

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.

Pored ovih paterna, postoje i drugi poput: Bridge, Adapter, Decorator, i Composite.

Adapter pattern bismo mogli primijeniti na naš sistem u budućnosti ako bismo htjeli integrisati neke vanjske servise čiji format podataka se razlikuje od našeg, kao na primjer servis za validaciju adresa ili nešto slično, međutim, trenutno nemamo potrebe za tim.

Bridge pattern bismo mogli primijeniti ako se odlučimo u budućnosti da dokumente vraćamo u drugim formatima osim PDF datoteka, na primjer DOCX ili XML. Međutim, kako naš sistem generiše samo 1 tip datoteka, nema potrebe.

Composite pattern bi teže mogli primijeniti, jedina mogućnost bi bila grupisanje dokumenata po nekim osobinama, međutim ne vidimo neku korist od toga.

Decorator pattern bismo mogli primijeniti u budućnosti ako htjednemo slati dokumente na različite načine, poput email, sms, in-app notifikacije i slično. Na taj način bismo nadograđivali mogućnosti slanja nekom baznom notifikacijskom objektu.

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:

