

Model podataka i perzistencije

# Planet Wars

*Članovi tima:*

*Dorđe Jovanović, 17150*

*Nevena Tufegdžić, 17506*

*Naziv tima:*

*Negatim*

*Datum: 09.01.2022.*

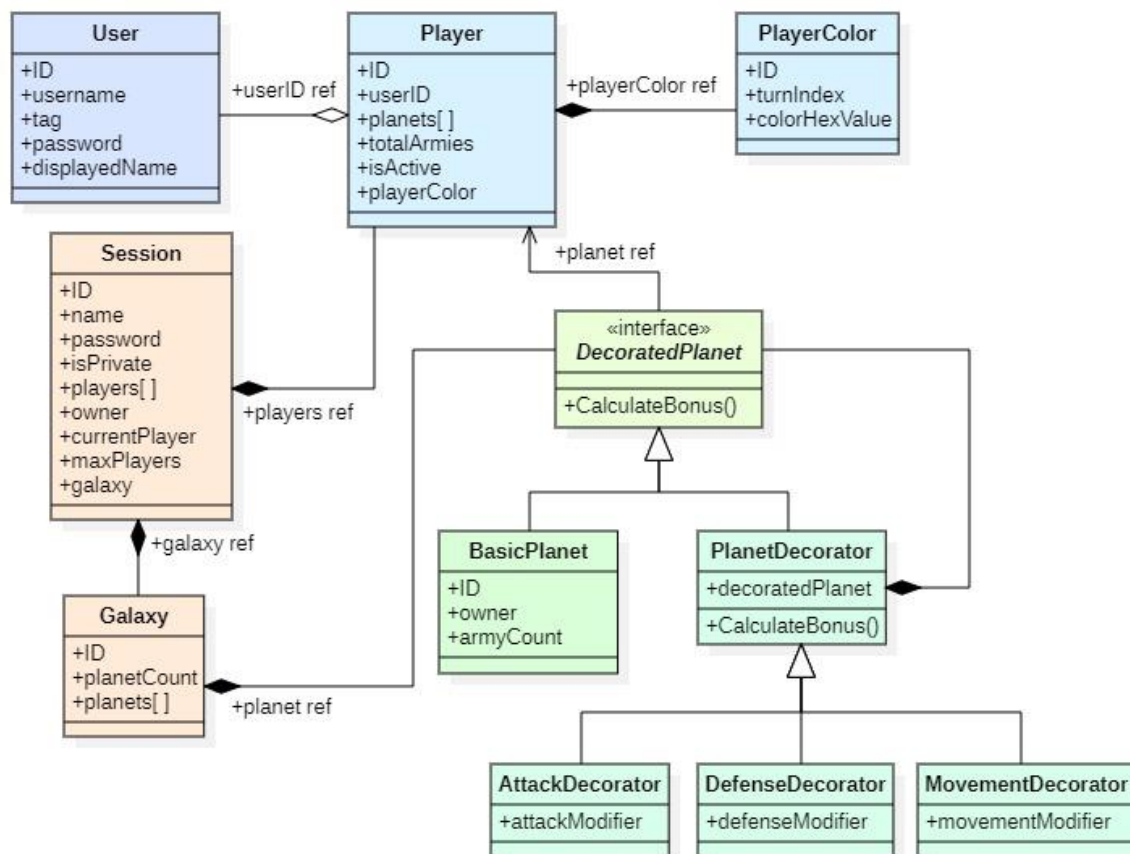
# SADRŽAJ

<b>1</b>	<b>MODEL PODATAKA .....</b>	<b>3</b>
1.1	DIJAGRAM KLASA.....	3
1.2	KLASE.....	3
1.2.1	<i>User</i> .....	3
1.2.2	<i>Player</i> .....	3
1.2.3	<i>PlayerColor</i> .....	4
1.2.4	<i>Session</i> .....	4
1.2.5	<i>Galaxy</i> .....	4
1.2.6	<i>Planete</i> .....	4
<b>2</b>	<b>MODEL PERZISTENCIJE.....</b>	<b>5</b>
2.1	DIJAGRAM ENTITETA .....	5
2.2	ENTITETI .....	6
2.2.1	<i>User</i> .....	6
2.2.2	<i>Player</i> .....	6
2.2.3	<i>PlayerColor</i> .....	6
2.2.4	<i>Session</i> .....	6
2.2.5	<i>Galaxy</i> .....	6
2.2.6	<i>Planet</i> .....	6
2.2.7	<i>PlanetPlanet</i> .....	6
<b>3</b>	<b>MEHANIZAM MAPIRANJA.....</b>	<b>7</b>

# 1 MODEL PODATAKA

## 1.1 DIJAGRAM KLASA

Na sledećem dijagramu klasa predstavljen je model podataka sistema Planet Wars:



## 1.2 KLASA

Model podataka sastoji se iz sledećih klasa.

### 1.2.1 User

User klasa ima zadatak da modeluje korisnika. Sadrži ID, onako kako se pamti u bazi. Sadrži korisničko ime, kao i tag koji je unikatan u celom sistemu za dato korisničko ime. Ovim principom se pribegava kako bi korisnici mogli da imaju bilo koje korisničko ime, čak i ako je ono zapravo zauzeto. Korisničko ime sa tagom, zajedno sa passwordom, se koristi za autentifikaciju korisnika. Pored toga, korisnik ima i atribut displayName.

### 1.2.2 Player

Player klasa služi kako bi se korisnik povezo sa pokrenutom igrom. Kada se korisnik priključi nekoj igri (u daljem tekstu sesija), dodeljuje mu se Player objekat. Jedan korisnik može istovremeno biti u više sesija, zbog mogućnosti snimanja igre i kasnijeg nastavljanja, a sesija ujedno uključuje više korisnika, tako da Player služi da poveže korisnike sa sesijama. Player ima svoju boju, koja je određena njegovim rednim brojem u sesiji. Pored toga, svaki igrač ima niz

