

Pogledi (1)

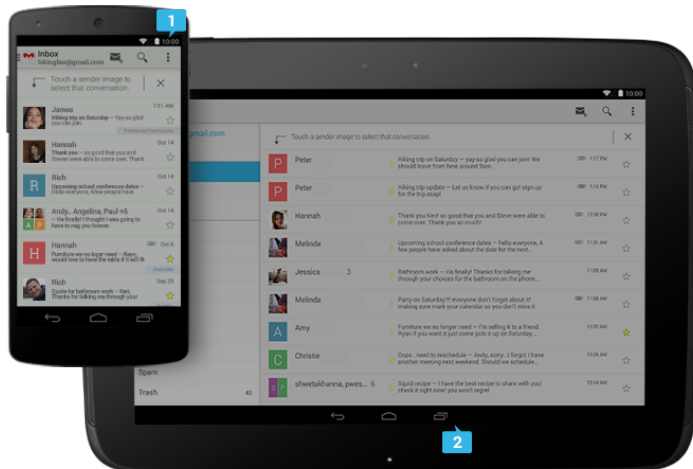
Android App Development

Pregled sadržaja

1 Grafički korisnički interfejs

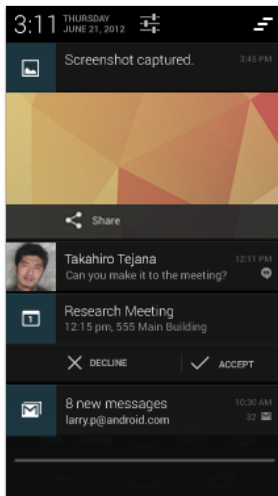
2 Pogledi

GUI



Slika 1: Statusna linija i navigaciona linija.

GUI

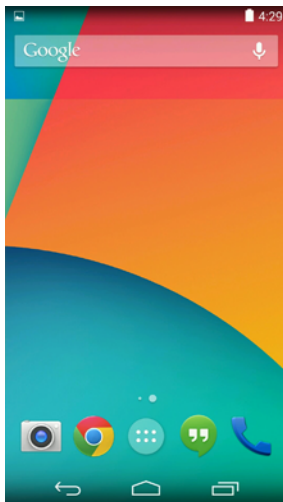


Slika 2: Obaveštenja.

GUI

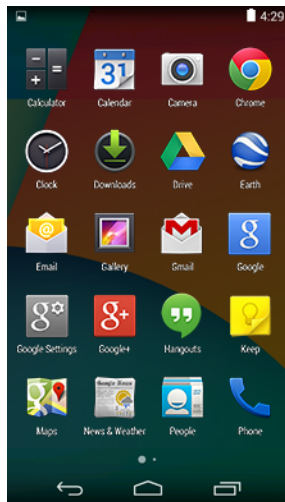
- Home Screen
- All Apps
- Overview Space (Recents Screen)
- Notifications
- App Screen (common app GUI)

GUI



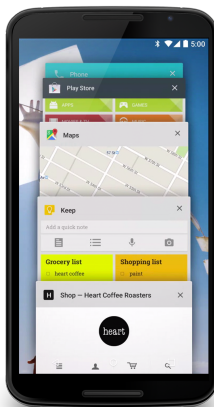
Slika 3: Home Screen.

GUI



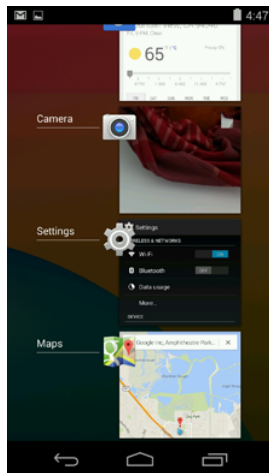
Slika 4: All Apps.

