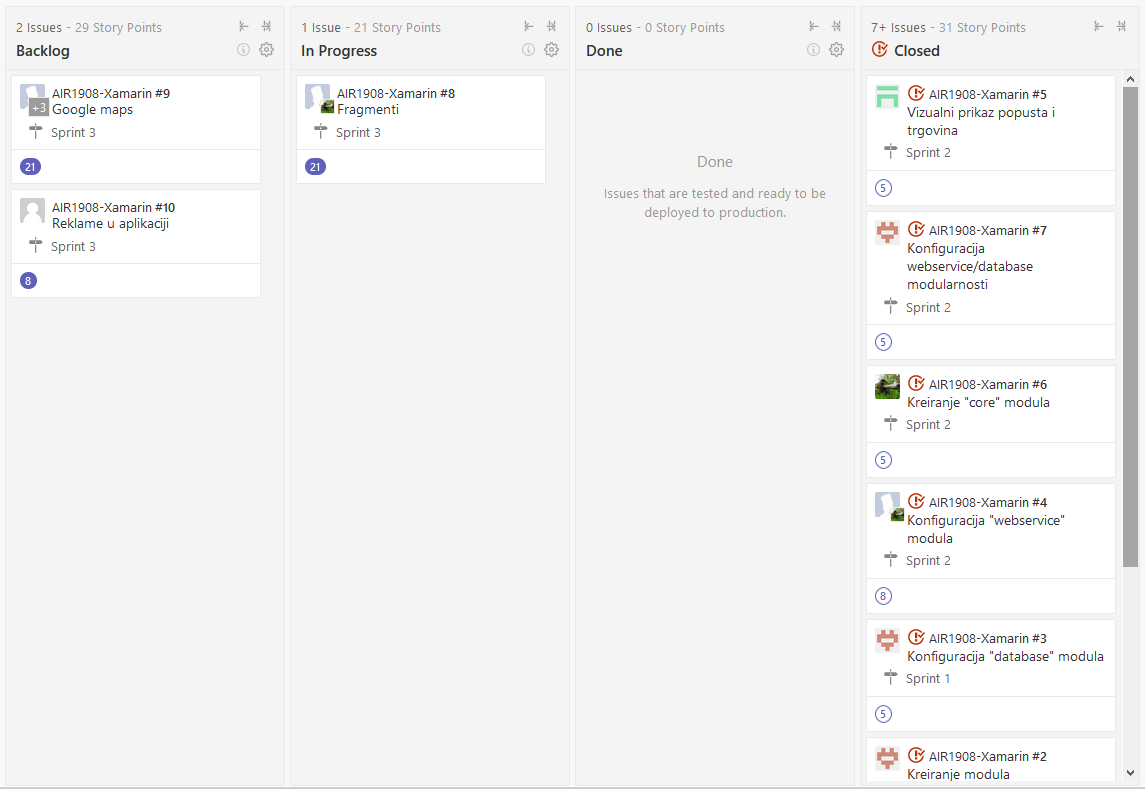
***Sažetak***

U prošloj vježbi bavili smo se prikazom detalja o odabranom popustu. Također smo implementirali brisanje popusta i prijevod aplikacije. U ovoj ćemo vježbi bazirat ćemo se na malo drugačiji način prikaza podataka. Naime, koristit ćemo fragmente. Nećemo više imati zasebne aktivnosti za prikaz dućana i detalja o popustu, već ćemo svaki pojedini prikaz kreirati kao fragment.

