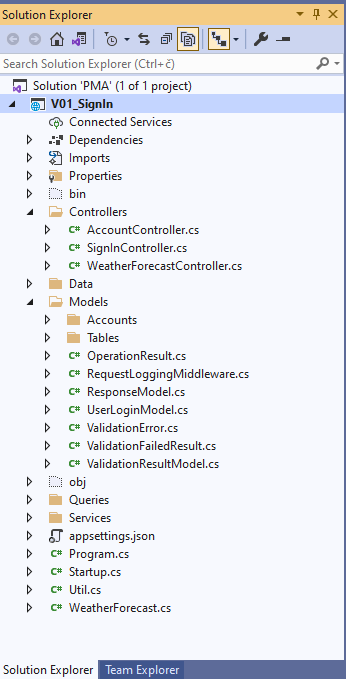
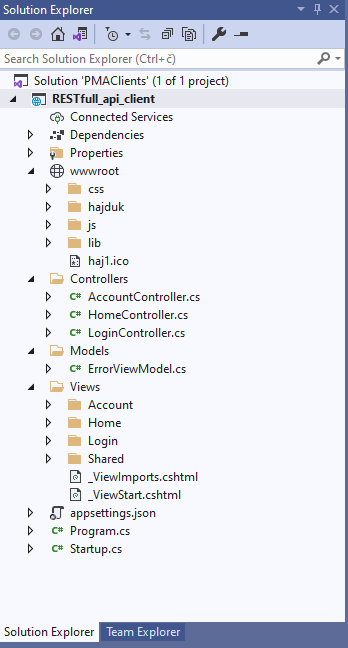
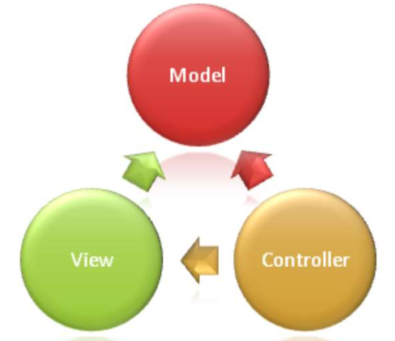
**DOKUMENTACIJA PROJEKTA IZ PMA**

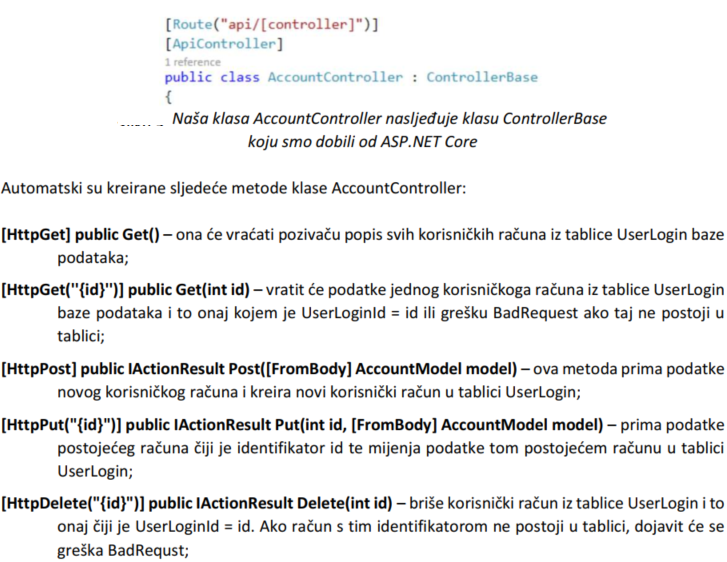
Cilj je napraviti web aplikaciju za prijavu članova HNK Hajduk Split utemeljenu na RESTful API. RESTful API je arhitektonski stil koji određuje pravila za programiranje web aplikacija pri čemu ***REST(REpresentational State Transfer)*** označava da se stanje aplikacije s klijenta prenosi serveru pri svakom pozivu funkcija s web servisa, a ***API(Application Interface)*** je skup definicija i protokola za izgradnju i interakciju aplikacijskog softvera. Aplikacija se sastoji od dva dijela: serverske i klijentske strane. Serversku stranu čine: operacijski sustav(Windows), web server(ISS), programski jezik za kodiranje poslovne logike(C#), sustav za upravljanje podacima(MS SQL Server), tehnologija za objektno predstavljanje baze podataka(Entity Framework) i jezik za interakciju sa poslužiteljem podataka(LINQ). Klijentsku stranu čine: jezik za kodiranje sadržaja korisničkog sučelja(HTML), jezik za kodiranje stila(izgleda) korisničkog sučelja(CSS),jezik za kodiranje interakcije sučelja s korisnikom(JavaScript), predefinirana knjižica stilova(Bootstrap) i bibliotečne funkcije interakcije s korisnikom(jQuery).

