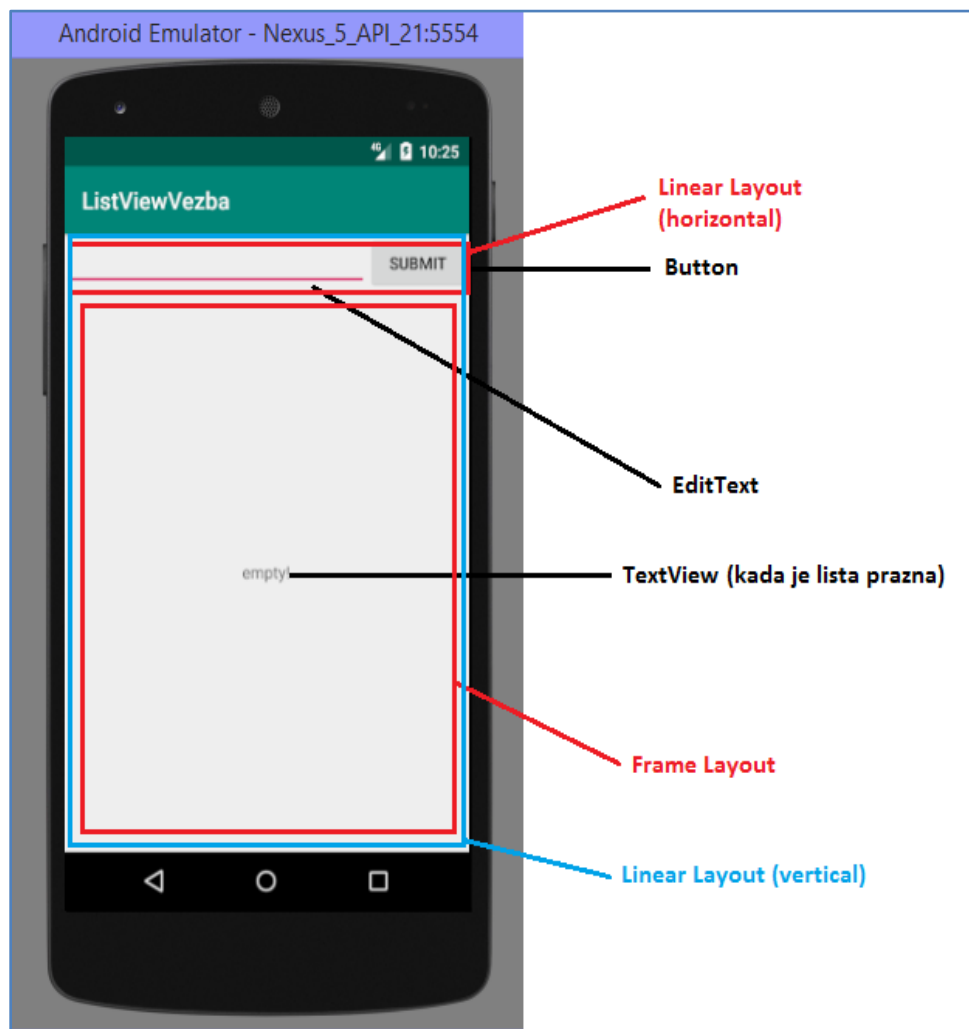


List View zadatak

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 postizemo upotrebom FrameLayout-a. Voditi računa o smeštanju stringova u strings.xml.