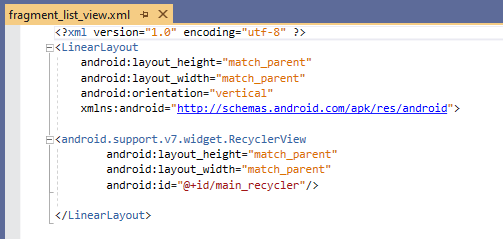
Zadnja verzija aplikacije: Commit 5a068be [<https://github.com/AIR-FOI-HR/AIR1908-Xamarin>]

***Što radimo?***

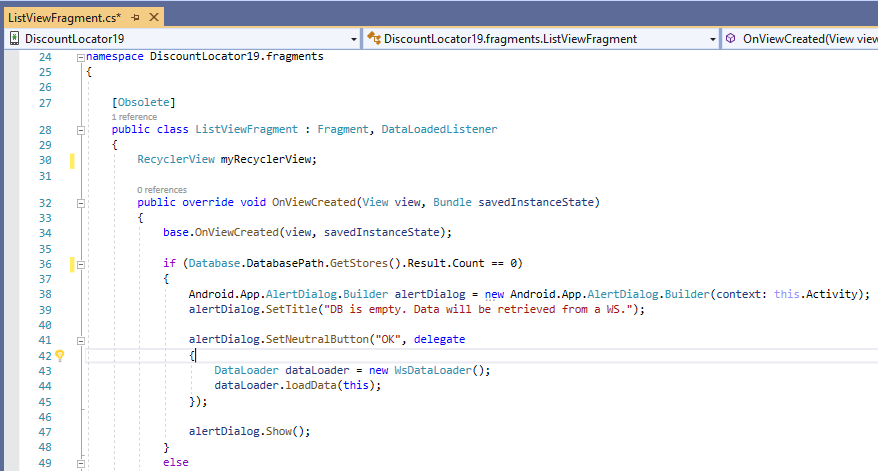


Prvo ćemo kreirati novu ListViewFragment klasu koja će biti zadužena za prikaz dućana s popustima. Ono što je dosad bilo prikazano na pokretanje MainActivity klase, sad ćemo prebaciti u ListViewFragment klasu. Desni klik na DiscountLocator19 -> Add -> New Folder -> fragments. Desni klik na mapu fragments -> Add -> New Item… -> Fragment -> ListViewFragment.

Odmah ćemo kreirati pripadajući xml za prikaz ovog fragmenta. DiscountLocator19 -> Resources -> desni klik na layout mapu -> Add -> New Item -> XML File -> fragment\_list\_view.xml.



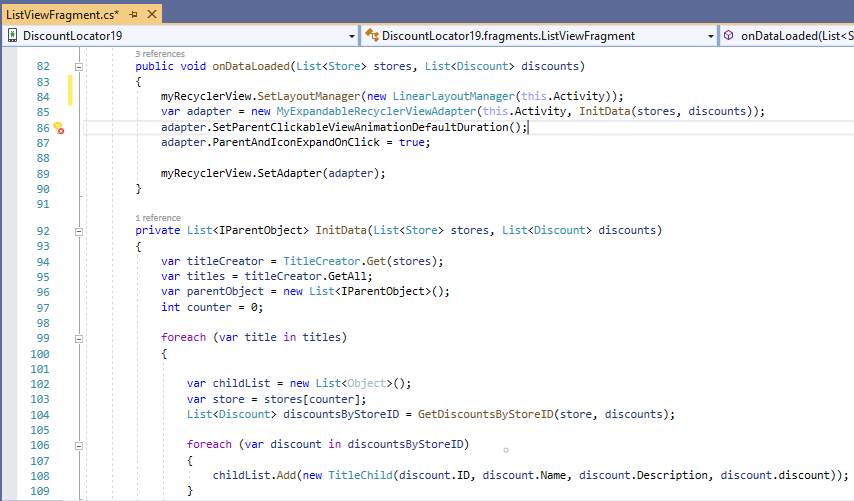
Sad moramo premjestiti kod iz MainActivity klase u novokreiranu ListViewFragment klasu. Premjestit ćemo samo onaj kod koji je zadužen za prikaz expandable recycler view elemenata. Kao i MainActivity, ListViewFragment nasljeđuje DataLoaderListener interface i implementira njegovu onDataLoaded metodu.





Dakle, na kreiranje fragmenta pokreće se kod koji smo imali u MainActivity klasi. U OnCreateView metodi pronalazi se fragment\_list\_view čiji smo morati definirali u fragment\_list\_view xml datoteci. Također pronalazimo naš recycler view element na nešto drugačiji način.

