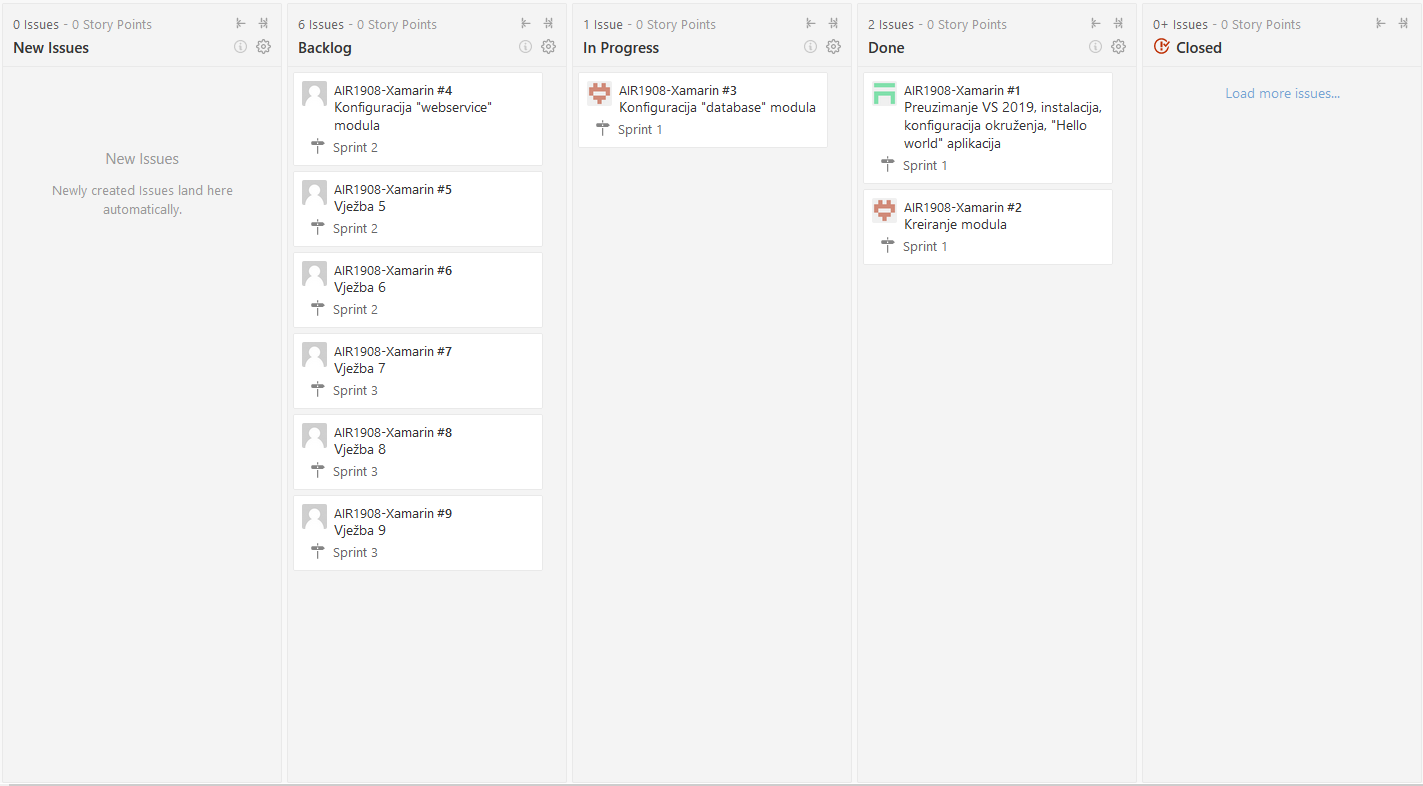
***Sažetak***

U ovoj vježbi konfigurirat ćemo database klasu. Kreirat ćemo klase entiteta „Store“ i „Discount“ i kreirat ćemo klase „MyDatabase“ i „Database“ koje će služiti za izvršavanje upita i za kreiranje lokalne baze. Također prikazat ćemo upisivanje podataka u bazu i prikaz istih na ekranu uređaja.

