

ANDROID platforma izgradnja korisničkog interfejsa

Mobilni i distribuirani informacijski sistemi

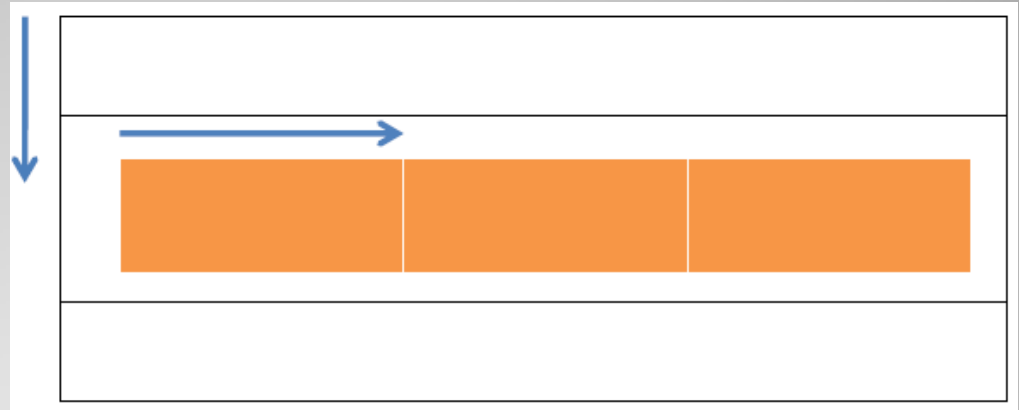
Mr Bratislav Predić

2012. godina



XML layout kontejneri

- *LinearLayout* je najčešće korišćen
- Može se ugnježdavati
- Najčešće korišćeni layout-i:
 - *LinearLayout*
 - *RelativeLayout*
 - *TableLayout*
 - *ScrollView*
 - *ListView*, *GridView*, *WebView*, *MapView*...
- *FrameLayout* je najjednostavniji i postavlja sve kontrole u gornji levi ugao jednu preko druge

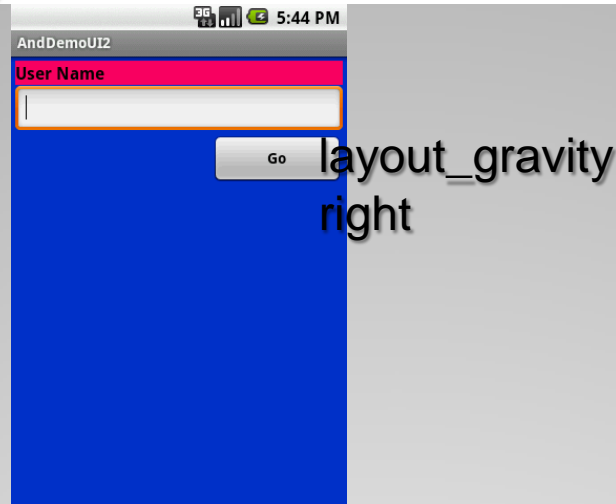
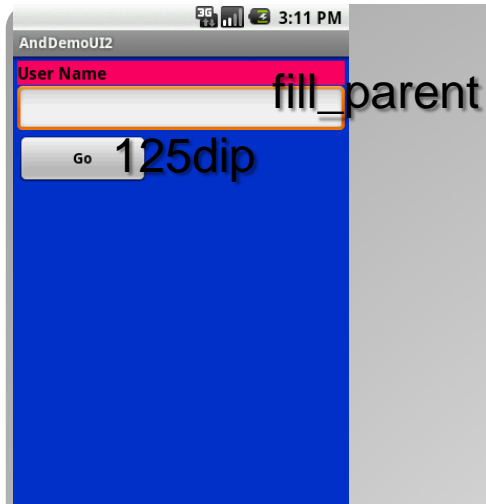


LinearLayout

- *LinearLayout* ima box model
- Elementi se ređaju u redovima ili kolonama
- 6 bitnih atributa određuju izgled:
 - orientation
 - horizontal ili vertical (*setOrientation()*)
 - layout_width ili layout_height - dimenzije
 - wrap_content ili fill_parent
 - layout_weight – proporcionalno dodeljen prostor
 - 1,2,3...
 - layout_gravity – poravnanje kontrole
 - left, center, right, top, bottom, ...
 - padding – margine u kontroli (*setPadding()*)
 - layout_margin – prostor oko kontrole



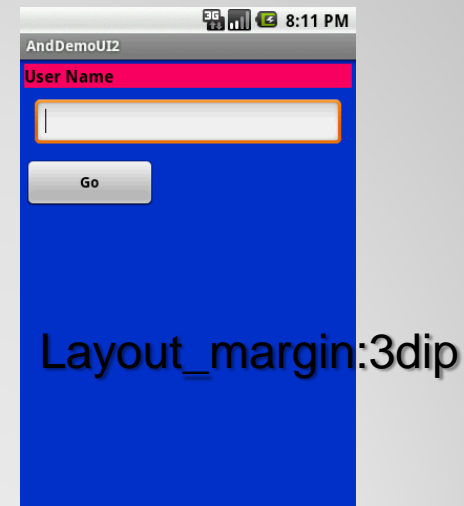
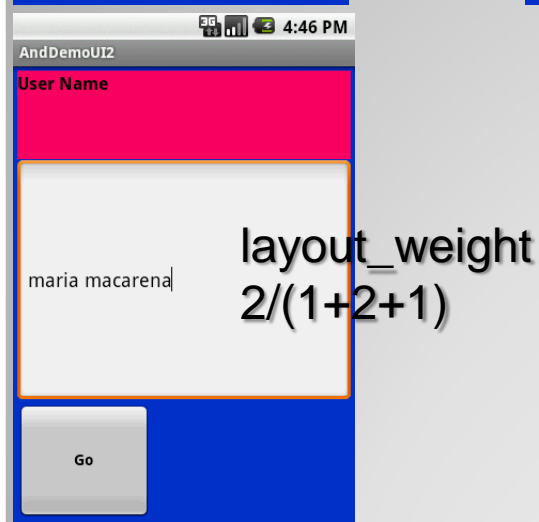
LinearLayout



gravity:center
(sadržaj kontrole)