planeta koje poseduje, kao i ukupan broj armija. Atribut `isActive` pokazuje da li igrač učestvuje u igri, odnosno da li je pobeđen.

### **1.2.3 PlayerColor**

Klasa koja čuva boju igrača u sesiji, i redni broj poteza (potezi se odvijaju ciklično među igračima u sesiji). Boja je vezana za redni broj igrača, a vrednosti su predefinisane.

### **1.2.4 Session**

Klasa Sesija ima centralnu ulogu u okviru igre. Sesija okuplja igrače koji učestvuju, a igrači se priključuju sesiji pomoću njenog imena i password-a. Sesija koja nije označena kao privatna (`isPrivate`) ima prazno polje za password. Sesija prihvata maksimalni broj igrača (`maxPlayers`), i čuva niz igrača koji su u sesiji. Čuva se i vlasnik sesije, odnosno korisnik koji je kreirao sesiju, kao i igrač koji je trenutno na potezu.

### **1.2.5 Galaxy**

Galaxy povezuje niz planeta sa datom sesijom. Sadrži niz planeta i broj planeta u galaksiji, a njena svrha je da se kreiranje planeta odvoji od same sesije.

### **1.2.6 Planete**

#### **1.2.6.1 DecoratedPlanet**

Interfejs koji predstavlja osnovu za planete. Implementiran je decorator obrazac koji služi da objedini osnovnu planetu sa željenim resursima.

#### **1.2.6.2 BasicPlanet**

Klasa koja predstavlja osnovnu planetu, koja sadrži vlasnika planete i broj armija koje se nalaze na toj planeti.

#### **1.2.6.3 AttackDecorator**

Dodaje Attack bonus (kao attack resurs) na planetu. Sadrži `attackModifier` atribut.

#### **1.2.6.4 DefenseDecorator**

Dodaje Defense bonus na planetu i sadrži `defenseModifier`.