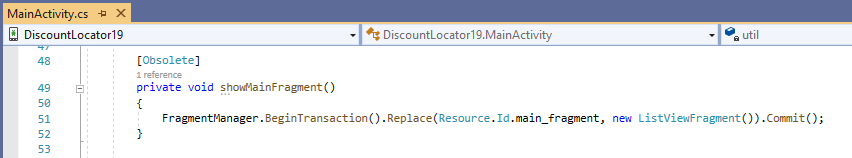
Još moramo dodati metodu onDataLoaded i metode koje služe za prikaz podataka u expandable recycler view elementu.



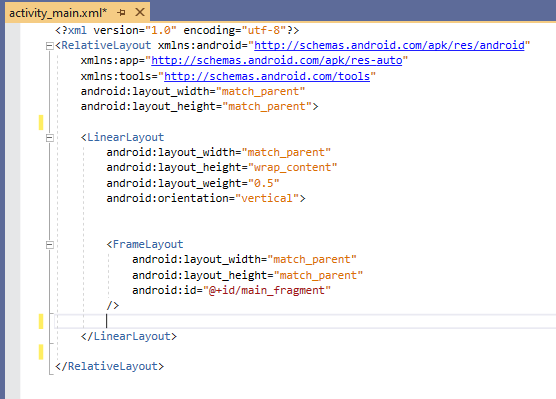
Sada ćemo se vratiti u MainActivity klasu i izbrisati kod koji smo prenijeli u ListViewFragment klasu. MainActivity više ne nasljeđuje DataLoadedListener niti implementira metodu onDataLoaded. To možemo maknuti. Također možemo maknuti metode InitData i GetDiscountsByStoreID. Maknuti možemo i logiku za dohvat podataka iz OnCreate metode.



Ovako sada izgleda OnCreate metoda. Dodali smo poziv metode showMainFragment koju ćemo sada implementirati.

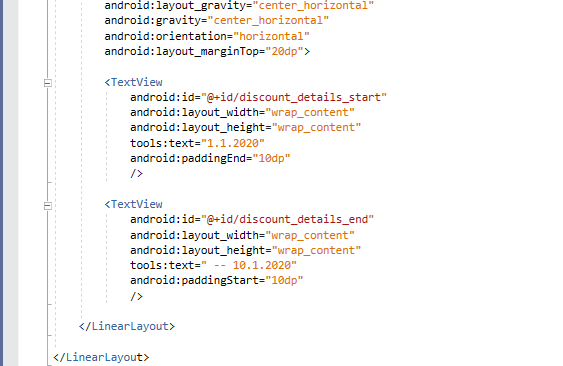
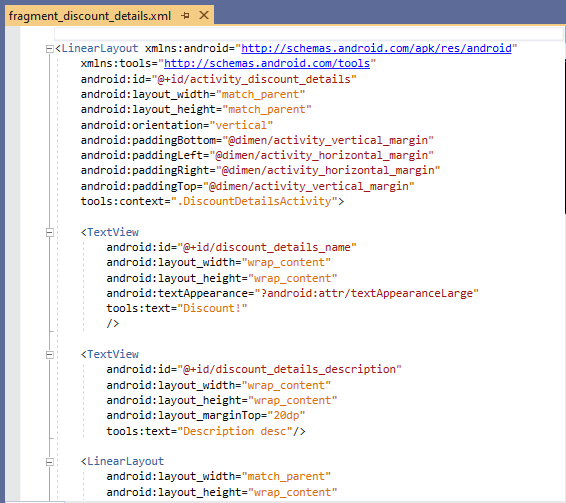


Također možemo maknuti RecyclerView element iz xml datoteke activity\_main. Ona će sada izgledati ovako:

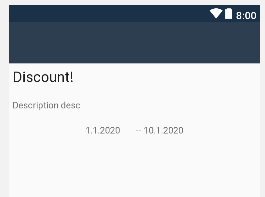


Isto ćemo sada napraviti i s DiscountDetailsActivity klasom. Prvo ćemo kreirati novu DiscountDetailsFragment klasu koja će biti zadužena za prikaz detalja o odabranom popustu. Ono što je dosad bilo prikazano na pokretanje DiscountDetailsActivity klase, sad ćemo prebaciti u DiscountDetailsFragment klasu. DiscountLocator19 -> fragments -> desni klik na mapu fragments -> Add -> New Item… -> Fragment -> DiscountDetailsFragment.

