

Model podataka i model perzistencije – EloChess

Autori: Milan Stanković, Djordje Stanković

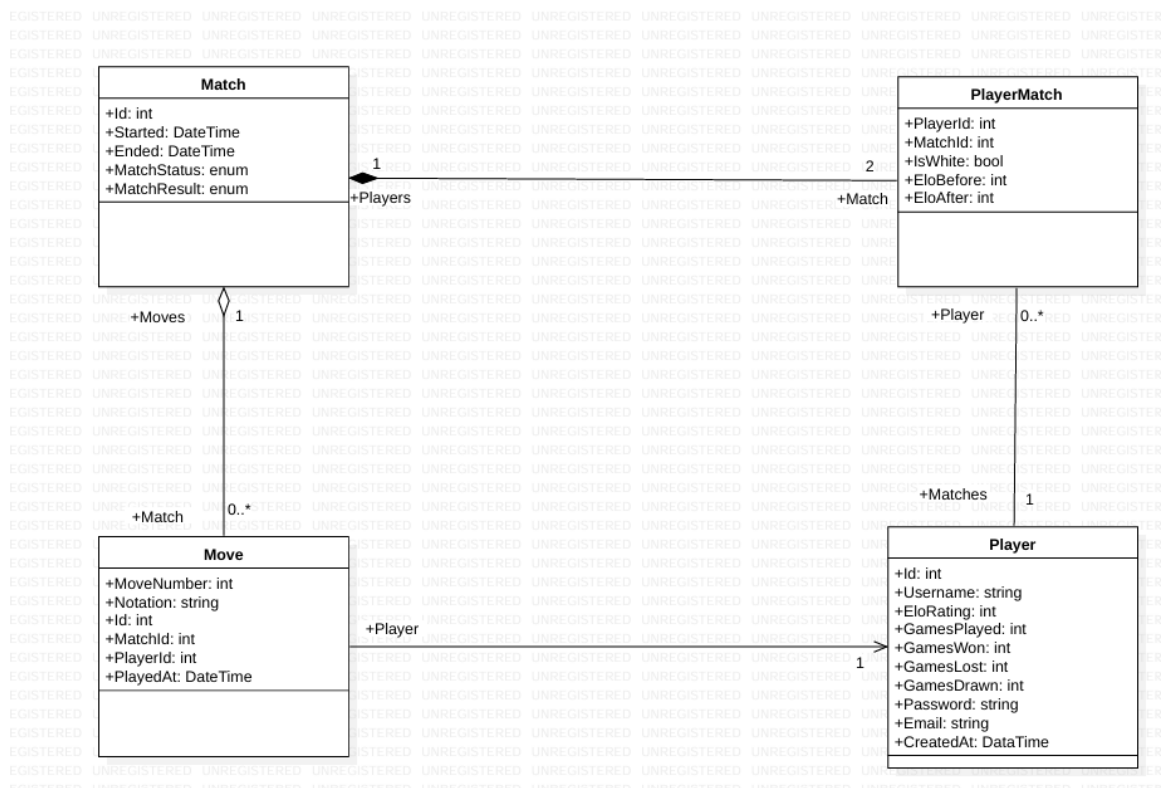
Projekat: EloChess

1. MODEL PODATAKA

Model podataka predstavlja konceptualni (business) model sistema i opisuje osnovne entitete šah aplikacije i njihove međusobne odnose, bez razmatranja načina perzistencije podataka.

1.1 Dijagram klasa

Dijagram klasa modela podataka obuhvata sledeće klase: Player, Match, Move i PlayerMatch, kao i enumeracije MatchStatus i MatchResult. Ovaj dijagram prikazuje logičku strukturu sistema i odnose između entiteta u realnom svetu.



1.2 Klase

1.2.1 *Player*

Klasa Player predstavlja korisnika sistema. Sadrži podatke o korisničkom imenu, ELO rejtingu i statistici odigranih partija. Jedan igrač može učestvovati u više mečeva.

1.2.2 *Match*

Klasa Match predstavlja šah partiju između dva igrača. Sadrži podatke o vremenu početka i završetka, statusu meča i konačnom rezultatu.

1.2.3 *Move*

Klasa Move predstavlja pojedinačan potez u partiji. Sadrži redni broj poteza, notaciju i informaciju o igraču koji je odigrao potez.

1.2.4 *PlayerMatch*

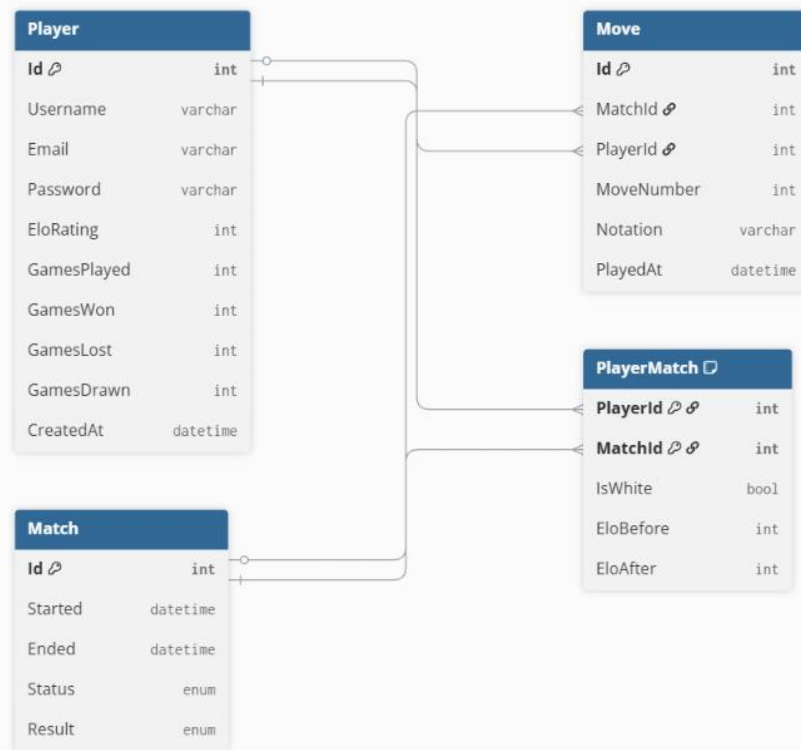
Klasa PlayerMatch predstavlja vezu između igrača i meča. Omogućava čuvanje dodatnih informacija kao što su boja figure i ELO rejting pre i posle meča.

2. MODEL PERZISTENCIJE

Model perzistencije predstavlja tehnički prikaz strukture baze podataka.

2.1 Dijagram entiteta

Dijagram entiteta prikazuje tabele Player, Match, Move i PlayerMatch, kao i njihove primarne i strane ključeve.



2.2 Entiteti

2.2.1 Player

Entitet Player sadrži primarni ključ Id i kolone za korisničke podatke. Predstavlja tabelu u bazi za čuvanje korisnika.

2.2.2 Match

Entitet Match sadrži informacije o partiji i povezan je sa entitetima PlayerMatch i Move.

2.2.3 Move

Entitet Move sadrži strane ključeve MatchId i PlayerId i predstavlja tabelu poteza.

2.2.4 *PlayerMatch*

Entitet *PlayerMatch* predstavlja spojnu tabelu između *Player* i *Match* entiteta.

3. MEHANIZAM MAPIRANJA

Za mapiranje između modela podataka i modela perzistencije koristi se Entity Framework Core. DTO klase se koriste za prenos podataka između API sloja i poslovne logike. Mapper klase omogućavaju konverziju između entiteta i DTO objekata.