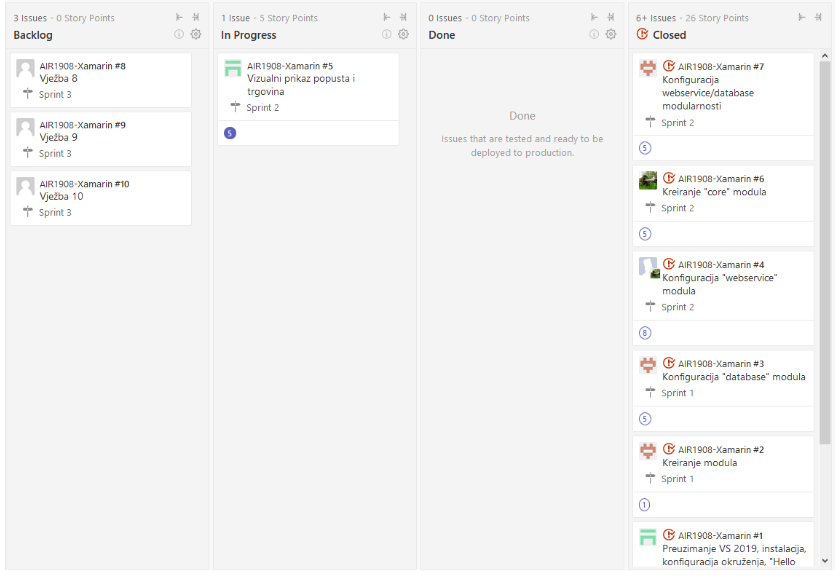
***Sažetak***

U prošloj vježbi omogućili smo prikaz dućana na ekranu. Na odabir dućana prikazali su nam se svi njegovi popusti s detaljima. U ovoj ćemo vježbi kreirati novu aktivnost koja će biti zadužena za prikaz detalja o svakom pojedinom popustu. Implementirat ćemo i brisanje odabranog popusta na duži klik. Također ćemo kreirati aktivnost na kojoj će biti prikazane postavke naše aplikacije. Na slici ispod možemo vidjeti kako se i dalje bavimo vizualnim prikazom, međutim sada se baziramo na popuste (njihove detalje) i postavke.

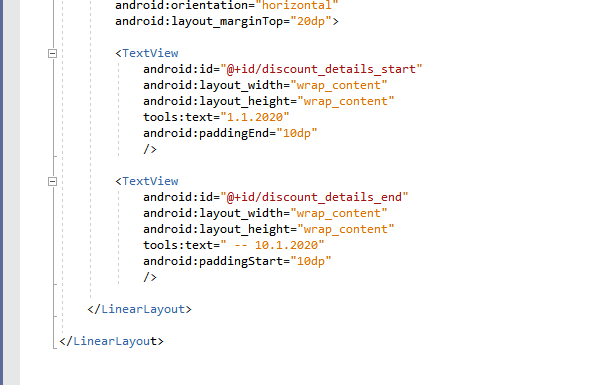
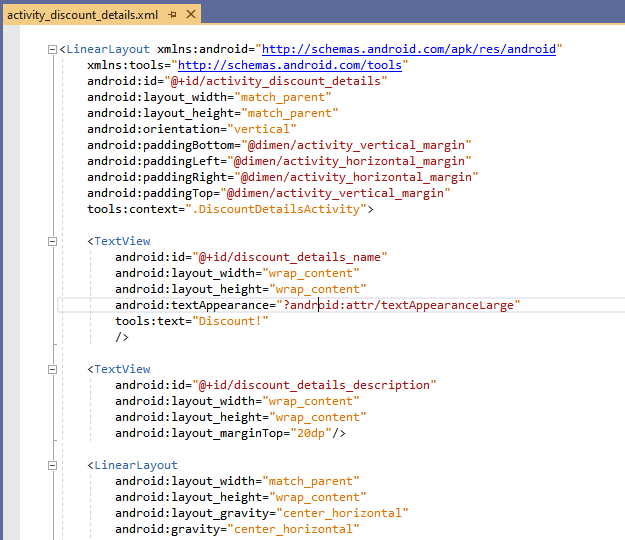
Zadnja verzija aplikacije: Commit d43f597 [<https://github.com/AIR-FOI-HR/AIR1908-Xamarin>]

***Što radimo?***

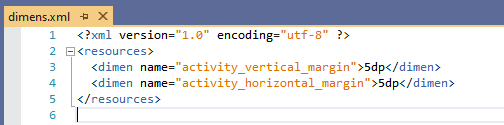


Za početak, dodat ćemo novu aktivnost DiscountDetailsActivity u DiscountLocator19 modul. Desni klik na DiscountLocatio19 -> Add -> New Item… -> Activity -> DiscountDetailsActivity.

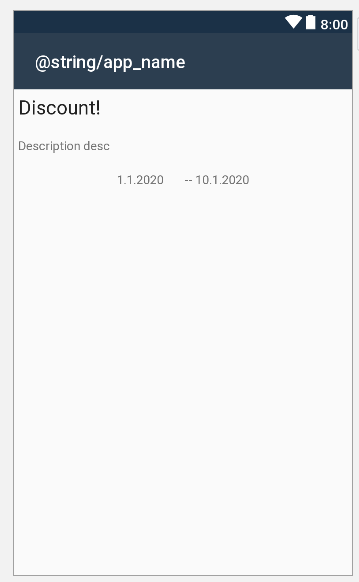
Nakon toga, moramo kreirati xml datoteku u kojoj ćemo definirati raspored elemenata na DiscountDetailsActivity aktivnosti. DiscountLocatior19 -> Resources -> desni klik na layout mapu -> Add -> New Item… -> XML File -> activity\_discount\_details.xml. Dodajemo sljedeće elemente:



Ovdje smo koristili dvije dimenzije: activity\_horizontal\_margin i activity\_vertical\_margin. Njih smo definirali u novom xml dokumentu dimens.xml. DiscountLocatior19 -> Resources -> desni klik na values mapu -> Add -> New Item… -> XML File -> dimens.xml.