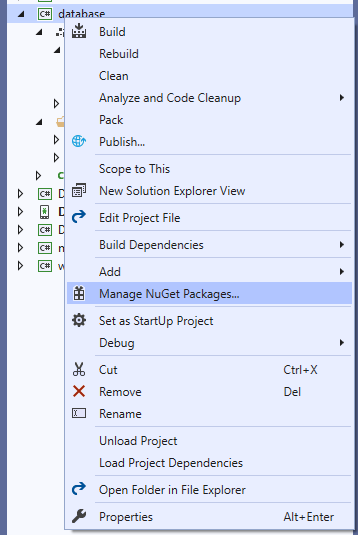
Zadnja verzija aplikacije: Commit 259dffc [<https://github.com/AIR-FOI-HR/AIR1908-Xamarin>]

***Što radimo?***

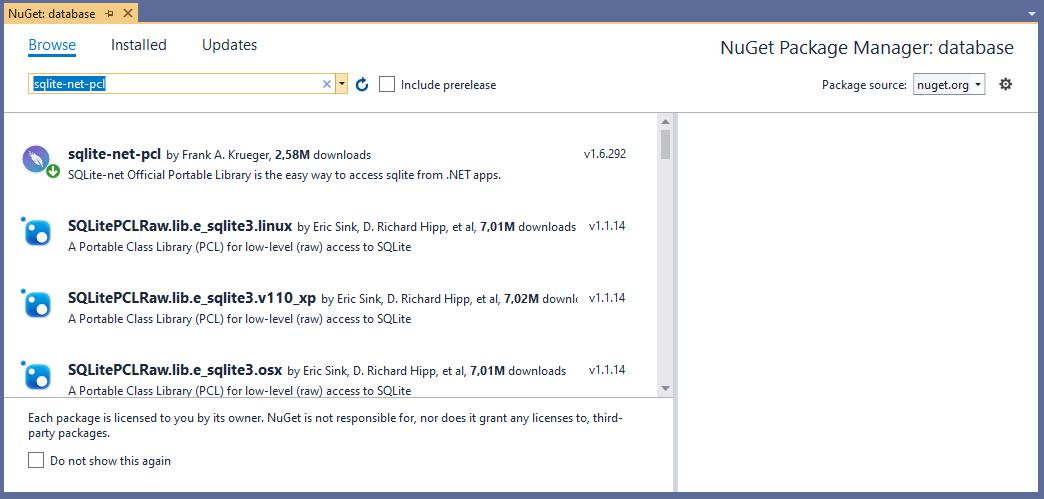


U ovom tutorialu obradit ćemo posljednju vježbu iz Sprinta 1. Nakon toga krećemo u Sprint 2 u kojem nas čeka konfiguracija webservice modula i ostale vježbe.

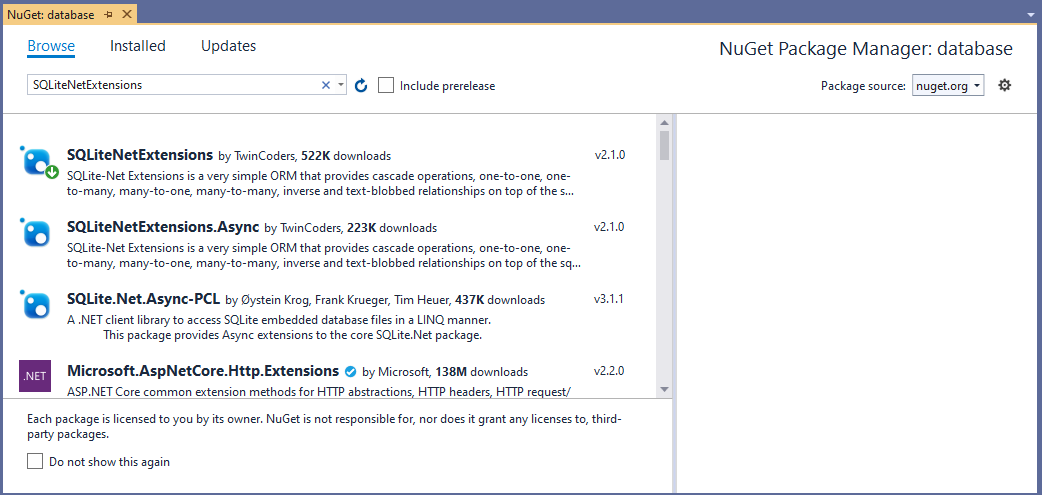
Kako bi kreirali bazu lokalno prvi korak je dodati „sqlite-net-pcl“ NuGet package.



Kliknemo desni klik na database klasu i „Manage NuGet Packages...“. Zatim upišemo „sqlite-net-pcl“ i instaliramo paket.