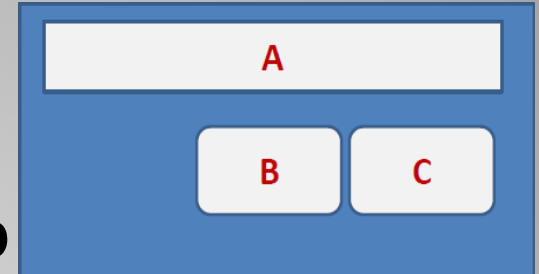


layout_gravity:center
(cela kontrola)



RelativeLayout

- *RelativeLayout* smešta widget-e relativno u odnosu na ostale elemente UI
- XML properties:
 - **android:layout_alignParentTop**
Vrh widget-a se poravnava sa vrhom kontejnera
 - **android:layout_alignParentBottom,**
android:layout_alignParentLeft,
android:layout_alignParentRight
 - **android:layout_centerInParent**
Centrira widget u kontejneru horizontalno i vertikalno
 - **android:layout_centerHorizontal,**
android:layout_centerVertical



RelativeLayout

- **android:layout_above,**
android:layout_below,
android:layout_toLeftOf,
android:layout_toRightOf

Pozicioniranje u odnosu na neki drugi widget u kontejneru.

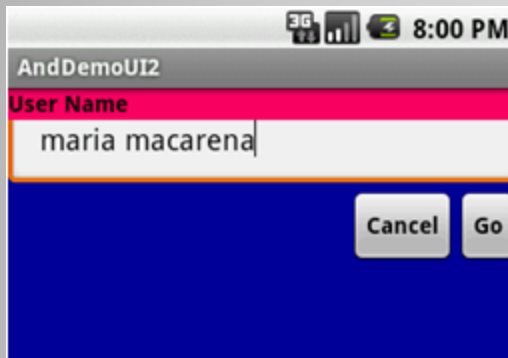
- **android:layout_alignTop,**
android:layout_alignBottom,
android:layout_alignLeft,
android:layout_alignRight,
android:layout_alignBaseline

Poravnanje u odnosu na neki drugi widget



RelativeLayout

- Kako bi koristili relativni layout svim widget-ima treba dodeliti id-jeve
android:id="@+id/ediUserName"
- Primer referenciranja
android:layout_below="@+id/ediUserName"



```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    android:id="@+id/myRelativeLayout"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="#ff000099"
    xmlns:android="http://schemas.android.com/apk/res/android">

    <TextView
        android:id="@+id/lblUserName"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:background="#ffff0066"
        android:text="User Name"
        android:textStyle="bold"
        android:textColor="#ff000000"
        android:layout_alignParentTop="true"
        android:layout_alignParentLeft="true">
    </TextView>
```

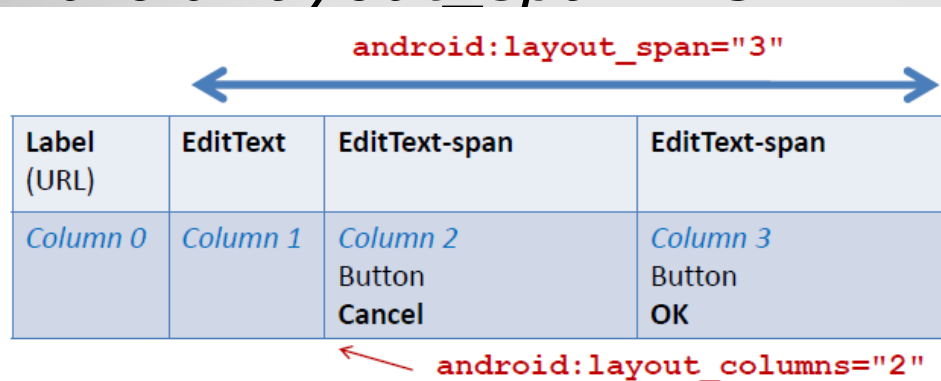
```
<EditText
    android:id="@+id/ediUserName"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/lblUserName"
    android:layout_alignParentLeft="true"
    android:layout_alignLeft="@+id/myRelativeLayout"
    android:padding="20dip">
</EditText>

<Button
    android:id="@+id/btnGo"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/ediUserName"
    android:layout_alignRight="@+id/ediUserName"
    android:text="Go"
    android:textStyle="bold">
</Button>

<Button
    android:id="@+id/btnCancel"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_toLeftOf="@+id/btnGo"
    android:layout_below="@+id/ediUserName"
    android:text="Cancel"
    android:textStyle="bold">
</Button>
</RelativeLayout>
```

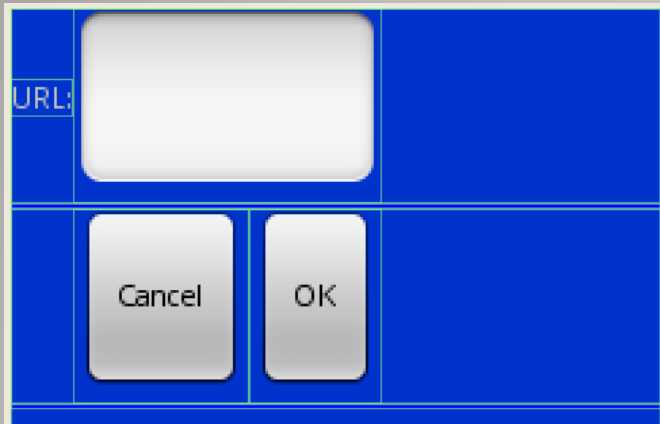
TableLayout

- *TableLayout* ima grid kolona i redova
- Kolone se šire i skupljaju u skladu sa dimenzijama widgeta koje sadrže
- *TableLayout* definiše generalno ponašanje tabele, a widget-i se dodaju u *TableRow*
- Broj kolona je automatski određen od strane sistema i broja dodatih widget-a
- Jedan widget može da zauzme više kolona
 - *android:layout_span="3"*



