

# Dokument: Analiza SOLID principa u dijagramu klasa

Ovaj dokument opisuje kako su SOLID principi primijenjeni na model-klase i ostale komponente sistema Matchletic u kontekstu klasnog dijagrama.

## 1. Single Responsibility Principle (SRP)

Svaka klasa ima **jednu** i samo jednu odgovornost:

- **Korisnik, Oglas, Mec, Recenzija, OglasKorisnik, MecKorisnik:**
  - Sadrže isključivo attribute i podatke vezane za svoju domenu (npr. `Korisnik` čuva korisničke podatke, `Recenzija` čuva ocjenu i komentar, `MecKorisnik` predstavlja vezu između meča i korisnika).
- **KontekstPodataka:**
  - Upravljanje `DbSetovima` model-klasa, centralno skladištenje i pristup podacima.
- **Kontroleri** (npr. `KorisnikKontroler`):
  - Imaju isključivo rute i logiku za HTTP zahtjeve/odgovore za svoju domenu.
- **Pogledi:**
  - Samo predstavljaju podatke za prikaz, nemaju poslovnu logiku.

**Obrazloženje:** Razdvajanje modela, pristupa podacima, logike i prezentacije omogućava lakše održavanje i testiranje.

## 2. Open/Closed Principle (OCP)

Klase su otvorene za **proširenje**, ali zatvorene za **modifikaciju**:

- Model-klase definišu osnovne attribute i mogu se proširiti dodatnim klasama bez promjene postojećih.
- Dodavanje nove logike ne zahtijeva izmjene u postojećim klasama.
- Kontroleri mogu biti prošireni novim rutama bez promjena postojećih metoda.

**Obrazloženje:** Sistem se može nadograđivati bez potrebe da se postojeće klase mijenjaju, čime se smanjuje rizik od uvođenja grešaka.

## 3. Liskov Substitution Principle (LSP)

Podklase mogu zamijeniti superklase bez promjene očekivanog ponašanja:

- U ovom sistemu ne koristi se nasljeđivanje između modela.
- Umjesto toga, odnosi su modelirani pomoću pomoćnih klasa kao što je `MecKorisnik`, čime se izbjegavaju problemi s nasljeđivanjem.

**Obrazloženje:** Zamjena nasljeđivanja agregacijom i veznim klasama doprinosi jasnijem modeliranju i poštivanju očekivanog ponašanja.

## 4. Interface Segregation Principle (ISP)

Klase ne bi trebale biti prisiljene da zavise o metodama koje ne koriste:

- Operacije su razdvojene po kontrolerima:
  - `KorisnikKontroler` sadrži samo metode za korisnike.
  - `RecenzijaKontroler` samo metode za recenzije itd.

**Obrazloženje:** Svaka klasa koristi samo metode koje su joj potrebne, bez viška zavisnosti. Iako interfejsi nisu implementirani u ovom sistemu, načelo je ispoštovano razdvajanjem funkcionalnosti u posebne kontrolere.

## 5. Dependency Inversion Principle (DIP)

Visokonivojske klase ne bi trebale zavisiti direktno od niskonivojskih:

- Kontroleri direktno koriste `KontekstPodataka`.
- U idealnom slučaju, između kontrolera i `KontekstPodataka` bi postojao sloj servisa i interfejsi, ali u ovoj verziji to nije implementirano.

**Obrazloženje:** Iako nema interfejsa ni servisa, struktura sistema može se lako nadograditi njihovim uvođenjem. U trenutnoj verziji, princip nije u potpunosti zadovoljen, ali postoji prostor za poboljšanje.

## Zaključak

U dijagramu klasa za sistem Matchletic većina SOLID principa je zadovoljena:

- Postoji jasna **separacija odgovornosti**.
- Sistem je **modularan** i spreman za proširenje.
- DIP i ISP nisu u potpunosti ostvareni zbog odsustva interfejsa i servisnog sloja, ali struktura omogućava njihovo naknadno uvođenje.

Ovaj dokument služi kao osnova za provjeru dizajna i buduće nadogradnje sistema u skladu sa principima objektno orijentisanog dizajna.