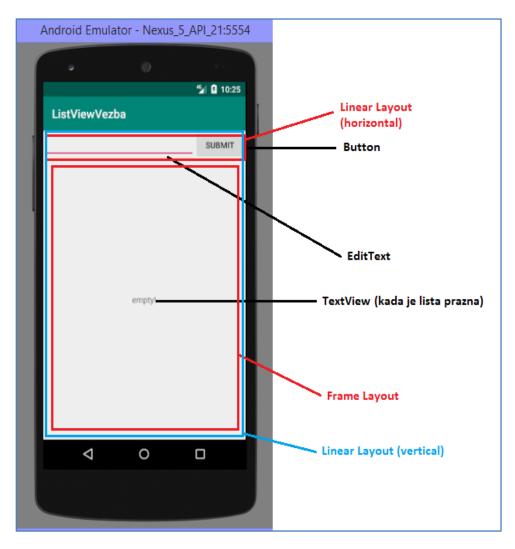
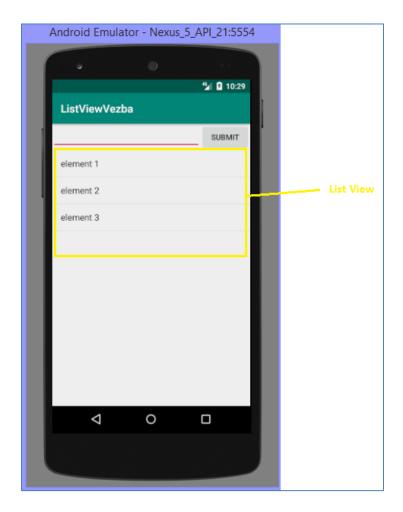
## List View zadatak

Kada smo se upoznali sa komponentom List View u Androidu, napravićemo jednu jednostavnu aplikaciju koja sadrži polje za unos teksta (EditText) i dugme *Submit* na čiji pritisak se tekst koji smo uneli dodaje kao element liste.

- 1. Kreiramo nov projekat **ListViewZadatak** (API nivo 21, Empty Activity).
- 2. Kreiramo layout koji treba da izgleda kao na slici (Slika 1 i Slika 2). Ukoliko je lista prazna treba da se ispisuje tekst "Empty List!", a ukoliko nije prikazuje se ListView. Ovo postižemo upotrebom FrameLayout-a. Voditi računa o smeštanju stringova u strings.xml.



Slika 1 - Prikaz aplikacije dok je lista prazna



Slika 2 - Prikaz aplikacije nakon dodavanja 3 elementa u listu

- 3. U okviru MainActivity –ja instanciramo EditText i dugme. Na dugme postavimo click listener.
- 4. Kako bismo rukovali listom, moramo kreirati i adapter. Radimo sa ArrayAdapter om koji rukuje tipom String i nazivamo ga *mListAdapter*. Elementi liste koju prikazujemo su jednostavni, tj. sastoje se samo iz TextView-ova. Iz ovog razloga, pri pozivanju ArrayAdapter konstruktora, drugi parametar je ugrađeni *simple\_list\_item\_1* layout. Ukoliko pogledamo kako ovaj layout izgleda (CTRL + click na ime) videćemo da je u pitanju samo TextView. Kada budemo radili sa kompleksnijim elementima liste (npr. Element liste sadrži i sliku i tekst) moraćemo sami da kreiramo izgled jednog elementa u okviru zasebnog layout-a i onda da ga prosleđujemo kao parametar.
- 5. Instanciramo i ListView i na njega postavimo adapter, kao i listener za pritisak na element liste. Želimo da pritiskom na element liste taj element uklonimo iz nje.

```
🜉 ListViewVezba 🕽 📑 app 🕽 🖿 src 🕽 🖿 main 🕽 🖿 java 🕽 🖿 com 🖯 🖿 example 🕽 🖿 listviewvezba 🤇 😅 MainActivity
                                       MainActivity.java ×
         ι¢:
                    activity_main.xml
   app
                         import ...
    manifests
                 12
    java
                 13
                         public class MainActivity extends AppCompatActivity {
      ▼ 🛅 com.e
                 14
            C M
                             private EditText mNewItem;
      ▶ a com.€ 16
                             private ArrayAdapter<String> mListAdapter;
                                                                                3.
      ▶ 🔤 com.€
                             @Override
    generated
                 19 0
                             protected void onCreate(Bundle savedInstanceState) {
    ▶ res
                 20
                                 super.onCreate(savedInstanceState);
    Gradle Script
                                 setContentView(R.layout.activity main);
                 22
                 23
                 24
                                 Button submitBtn = findViewById(R.id.submitBtn);
                                 submitBtn.setOnClickListener(new MyOnClickListener());
                 25
                 26
                 27
                                 mNewItem = findViewById(R.id.newItem);
                                                                                      3.
                 28
                 29
                                 mListAdapter = new ArrayAdapter<String>( context: this,
                 30
                                          android.R.layout.simple list item 1);
                                                                                         4.
                 31
                                 ListView list = findViewById(R.id.list);
                 32
                 33
                                  list.setAdapter(mListAdapter);
                 34
                                 list.setOnItemClickListener(new MyOnItemClickListener());
                 35
                                  list.setEmptyView(findViewById(R.id.emptyView));
                 36
                 37
```