Label (URL)	EditText	EditText-span	EditText-span
Column 0	Column 1	Column 2 Button Cancel	Column 3 Button OK

TableLayout - primer



```
<?xml version="1.0" encoding="utf-8"?>
<TableLayout
    android:id="@+id/myTableLayout"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="#ff0033cc"
    android:orientation="vertical"
    xmlns:android="http://schemas.android.com/apk/res/android"
>
    <TableRow>
        <TextView
            android:text="URL:" />
        <EditText android:id="@+id/ediUrl"
            android:layout_span="3"/>
        </TableRow>
        <View
            android:layout_height="3dip"
            android:background="#0000FF" />
        <TableRow>
            <Button android:id="@+id/cancel"
                android:layout_column="2"
                android:text="Cancel" />
            <Button android:id="@+id/ok"
                android:text="OK" />
        </TableRow>
        <View
            android:layout_height="3dip"
            android:background="#0000FF" />
    </TableLayout>
```

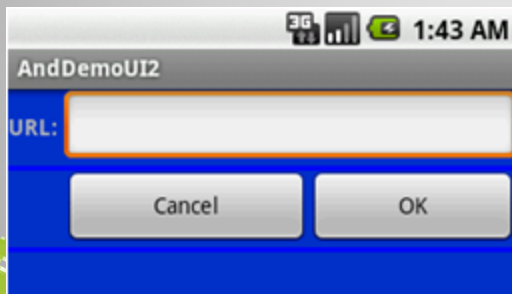
Stretch up to column 3

Skip columns: 0, 1



TableLayout

- Podrazumevano ponašanje kolone je da prilagodi svoju veličinu prirodnoj veličini widget-a koji sadrži
- Ukoliko je sadržaj manji od dostupnog prostora kolona može da popuni preostali prostor
 - *android:stretchColumns="..."*
 - Spisak kolona koje će se proširiti da zauzmu preostali prostor



```
<TableLayout
    android:id="@+id/myTableLayout"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="#ff0033cc"
    android:orientation="vertical"
    android:stretchColumns="2,3,4"
    xmlns:android="http://schemas.android
>
```

ScrollView Layout

- Ukoliko layout sadrži više nego što može da se prikaže na ekranu
- Korisnik može da skroluje layout

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView
    android:id="@+id/myScrollView1"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="#ff009999"
    xmlns:android="http://schemas.android.com/apk/res/android"
>
    <LinearLayout
        android:id="@+id/myLinearLayoutVertical"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:orientation="vertical"
    >
        <LinearLayout
            android:id="@+id/myLinearLayoutHorizontal1"
            android:layout_width="fill_parent"
            android:layout_height="fill_parent"
            android:orientation="horizontal"
        >
            <ImageView
                android:id="@+id/myPicture"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:src="@drawable/icon" />
            <TextView
                android:id="@+id/textView1"
                android:layout_width="fill_parent"
                android:layout_height="wrap_content"
                android:text="Line1"
                android:textSize="70dip" />
        </LinearLayout>
        <View
            android:layout_width="fill_parent"
            android:layout_height="6dip"
            android:background="#ffccffcc" />
    </ScrollView>
```

```
<TextView
    android:id="@+id/textView2"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Line2"
    android:textSize="70dip" />
<View
    android:layout_width="fill_parent"
    android:layout_height="6dip"
    android:background="#ffccffcc" />
<TextView
    android:id="@+id/textView3"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Line3"
    android:textSize="70dip" />
<View
    android:layout_width="fill_parent"
    android:layout_height="6dip"
    android:background="#ffccffcc" />
<TextView
    android:id="@+id/textView4"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Line4"
    android:textSize="70dip" />
<View
    android:layout_width="fill_parent"
    android:layout_height="6dip"
    android:background="#ffccffcc" />
<TextView
    android:id="@+id/textView5"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Line5"
    android:textSize="70dip" />
</LinearLayout>
</ScrollView>
```



AbsoluteLayout

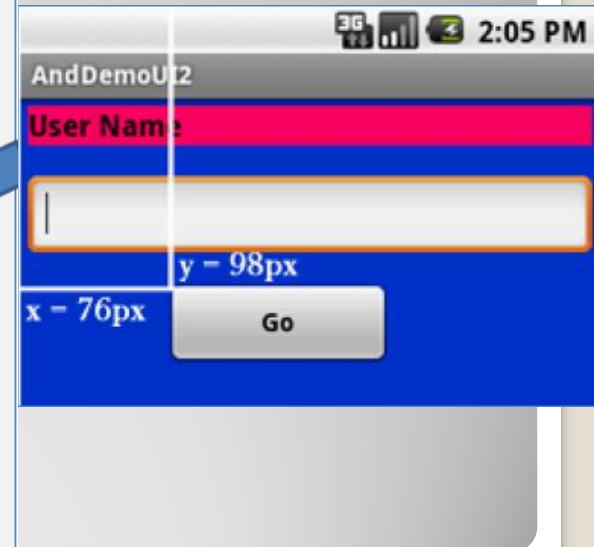
- Ređe se koristi
- Svakom widget-u možemo eksplicitno da odredimo poziciju (u pikselima)
- Ovakvi layout-i se teško održavaju i prilagođavaju većem broju uređaja

```
<?xml version="1.0" encoding="utf-8"?>
<AbsoluteLayout
    android:id="@+id/myLinearLayout"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="#ff0033cc"
    android:padding="4dip"
    xmlns:android="http://schemas.android.com/apk/res/android"
>

    <TextView
        android:id="@+id/tvUserName"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:background="#ffff0066"
        android:text="User Name"
        android:textSize="16sp"
        android:textStyle="bold"
        android:textColor="#ff000000"
        android:layout_x="0dip"
        android:layout_y="10dip"
    >

        </TextView>
        <EditText
            android:id="@+id/etName"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:textSize="18sp"
            android:layout_x="0dip"
            android:layout_y="38dip"
        >
        </EditText>