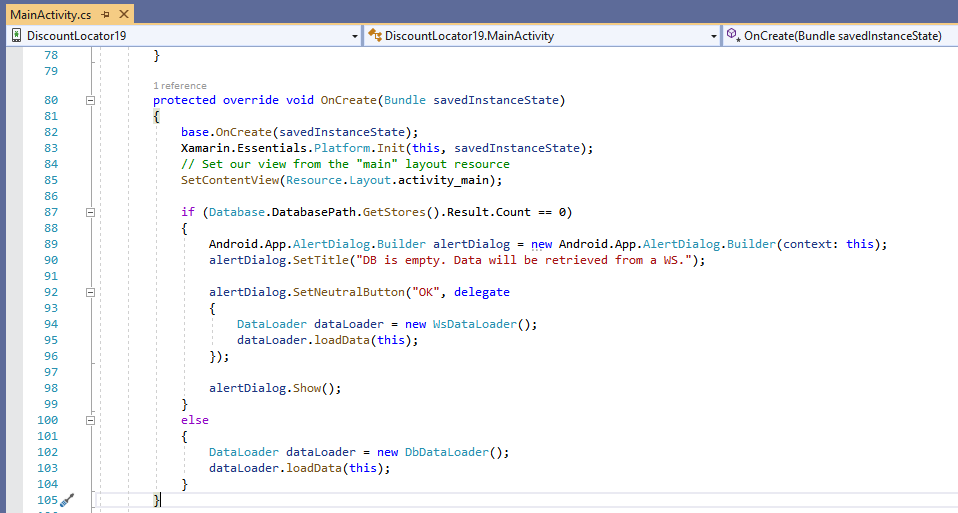


Ovako bi trebao izgledati prikaz detalja o popustu na temelju gore definiranih elemenata:

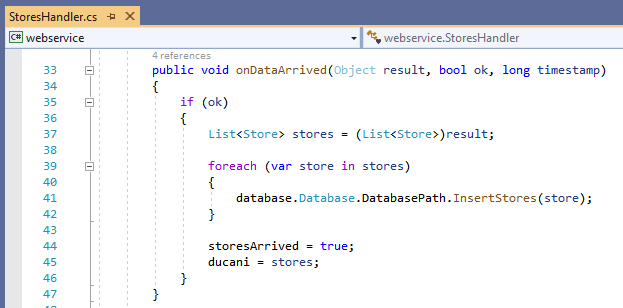


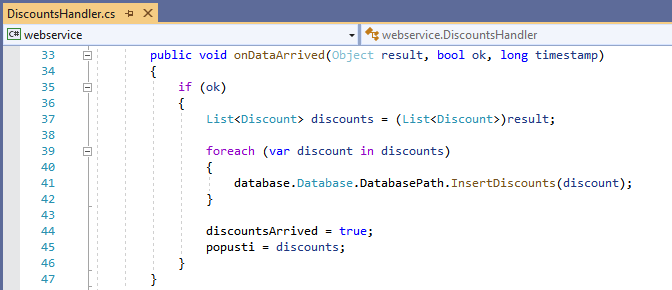
Sada ćemo implementirati logiku koja će biti zadužena za izbor izvora podataka prilikom pokretanja aplikacije. To ćemo dodati u MainActivity klasi. U OnCreate metode MainActivity klase provjerit ćemo stanje u lokalnoj bazi podataka. Ako se u bazi ne nalazi niti jedan dućan, onda ćemo obavijestiti korisnika da će se podaci dohvatiti s web servisa. Korisniku će se pojaviti jednostavna poruka na ekranu. Tu koristimo AlertDialog iz Android.App biblioteke. Moguće se postaviti 3 gumba (pozitivni odgovor, negativni odgovor i neutralni odgovor). Mi smo koristili samo gumb s neutralnim odgovorom kojim će korisnik zatvoriti AlertDialog.

Ako u bazi nema podataka, za dohvat će biti zadužen WsDataLoader. Ako u bazi ima podataka, za dohvat će biti zadužen DbDataLoader.



Sad kad znamo koji će DataLoader biti zadužen za dohvat podataka za ispis, moramo osigurati da su podaci aplikaciji uvijek dostupni. Kad se podaci dohvate s web servisa, potrebno ih je spremiti u bazu podataka. To ćemo napraviti u DiscountsHandler i StoresHandler klasama u webservice modulu. Svaki popust i svaki dućan spremit ćemo u bazu podataka.



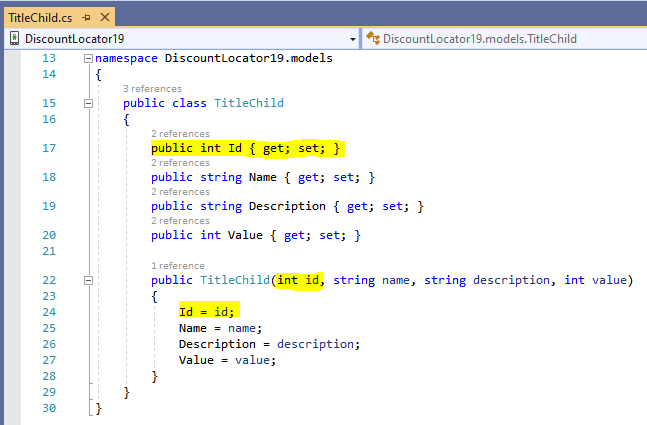


Sada moramo podatke proslijediti novokreiranoj DiscountDetailsActivity aktivnosti. Zapravo moramo implementirati klik na svaki popust u našem proširenom pogledu. Kada korisnik odabere neki popust, otvorit će mu se novi ekran na kojem će biti svi detalji odabranog popusta. S obzirom da smo definirali raspored elemenata na DiscountDetailsActivity aktivnosti, sad moramo odrediti način na koji će aktivnost znati koji popust je korisnik odabrao.

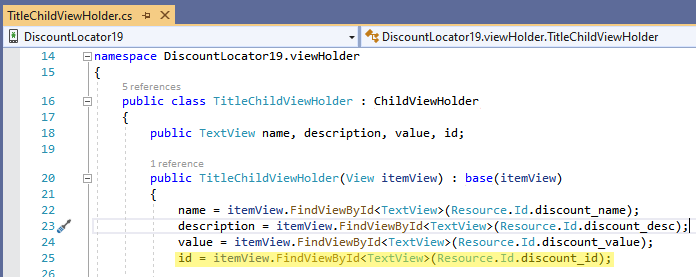
U discount\_list\_item.xml dodat ćemo još jedan element koji će sadržavati identifikator svakog popusta. Taj element nećemo prikazivati na ekranu, već će nam služiti samo za identifikaciju odabranog popusta. Dakle, u spomenuti xml dodajemo jedan TextView element koji se ne smije vidjeti na ekranu.



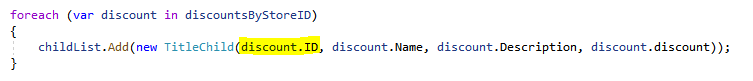
U TitleChild klasi koja nam služi kao dijete u expandable recycler view elementu moramo dodati još jedan atribut u koji ćemo spremati identifikator djeteta.