Slika 3 - Main Activity

- 6. U zasebnoj klasi smo implementirali OnClickListener i *onClick* metodu dugmeta **Submit.** Pritiskom na dugme preuzimamo vrednost iz EditText-a, pozivom metode *getText()*. Da bismo dobili String vrednost, moramo dodatno obaviti konverziju pozivom *toString()* (**Napomena**: pogledati šta *getText()* vraća).
- 7. Ukoliko EditText polje nije prazno, preuzeti string dodajemo kao element liste pozivom *add* metode adaptera. <u>Rukovanje listom se, dakle, uvek obavlja kroz adapter</u>.
- 8. Takođe u odvojenoj klasi MyOnItemClickListener implementiramo *onItemClick* metodu. Da bismo neki element uklonili iz liste, nad adapterom pozivamo *remove* metodu. Njoj prosleđujemo element na koji smo kliknuli, koji preuzimamo pozivom *getItem* koja za prosleđenu poziciju (koju imamo kao atribut onItemClick metode) vraća vrednost elementa, u našem slučaju string.

```
ListViewVezba > 📑 app > 🖿 src > 🖿 main > 🖿 java > 🖿 com > 🖿 example > 🖿 listviewvezba > 💿 MainActivity
                   activity_main.xml >

    MainActivity.java

     app
     ▶ manifest: 38
     ▼ 📄 java
                  39
                               private class MyOnClickListener implements View.OnClickListener {
        ▼ 🖿 com.e
Resource Manager
                   41
             C M 42
                                   @Override
        ▶ 🔤 com.€ 43 🔊
                                   public void onClick(View v) {
        ▶ 🖿 com.€ 44
                                       String item = mNewItem.getText().toString();
     generated 45
                                       if (TextUtils.isEmpty(item)) {
                   46
     ▶ res
                                           mNewItem.setError(getString(R.string.error_empty_input));
  ▶ ⋒ Gradle Script
                   48
                                            return;
49
                                       1
                   50
                                       mListAdapter.add(item);
                  51
                  52
                                                                                                       6.
                  53
                  54
                  55
                               private class MyOnItemClickListener implements AdapterView.OnItemClickListener {
                  56
                  57
                  58
                                   public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
                  59
                                       String item = mListAdapter.getItem(position);
                  60
                                       mListAdapter.remove(item);
                  61
                                                                                                    7.
                  62
                  63
                  64
```

Slika 4 - MainActivity

9. **Prevesti i pokrenuti aplikaciju**. Testirati na emulatoru, dodavanjem i uklanjanjem elemenata. Kompletno rešenje biće dostupno nakon online konsultacija.