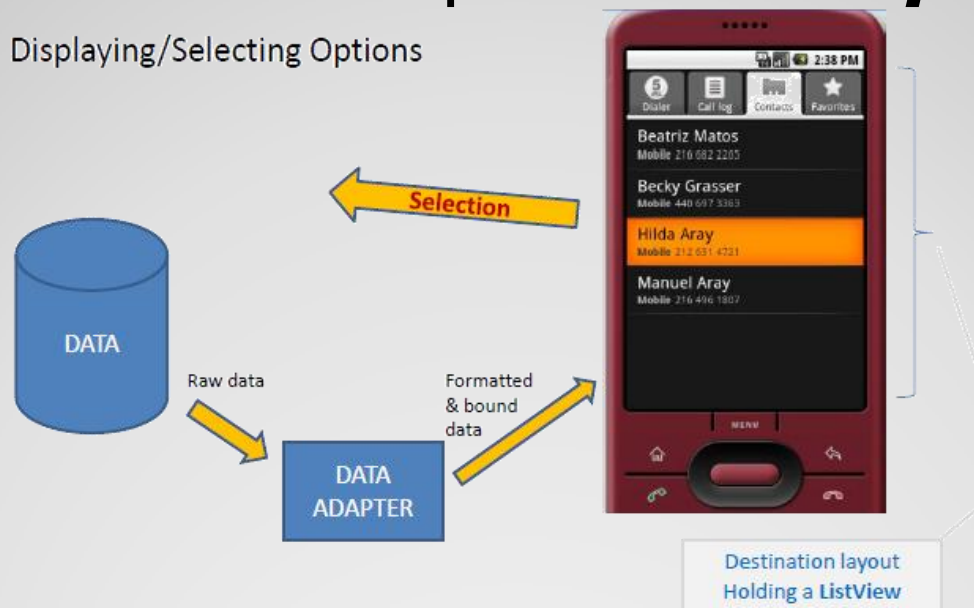
        <Button
            android:layout_width="120dip"
            android:text="Go"
            android:layout_height="wrap_content"
            android:textStyle="bold"
            android:id="@+id/btnGo"
            android:layout_x="100dip"
            android:layout_y="170dip" />
    </AbsoluteLayout>
```



Widget-i za selekciju

- Za manji broj opcija
 - *RadioButton* i *CheckBox*
- Za veći broj opcija
 - listbox, combobox, drop-down liste, galerije slika
- U pozadini ovih kontrola je model podataka povezan kroz koncept ***data adapter-a***

Displaying/Selecting Options



Data adapter-i

- Najjednostavniji primer : **ArrayAdapter**
 - Wrapuje se oko niza ili `java.util.List`

```
String[] items={"this", "is", "a","really", "silly", "list"};

new ArrayAdapter<String>(this,
                        android.R.layout.simple_list_item_1,
                        items);
```

- Parametri konstruktora
 - Context – tipično activity
 - Resource ID view-a koji se koristi za red
 - *android.R.layout.simple_list_item_1* je jedan od predefinisanih
 - Array ili List koji se povezuje



Jednostavna lista - ListActivity

- Koristimo *ListActivity*
- Layout može i da se prilagodi
 - *@android:id/list* view za prikaz liste
 - *@android:id/empty* se prikazuje kada nema podataka

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >
    <TextView
        android:id="@+id/selection"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:background="#ff0000cc"
        android:textStyle="bold" />
    <!-- Here is the list. Since we are using a ListActivity, we have
    <ListView
        android:id="@android:id/list"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:drawSelectorOnTop="false" />

    <TextView android:id="@android:id/empty"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Empty set" />
</LinearLayout>
```



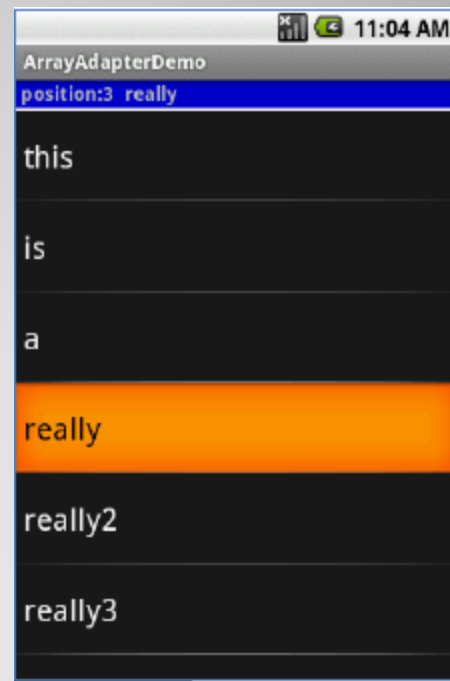
Jednostavna lista – ListActivity

- Koristimo *ListActivity* umesto *Activity*

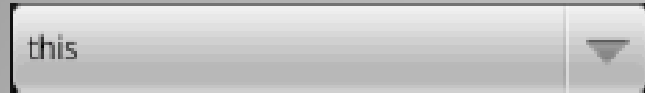
```
public class ArrayAdapterDemo extends ListActivity {  
  
    TextView selection;  
    String[] items = { "this", "is", "a", "really",  
                      "really2", "really3", "really4",  
                      "really5", "silly", "list" };  
}
```

- Povezujemo sa podacima korišćenjem adaptera

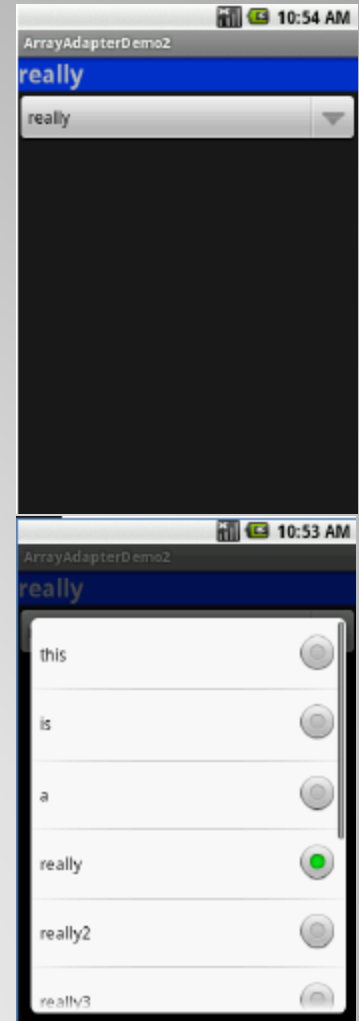
```
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.main);  
  
    setListAdapter(new ArrayAdapter<String>(  
        this,  
        android.R.layout.simple_list_item_1,  
        items));  
  
    selection=(TextView) findViewById(R.id.selection);  
}
```