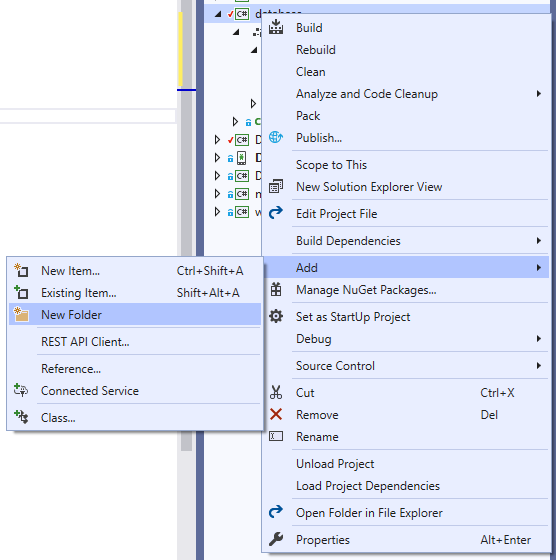
U paketima vidimo da se instalirao paket. Ali, potreban nam je još jedan paket kako bi mogli koristiti sve funkcionalnosti SQLite rada sa bazom. Na isti način možemo instalirati i taj drugi paket.



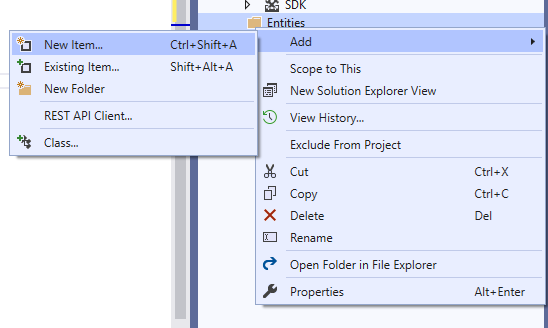
Time smo dodali još jedan paket. Zadnji korak konfiguracije je taj da moramo dodati „Dependencies“ u glavni projekt na database modul. Otvorimo projekt DiscountLocator19, desni klik na **Dependencies -> Add reference,** odaberemo database i kliknemo OK.

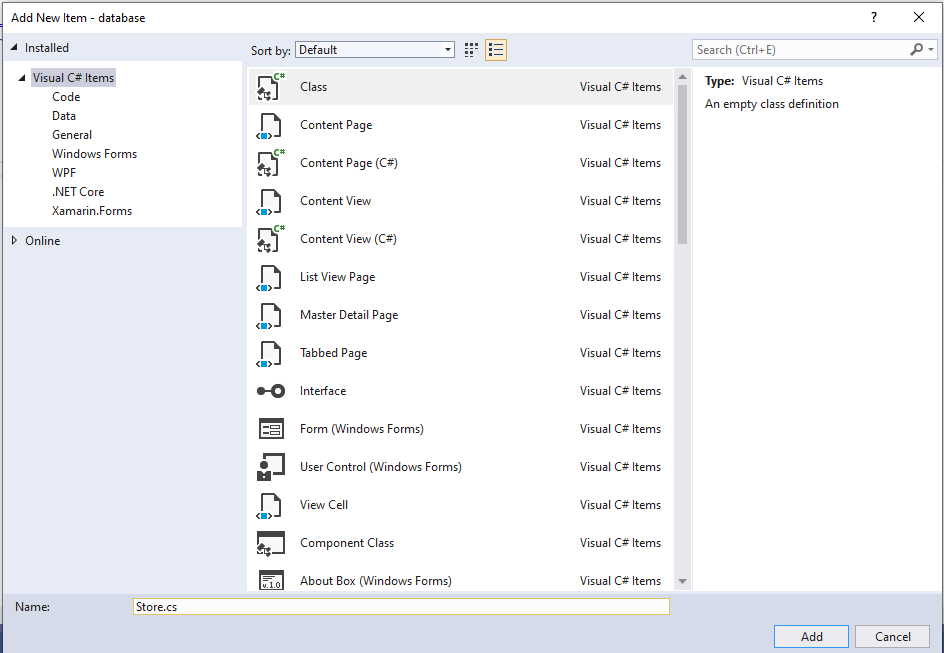
Kada smo konfigurirati sve što nam treba za database modul možemo početi sa kreiranjem entitetskih klasa.

Desni klik na database modul i dodamo novi folder po nazivom „Entities“.