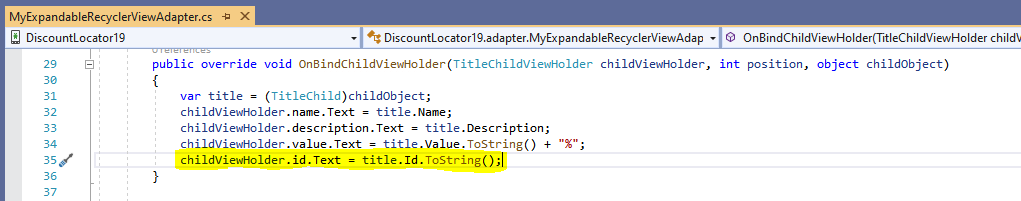
U TitleChildViewHolder klasi moramo kreirati varijablu identifikatora.



U MainActivity klasi, u InitData metodi koja priprema podatke za expandable recycler view, potrebno je dodati naš atribut kod dodavanja novog djeteta.



Još je potrebno u MyExpandableRecyclerViewAdapter klasi povezati naš novi atribut objekta djeteta s varijablom iz TitleChildViewHolder klase.

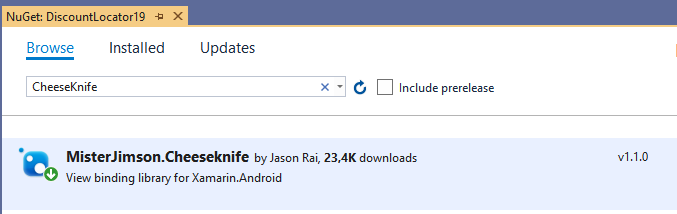


Sada možemo implementirati klik na popust. To ćemo napraviti TitleChildViewHolder klasi.



Na klik stavke TitleChildViewHolder-a stvaramo novi *intent* objekt kojeg koristimo kako bismo pozvali novu aktivnost DiscountDetailsActivity. Preko njega povezujemo dvije aktivnosti, MainActivity i DiscountDetailsActivity. Također možemo slati podatke između aktivnosti. Mi smo u ovom slučaju novoj aktivnosti proslijedili identifikator popusta koji je tipa *integer*. Nakon toga, pokrećemo DiscountDetailsActivity.

Za povezivanje varijabli i elemenata definiranih u xml-u koristit ćemo Cheeseknife paket. Desni klik na DiscountLocator19 -> Manage NuGet Packages… U tražilicu upišemo CheeseKnife i instaliramo MisterJimson.Cheeseknife paket.

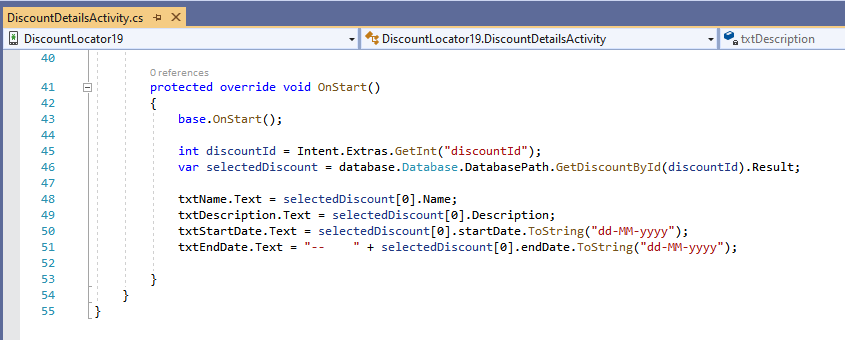


Slijedi implementacija prikaza detalja o popustu u aktivnosti DiscountDetailsAcitivty.

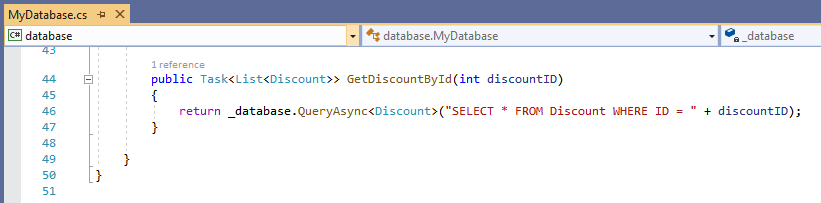
Ovdje smo odredili koje elemente iz xml datoteke povezujemo s novokreiranim varijablama tipa TextView. U OnCreate metodu pozivamo Cheeseknife.Bind metodu koja ih povezuje



U OnStart metodu dohvaćamo podatke koje smo proslijedili kroz *intent.* Spremamo identifikator u novu varijablu discountId tipa *integer.* Dohvaćamo popust iz baze podataka na temelju proslijeđenog identifikatora. U varijable txtName, txtDescription, txtStartDate, txtEndDate zapisujemo detalje odabranog popusta kojeg smo dohvatili iz baze.

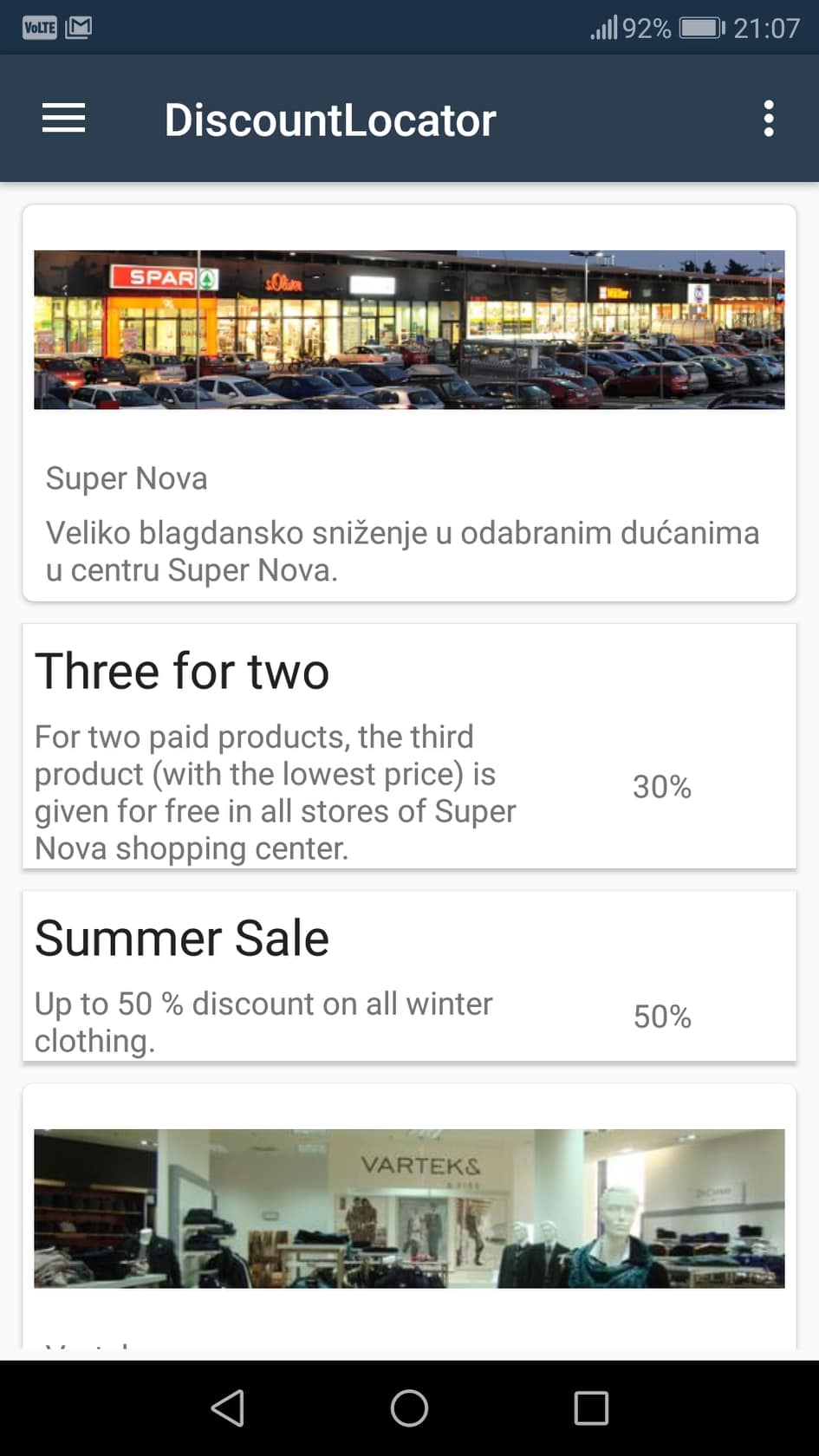
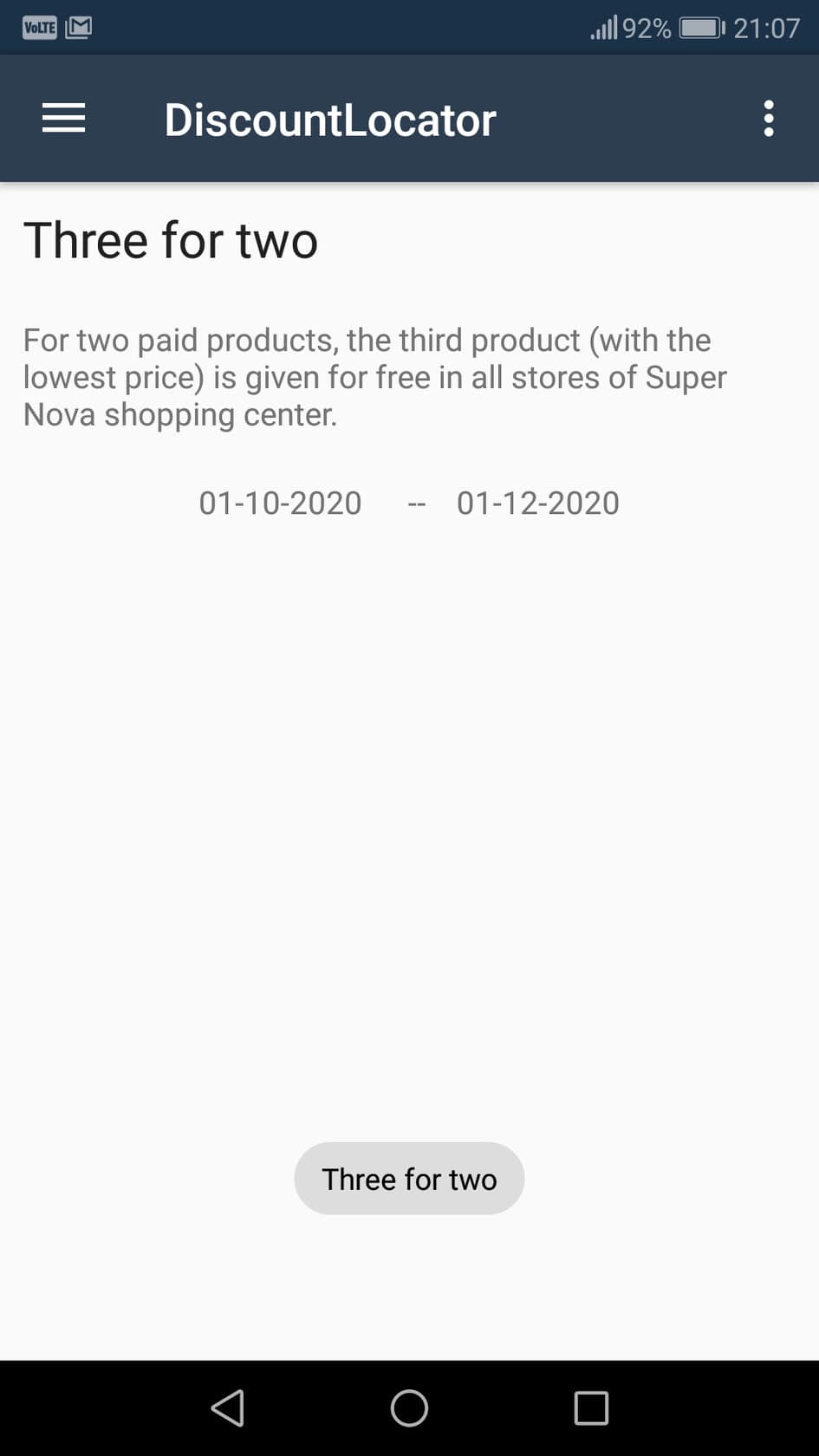


Ispod je prikazan upit kojeg smo dodali u MyDatabase klasu database modula.