SpinControl



- Kao i sa *ListView* podaci se povezuju adapterom sa *setAdapter()*
- Izbor se hvata u metodi *setOnItemSelectedListener()*
- Metoda *setDropDownViewResource()* postavlja view za izbor stavki



Ostali selection Widget-i

```
@Override
public void onCreate(Bundle icle) {
    super.onCreate(icle);
    setContentView(R.layout.main);

    selection = (TextView) findViewById(R.id.selection);

    Spinner spin = (Spinner) findViewById(R.id.spinner);
    spin.setOnItemSelectedListener(this);
    // set a clickable right push-button comboBox to show items
    ArrayAdapter<String> aa = new ArrayAdapter<String>(
        this, android.R.layout.simple_spinner_item, items);
    // provide a particular design for the drop-down lines
    aa.setDropDownViewResource(
        android.R.layout.simple_spinner_dropdown_item);
    // associate GUI spinner and adapter
    spin.setAdapter(aa);
}

////////////////////////////////////
public void onItemSelected(
    AdapterView<?> parent, View v, int position, long id) {
    selection.setText(items[position]);
}

public void onNothingSelected(AdapterView<?> parent) {
    selection.setText("");
}
}
```



GridView

- Prikazuje stavke u dvodimenzionalnom grid-u koji može da se skroluje
- Takođe koristi *ListAdapter*
- Neki bitni atributi (properties)
 - *android:numColumns* – br. kolona
 - *android:verticalSpacing*
android:horizontalSpacing
razmak između stavki
 - *android:stretchMode*
Šta se dešava sa viškom prostora



GridView

- Primer

- Ekran širine 320px
- *android:columnWidth* – 10px
- *android:horizontalSpacing* – 5px
- GridView zauzima 310px
- *android:stretchMode* postavljen na *columnWidth* će povećati svaku kolonu za 3-4px
- *android:stretchMode* postavljen na *spacingWidth* će svaki razmak povećati sa 5 na 10px

```
String[] items= { "this", "is", "a", "really", "really2",  
                  "really3","really4", "really5", "silly", "list"};  
GridViewgv= (GridView) findViewById(R.id.grid);  
ArrayAdapter<String> aa= new ArrayAdapter<String>(this,  
                                                  android.R.layout.simple_list_item_1,items);  
gv.setAdapter(aa);  
gv.setOnItemClickListener(this);
```

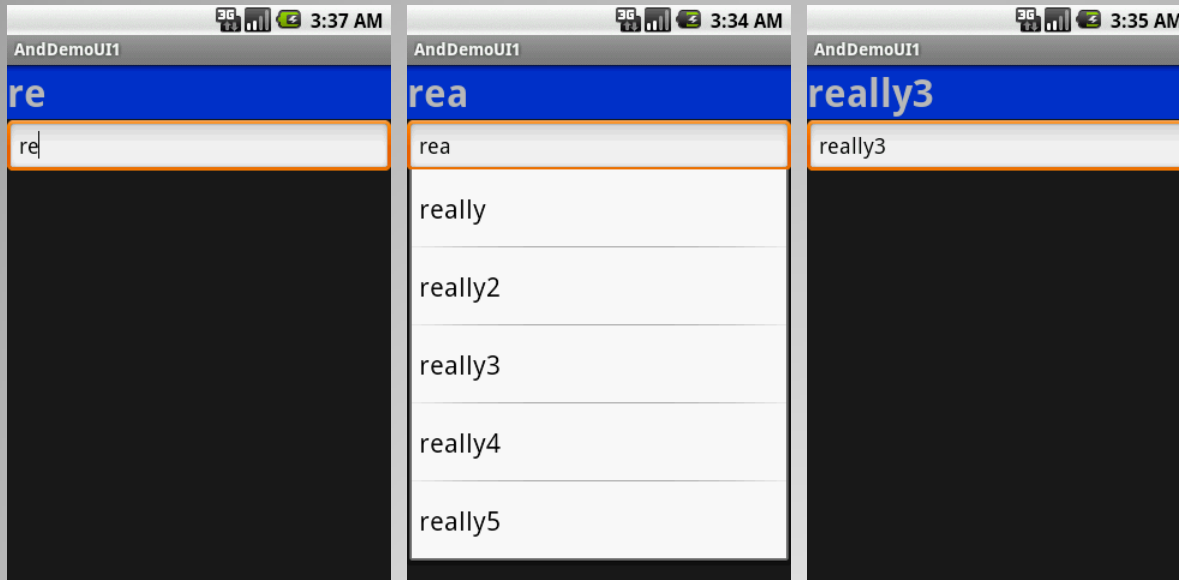


AutoCompleteTextView

- Kako korisnik unosi tekst, uneti tekst se tretira kao prefiks filter za listu kandidata
- Lista kandidata se prikazuje kao selection lista (slično kao kod Spinner-a)
- Korisnik može da izabere stavku ponuđenu u listi
- Podklasa *EditText* klase tako da nasleđuje sve look-and-feel attribute (font, boja itd.)
- Bitan property:
android:completionThreshold – posle koliko unetih karaktera se pokazuje lista ponuđenih



AutoCompleteTextView



```
<AutoCompleteTextView
    android:id="@+id/edit"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:completionThreshold="3"/>
```

```
String[] items= { "this", "is", "a", "really", "really2", "really3",
    "really4", "really5", "silly", "list"};
edit= (AutoCompleteTextView) findViewById(R.id.edit);
edit.addTextChangedListener(this);
edit.setAdapter(new ArrayAdapter<String>(this,
    android.R.layout.simple_dropdown_item_1line, items));
```

