

## **S - Single Responsibility Principle**

Svaka klasa ima jednu odgovornost. Na primjer, klasa `RegistrovaniKorisnik` upravlja samo podacima o korisniku.

Svaka klasa ima jasno definisanu svrhu:

**RegistrovaniKorisnik:** čuva podatke o korisniku.

**Event:** opisuje događaj.

**Recenzija:** čuva korisničku ocjenu i komentar za događaj.

**Kupovina:** evidentira kupovinu ulaznica.

**Bookmark:** bilježi favorite.

Ne postoji klasa koja ima više nepovezanih odgovornosti.

## **O - Open/Closed Principle**

Model omogućava prosirenje bez izmjene postojećih klasa, npr. dodavanjem novih funkcionalnosti.

## **L - Liskov Substitution Principle**

Iako trenutni model ne uključuje nasljeđivanje, dizajn dozvoljava buduće proširenje, npr

`AdminKorisnik` i `RegistrovaniKorisnik` mogu biti izvedene klase iz bazne `Korisnik` klase.

## **I - Interface Segregation Principle**

Klase imaju samo one attribute koji su im potrebni. Sistem se može lako proširiti interfejsima, npr `IKupac`, `IRecenzije` čime direktno možemo ISP primijeniti.

## **D - Dependency Inversion Principle**

Model je spreman za korištenje apstrakcija i razdvajanje slojeva, što podržava nezavisnost visokonivojskih komponenti.

## **Zaključak**

Model je uskladjen sa SOLID principima i spreman za proširenje i održavanje