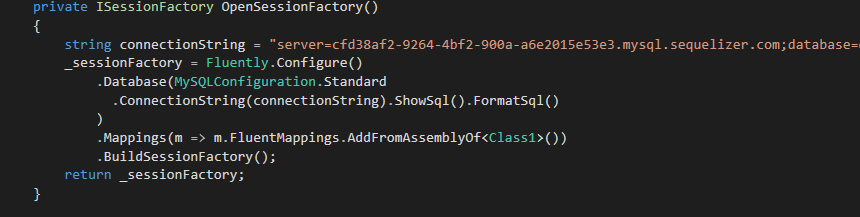
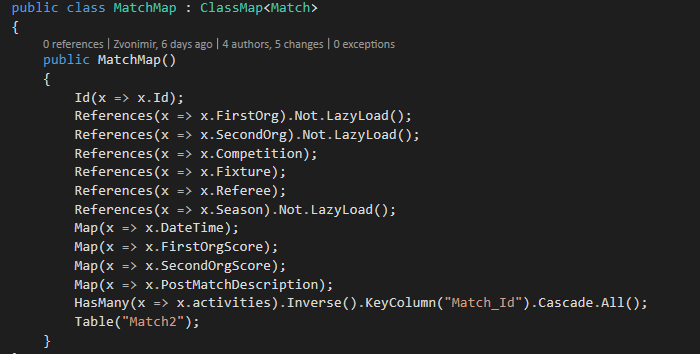
Fluent NHibernate

Za potrebe OR mapiranja koristili smo Fluent NHibernate za razliku od običnog NHibernatea koji koristi XML za konfiguriranje i mapiranje klasa.



Na slici vidimo isječak koda iz naše klase koja kreira session factory te služi za otvaranje sessiona. Prilikom konfiguracije session factorya naveli smo tip baze koju koristimo, connection string za spajanje na bazu, dodatne parametre (npr ShowSql i FormatSql koji nam omogućuju ispis SQL naredbi koje NHibernate kreira i provodi nad bazom) te naveli gdje da se gleda mapiranje klasa.



Prilikom mapiranja klasa koristili smo posebne klase koje kao baznu klasu imaju klasu definiranu u Fluent NHibernateu ClassMap. Tamo smo popisali mapiranje za sve atribute pojedine klase kao što se vidi na slikama. Oznaka Id označava da je ta kolona odnosno atribut identifikator klase u bazi. References označava strani ključ na neki drugi razred odnosno tip objekta. Map označava običan atribut pojedine klase. Oznaka HasMany označava master detail odnos između dvije klase odnosno tablice u bazi i olakšava nam spremanje podataka u bazu na način ako idemo spremiti master podatak uz njegove detail podatke NHibernate će ih sam spremiti automatski u obje tablice, svaku u svoju.

Repozitorije smo organizirali na način da imamo repozitorij za svaku važniju komponentu u sustavu. U repozitoriju se nalaze metode koje služe za dohvat i spremanje podataka u bazu odnosno služe za interakciju s bazom podataka. Tu su uz osnovne operacije definirani i malo složeniji upiti na bazu. Svaki repozitorij je definiran sučeljem koje govori koje sve metode mora repozitorij sadržavati kako bi zadovoljio sučelje. Svaka od metoda koristi svoju transakciju za spremanje i dohvat podataka. Nakon obavljanja upita transakcija se commita.

Najosnovniji način za dohvat objekta iz baze je uz pomoć funkcije Get<Tip objekta>(id) koja dohvaća objekt preko njegovom ID-a.

Najčešće smo podatke spremali uz pomoć SaveOrUpdate(tip objekta) funkcije koja govori NHibernate-u da ukoliko objekt ima ID potraži u bazi i izmijeni ga, u protivnom unese novi objekt u bazu podataka.

Za dohvat objekata ili liste objekata koristili smo LINQ upite koje NHiberante podržava.

Va result = (List<int>)session.QueryOver<OrgCompetition>().Where(orgC => orgC.Competition.Id == competitionId && orgC.Season.Id == seasonId).Select(OrgC => OrgC.Organization.Id).List<int>();

Primjer upita nad tablicim OrgCompetition koja dohvaća objekte koji imaju id natjecanja i sezone jednake zadanima. Prilikom dohvaćanja uzimaju se samo id-evi organizacija koje zadovoljavaju te uvijete odnosno kombinira se SELECT i WHERE dio SQL naredbe kako bi se dobila lista integera.