GUI



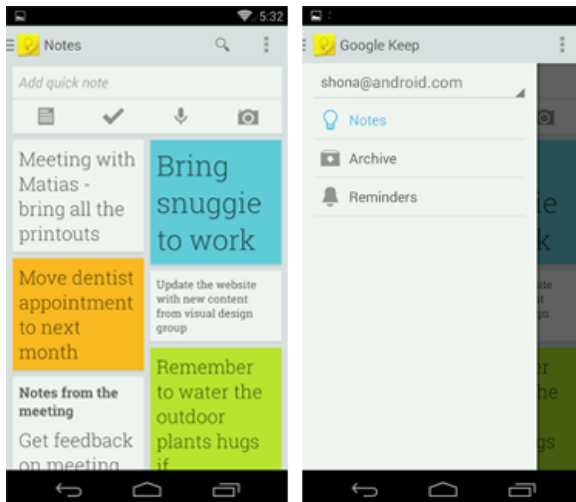
Slika 5: Overview Space

GUI



Slika 6: Recents Screen.

GUI



Slika 7: Toolbar, Navigation Drawer & Content Area.

GUI



Slika 8: Toobar (App Bar).

- ① app icon
- ② view control
- ③ action buttons
- ④ action overflow

GUI

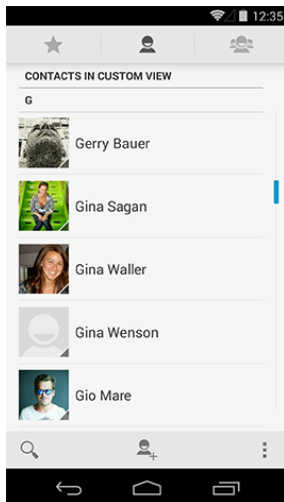
Gestovi

- Touch
- Long press
- Swipe/Drag
- Long press drag
- Double touch
- Double touch drag
- Pinch open
- Pinch close

Navigacija

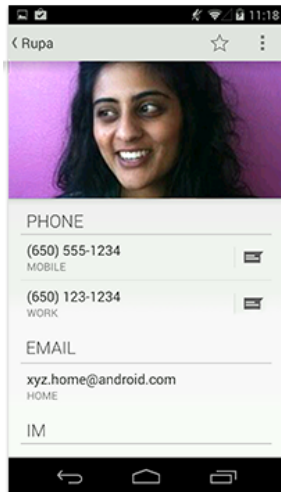
- Modelovati podatke (npr. dijagram klasa, dijagram entiteta i veza, itd.)
- Skicirati korisnički interfejs kao skup ekrana i njihovih veza (npr. koristeći mock-up alat)
- Implementirati korisnički interfejs

Navigacija



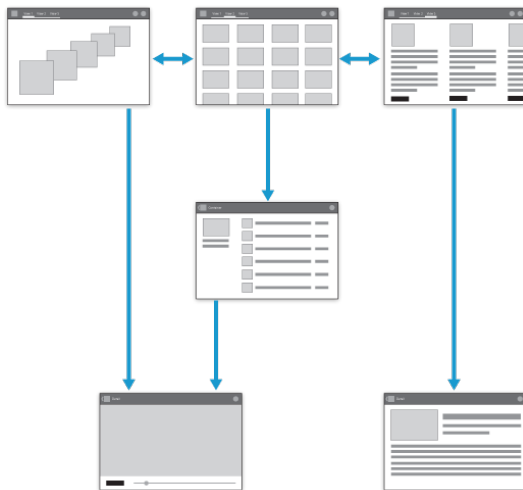
Slika 9: Kolekcija entiteta.

Navigacija



Slika 10: Pojedinačni entitet.

Navigacija



Slika 11: Struktura GUI-a.