GalleryWidget

- Skup opcija predstavljenih slikama

```
<Gallery
    android:id="@+id/myGallery"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:gravity="bottom"
/>
```



- Mogu se koristiti slike i kod GridView-a
 - Treba implementirati *ImageAdapter*

```
<GridView
    android:id="@+id/gridview"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:numColumns="auto_fit"
    android:verticalSpacing="10dp"
    android:horizontalSpacing="10dp"
    android:columnWidth="90dp"
    android:stretchMode="columnWidth"
    android:gravity="center"
    android:background="#ff777777" />
```



GridView - ImageAdapter

```
public class ImageAdapter extends BaseAdapter {
    private Context mContext;
    public ImageAdapter(Context c) {
        mContext = c;
    }
    public int getCount() {
        return mThumbIds.length;
    }
    public Object getItem(int position) {
        Toast.makeText(getApplicationContext(), "Pos: " + position, 1).show();
        return null;
    }
    public long getItemId(int position) { return 0; }

    public View getView(int position, View convertView, ViewGroup parent) {
        ImageView imageView;
        if (convertView == null) {
            imageView = new ImageView(mContext);
            imageView.setLayoutParams(new GridView.LayoutParams(85, 85));
            imageView.setScaleType(ImageView.ScaleType.CENTER_CROP);
            imageView.setPadding(8, 8, 8, 8); }
        else {
            imageView = (ImageView) convertView;
        }
        imageView.setImageResource(mThumbIds[position]);
        return imageView;
    }
}
```


Custom liste

- Koriste se u slučajevima
 - Različiti redovi imaju različit layout (npr. Neki sa jednom, neki sa dve linije teksta)
 - Kada reovi sadrže specifične widget-e
- U ovom slučaju se kreira posebna podklasa Adapter-a
- Kada se nasleđuje Adapter treba:
 - Predefinisati *getView()* metodu
 - Napraviti ručno redove
- Metoda *getView()* treba da vrati View koji predstavlja red na zadatoj poziciji



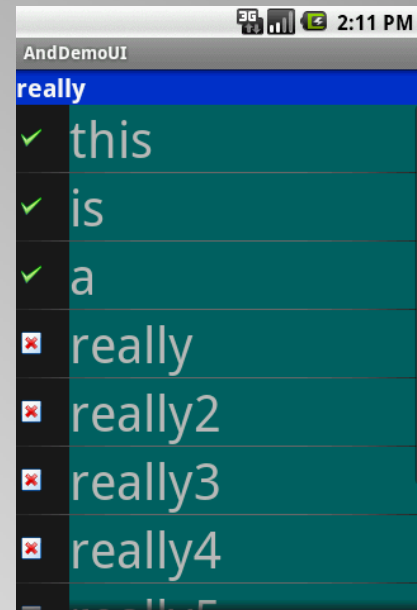
Custom liste - primer

- Različite ikone za redove u zavisnosti od dužine reči
- Pravimo layout za red

```
<ImageView android:id="@+id/icon"  
    android:layout_width="22px"  
    android:paddingLeft="2px"  
    android:paddingRight="2px"  
    android:paddingTop="2px"  
    android:layout_height="wrap_content"  
    android:src="@drawable/ok"/>
```

```
<TextView  
    android:id="@+id/label"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:textSize="40sp"/>
```

```
Public class AndDemoUI extends ListActivity{  
    ...  
    public void onCreate(Bundle icicle) {  
        ...  
        setListAdapter(new IconicAdapter(this));  
        ...  
    }  
}
```



Custom liste - primer

- IconicAdapter

```
Class IconicAdapter extends ArrayAdapter {  
    Activity context;
```

```
    IconicAdapter(Activity context) {  
        super(context, R.layout.myrow, items);  
        this.context= context;  
    }
```

```
    public View getView(int position, View convertView, ViewGroup parent) {  
        LayoutInflater inflater = context.getLayoutInflater();  
        View row = inflater.inflate(R.layout.myrow, null);  
        TextView label = (TextView) row.findViewById(R.id.label);  
        ImageView icon = (ImageView) row.findViewById(R.id.icon);  
        label.setText(items[position]);  
        if(items[position].length() > 4)  
            icon.setImageResource(R.drawable.delete);  
        else  
            icon.setImageResource(R.drawable.ok);  
        return (row);  
    } //getView  
}
```

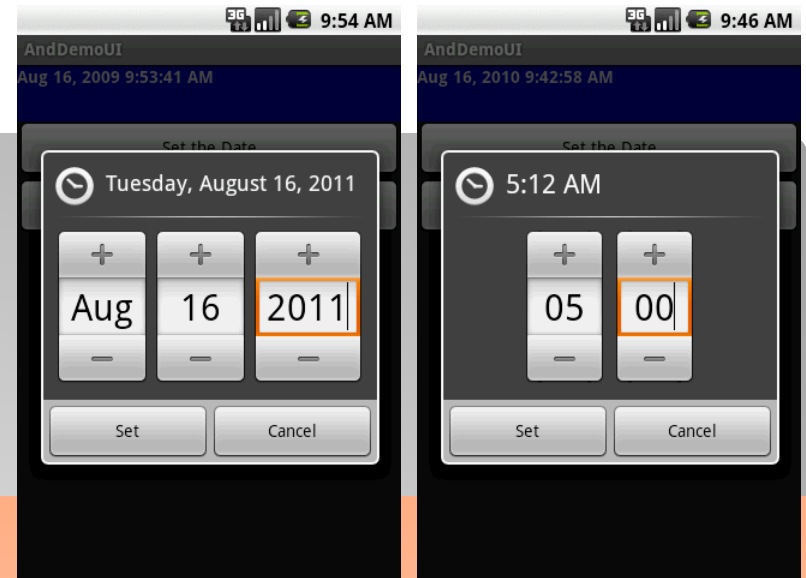
IconicAdapter - objašnjenje

- LayoutInflater se koristi da "raspakuje" XML opis layout-a i od njega napravi stablo View objekata
- ArrayAdapter uzima tri argumenta
 - Trenutni context
 - Layout za prikaz reda
 - Podatke za popunjavanje redova
- Predefinisana *getView()* metoda
 - Kreira (inflates) layout
 - Specijalizira novokreirani view tekstem i ikonom
 - Vraća kreirani view (red)