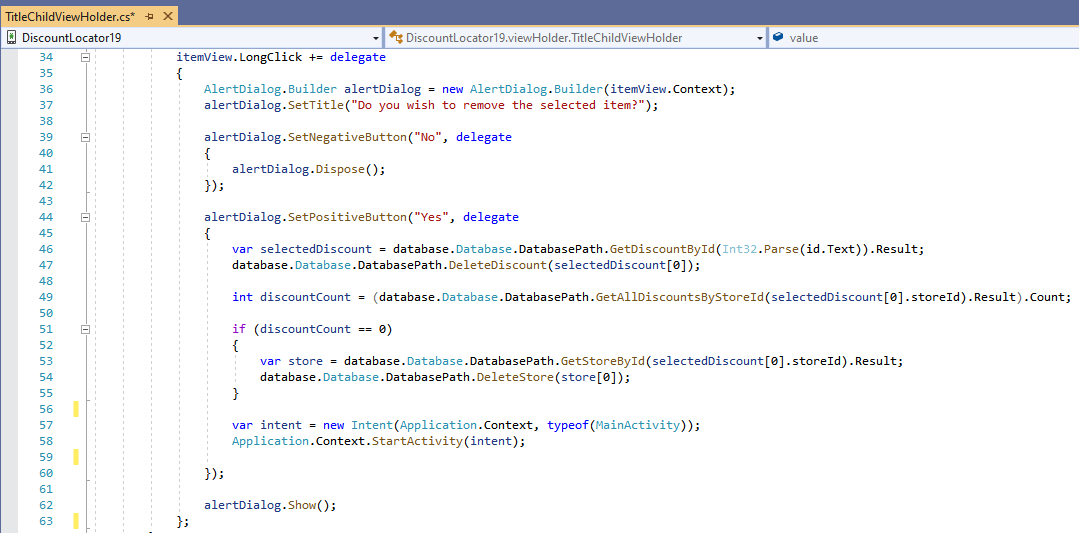


U entitetskim klasama Store i Discount potrebno je maknuti AutoIncrement primarnih ključeva kako bi se u bazu podataka spremali ispravni identifikatori.

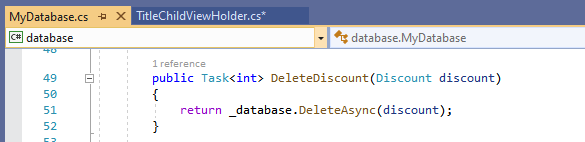
Sada ćemo napraviti Rebuild projekta i pokrenuti aplikaciju. Klikom na određeni popust, prikazat će nam se novi ekran s detaljima odabranog popusta.

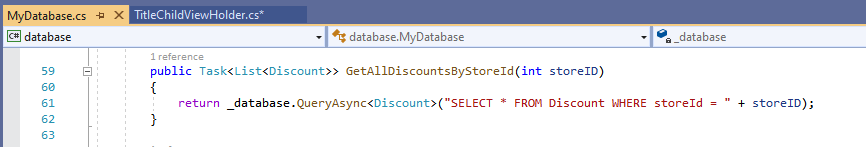
Sada ćemo implementirati brisanje odabranog popusta. To ćemo napraviti na korisnikov duži klik na odabrani popust. Vraćamo se u TitleChildViewHolder klasu i implementiramo *LongClick* na stavku.



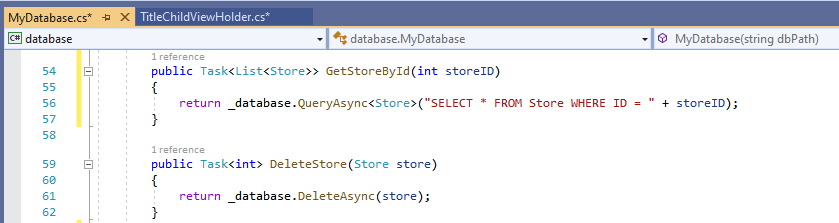
Na duži klik prvo će se na ekranu pojaviti poruka AlertDialog. Ona upozorava korisnika da je odabrao brisanje popusta. AlertDialog ima dva moguća izbora (pozitivni i negativni). Na odabir negativnog, poruka se zatvara i ništa se dodatno ne događa. Međutim, ako korisnik potvrdi brisanje, popust će se obrisati iz baze podataka. Prvo se koristi metoda GetDiscountById kojoj proslijedimo identifikator popusta. Ona nam vraća objekt popusta kojeg želimo obrisati. Zatim se izvršava metoda DeleteDiscount koju valja definirati u MyDatabase klasi database modula.



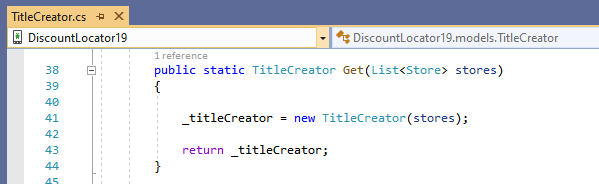
Zatim se provjerava trenutni broj popusta dućana čiji smo popust upravo izbrisali. Tu metodu također valja napisati u MyDatabase klasu.



