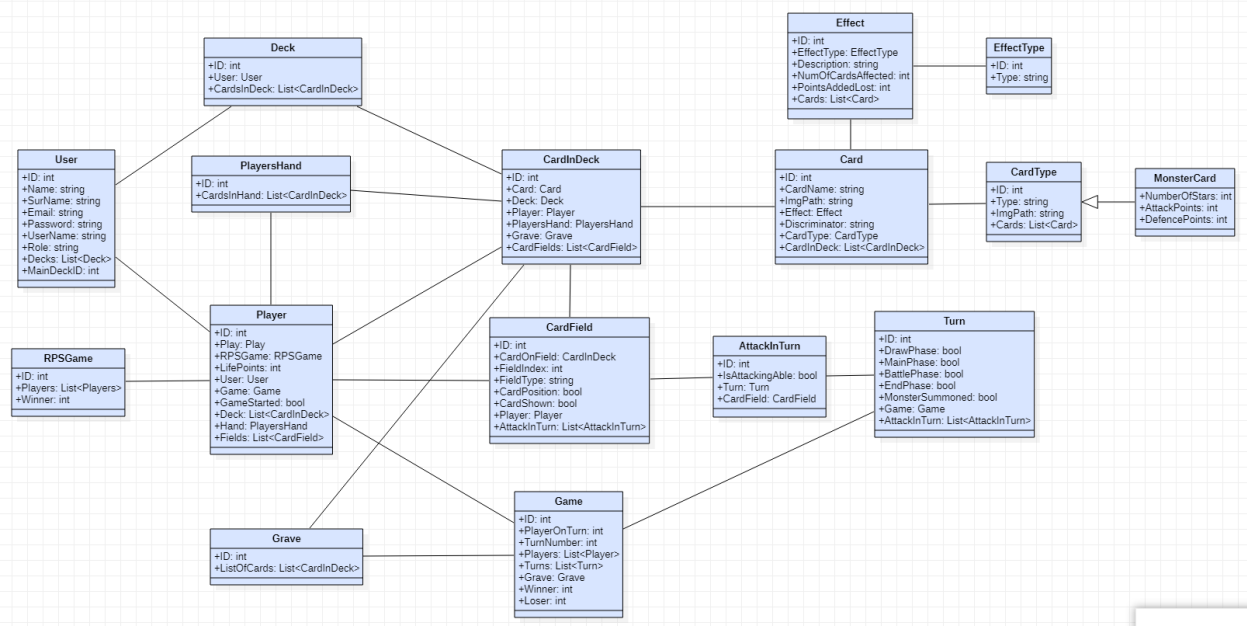
KNIGHTS&DIAMONDS

MODEL PODATAKA I MODEL PERZISTENCIJE

Dušan Spasić 16868
Danica Antić 16473

1. Model podataka

Model podataka u aplikaciji Knights&Diamonds predstavljen je sledećim klasnim dijagramom:



Slika 1. Dijagram modela podataka

Card – Osnovna klasa koja se koristi za predstavljanje karte, koja može biti Spell ili Trap.

MonsterCard – Klasa izvedena iz klase Card, od dodatnih property-ja ima bodove za napad, bodove za odbranu i level karte.

CardType – Klasa povezana sa klasama Card i MonsterCard, može biti Spell, Trap ili Monster.

Effect – Svaka karta je povezana klasom Effect, sam efekat se generiše prilikom kreiranja karte, ukoliko takav efekat već postoji, on će se dodeliti karti koju kreiramo. Na osnovu parametara koje šaljemo efektu, generišemo deskripciju same karte preko ***Factory*** patterna, a na osnovu upamćenih parametara se taj efekat u igrici izvršava pomoću ***Strategy*** patterna.

EffectType – Na osnovu tipova efekata se bira koji ConcreteFactory i ConcreteStrategy će se koristiti.

CardInDeck – Predstavlja vezu između karte i špila.

Deck – Sastoji se od liste CardInDeck objekata.

CardField – Na početku svake partije, za svakog Player-a se kreiraju 10 polja, 5 za Monster karte, 5 za Spell/Trap karte. Na osnovu parametra *Position* znamo da li je Monster karta u napadu ili odbrani, a na osnovu parametra *IsCardShown* znamo da li je karta neotkrivena ili otkrivena. Svako polje ima vezu sa klasom CardInDeck koje predstavlja koja karta se nalazi na tom polju.

RPSGame – Klasa koja se kreira kada izazvani korisnik prihvati zahtev za partiju. Pobednik RPS game-a je prvi na potezu u Game-u.

Game – Kreira se u isto kada i RPSGame, ima dva Player-a u game-u koja igraju jedan protiv drugog. Na početku igre životni bodovi igrača su 4000. Gubitnik je onaj Player koji prvi spadne na 0 životnih bodova.

Player – Klasa Player je veza između User i Game klase, sam Player se kreira kada User prihvati zahtev za partiju. Svaki Player ima svojih 10 polja, špil sa izabranim kartama, i ruku iz koje odigrava karte.