Obratimo sada pozornost na serversku stranu. Arhitektonski uzorak MVC(Model-View-Controller) razdvaja aplikaciju na strani servera na tri glavne skupine komponenti i on nam koristi da se postigne razdvajanje problema. Prema ovom obrascu, korisnički zahtjevi se preusmjeravaju na kontroler koji je odgovoran za rad s modelom radi izvršavanja korisničkih radnji i/ili dohvaćanja rezultata upita tj. Kontroler prihvaća zahtjeve klijenta, poziva metode podatkovnog sloja s parametrima prema modelu, podatkovni sloj vraća model s podacima i tada kontroler poziva view i šalje mu model radi stvaranja HTML koda s podacima iz modela. Svaka od ovih komponenti ima svoju zadaću.

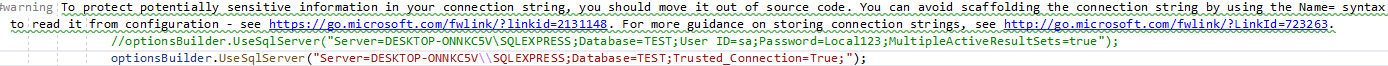


U RESTful usluzi se očekuje odgovor na zahtjeve, a najčešći protokol na zahtjeve i odgovore je HTTP koji nam pruža operacije(metode):GET(dobijanje podataka o objektu ili objektima koji postoje na serveru-upit tipa SELECT), POST(stvaranje novog objekta na serveru-operacija tipa INSERT), PUT(ažuriranje postojećeg objekta na serveru, zahtjeva vrijednosti svih atributa objekta-UPDATE), DELETE(brisanje objekta-DELETE) i PATCH(ažuriranje jednog dijela vrijednosti postojećeg objekta na serveru).

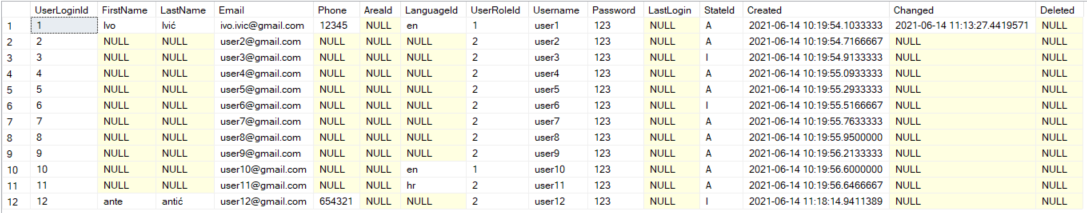
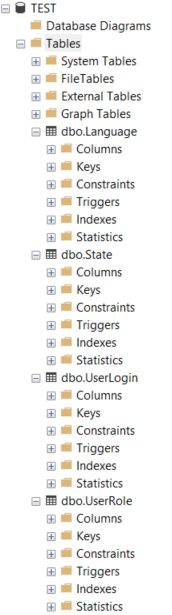


Sve operacije prema tablici UserLogin baze podataka izvršavat će klasa AccountManager koju smještamo u direktorij Services. Polja koja ima klasa AccountModel odgovaraju atributima tablice UserLogin. Zadaća objekata ove klase je da prenose podatke od AccountController prema AccountManager (metodama POST i PUT) ili od manager prema kontroleru metodom GET.

**BAZA PODATAKA**

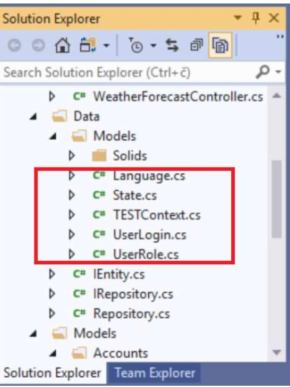
****

Komanda koja će nam u direktoriju Data/Models stvoriti klase koje opisuju bazu podataka TEST. Ovu komandu kopiramo u konzolu projekta svaki put kada promjenimo tablice ili njihova ograničenja u bazi podataka.