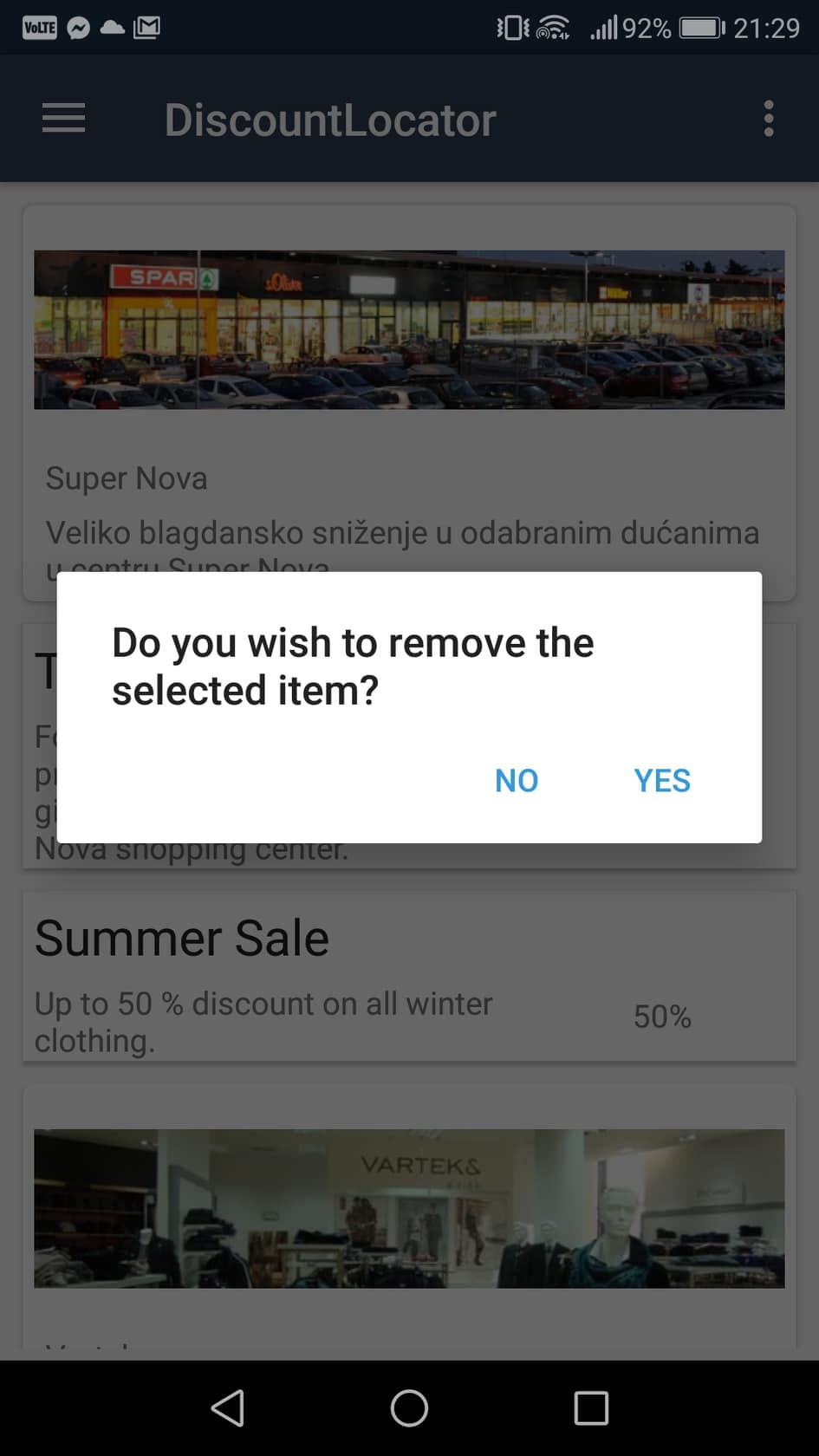
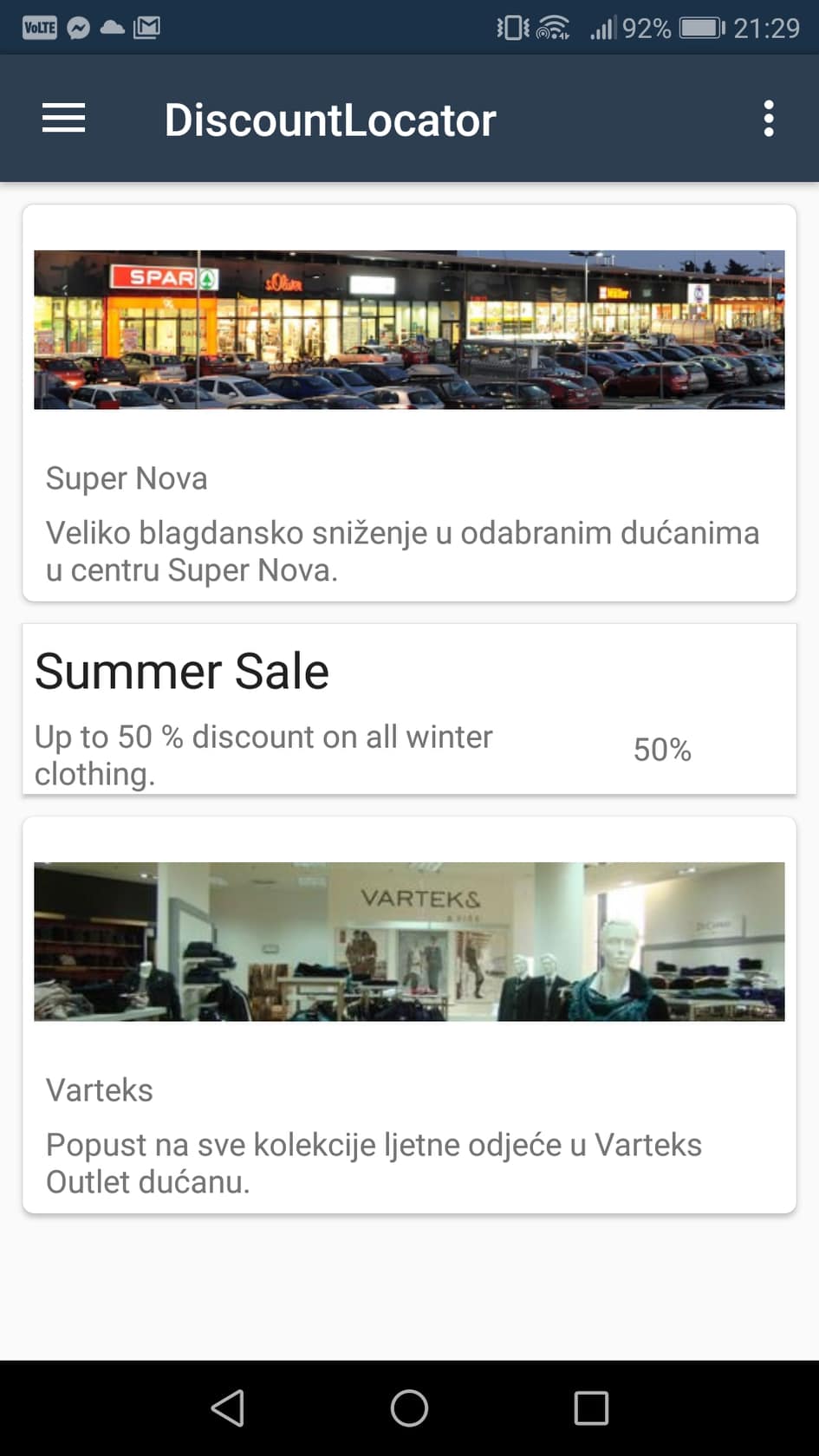
Ako je broj popusta za taj dućan jednak nula, briše se i dućan iz baze podataka. To radi metoda DeleteStore. Njoj je potreban identifikator dućana kojeg dohvaća metoda GetStoreById. Obje metode implementirat ćemo u MyDatabase klasi.



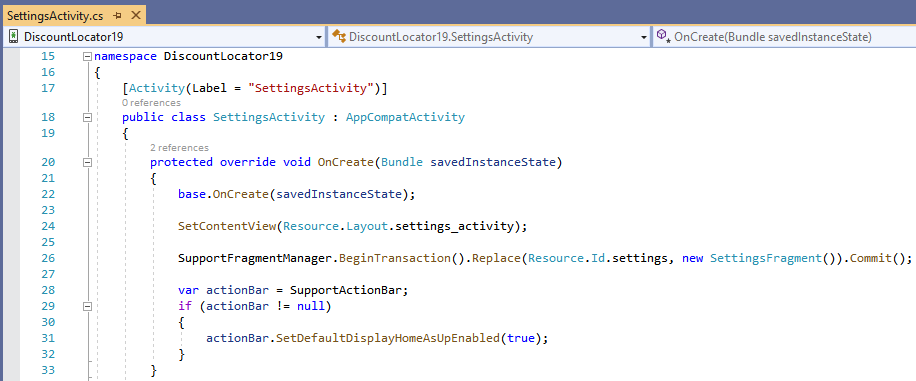
Nakon toga, ponovno pokrećemo MainActivity s ažuriranim prikazom dućana i popusta. U TitleCreator klasi, u Get metodi potrebno je maknuti provjeravanje null vrijednosti \_titleCreator varijable kako bi expandable recycler view mogao prikazati ažurirane podatke.