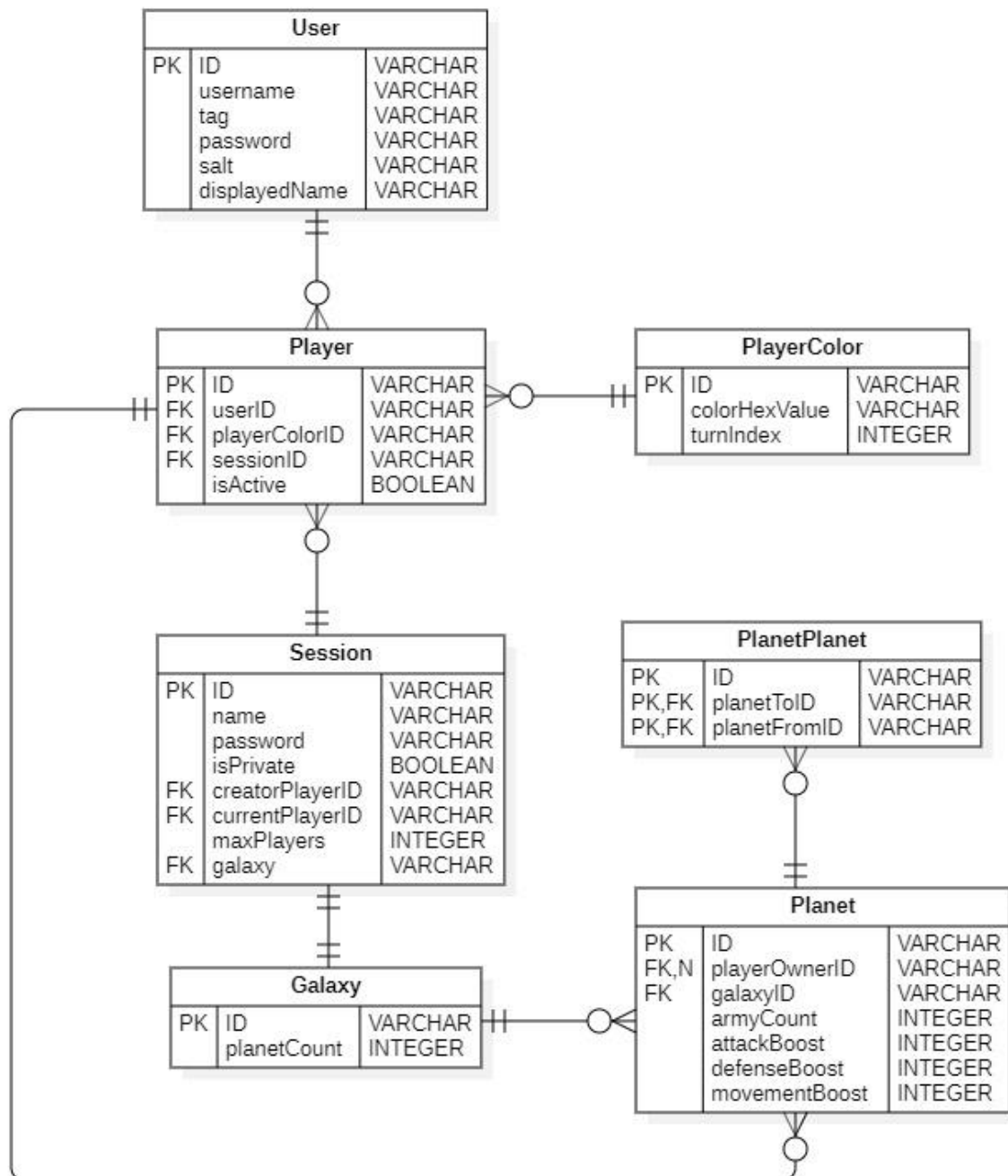
#### **1.2.6.5 MovementDecorator**

Dodaje Movement bonus na planetu pomoću `movementModifier`.