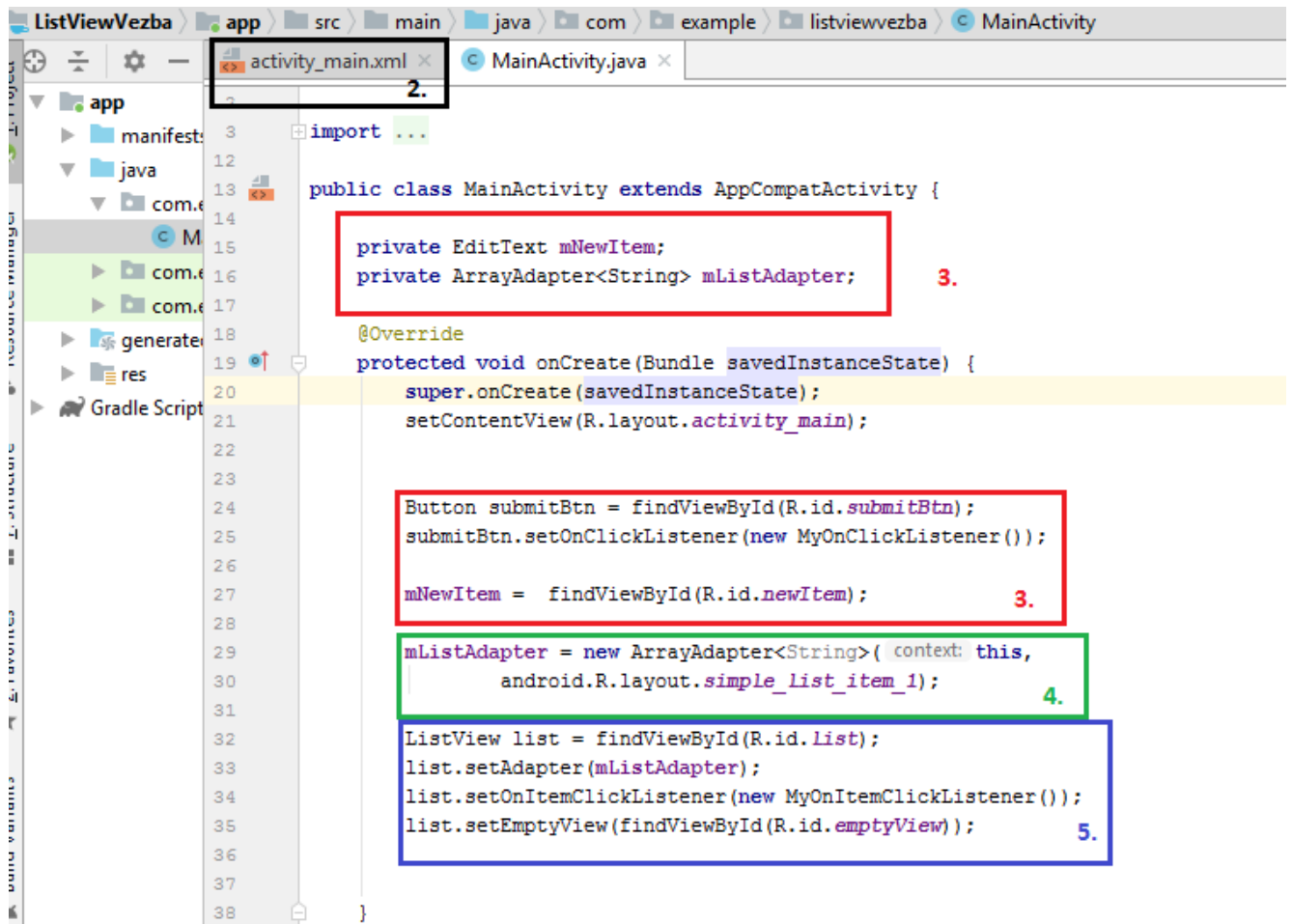


Slika 1 - Prikaz aplikacije dok je lista prazna



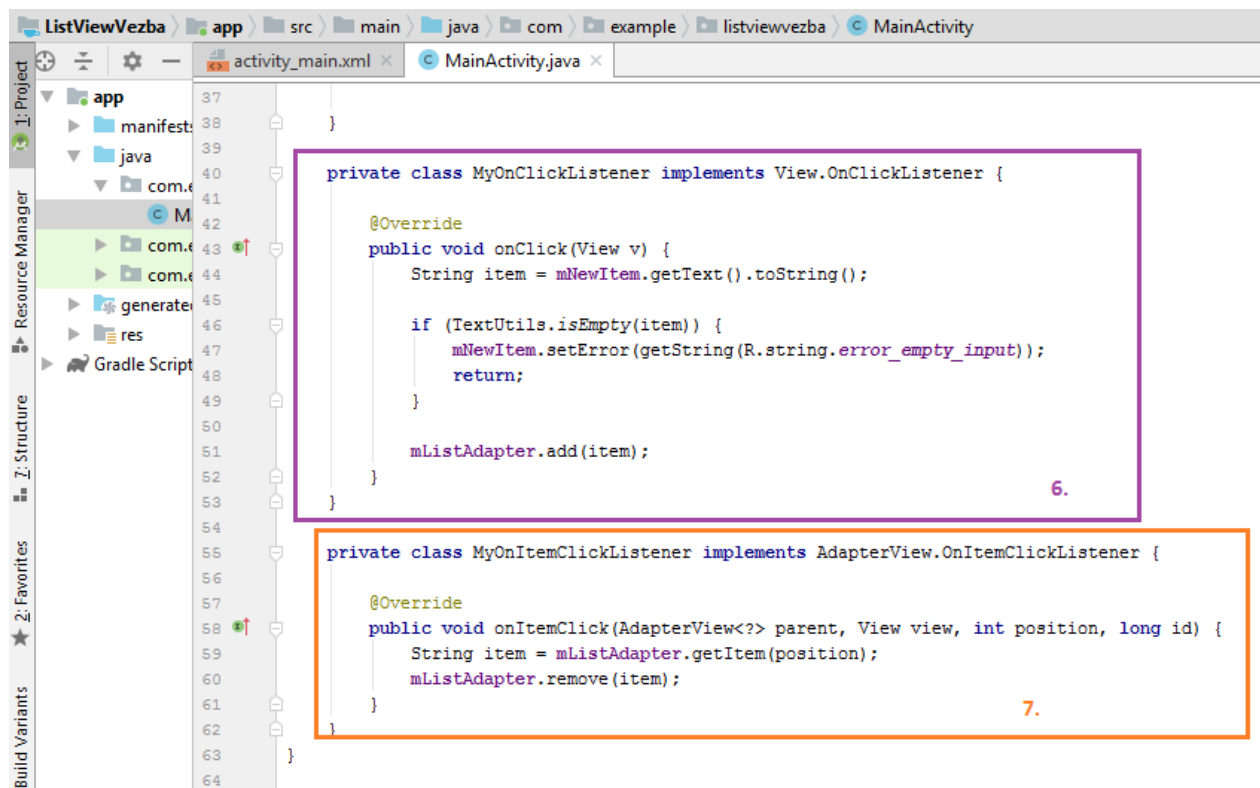
Slika 2 - Prikaz aplikacije nakon dodavanja 3 elementa u listu

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.



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. Rukovanje listom se, dakle, uvek obavlja kroz adapter.
8. Takođe u odvojenoj klasi MyItemClickListener implementiramo *onItemClick* metodu. Da bismo neki element uklonili iz liste, nad adapterom pozivamo **remove** metodu. Njoj prosledujemo element na koji smo kliknuli, koji preuzimamo pozivom **getItem** koja za prosledenu poziciju (koju imamo kao atribut onItemClick metode) vraća vrednost elementa, u našem slučaju string.



Slika 4 - MainActivity

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