Turn – Klasa od proporty-ja ima *DrawPhase*, *MainPhase*, *BattlePhase* i *EndPhase*. DrawPhase izvlači jednu kartu iz špila. MainPhase služi za odigravanje karata iz ruke, prilikom koje se može odigrati jedna Monster karta i neograničen broj Spell ili Trap karti. U BattlePhase-u se vrši napadanje protivnika. Klikom na EndPhase se potez završava i kreira se novi potez za protivničkog igrača.

Grave – Klasa koja se sastoji od liste objekata CardInDeck. Predstavlja karte koje su otišle na groblje.

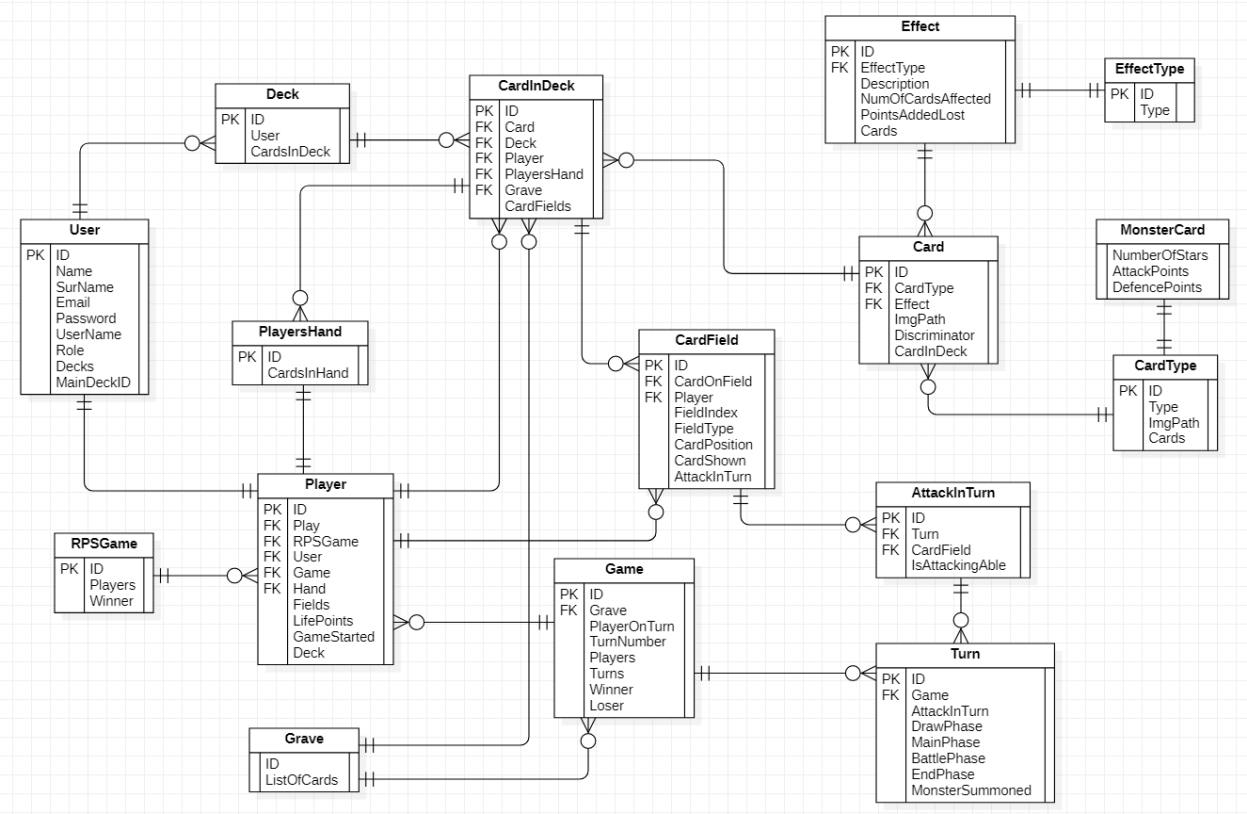
User – Klasa koja predstavlja samog korisnika aplikacije i sadrži sve potrebne informacije o njemu i podatke koji se koriste za autentifikaciju i autorizaciju.

AttackInTurn – Kreira se ulaskom u *BattlePhase* tokom poteza ukoliko igrač ima Monster kartu u napadačkoj poziciji.

PlayersHand – Klasa koja se sastoji od liste objekata CardInDeck koja predstavlja karte u ruci koje korisnik može da igra u toku poteza.

2. Model perzistencije

Prethodno predstavljeni model podataka se na odgovarajući način perzistira u bazi podataka, predstavljen u vidu modela entiteta:



Slika 2. Dijagram modela perzistencije

3. Mehanizmi mapiranja

Za perzistenciju i rad sa bazom podataka je korišćen Microsoft SQL Server, dok je za mapiranje između objekata klasa iz modela podataka i entiteta baze podataka korišćen objektno-relacioni maper *Entity Framework Core*. Princip koji se koristi za mapiranje je *code-first*, gde se na osnovu Entity klasa, kojima je prestavljen model podataka, kreiraju tabele relacione baze podataka. Prilikom mapiranja su primenjeni i DataLayer obrasci **Repository** i **UnitOfWork**. Takođe, pored modela podataka, iskorišćene su i *DTO klase* za prenos podataka između slojeva aplikacije.