Calendar widget-i

- DatePickerDialog
- TimePickerDialog



```
Calendar myCalendar = Calendar.getInstance();
...
DatePickerDialog.OnDateSetListener d= new DatePickerDialog.OnDateSetListener() {
    public void onDateSet(DatePicker view,int year,int monthOfYear,int dayOfMonth) {
        myCalendar.set(Calendar.YEAR, year);
        myCalendar.set(Calendar.MONTH, monthOfYear);
        myCalendar.set(Calendar.DAY_OF_MONTH, dayOfMonth);
        updateLabel();
    }
};

TimePickerDialog.OnTimeSetListener t = new TimePickerDialog.OnTimeSetListener() {
    public void onTimeSet(TimePicker view, int hourOfDay, int minute) {
        myCalendar.set(Calendar.HOUR_OF_DAY, hourOfDay);
        myCalendar.set(Calendar.MINUTE, minute);
        updateLabel();
    }
};
```

Calendar widget-i

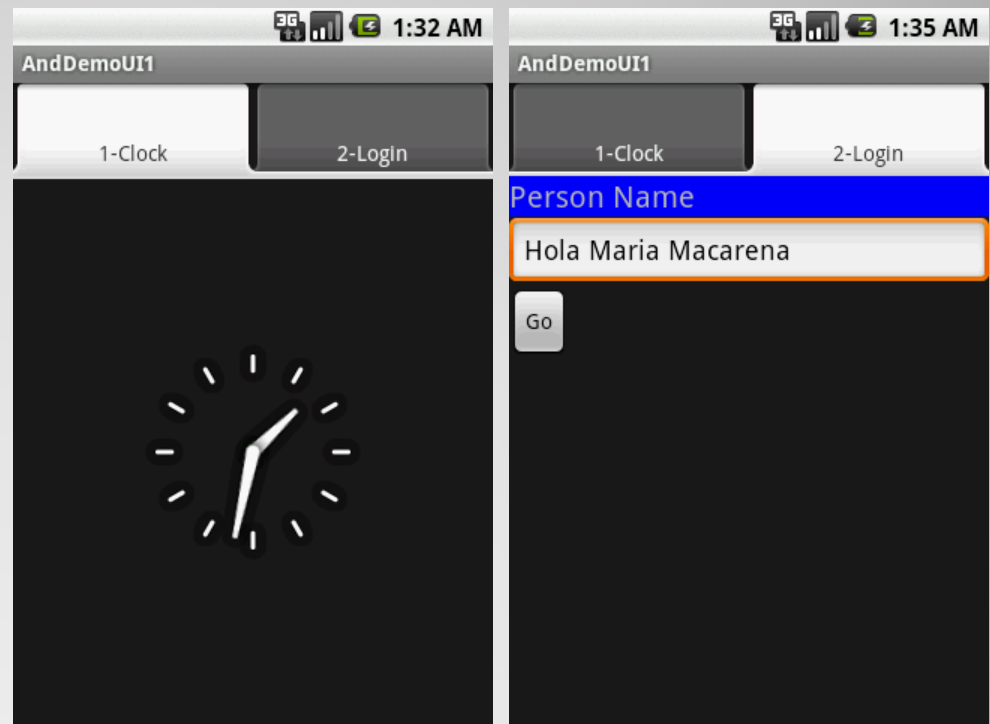
- Aktiviramo picker-e klikom na dugme

```
...
Button btnDate= (Button) findViewById(R.id.btnDate);
btnDate.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        new DatePickerDialog(AndDemoUI.this, d,
            myCalendar.get(Calendar.YEAR),
            myCalendar.get(Calendar.MONTH),
            myCalendar.get(Calendar.DAY_OF_MONTH)).show();
    }
});
Button btnTime= (Button) findViewById(R.id.btnTime);
btnTime.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        new TimePickerDialog(AndDemoUI.this, t,
            myCalendar.get(Calendar.HOUR_OF_DAY),
            myCalendar.get(Calendar.MINUTE), true).show();
    }
});
```

