

FACADE

Facade pattern možemo koristiti prilikom kreiranja novih korisničkih zahtjeva ili žalbi. Prilikom kreiranja zahtjeva potrebno je izvršiti validaciju podataka, kreirati novi zahtjev ili žalbu, te isti dodati u list zahtjeva i žalbi sistema kojoj administrator može pristupiti. Ovdje je korisniku potrebna samo informacija o tome da li je njegov zahtjev ili žalba kreirana, a ono što se događa u pozadini se korisnika ne treba da se tiče. Uslijed toga, ovdje bi bilo prikladno koristiti Facade pattern.

DECORATOR

Decorator pattern bi mogli koristiti u slučaju da želimo korisniku omogućiti da uredi svoj profil npr. da stavi novu profilnu sliku ili za prikaz recenzija (prosječne ocjene ili slično) za korisnika.

COMPOSITE

Budući da posjedujemo različite vrste zahtjeva, možemo implementirati Composite pattern tako što u apstraktnoj klasi Zahtjev implementiramo metodu `dajPodatke()`. U tom slučaju, navedena metoda bi kao povratnu vrijednost imala bitne podatke o zahtjevu.

PROXY

Možemo koristiti kako bi Volonteru ili Građaninu omogućili da samo vide svoje zahtjeve koje su započeli, dok administrator ima pravo pristupa svim zahtjevima u sistemu.

ADAPTER

Možemo napraviti klasu Izvještaj. Ova klasa bi nam služila za kreiranje izvještaja o korisniku ili o nekom zahtjevu. Kako ne bi morali praviti dvije različite klase izvještaja možemo implementirati adapter koji omogućava da ova klasa može kreirati i izvještaj o korisniku i o zahtjevu.

BRIDGE

Bridge adapter možemo koristiti u slučaju da želimo dodati važnost zahtjeva (hitna i obična). Umjesto da kreiramo nove klase `HitanZahtjevZaKupovinu`, `ObičanZahtjevZaKupovinu`, `HitanZahtjevZaBanku` itd., kreirati ćemo podklasu `Važnost` iz koje su izvedene klase `Hitan` i `Običan` i njima ćemo staviti kao atribut klase `Zahtjev`, tj. klasa `Zahtjev` contains klasu `Važnost`.

FLYWEIGHT

Recimo da imamo klasu Okvir koja služi za čuvanje podataka o okviru profilne slike korisnika (cover photo). Ako bi za svakog korisnika kreirali novi okvir ovo bi zauzelo mnogo memorije. Dosta bolje rješenje bi bilo da prilikom dodavanja okvira imamo neki pool u kojem čuvamo sve do tad kreirane okvire. Ako okvir sa atributima koje je korisnik odabrao već postoji u poolu onda korisniku dodjeljujemo već postojeći okvir, ako takav okvir ne postoji onda kreiramo novi i dodajemo u pool.