## 2 MODEL PERZISTENCIJE

### 2.1 DIJAGRAM ENTITETA

Na sledećem dijagramu prikazani su entiteti sistema:



## 2.2 ENTITETI

### 2.2.1 User

User entitet služi za skladištenje podataka o korisniku. Sadrži ID korisnika, username, tag, password u heširanom obliku radi sigurnosti, SALT podatak kao dodatak password-u radi sigurnosti, i displayName.

### 2.2.2 Player

Player entitet služi za skladištenje podataka o igračima. Sadrži User ID koji povezuje igrača sa korisnikom. Jedan korisnik može biti u više sesija, stoga se njegov ID može naći u više igrača. Igrač ima svoju boju, tako da sadrži i PlayerColor ID. Jedan igrač može imati samo jednu boju. Igrač je povezan sa sesijom tako što ima Session ID. Sesija sadrži više igrača, ali jedan igrač može da pripada samo jednoj sesiji.

### 2.2.3 PlayerColor

PlayerColor entitet sadrži predefinisani skup boja za redni broj korisnika u sesiji. Kada korisnik pristupa sesiji, kreira se njegov Player koji se na osnovu trenutnog broja korisnika u toj sesiji povezuje sa odgovarajućom torkom iz PlayerColor. Broj igrača u sesiji je ograničen.

### 2.2.4 Session

Sesija sadrži ID, name, password, i podatak da li je privatna sesija (isPrivate). Takođe sadrži i maksimalni broj igrača. Povezana je sa entitetima Player, tako što čuva ID igrača na potezu, kao i ID igrača koji je kreator sesije. Sa druge strane, povezana je i sa entitetom Galaxy, tako što čuva ID galaksije koja toj sesiji pripada.

### 2.2.5 Galaxy

Galaxy entitet povezuje sesiju sa skupom planeta koje se nalaze u sesiji. Sadrži ukupan broj planeta.

### 2.2.6 Planet

Planet entitet služi za perzistiranje planeta u igri. Planeta pripada jednoj galaksiji, i to je određeno pomoću galaxyID. Planeta može da ima vlasnika, nekog igrača, i u tom slučaju je playerOwnerID postavljen na ID tog igrača. Ovaj atribut je nullable, jer planeta može biti bez vlasnika, što je slučaj na početku igre.

Planete mogu imati određene resurse. Svaki resurs ima sopstveni atribut čija se vrednost postavlja na vrednost modifikatora kojeg taj resurs nudi armijama na planeti. Planeta takođe pamti i broj armija koji se trenutno nalazi na njoj.

### 2.2.7 PlanetPlanet

Planete su u galaksiji organizovane u vidu neorijentisanog grafa. Preslikano na model entiteta, ovo znači da entitet planeta mora biti u vezi sa samim sobom i to u many to many formatu. Kako bi se ovo formiralo, postoji PlanetPlanet entitet koji čuva međusobne relacije planeta. Planete koje su direktno povezane u grafu imaju svoju torku u tabeli, gde se povezuju perzistiranjem njihovih ID-jeva.

### 3 MEHANIZAM MAPIRANJA

---

Planet Wars koristi MS SQL bazu podataka, formiranu code-first pristupom. Kao ORM alat, korišćen je Entity Framework Core za uspostavljanje entiteta u bazi i komunikaciju sa bazom.

Za implementaciju nivoa perzistencije korišćeni su Repository obrazac i Unit Of Work. Repository je iskorišćen u generičkom obliku, gde svaki entitet koristi svoj Repository kako bi izvršavao upite. Repository nivo koristi Service nivo, koji je ubačen kako bi se kontrolerske klase odvojile od samog rada sa modelima. Kontroleri imaju zadatak pružanja usluga klijentu.

Kako bi se viši slojevi odvojili od sloja perzistencije, implementirani su Data Transfer Objects koji imaju zadatak da predstave modele iz perzistencije na način koji poznaje klijent. Prevođenje iz DTO u modele vrši se na nivou servisnih klasa, tako da kontroleri poznaju rad sa DTO-ima, dok repozitorijumi ostaju na nivou modela.