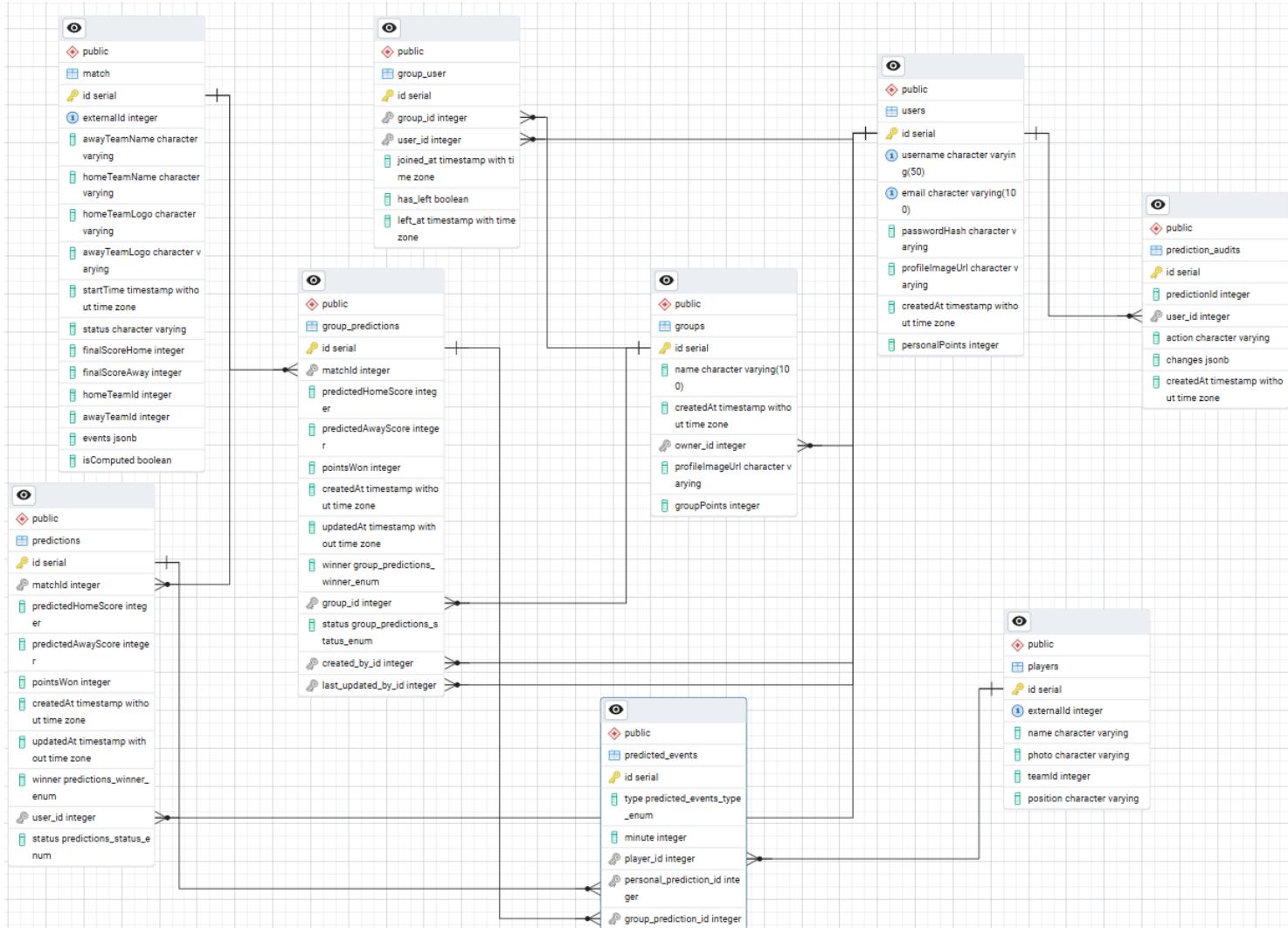


Druga faza

1. Model egzistencije (Entity Model)

Model perzistencije je realizovan u okviru PostgreSQL relacione baze podataka. Sastoji se od 9 povezanih tabela koje obezbeđuju čuvanje podataka o korisnicima, mečevima, grupama i predikcijama.



1.1. Tabele

1.1.1. Users

Centralna tabela koja čuva podatke o korisnicima.

- **Ključne kolone:** id, username, email, passwordHash, personalPoints (ukupni bodovi korisnika).
- **Uloga:** Osnova za autentifikaciju i rangiranje igrača.

1.1.2. Match

Sadrži sve informacije o sportskim događajima.

- **Ključne kolone:** id, homeTeamName, awayTeamName, startTime, status (NS, LIVE, FT), finalScoreHome/Away, events (JSONB polje za tok meča), isComputed (flag za obradu predictions koji su za dati mec).
- **Uloga:** Izvor podataka za simulaciju i validaciju predikcija.

1.1.3. Groups

Tabela za takmičarske grupe (lige unutar aplikacije).