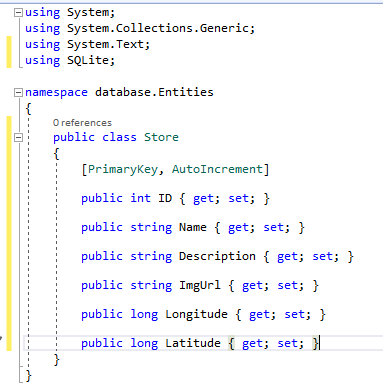


U taj folder stavit ćemo entitetske klase. Desni klik na novokreirani folder i dodamo novi item, entitetsku klasu s nazivom „Store“.

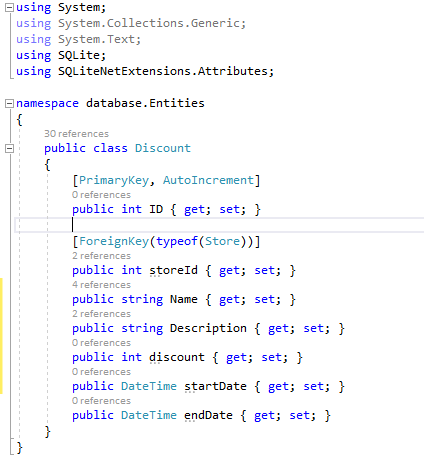




U tu klasu stavit ćemo atribute koje ima tablica Store u bazi.

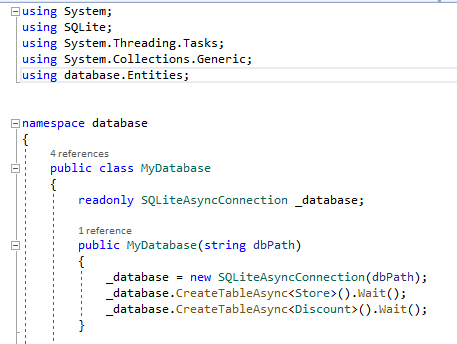


Na isti način kreirat ćemo klasu Discount i dodati joj atribute, ali s jednim vanjskim ključem koji povezuje Discount i Store.



Kada smo to završili, preimenujemo Class1.cs u MyDatabase.cs i možemo kreirati bazu, složiti pisanje u bazu i čitanje iz baze pomoću asinkronih SQLite upita koji se izvode u pozadini pomoću dretvi.

U konstruktor klase MyDatabase smještamo putanju gdje će baza biti lokalno spremljena i u nju asinkrono kreiramo tablice.

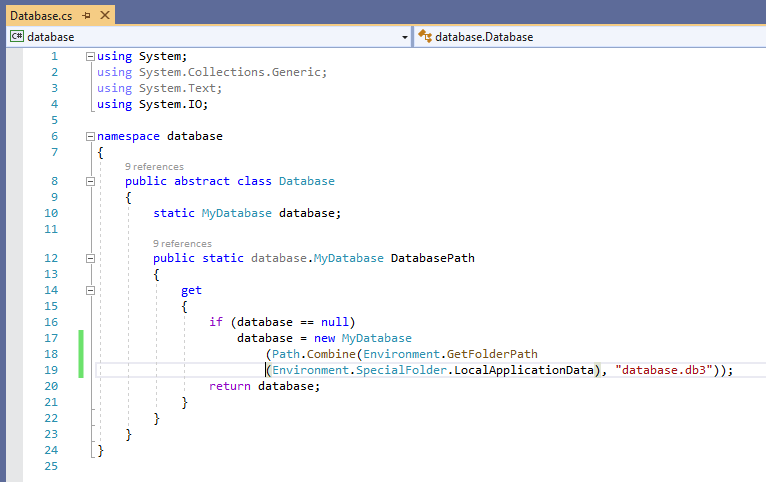


Nakon toga u istoj klasi kreiramo upite nad bazom odnosno implementiramo metode GetStores, InsertStores, InsertDiscounts, GetDiscounts i GetDiscountsById.