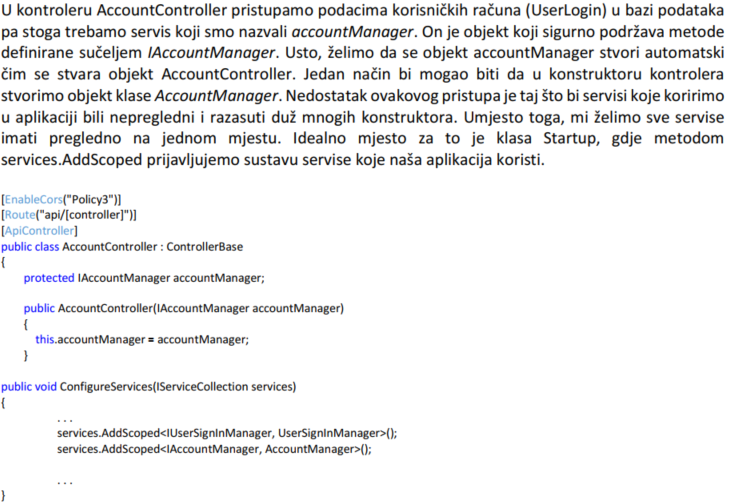
Mnoge web aplikacije usmjerene su prema podacima u bazi podataka. Dakle,  
na klijentu se zadaju podaci koje treba upisati u bazu, a isto tako se zadaju kriteriji pretraživanja baze podataka. Podaci zadani na klijentu u obliku zahtjeva putuju mrežom prema serveru, server ih analizira (engl. parsira) te prema tom zahtjevu postupi prema serveru baze podataka. Server baze podataka obavi operacije nad spremištem podataka te rezultat vrati istim tragom natrag prema pozivaču, sve do klijenta. Izgradanja baze podataka pomoću MS SQL Server nam čini podatkovni sloj, a predstavljanje baze podataka objektno se izvršava pomoću EF(Entity Framework) OEM sustava pristupom database.first(najprije oblikujemo bazu podataka i odredimo objekte baze podataka:kolone tablica i njihove tipove podataka, ograničenja null vrijednosti, primarni ključ svake tablice i strane ključeve(referencijalni integritet) pa onda automatski generiramo klase.

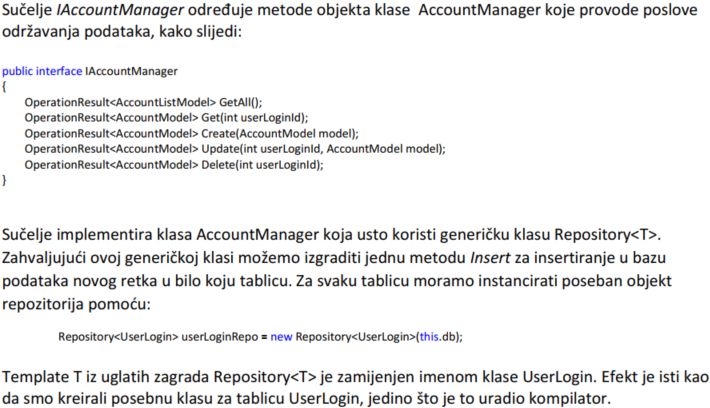
OEM EF Core kreirao je u direktoriju Data/Models sljedeće klase:

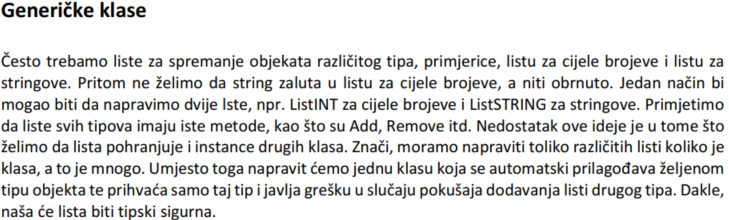


***INJEKTIRANJE***

Injektiranje je tehnika u kojoj objekt prima druge objekte o kojima ovisi. Ti se drugi objekti nazivaju ovisnostima, a sama se tehnika odnosi na prenošenje servisa u objekt koji ih koristi. Na ovaj se način programer koncentrira na klasu koju stvara i ne mora razmišljati o drugim klasama. Npr, AccController koristi uslugu AccManagera za poslove pristupanja i održavanja u bazi podataka.







Ante Barbarić, III.godina(Matematika i informatika)