Pregled sadržaja

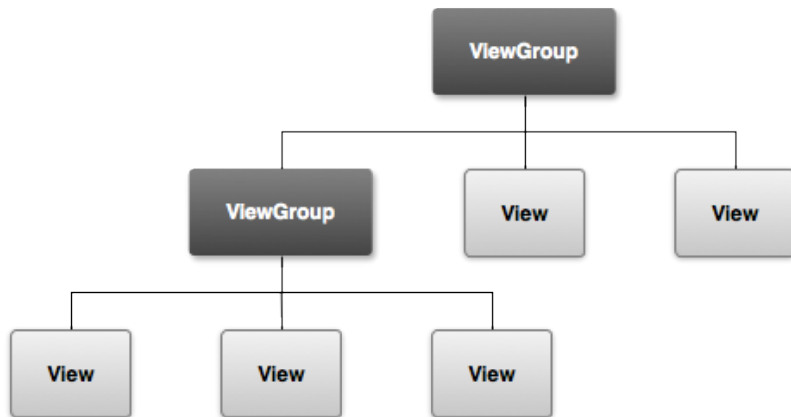
1 Grafički korisnički interfejs

2 Pogledi

Pogledi i rasporedi

- Grafički korisnički interfejs bilo koje aktivnosti može se predstaviti hijerarhijom pogleda (view) i rasporeda (layout)
- Pogledi predstavljaju elemente GUI-a, a rasporedi poglede koji sadrže druge poglede i određuju kako se oni raspoređuju na ekranu

Pogledi i rasporedi



Slika 12: Hijerarhija pogleda i rasporeda GUI-a.

Pogledi

- Elementi Android GUI-a su pogledi
- Android sadrži predefinisane tipove pogleda (npr. labele, tekstualna polja, dugmad, itd.)
- Moguće je definisati nove tipove pogleda (mada se obično koriste predefinisani)

Svojstva pogleda

- Stanje pogleda određeno je njegovim svojstvima
- Postoje svojstva koja su zajednička za sve tipove pogleda (npr. vidljivost, transparentnost, itd.), a neki tipovi pogleda mogu da sadrže i posebna svojstva

Svojstva pogleda

Svojstvo	Opis
id	supply an identifier name for this view
clickable	defines whether this view reacts to click events
focusable	boolean that controls whether a view can take focus
visibility	controls the initial visibility of the view
alpha	controls the initial transparency of the view

Tabela 1: Važnija svojstva pogleda.

Događaji

- Pogledi mogu obrađivati različite događaje (dodir, pritisak tastera, promenu fokusa, itd.)

Pogledi

Događaj	Opis
onTouch()	the user performs an action qualified as a touch event)
onClick()	the user touches the item
onLongClick()	the user either touches and holds the item
onFocusChange()	the user navigates onto or away from the item
onKey()	the user is focused on the item and presses or releases a hardware key on the device
onCreateContextMenu()	this is called when a Context Menu is being built

Tabela 2: Važniji događaji.

Pravljenje pogleda

- Pogledi se mogu definisati instanciranjem objekata u Java kodu ili dodavanjem elemenata u XML kodu
- Na sličan način mogu se postaviti svojstva i obrađivači događaja pogleda

Pravljenje pogleda

ExampleActivity.java

```
1  public class ExampleActivity extends Activity {  
2      protected void onCreate(Bundle state) {  
3          super.onCreate(state);  
4  
5          setContentView(R.layout.main);  
6  
7          Button button = new Button(this);  
8  
9          LinearLayout layout = (LinearLayout) findViewById(R.id.linear_layout);  
10         layout.addView(button);  
11     }  
12 }
```

Pravljenje pogleda

main.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      android:id="@+id/linear_layout"
4      android:orientation="vertical">
5
6      <Button android:id="@+id/button" />
7
8  </LinearLayout>
```

Postavljanje svojstava

ExampleActivity.java

```
1  public class ExampleActivity extends Activity {  
2      protected void onCreate(Bundle state) {  
3          super.onCreate(state);  
4  
5          setContentView(R.layout.main);  
6  
7          Button button = (Button) findViewById(R.id.button);  
8          button.setLayoutParams(new LayoutParams(  
9              ViewGroup.LayoutParams.WRAP_CONTENT,  
10             ViewGroup.LayoutParams.WRAP_CONTENT));  
11         button.setText(R.string.button_text);  
12     }  
13 }
```

Postavljanje svojstava

main.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      android:id="@+id/linear_layout"
4      android:orientation="vertical">
5
6      <Button
7          android:id="@+id/button"
8          android:layout_width="wrap_content"
9          android:layout_height="wrap_content"
10         android:text="@string/button_text" />
11
12  </LinearLayout>
```

