

## Single Responsibility Principle

U prethodnoj verziji smo imali eksplicitno naglašene metode za provjeru konzistencije koje smo u novoj verziji uklonili jer se pretpostavlja da svaka klasa vodi računa da drži samu sebe konzistentnom. Operacije spašavanja (npr. nedovršenog drafta ponude koju kreira turistička agencija) smo izdvojili od samih pojedinačnih klasa te smjestili u posebni StagingDAO kako bi smo ispoštovali SRP. Uveli smo niz novih klasa koje upotpunjuju dijagram klasa i ispunjavaju SRP poput klasa: FinansijskaAnaliza, Transakcija, Pretraga, Poruka, Rezervacija – gdje svaka od njih ima jasno definiranu odgovornost i domen.

## Open / Closed Principle

Uveli smo tri interfejsa koji predstavljaju ključne apstrakcije našeg modela: Posao, Korisnik, Rezervabilno. Posao predstavlja sve potencijalne subjekte koji bi mogli izdavati turističke usluge i kao takav podrazumijeva pravni minimum za obavljanje poslova. Trenutačno je to Agencija, no u budućnosti bi mogle biti i fizička lica, lanci hotela, prevozne kompanije itd. Interfejs Rezervabilno je namijenjen da modelira sve usluge koje bi naš sistem potencijalno trebao da podrži, trenutno su to podklase Usluge te agregacije istih kao Ponude, no upotreba interfejsa omogućava proširenja na nove vrste proizvoda. Isto je urađeno i sa interfejsom Korisnik – sada se otvara mogućnost elegantnijeg proširenja na nove vrste korisnika npr. VIP korisnik, Premium korisnik, Free Trial korisnik itd.

## Liskov Substitution Principle

Nasljeđivanja iz Usluge, odnosno sve implementacije interfejsa Rezervabilno ispunjavaju ovaj princip. Metode su deklarirane na vrhu, odnosno na nivou interfejsa, dok specijalizacije uvode dodatne attribute izvedenim klasama. Interfejsi Posao i Korisnik imaju trenutno samo po jednu „praznu“ klasu koja ih implementira pa je ovaj princip trivijalno zadovoljen.

## Interface – Segregation Principle

Interfejsi u našem dijagramu nisu veliki, nemaju puno metoda, nema ih mnogo a i poštuju SRP, pa stoga ih nismo smatrali da ima potrebe da ih razbijamo na specifičnije interfejse u ovom trenutku. Stoga smatramo da je naš dijagram u skladu sa ovim principom.

## Dependency Inversion Principle

Centralni koncepti naše aplikacije (Posao, Rezervabilno i Korisnik) su sve apstrakcije čime je ovaj princip ispoštovan kada je riječ o pomenutim klasama. Sve ostale klase višeg reda se oslanjaju na ove apstrakcije čime je princip ispoštovan i za takve klase. Naš dijagram bi dopuštao i uvođenje dodatnih apstrakcija za još „labaviji“ coupling klasa - recimo kod klasa nižeg reda poput Poruka, Transakcija ili klasa Administrator, Pretraga, FinansijskaAnaliza. No međutim odlučili smo u ovom trenutku da ne uvodimo dodatne apstrakcije s obzirom da se one ne odnose na suštinu aplikacije (turizam) a uvećali bi kompleksnost samog dijagrama.