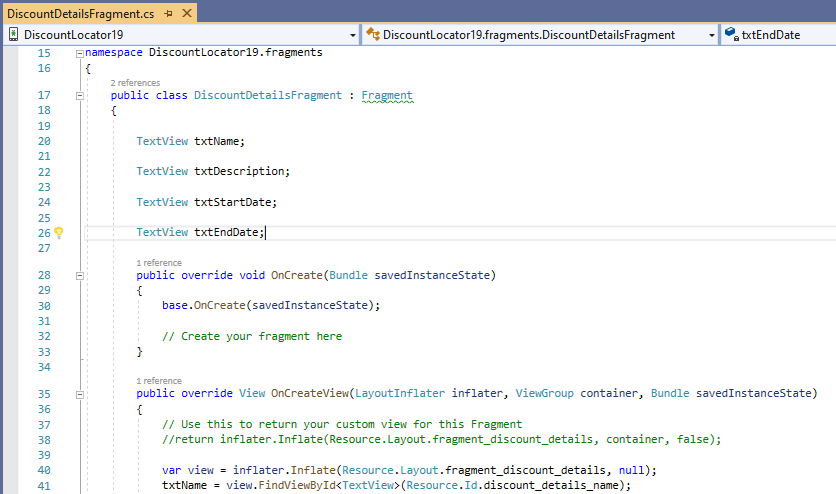
Odmah ćemo kreirati pripadajući xml za prikaz ovog fragmenta. DiscountLocator19 -> Resources -> desni klik na layout mapu -> Add -> New Item -> XML File -> fragment\_discount\_details.xml.



Ovako bi trebao izgledati prikaz na temelju xml-a:



Sad moramo premjestiti kod iz DiscountDetailsActivity klase u novokreiranu DiscountDetailsFragment klasu.

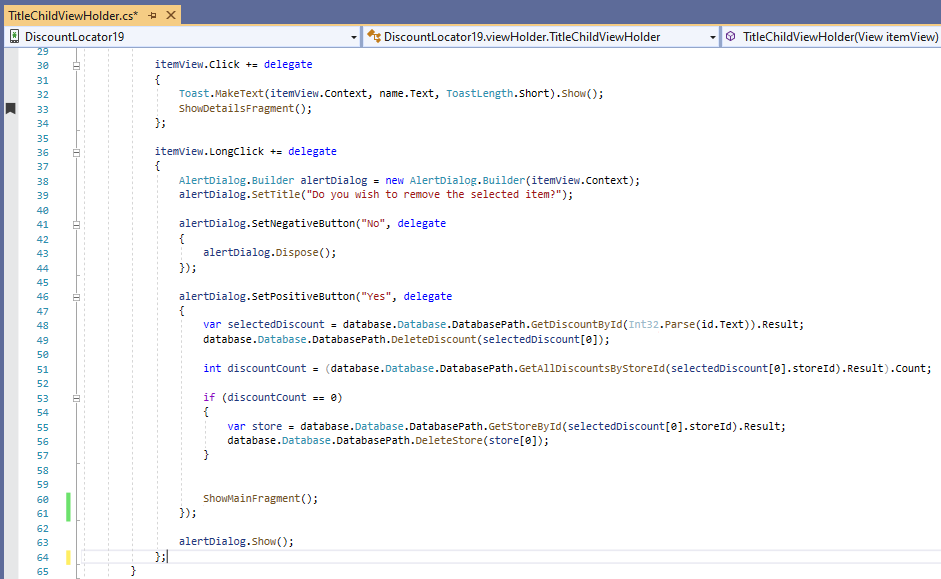




U OnViewCreated pozivamo ShowDiscountDetails metodu koja je zaslužena za povezivanje elemenata na ekranu i detalja odabranog popusta. Također više ne koristimo Cheeseknife za povezivanje elemenata i vrijednosti već to radimo pomoću FindViewById metode.

