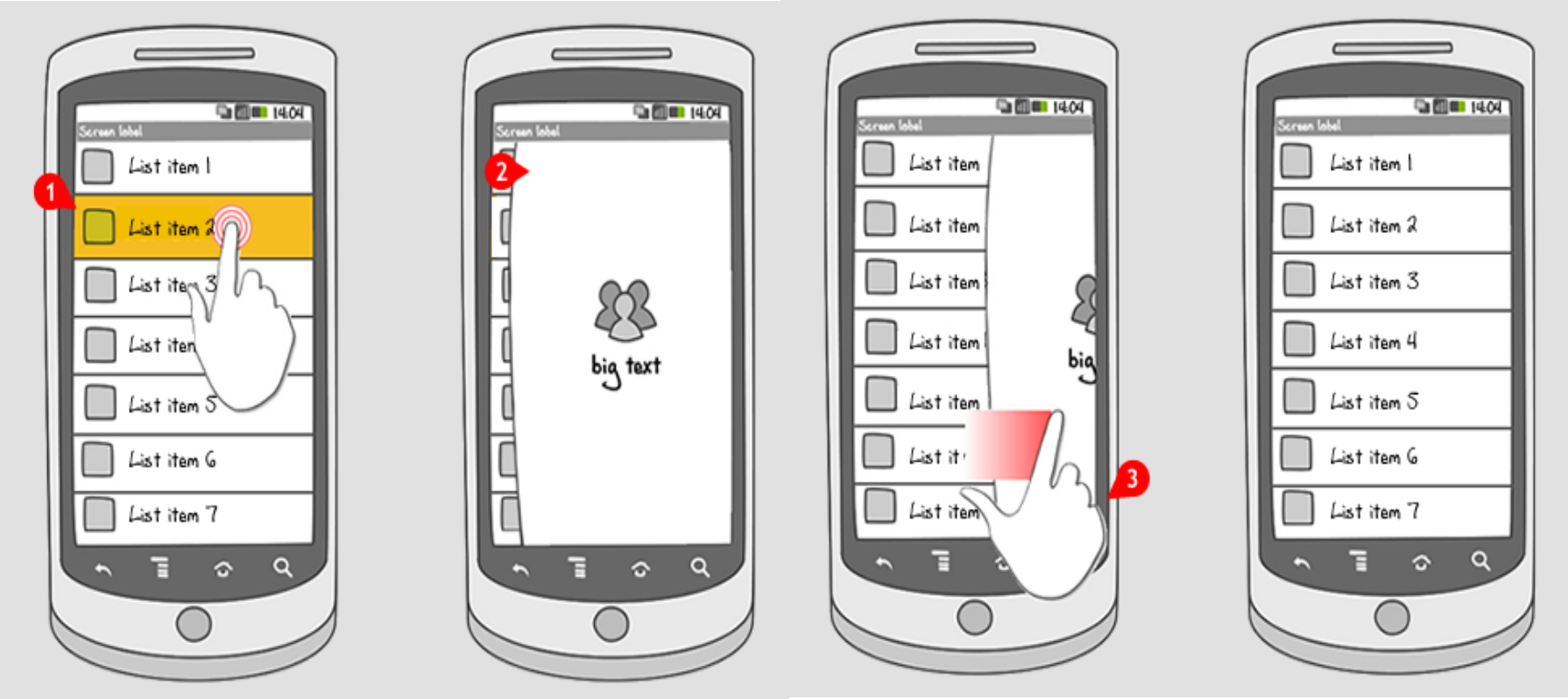

1
2
3
List item 1
List item 2
List item 3
List item 4
List item 5
List item 6

Tab bar

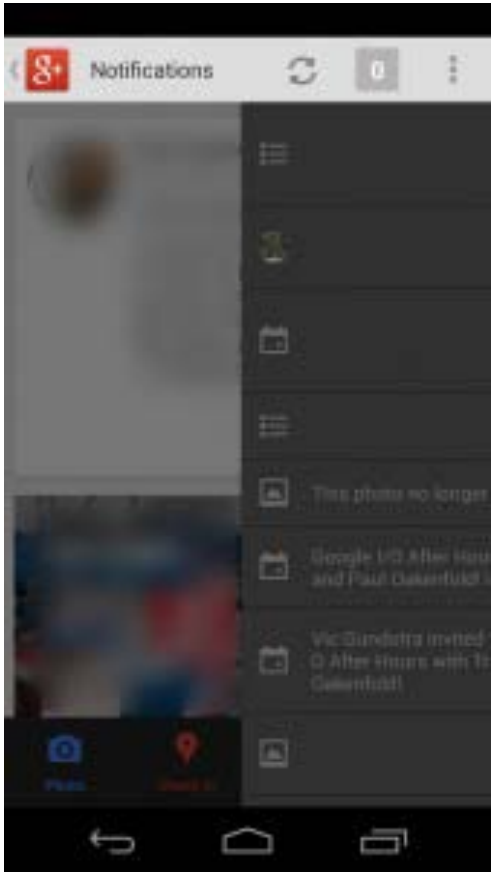

Worlds
in Lijst
op Kaart
in Cam
World
Nearby
Booking.com
1. Aadam Hotel Wilhelmina
80 m
Class: 2
Score from 82 reviews: 6.45
Booking.com
2. Apple Inn Hotel
176 m
Class: 2
Score from 38 reviews: 5.8
Booking.com
3. Hotel Toro Hampshire Clas...
216 m
Class: 4
Score from 282 reviews: 7.95
Booking.com
4. Borgmann Villa Hotel
238 m
Class: 3
Score from 39 reviews: 7.325
Booking.com

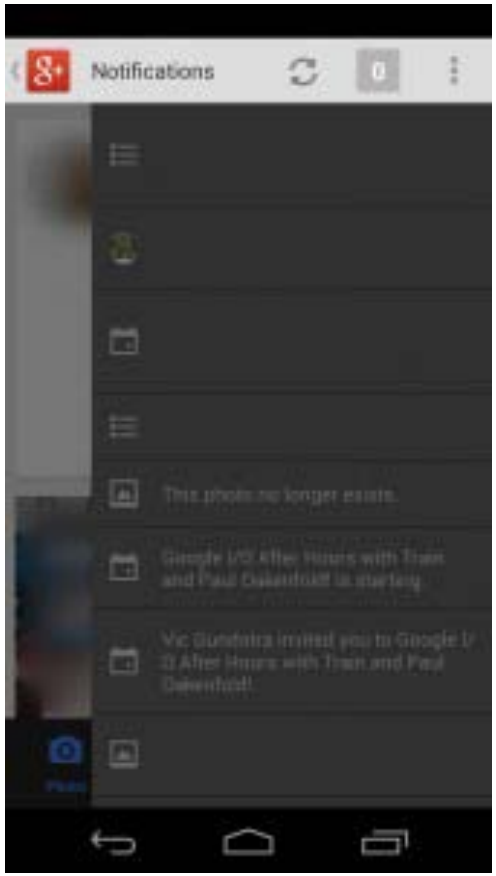

Worlds
in Lijst
op Kaart
in Cam

Sliding Layer



Sliding Layer





Android Material Design

https://developer.android.com/design/material/index.html
https://developer.android.com/training/material/get-started.html

Tugas 1.

Buat Layout Applikasi.

Compress your File (.rar/zip) that contain
(layout XMLfile & screenshot .PNG).

Documents format : PAPB-Class-NIM-T(x)
PAPB-D-0710683021-T2.rar

Collaborate : andra.course [at] gmail.com

If you are not following this rule, I will assume that you are not complete the assignment

Deadline : 4 Oktober 2016, 23:59 AM (WIB)

Thanks!

QUESTIONS?

JOIN !!