Obrada događaja

ExampleActivity.java

```
1  public class ExampleActivity extends Activity {  
2      protected void onCreate(Bundle bundle) {  
3          super.onCreate(state);  
4  
5          setContentView(R.layout.main);  
6  
7          Button button = new Button(this);  
8          button.setOnClickListener(new View.OnClickListener() {  
9              public void onClick(View v) {  
10                  Toast toast = Toast.makeText(this, "Button clicked!", Toast.LENGTH_SHORT)  
11                  toast.show();  
12              }  
13          })  
14      }  
15  }
```

Obrada događaja

layout.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      android:id="@+id/linear_layout"
4      android:orientation="vertical">
5
6      <Button
7          android:id="@+id/button"
8          android:layout_width="wrap_content"
9          android:layout_height="wrap_content"
10         android:text="@string/button_text"
11         android:onClick="onClick" />
12
13  </LinearLayout>
```

Obrada događaja

ExampleActivity.java

```
1 public void onClick(View view) {  
2     Toast toast = Toast.makeText(  
3         this, "Button clicked!", Toast.LENGTH_SHORT);  
4     toast.show();  
5 }
```


Tipovi pogleda

- TextView
- ImageView
- EditText
- Button
- RadioButton
- ToggleButton
- Checkbox

Hello Android!

Slika 13: TextView.

- Pogled TextView prikazuje tekst i omogućava njegovo kopiranje

TextView

```
1  <TextView
2      android:id="@+id/email_address"
3      android:layout_width="fill_parent"
4      android:layout_height="wrap_content"
5      android:text="@string/hello_world" />
```

ImageView



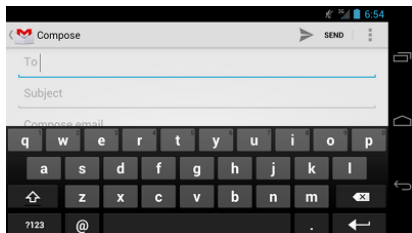
Slika 14: ImageView.

- Pogled ImageView prikazuje proizvoljnu sliku iz različitih izvora
- Omogućava i skaliranje, odsecanje, primenu filtera, itd.

ImageView

```
1 <ImageView
2   android:id="@+id/icon"
3   android:layout_width="wrap_content"
4   android:layout_height="wrap_content"
5   android:src="@drawable/my_image" />
```

EditText



Slika 15: EditText.

- Pogled EditText omogućava unos teksta
- Pored unosa teksta, omogućava i niz drugih aktivnosti kao što su označavanje, isecanje, kopiranje, itd.
- Moguće je specificirati tip tastature (normalna, numerička, telefonska, itd.) ili ponašanje tastature (automatsko pretvaranje početnih slova reči u velika slova, itd.)

EditText

```
1  <EditText
2      android:id="@+id/email_address"
3      android:layout_width="fill_parent"
4      android:layout_height="wrap_content"
5      android:hint="@string/password_hint"
6      android:inputType="text|textPassword" />
```

EditText

Vrednost	Opis
text	normalna tastatura
textEmailAddress	normalna tastatura sa @ znakom
textUri	normalna tastatura sa / znakom
number	osnovna numerička tastatura
phone	telefonska tastatura

Tabela 3: Vrednosti inputType atributa.

EditText

Vrednost	Opis
textCapWords	normalna tastatura koja automatski pretvara početno slovo rečenice u veliko slovo
textAutoCorrect	normalna tastatura koja ispravlja česte pravopisne greške
textPassword	normalna tastatura koja unesene znakove prikazuje kao tačke
textMultiLine	normalna tastatura koja omogućava korisnicima da unose tekst u više redova

Tabela 4: Vrednosti inputType atributa.

Button



Slika 16: Button.

- Pogled Button prikazuje tekst ili sliku koja simbolizuju akciju
- Kada korisnik pritisne dugme generiše se click događaj
- Metoda koja obrađuje ovaj događaj specificira se onClick atributom i mora biti sadržana u aktivnosti kojoj je dugme pridruženo

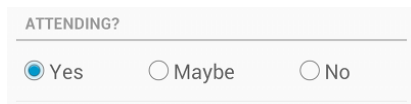
Button