- **Ključne kolone:** id, name, ownerId (FK ka users), groupPoints.
- **Uloga:** Omogućava grupisanje korisnika i privatna takmičenja.

1.1.4. Group_user

Pivot tabela koja povezuje korisnike i grupe.

- **Ključne kolone:** groupId, userId, joinedAt, hasLeft.
- **Uloga:** Implementira Many-to-Many vezu; jedan korisnik može biti u više grupa, i jedna grupa ima više korisnika.

1.1.5. Predictions (Personalne predikcije)

Korisnikove prognoze za mečeve van grupe. Samo korisnik može da menja svoju prognozu

- **Ključne kolone:** userId, matchId, predictedHome/AwayScore, pointsWon, status (SUBMITTED, PROCESSED).

- Uloga: Čuva "tikete" koje pojedinac odigra.

1.1.6. Group_predictions

Prognoze koje se daju u okviru određene grupe. Svi članovi grupe mogu da menjaju prognozu(samo jedan u jednom trenutku).

- Ključne kolone: groupId, matchId, createdBy, predictedHome/AwayScore, pointsWon, status(SUBMITTED, PROCESSED,DRAFT).
- Uloga: Specifične predikcije čiji bodovi idu na konto grupe/korisnika unutar te grupe.

1.1.7. Players

Podaci o igračima (fudbalerima).

- Ključne kolone: id, name, teamId, position.
- Uloga: Koristi se za predikciju specifičnih događaja (npr. ko će dati gol).

1.1.8. Predicted_events

Detaljne prognoze unutar meča.

- Ključne kolone: type (GOAL,ASSIST), playerId, personalPredictionId, groupPredictionId.
- Uloga: Omogućava dodatne bodove ako korisnik pogodi strelca ili strelca i minut.

1.1.9. Prediction_audits

Tabela za istoriju izmena predikcija.

- Ključne kolone: predictionId, userId, action, changes (JSONB).
- Uloga: "Log" sistem koji prati ko je i kada menjao svoju prognozu (bitno zbog regularnosti).

1.2. Relacije

1.2.1. One-to-Many (1:N):

- users → predictions: Jedan korisnik ima mnogo prognoza.

- match → predictions: Jedan meč se nalazi na mnogo različitih listića.
- groups → group_predictions: Jedna grupa sadrži mnoštvo grupnih prognoza.
- players → predicted_events: Jedan igrač može biti deo mnogih prognoza za strelca.

1.2.2. Many-to-Many (N:M):

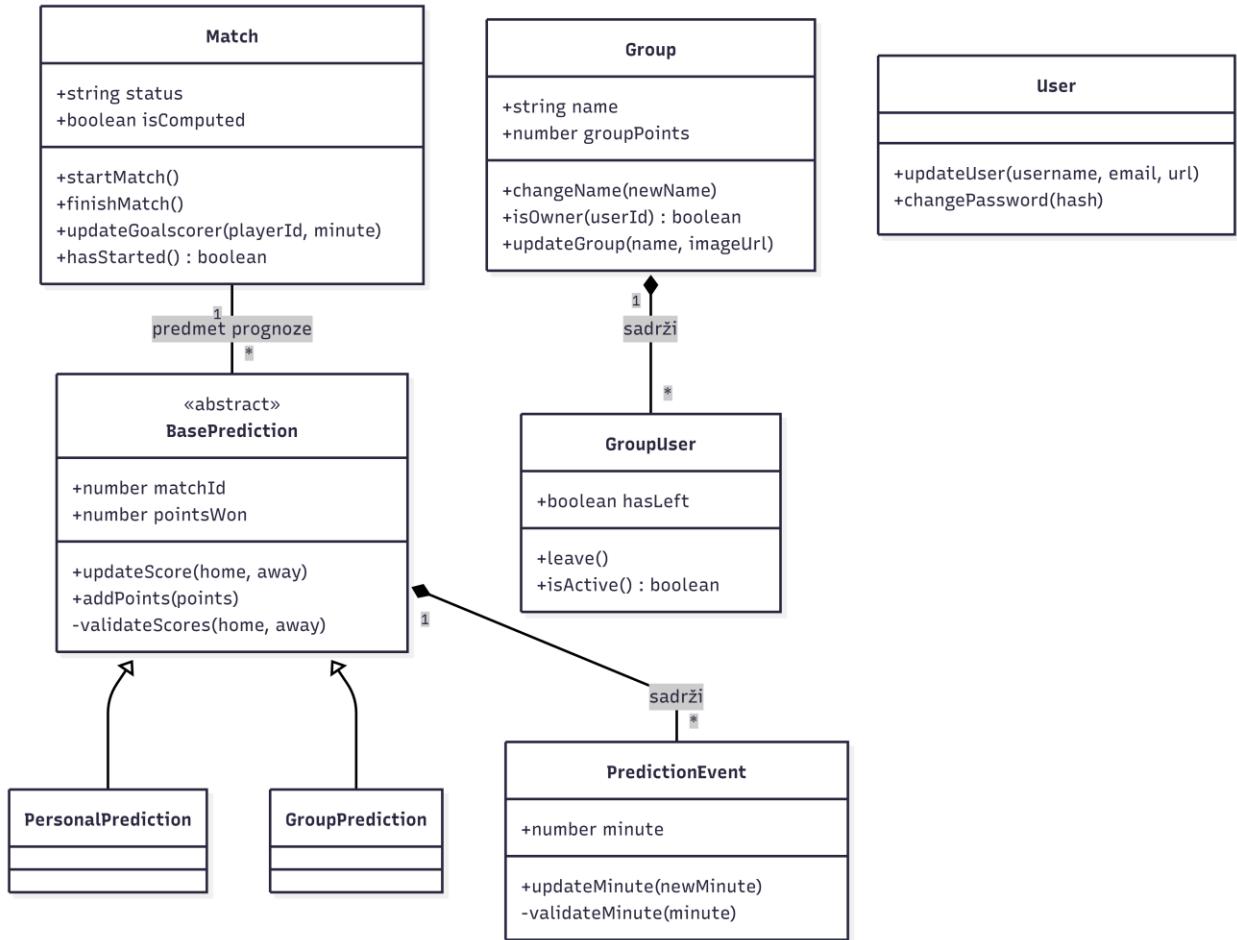
- users → groups: Realizovano preko group_user tabele. Ovo je ključno za socijalni aspekt aplikacije.