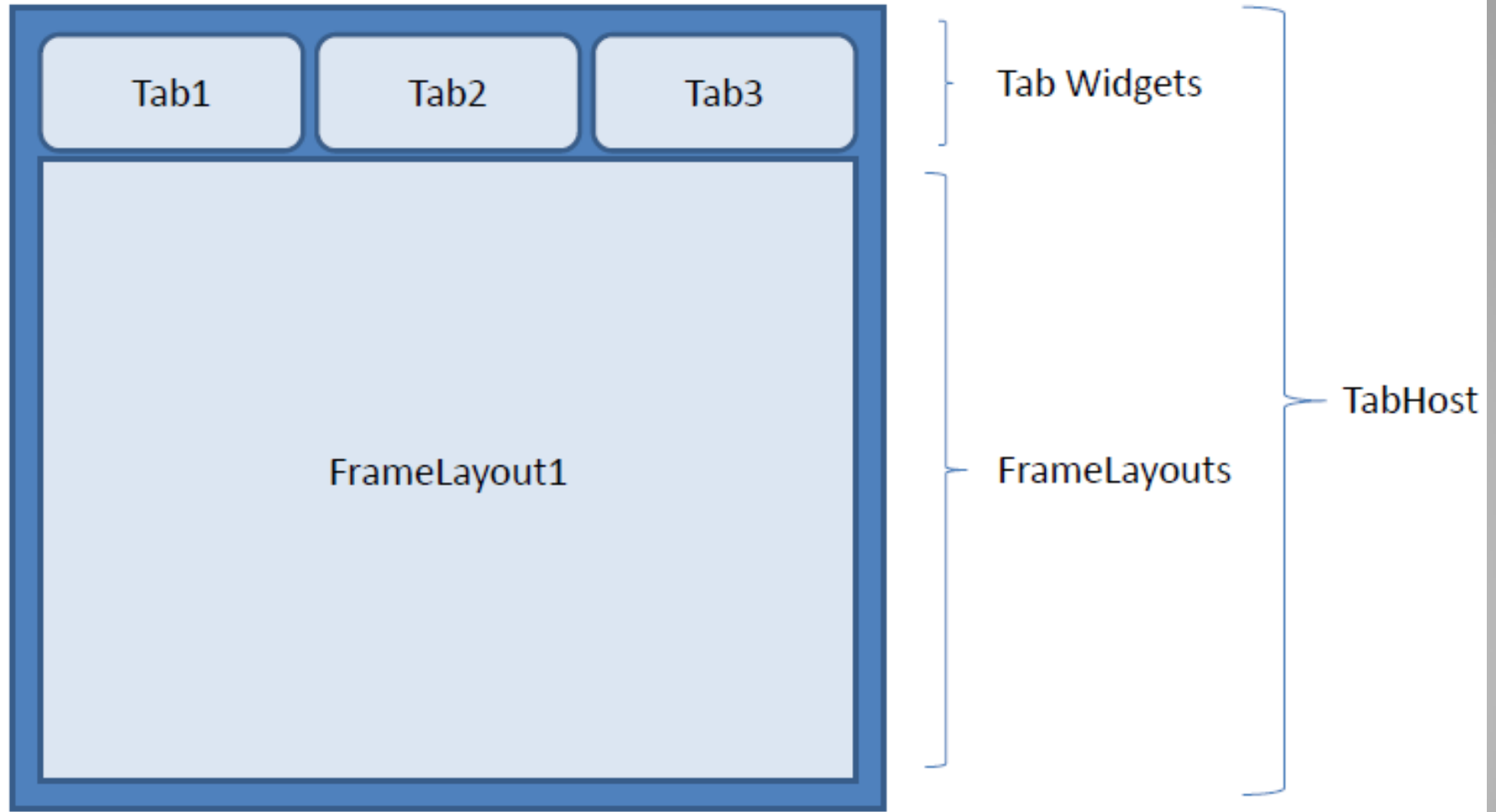


Tab selector widget

- Preporuka je da Android UI bude što je moguće jednostavniji
- Ako imamo mnogo informacija za prikaz tab widget se koristi da se te informacije logički podele
- U kreiranju tab widget-a učestvuju
 - **TabHost**
Glavni kontejner
 - **TabWidget**
Dugmići za tabove
 - **FrameLayout**
Sadržaj tab-a



Tab selector widget



Tab selector widget

- Layout

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android=http://schemas.android.com/apk/res/android
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <TabHost android:id="@+id/tabhost"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent">
        <TabWidget android:id="@android:id/tabs"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"/>
        <FrameLayout android:id="@android:id/tabcontent"
            android:layout_width="fill_parent"
            android:layout_height="fill_parent"
            android:paddingTop="62px">

            <!--PUT HERE FrameLayout1 -->
            <!--PUT HERE FrameLayout2 -->

        </FrameLayout>
    </TabHost>
</LinearLayout>
```

Tab selector widget

- **FrameLayout1 i FrameLayout2 je najbolje include-ovati i držati u posebnom layout-u**

```
<LinearLayout android:id="@+id/tab2"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical"
    xmlns:android="http://schemas.android.com/apk/res/android">
    <TextView android:id="@+id/caption1"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:background="#ff0000ff"
        android:text="Person Name"
        android:textSize="20px">
    </TextView>
    <EditText android:id="@+id/txtPerson"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="txtPerson"
        android:textSize="18sp">
    </EditText>
    <Button
        android:id="@+id/btnGo"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Go">
    </Button>
</LinearLayout>
```

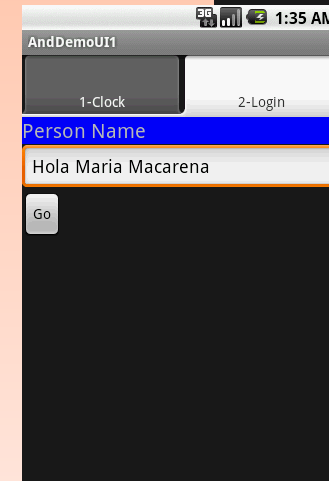
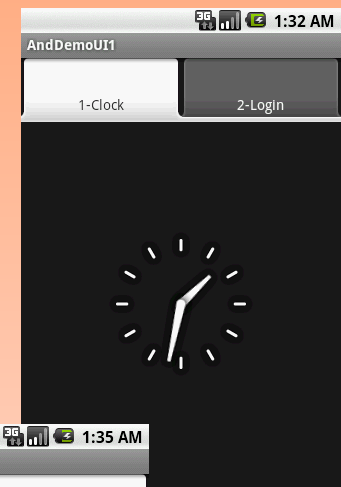
Upotreba tab-ova

```
public class AndDemoUI1 extends Activity {
    @Override
    public void onCreate(Bundle icle) {
        super.onCreate(icle);
        setContentView(R.layout.main);
        TabHost tabs=(TabHost)findViewById(R.id.tabhost);
        tabs.setup();

        TabHost.TabSpec spec;
        spec =tabs.newTabSpec("tag1");
        spec.setContent(R.id.tab1);
        spec.setIndicator("1-Clock");
        tabs.addTab(spec);

        spec=tabs.newTabSpec("tag2");
        spec.setContent(R.id.tab2);
        spec.setIndicator("2-Login");
        tabs.addTab(spec);

        tabs.setCurrentTab(0);
    }
}
```



- Možemo postaviti i ikonu

```
spec.setIndicator("2-Login",
    getResources().getDrawable(R.drawable.ic_menu_info_details));
```

Upotreba tab-ova

- Ukoliko želimo da detektujemo promenu tab-a
 - Registrujemo odgovarajući listener

```
tabs.setOnTabChangeListener(  
    new OnTabChangeListener() {  
        @Override  
        public void onTabChanged(String tagId) {  
            String text = "Imcurrently in: "+ tagId+ "\nindex: "+  
                tabs.getCurrentTab();  
            Toast.makeText(getApplicationContext(), text, 1).show();  
        }  
    });
```

