

# **SOLID principi u sistemu eApoteka**

## **S - Princip pojedinačne odgovornosti (SRP)**

Osoba i njezine podklase (korisnik, apotekar, dostavljač): svaka klasa ima jednu primarnu odgovornost. Osoba raspolaže osnovnim osobnim podacima (id, ime i prezime). Korisnik je odgovoran za radnje koje se odnose na kupca kao što su narudžbe i kreiranje upita.. Farmaceut upravlja zadacima specifičnim za apoteku kao što je odgovaranje na upite i rukovanje odobrenjima za narudžbe. Dostavljač se fokusira na aspekt dostave.

Narudžba, StavkaNarudžbe, DetaljiDostave, DetaljiPlaćanja, StatusNarudžbe: Svaka klasa odgovorna je za poseban dio procesa narudžbe. Narudžba upravlja agregacijom stavki narudžbe i ukupnim podacima o narudžbi. StavkaNarudžbe obrađuje informacije koje se odnose na pojedinačne stavke u narudžbi. DetaljiPlaćanja i DetaljiDostave upravljaju posebnostima obrade plaćanja i logistike isporuke. StatusNarudžbe prati promjene statusa narudžbe.

## **O - Otvoreno/zatvoreno princip (OCP)**

Klase proizvoda, narudžbe i korisnika: ove su klase otvorene za proširenje, ali zatvorene za izmjene. Na primjer, nova funkcionalnost može se dodati proizvodima ili narudžbama bez mijenjanja postojeće baze kodova, poput dodavanja novih vrsta proizvoda ili dodatnih pravila obrade za narudžbe. Slično tome, dodatne pojednostiosti kao "loyalty program" mogu se dodati Korisniku bez izmjene postojećih funkcionalnosti.

Proširenje osobe: Mogu se dodati nove vrste osoblja ili korisničkih uloga (npr. menadžer ili tehničara) proširenjem klase osoba, čime se pridržava OCP-a bez utjecaja na postojeću funkcionalnost.

## **L - Liskov princip zamjene (LSP)**

Nasljeđivanje od klase Osoba: Podklase osobe (korisnik, apotekar, dostavljač) mogu se koristiti naizmjenično tamo gdje se očekuju instance osobe, bez utjecaja na ponašanje programa. To je osigurano dosljednim interfejsima klasa i ponašanjem naslijeđenim od klase Osoba.

## **I - Princip izoliranja interfejsa (ISP)**

Specijalizirani interfejsi: Svaka klasa implementira interfejse koji su relevantni samo za njezinu funkcionalnost, osiguravajući da klase nisu prisiljene implementirati metode koje ne koriste. Na primjer, Korisnik se ne bavi direktno metodama inventara proizvoda kojima bi mogao upravljati Administrator.

## **D - Princip inverzije ovisnosti (DIP)**

Moduli visokog nivoa ne ovise o modulima niskog nivoa: oba ovise o apstrakcijama. Na primjer, Narudžba vrši interakciju s StavkaNarudžbe,, DetaljiDostave, DetaljiPlaćanja i StatusNarudžbe putem njihovih apstraktnih prikaza, a ne konkretnih implementacija. To podstiče odvajanje i fleksibilnost u arhitekturi sistema.

### **Praktične primjene u sistemu eApoteka**

Pridržavajući se ovih principa, sistem eApoteka promoviše robusnu, održivu i skalabilnu arhitekturu. Odvajanje u dobro definirane klase osigurava da promjene u jednom dijelu sistema imaju minimalan utjecaj na druge. Na primjer, modificiranje procesa plaćanja ili dodavanje novih vrsta proizvoda može se postići bez ometanja osnovne funkcije obrade narudžbi ili upravljanja korisnicima. To dovodi do lakših nadogradnji, otklanjanja greški i usklađenosti s novim zahtjevima kako složenost sistema eApoteka raste.

Zaključno, primjena principa SOLID-a u dizajnu sistema eApoteka ne samo da pojednostavljuje razvoj i buduće održavanje, već također olakšava lakše proširenje i skaliranje sistema kako bi se zadovoljile rastuće potrebe.