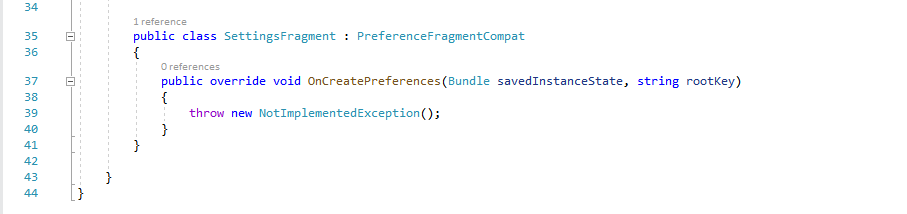


Sada možemo napraviti Rebuild projekta i pokrenuti aplikaciju. Obrisat ćemo prvi popust prvog dućana i ponovno će nam se prikazati MainActivity bez obrisanog popusta.

Sada ćemo napraviti novu aktivnost SettingsActivity koja će služiti za mijenjanje postavki naše aplikacije. DiscountLocator19 -> Add -> Class… -> SettingsActivity.

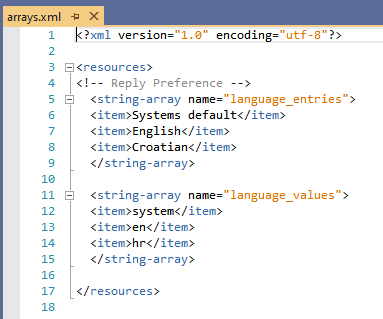




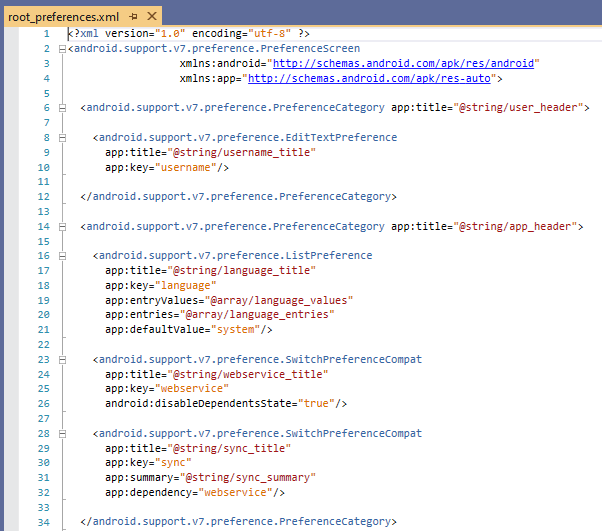
Sada je potrebno dodati xml datoteku za SettingsActivity. DiscountLocatior19 -> Resources -> desni klik na layout mapu -> Add -> New Item… -> XML File -> settings\_activity.xml.



Sada ćemo dodati xml dokument u kojem ćemo definirati moguće jezike aplikacije. DiscountLocatior19 -> Resources -> desni klik na values mapu -> Add -> New Item… -> XML File -> arrays.xml.



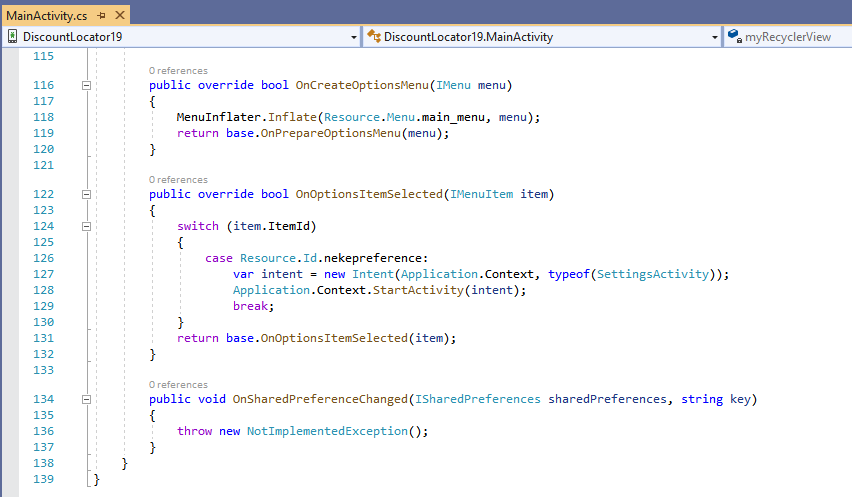
Sada je potrebno dodati layout za postavke. DiscountLocatior19 -> Resources -> Add -> New Folder -> xml. Desni klik na xml mapu -> Add -> New Item… -> XML File -> root\_preferences.xml.



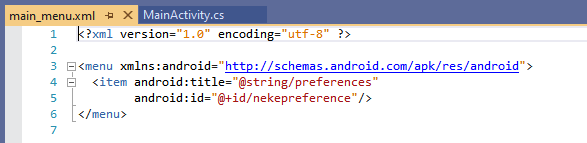
Potrebno je u Resources -> values -> strings xml dokumentu definirati *preferences,* *title\_activity\_settings, user\_header, app\_header, username\_title, sync\_title, sync\_summary, language\_title, webservice\_title*. Njih koristimo ovdje.



Sada ćemo dodati izbornik za postavke u MainActivity klasi. MainAcitivity klasa mora nasljeđivati ISharedPreferencesOnSharedPreferenceChangeListener interface. Potrebno je implementirati metode interface-a. To su: OnCreateOptionsMenu, OnOptionsItemSelected, OnSharedPreferenceChanged.



