

# DESIGN PATTERN

## 1. ADAPTER PATTERN

Ovaj design patern služi da predstavljamo neki tip podataka drugacije u odnosu kako je sacuvano. U nasoj aplikaciji adapter pattern bi se mogao iskoristiti pri implemetaciji interface za placanje, ukoliko firma koju primarno koristimo za izvorsavanje transakcija propadne, mozemo adaptirati tako da se koristi druga firma.

## 2. FACADE PATTERN

Facade patern se koristi kada sistem ima više identificiranih podsistema (subsystems) pri čemu su apstrakcije i implementacije podsistema usko povezane. U nasoj aplikaciji ovaj patern smo primjenili za editovanje terena od strane vlasnika terena i admina jer editovanje zapravo predstavlja usko povezane podsisteme npr. (editovanje naziva, editovanje lokacije, editovanje opisa, editovanje slika...) Svi ovi podsistemi se mogu zasebno raditi a usko su povezani jedni sa drugima

## 3. DECORATOR PATTERN

Osnovna namjena Decorator paterna je da omogućiti dinamičko dodavanje novih elemenata i ponašanja (funktionalnosti) postojećim objektima. U nasem projektu ovaj patern bi mogli iskoristiti pri kreiranju terena, dodavanje broja sala, lokacije, sportova koji se mogu održavati na terenu/sali.

## 4. BRIDGE PATTERN

Osnovna namjena Bridge patterna je da omogući odvajanje apstrakcije i implementacije neke klase tako da ta klasa može posjedovati više različitih apstrakcija i više različitih implementacija za pojedine apstrakcije. U našoj aplikaciji bridge pattern možemo koristiti pri računanju cijene za rezervaciju u ovisnosti od Terena i termina

## 5. PROXY PATTERN

Namjena Proxy patterna je da omogući pristup i kontrolu pristupa stvarnim objektima. Proxy je obično mali javni surogat objekat koji predstavlja kompleksni objekat čija aktivizacija se postiže na osnovu postavljenih pravila. U našoj aplikaciji proxy pattern smo implementirali za provjeravanje stanja transakcije, odnosno da pri rezervaciji korisnik ne može ručno postaviti stanje transakcije da bude gotovo već mora proći kroz proxy koji će provjeriti tu verifikaciju

## 6. COMPOSITE PATTERN

Osnovna namjena Composite patterna (kompozitni pattern) je da omogući formiranje strukture stabla pomoću klasa, u kojoj se individualni objekti (listovi stabla) i kompozicije individualnih objekata (korijeni stabla) jednako tretiraju. U našoj aplikaciji ovaj pattern smo mogli iskoristiti za dodavanje odnosno mijenjanje termina za određeni teren, gdje nam list predstavlja jedna instanca termina dok nam je korijen lista slobodnih termina za dati teren.

## 7. FLYWEIGHT PATTERN

Postoje situacije u kojima je potrebno da se omogući razlikovanje dijela klase koji je uvijek isti za sve određene objekte te klase (tzv. glavno stanje (engl.

intrinsic state)) od dijela klase koji nije uvijek isti za sve određene objekte te klase (tzv. sporedno stanje (engl. extrinsic state)). Osnovna namjena Flyweight paterna je upravo da se omogući da više različitih objekata dijele isto glavno stanje, a imaju različito sporedno stanje. U našoj aplikaciji ovaj patern se može iskoristiti za prikaz filtriranih terena, filter po kojem se sortira može biti po sportu ili po lokaciji.