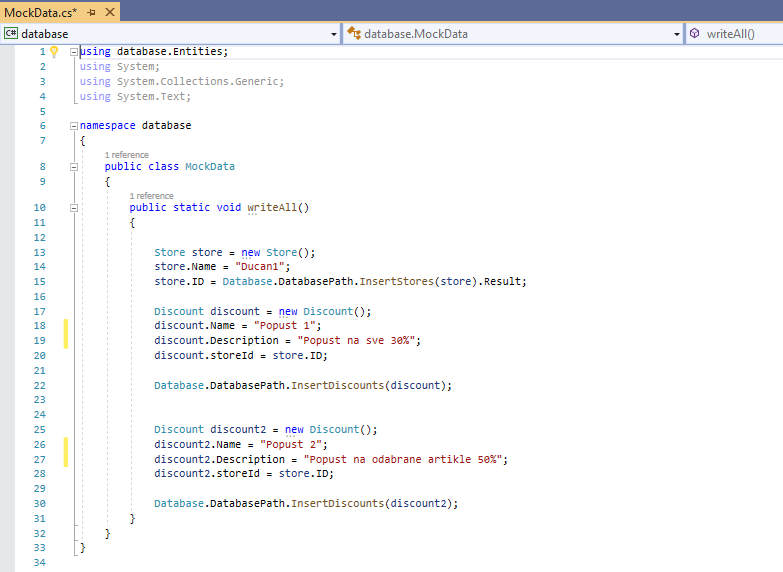


Mi smo za početak koristili prikazane operacije u daljnjim vježbama ćemo dodati upite koji će nam biti potrebni.

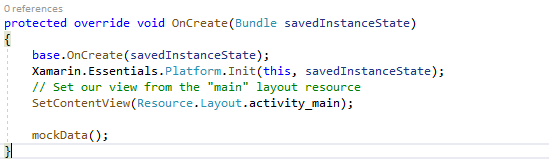
Kako bi kreirali bazu moramo otići u database modul i kreirati novu klasu koju ćemo nazvati Database. U njoj kreiramo property koji kreira instancu baze, lokalni dokument i ime dokumenta koji predstavlja bazu.



S ovime smo kreirali sve što je potrebno za bazu. Sljedeći korak je pokazati kako prikazati podatke iz baze. Napravit ćemo novu klasu u database modulu koju ćemo nazvati MockData. U ovoj klasi dodat ćemo jedan dućan s dva popusta koje ćemo prikazati na ekranu.



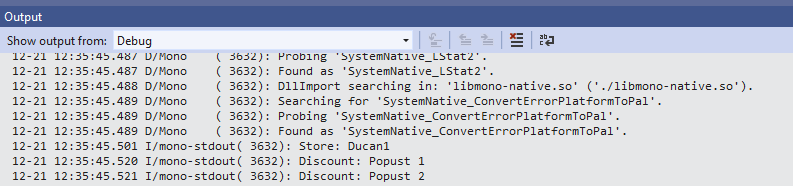
Sada ćemo testirati bazu ispisom njenih podataka u konzolu. U MainActivity klasi pozvat ćemo metodu mockData koja će ispisati podatke iz baze u konzolu. U OnCreate metodu dodajemo poziv mockData metode.



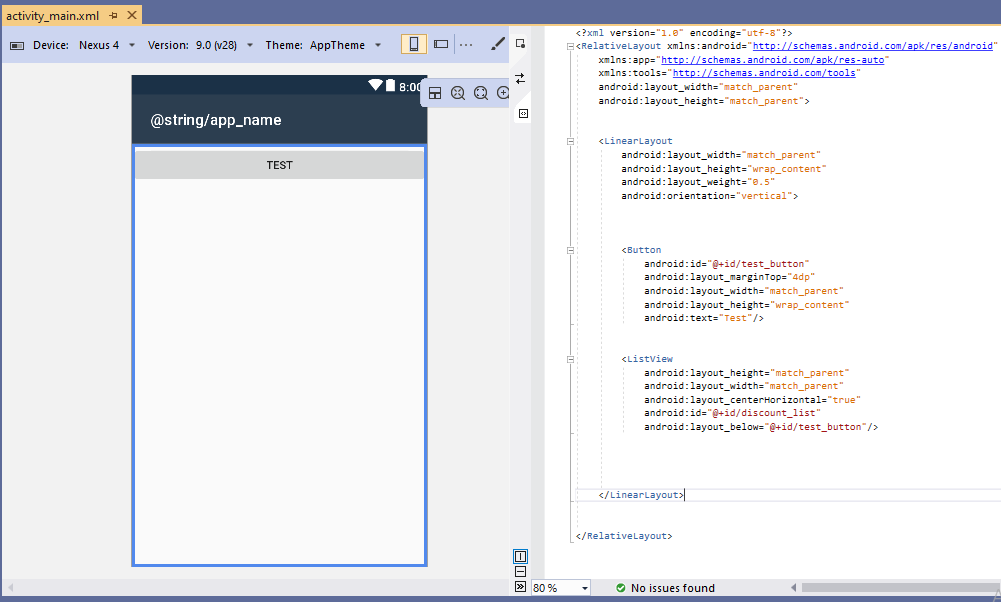
Sada implementiramo mockData metodu. Stvaramo novu listu dućana u koju spremamo dućane iz baze pomoću funkcije GetStores definirane u MyDatabase klasi. Za svaki dohvaćeni dućan u konzolu ispisujemo njegov naziv. Stvarano novu listu popusta u koju spremamo sve popuste trenutnog dućana funkcijom GetDiscountsByStoreId. U konzolu ispisujemo naziv svakog dućana.