```
1  <Button
2    android:id="@+id/button_id"
3    android:layout_width="wrap_content"
4    android:layout_height="wrap_content"
5    android:text="@string/button_text"
6    android:drawableLeft="@drawable/button_icon"
7    android:onClick="alarm" />
```

Button

```
1  /* Called when the user touches the button */
2  public void alarm(View view) {
3      Uri notification = RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION);
4      Ringtone r = RingtoneManager.getRingtone(getApplicationContext(), notification);
5      r.play();
6  }
```

RadioButton



Slika 17: RadioButton.

- Pogled `RadioButton` omogućava korisniku da izabere jednu opciju iz skupa više opcija
- Svaka opcija predstavljena je objektom klase `RadioButton` koji su grupisani objektom klase `RadioGroup`

RadioButton

```
1 <RadioButton
2   android:id="@+id/yes"
3   android:layout_width="wrap_content"
4   android:layout_height="wrap_content"
5   android:text="@string/yes"
6   android:onClick="onRadioButtonClicked" / >
```

RadioButton

```
1  <RadioGroup
2      android:layout_width="fill_parent"
3      android:layout_height="wrap_content"
4      android:orientation="horizontal">
5
6      <RadioButton ... />
7      <RadioButton ... />
8
9  </RadioGroup>
```

RadioButton

```
1  public void onRadioButtonClicked(View view) {  
2      // Is the button now checked?  
3      boolean checked = ((RadioButton) view).isChecked();  
4      // Check which radio button was clicked  
5      switch(view.getId()) {  
6          case R.id.yes:  
7              if (checked)  
8                  // ...  
9              break;  
10         case R.id.maybe:  
11             if (checked)  
12                 // ...  
13             break;  
14     }  
15 }
```


ToggleButton



Slika 18: ToggleButton.

- Pogled ToggleButton omogućava korisniku da promeni podešavanje između dva stanja

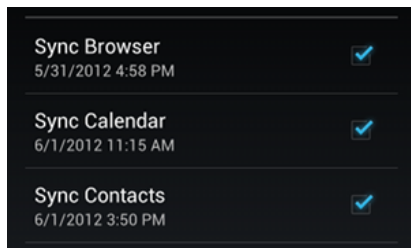
ToggleButton

```
1 <ToggleButton
2   android:id="@+id/togglebutton"
3   android:layout_width="wrap_content"
4   android:layout_height="wrap_content"
5   android:textOn="Vibrate on"
6   android:textOff="Vibrate off"
7   android:onClick="onToggleClicked" / >
```

ToggleButton

```
1  public void onToggleClicked(View view) {  
2      // Is the toggle on?  
3      boolean on = ((ToggleButton) view).isChecked();  
4      if (on) {  
5          // Enable vibrate  
6      } else {  
7          // Disable vibrate  
8      }  
9  }
```

CheckBox



Slika 19: CheckBox.

- Pogled CheckBox omogućava korisniku da izabere jednu ili više opcija iz skupa opcija
- Opcije se obično prikazuju u vertikalnoj listi.

CheckBox

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      android:orientation="vertical"
4      android:layout_width="fill_parent"
5      android:layout_height="fill_parent">
6
7      <CheckBox
8          android:id="@+id/checkbox_browser"
9          android:layout_width="wrap_content"
10         android:layout_height="wrap_content"
11         android:text="@string/meat"
12         android:onClick="onCheckboxClicked" />
13
14     <CheckBox
15         android:id="@+id/checkbox_calendar"
16         android:layout_width="wrap_content"
17         android:layout_height="wrap_content"
18         android:text="@string/cheese"
19         android:onClick="onCheckboxClicked" />
20
21 </LinearLayout>
```

CheckBox

```
1  public void onCheckboxClicked(View view) {  
2      // Is the view now checked?  
3      boolean checked = ((CheckBox) view).isChecked();  
4  
5      // Check which checkbox was clicked  
6      switch(view.getId()) {  
7          case R.id.checkbox_browser:  
8              if (checked)  
9                  // Synchronize browser  
10                 break;  
11         case R.id.checkbox_calendar:  
12             if (checked)  
13                 // Synchronize calendar  
14                 break;  
15         }  
16     }
```



All images copyrighted by Android Open Source Project (CC BY)