Sada možemo izbrisati DiscountDetailsActivity i pripadajući xml file jer nam više ne trebaju. U TitleChildViewHolder klasi možemo izbrisati poziv DiscountDetailsActivity aktivnosti. Zapravo ćemo obrisati kod koji se izvršava na klik na popust. Dodat ćemo nešto drugačiji kod. Obavijestit ćemo korisnika koji je popust odabrao pomoću Toast poruke i pozvati ShowDetailsFragment metodu.

Isto ćemo napraviti i s kodom koji se izvršava na duži klik. Samo ćemo pozvati metodu ShowMainFragment umjesto kreiranja novog *intenta*.



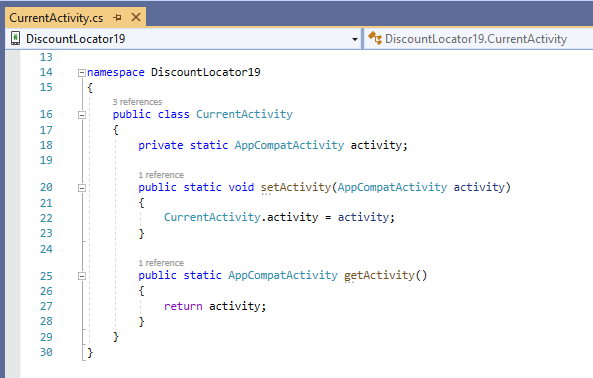
Sada ćemo implementirati metodu ShowDetailsFragment. Kreiramo novu instancu klase Bundle koja će nam služiti za prijenos podataka u fragment DiscountDetailsFragment. Kreiramo novu instancu DiscountDetailsFragment i proslijedimo joj novokreiranu objekt bundle kao argument. Prelazimo iz jednog fragmenta u drugi (iz ListViewFragment u DiscountDetailsFragment).



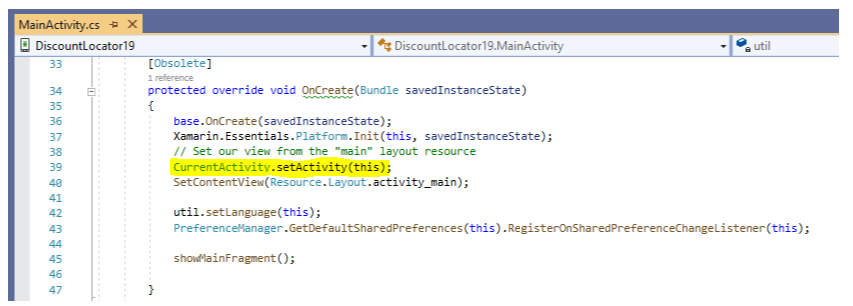
A sada metodu ShowMainFragment.



Sad ćemo dodati novu aktivnost CurrentActivity u DiscountLocator19 modul. Ona će sadržavati referencu na bilo koju trenutno vidljivu aktivnost. Desni klik na DiscountLocator19 -> Add -> Class… -> CurrentActivity.

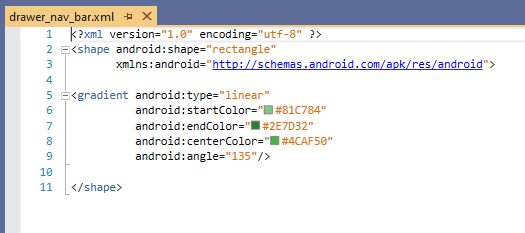


U MainActivity klasi postavit ćemo trenutnu aktivnost pomoću novo kreirane metode setActivity.