Ovako izgleda ispis u konzoli:

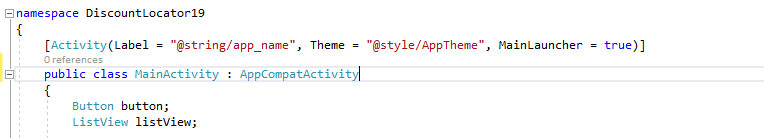


Sada ćemo pripremiti korisničko sučelje za prikaz podataka koje smo kreirali. Korisničko sučelje MainActivity klase konfigurira se u activity\_main.xml dokumentu. DiscountLocator19 > Resources > layout > activity\_main.xml. U jednom dijelu otvorit će nam se prikaz ekrana mobitela, a u drugom ćemo pisati xml da bismo rasporedili elemente na ekranu. Dodat ćemo LinearLayout element unutar kojeg ćemo smjestiti Button i ListView.

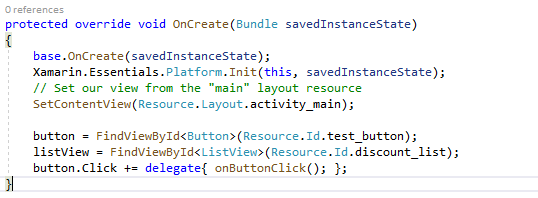


U xml dijelu svakom elementu dodajemo potrebne atribute koju mu mijenjaju stil. Dodali smo im i jedinstveni identifikator (android:id) kako bi ga mogli pronaći u kodu. Trenutno se na ekranu ne vidi ListView kojeg smo dodali, međutim on postoji.

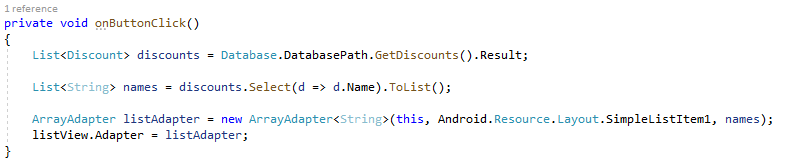
Sada je red na testiranju. U MainActivity klasi deklarirat ćemo gumb i ListView.



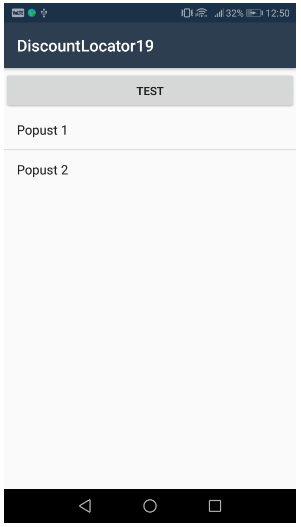
U OnCreate metodi u Main Activity klasi povezat ćemo gumb i ListView s onime što smo napravili u activity\_main.xml. Na klik gumba pozvat ćemo metodu onButtonClick.



Na klik gumba napravit ćemo novu listu popusta u koju ćemo spremiti popuste iz baze funkcijom GetDiscounts iz MyDatabase klase. Stvorit ćemo i listu naziva popusta u koju ćemo spremiti nazive. Koristimo Select metodu nad listom popusta. Da bismo ju koristili potrebno je uključiti biblioteku Linq tako da na početak MainActivity klase napišemo 'using System.Linq'. Stvaramo novu instancu ArrayAdapter-a u koju spremamo listu naziva. Zatim pozivamo metodu Adapter nad našim ListView elementom.



Sada pokrenemo aplikaciju i trebali bismo dobiti dva popusta na klik gumba 'Test'.



Ovom smo demonstracijom završili drugu vježbu u kojoj smo pokrili rad s bazom podataka. U sljedećoj ćemo vježbi podatke dohvaćati s web servisa.