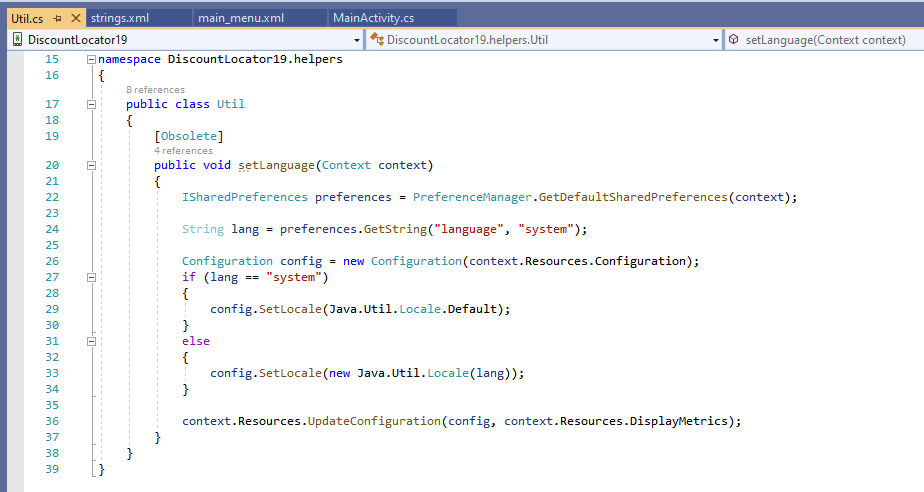
Moramo kreirati layout za glavni izbornik. DiscountLocatior19 -> Resources -> Add -> New Folder -> menu. Desni klik na menu mapu -> Add -> New Item… -> XML File -> main\_menu.xml.



Sada ćemo dodati prijevode za hrvatski jezik. To ćemo napraviti u novoj xml datoteci strings, u novoj mapi values-hr. . DiscountLocatior19 -> Resources -> Add -> New Folder -> values-hr. Desni klik na values-hr mapu -> Add -> New Item… -> XML File -> strings.xml.

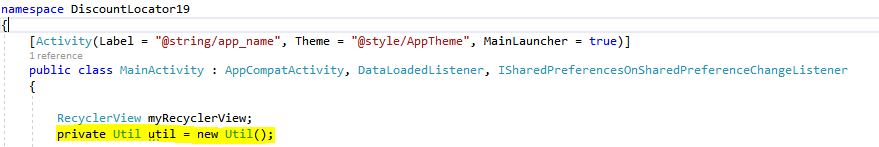


Sada ćemo kreirati pomoćnu klasu za promjenu jezika aplikacije. DiscountLocatior19 -> New Folder -> helpers. Desni klik na helpers mapu -> Add -> Class… -> Util.cs.

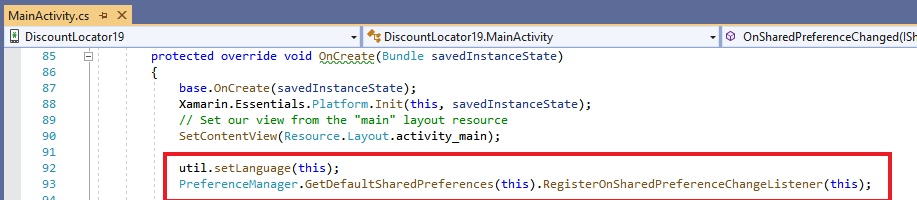


U ovoj klasi mijenjaju se konfiguracijske postavke aplikacije, tj. mijenja se jezik aplikacije.

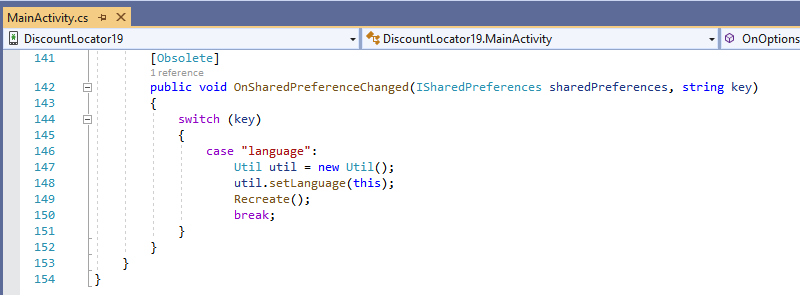
Sada moramo omogućiti promjenu jezika u MainActivity i SettingsActivity klasama. U MainActivity klasi kreiramo novu instancu Util klase.



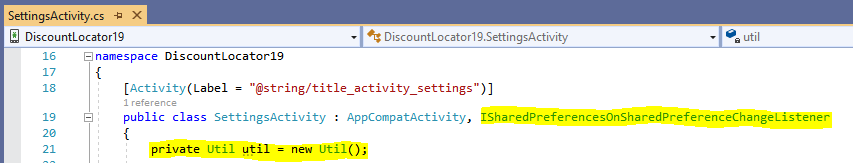
U OnCreate metodi postavljamo jezik.



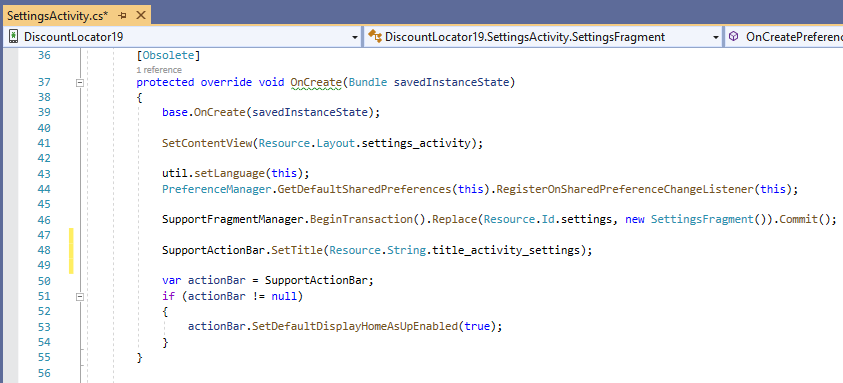
Implementiramo OnSharedPreferenceChanged.

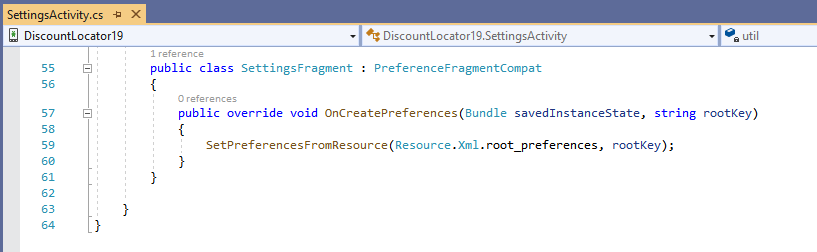


Isto ćemo napraviti i u SettingsActivity klasi. Ona mora nasljeđivati ISharedPreferencesOnSharedPreferenceChangeListener interface.



U OnCreate metodi postavljamo jezik.





Sada je potrebno napraviti Rebuild projekta i pokrenuti aplikaciju. U gornjem desnom uglu pojavit će se izbornik koji vodi na ekran s postavkama aplikacije.