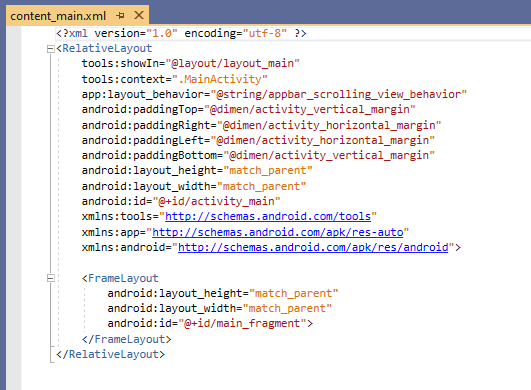


Sada ćemo napraviti izbornik koji će se otvarati na lijevoj strani ekrana. U Resources mapu DiscountLocator19 modula potrebno je dodati sljedeće xml datoteke: drawer\_nav\_bar.xml u novokreiranu drawable mapu, content\_main.xml u layout mapu, drawer\_header.xml u layout mapu, layout\_main.xml u layout mapu, drawer\_menu.xml u menu mapu, styles.xml u novokreiranu values-v21 mapu.

Desni klik na DiscountLocator19 -> Resources -> Add -> New Folder -> drawable. Desni klik na mapu drawable -> Add -> New Item… -> XML File -> drawer\_nav\_bar.xml.



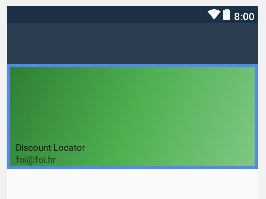
DiscountLocator19 -> Resources -> desni klik na mapu layout -> Add -> New Item… -> XML File -> content\_main.xml.



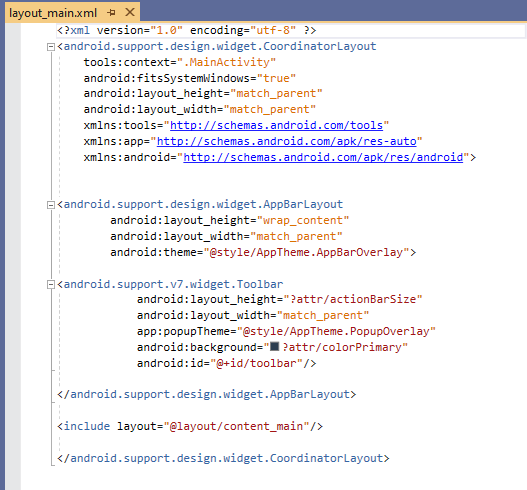
DiscountLocator19 -> Resources -> desni klik na mapu layout -> Add -> New Item… -> XML File -> drawer\_header.xml.



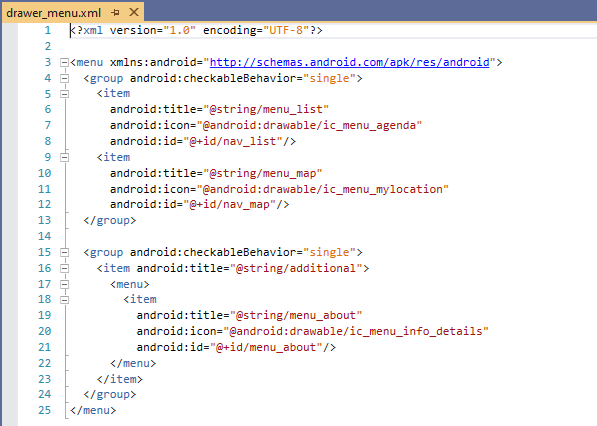
Ovako izgleda prikaz na temelju xml-a:



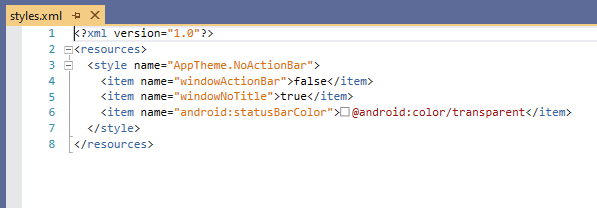
DiscountLocator19 -> Resources -> desni klik na mapu layout -> Add -> New Item… -> XML File -> layout\_main.xml.



DiscountLocator19 -> Resources -> desni klik na mapu menu -> Add -> New Item… -> XML File -> drawer\_menu.xml.



