

FAZA II - Perzistencija

1. Business model (domain)

Implementirane su sledeće klase:

- User
- Trip-plan
- Trip-member
- Activity
- Trip-lock

2. Entity model (persistence model)

Perzistencija je realizovana korišćenjem **PostgreSQL** baze i **TypeORM**-a.

Entity klasa	Naziv tabele u bazi
user.entity	users
trip-plan.entity	trip_plans
trip-member.entity	trip_members
activity.entity	activities
trip-lock.entity	trip_locks

3. Mapiranje business model ↔ entity model

Mapiranje je realizovano kroz Mapper klase:

- User.mapper
- Trip-plan.mapper
- Trip-member.mapper
- Activity.mapper
- Trip-lock.mapper

Svaki mapper sadrži dve osnovne funkcije:

- toEntity(domainObject)
- toDomain(entityObject)

4. Biblioteka za snimanje i učitavanje (save/load)

Sloj perzistencije koristi **Repository Pattern** koji obezbeđuje TypeORM.

Primer operacija:

- repository.save(entity)
- repository.find()
- repository.findOne(...)

5. Korišćeni projektni obrasci

5.1 Repository Pattern

Repository Pattern se koristi za pristup bazi podataka i izolaciju perzistentnog sloja.

U projektu se primenjuje nad entity klasama navedenim u poglavlju 2 (user.entity, trip-member.entity...), korišćenjem **TypeORM** repozitorijuma u servisima.

Repository omogućava osnovne **CRUD** operacije:

- repository.**save**(entity)
- repository.**find**()
- repository.**findOne**(...)

Na ovaj način poslovna logika ostaje nezavisna od konkretne baze podataka.

5.2 Data Mapper Pattern

Data Mapper Pattern se koristi za razdvajanje business modela i entity modela.

U projektu je realizovan kroz Mapper klase koje povezuju već definisane domenske i entity klase: **User, Trip-plan, Trip-member, Activity, Trip-lock**.

Svaki mapper sadrži dve osnovne metode:

- toEntity(domainObject)
- toDomain(entityObject)

Ovim obrascem omogućeno je da domenske klase ostanu potpuno nezavisne od ORM-a.

19101 Nađa Dinić
18549 Nađa Stošić