1.2.3. Polimorfne / Opcione veze:

- predicted_events je povezan i sa personal_predictions i sa group_predictions. To znači da jedan "event" može pripadati ili privatnom ili grupnom listiću.

2. Model podataka

Biznis model aplikacije ScoreMinds je implementiran kao set čistih TypeScript klasa koje enkapsuliraju pravila igre, validacije i logiku bodovanja. Za razliku od entiteta perzistencije, ovi modeli su „živi“ objekti koji garantuju integritet podataka pre nego što oni stignu do baze. Svi atributi su privatni, a pristup podacima se vrši isključivo preko get metoda. Modeli ne dozvoljavaju kreiranje nevalidnih stanja (npr. negativni rezultati ili minuti utakmice van opsega). Klase sadrže metode za promenu stanja (npr. finishMatch, leaveGroup) koje automatski ažuriraju povezana polja (poput updatedAt ili winner statusa).



2.1. Modeli

- User predstavlja takmičara. Sadrži logiku za praćenje ukupnog broja poena i učešće u grupama.
- Match centralni objekat simulacije. Upravlja stanjem meča (statusom) i čuva tok događaja (golovi, kartoni). Utakmica prolazi kroz strogta stanja: NS (nije počela) -> LIVE (u toku) -> FT. Metode updateGoalscorer i updateAssistant dinamički pune niz događaja koji su ključni za kasnije obračunavanje bonus poena.
- Prediction predstavlja „intelektualnu svojinu“ korisnika. Sadrži prognozirani rezultat i metode za poređenje sa stvarnim rezultatom. validateMinute striktno ograničava unos minuta na opseg od 0 do 130, čime se pokrivaju regularno vreme, produžeci i sudska nadoknada.
- Group definiše zatvoreno takmičarsko okruženje sa sopstvenim pravilima rangiranja. GroupUser model sprečava višestruko napuštanje grupe proverom `_hasLeft` stanja.

3. Biblioteka za perzistenciju i mapiranje

Koriste se Data mapperi i Repository sloj.

3.1. Data Mapper pattern

Mapperi pretvaraju enttiy u model i obrnuto.

- **toDomain()**: Funkcija koja uzima „sirovi“ entitet iz baze i pretvara ga u pametni biznis objekat spremjan za logičke operacije.
- **toPersistence()**: Funkcija koja biznis objekat priprema za snimanje, pretvarajući ga nazad u entitet koji TypeORM razume.

Tok podataka pri snimanju predikcije:

1. Korisnik šalje podatke (DTO).
2. Servis kreira Prediction Business Model i proverava biznis pravila (npr. da li je meč već počeo).
3. PredictionRepository prima biznis model.
4. PredictionMapper pretvara model u PredictionEntity.
5. Repozitorijum vrši perzistenciju u PostgreSQL bazu.

3.2. Repository sloj

- Servisi u aplikaciji (poput CalculatingService) ne znaju za bazu. Oni od Repozitorijuma traže Model.
- Repozitorijum interno koristi Mapper i TypeORM da bi izvršio zadatak.

4. Biznis pravila

Pravila koriste modele za rad. Neka od njih su:

- **Pravilo bodovanja:** Predikcija ne može dobiti bodove dok status meča u modelu nije FT (Finished).
- **Pravilo zaključavanja:** Rezultat u predikciji se ne može menjati nakon što je startTime meča prošao.
- **Pravilo vlasništva:** Samo vlasnik grupe (ownerId) u modelu može da inicira brisanje ili promenu imena grupe.