Desni klik na DiscountLocator19 -> Resources -> Add -> New Folder -> values-v21. Desni klik na mapu values-v21 -> Add -> New Item… -> XML File -> styles.xml.



Svi su nam ovi xml dokumenti potrebni za prikaz navigacijskog izbornika. Sada ćemo dodati dimenzije, stilove i naslove potrebne za prikaz elemenata u navigacijskom izborniku.

Prvo ćemo dodati nove naslove u strings xml za hrvatski jezik. DiscountLocator19 -> Resources -> values-hr -> strings.xml.



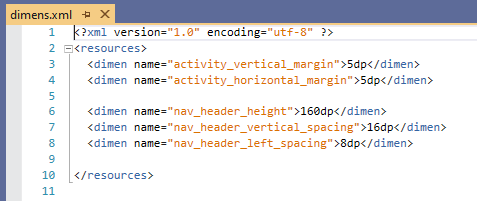
Sada za engleski jezik. DiscountLocator19 -> Resources -> values -> strings.xml.



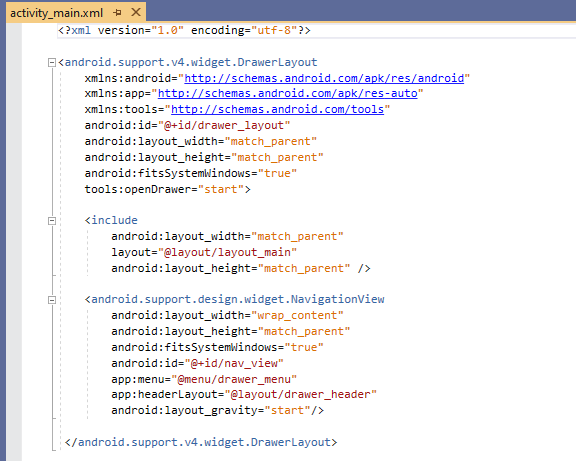
Sada ćemo dodati stilove. DiscountLocator19 -> Resources -> values -> styles.xml.



Sad su na redu dimenzije. DiscountLocator19 -> Resources -> values -> dimens.xml.



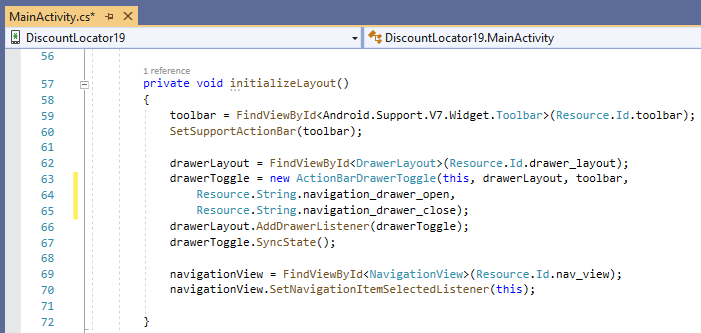
Sada ćemo izmijeniti activity\_main.xml datoteku. DiscountLocator19 -> Resources -> layout -> activity\_main.xml.



Sada ćemo u MainActivity klasi inicijalizirati navigacijski izbornik. MainActivity sada nasljeđuje NavigationView.IOnNavigationItemSelectedListener interface. Dodat ćemo sljedeće varijable: toolbar tipa Toolbar, drawerLayout tipa DrawerLayout, draweToggle tipa ActionBarDrawerToggle i navigationView tipa NavigationView. Za svaki je potrebno dodati reference kako bismo ih mogli deklarirati.



U OnCreate metodu, treba dodati poziv initializeLayout metode koju ćemo sada implementirati.



Pronalazimo toolbar prema identifikatoru definiranom u layout\_main.xml. Na isti način pronalazimo i ostale elemente koje trebamo povezati. Varijabla drawerToggle objekt je ActionBarDrawerToggle klase.

Sada je potrebno napraviti Rebuild projekta i pokrenuti aplikaciju. Aplikacija će biti slična dosadašnjoj, uz navigacijski izbornik koji se otvara na klik ikone u gornjem lijevom uglu ekrana.

