

STRUKTURALNI PATERNI

Za naš projekat odlučili smo se da dodamo dizajn paterne **Facade** i **Proxy**. Prvo da navedemo njihovu generalnu namjenu.

Zaštitni proxy omogućava pristup i kontrolu pristupa stvarnim objektima.

Facade patern se koristi kada sistem ima više podsistema (subsystems) pri čemu su apstrakcije i implementacije podsistema usko povezane. Osnovna namjena Facade paterna je da osigura više pogleda (interfejsa) visokog nivoa na podsisteme čija implementacija je skrivena od korisnika.

Sada navodimo konkretnu primjenu u našem sistemu.

Façade pattern ćemo iskoristiti da bismo povezali sve klase koje su neophodne za pružanje svih funkcionalnosti koje naš sistem nudi korisniku. Pošto korisnik ima više operacija koje može izvršiti a one su dio različitih klasa (npr. pregled zahtjeva iz Korisnik klase, ObradiZahtjev iz Zahtjev klase I sl.) iskoristićemo fasadu da napravimo jedan high-level pogled na sistem da sakrijemo unutrašnje implementacije funkcionalnosti.

Zaštitni proxy ćemo iskoristiti jer naš sistem podržava login na korisničke račune, i podržava različite tipova korisnika sa različitim pravima pristupa, te moramo omogućiti kontrolu pristupa. Iskoristićemo zaštitni proxy kao način provjere ulaznih podataka prilikom pristupa računu, te omogućiti ili onemogućiti korištenje funkcionalnosti sistema na osnovu istih.

Novi dijagram klasa sa dodanim paternima izgleda kao na slici:

