### **Uvod**

S obzirom na dinamiku korisničkih akcija i potrebnu fleksibilnost, uvođenje paterna ponašanja (Behavioral Patterns) predstavlja optimalan pristup za povećanje skalabilnosti, održavanja i organizacije softverskog koda.

### **1. Strategy Pattern – za prilagodbu rangiranja i statistike**

**Mjesto u sistemu:** Rangiranje korisnika, dodjela postignuća  
 **Zašto:** Algoritmi za bodovanje, rangiranje i dodjelu priznanja se mogu mijenjati u zavisnosti od sporta, učestalosti, ponašanja korisnika itd.  
 **Korist:** Omogućava da lako dodajemo nove algoritme bez mijenjanja osnovne logike sistema.

### **2. Observer Pattern – za obavijesti i notifikacije**

**Mjesto u sistemu:** Slanje zahtjeva, prihvatanje/odbijanje, održavanje meča  
 **Zašto:** Više entiteta treba da bude obaviješteno o događaju (npr. korisnik šalje zahtjev → kreator oglasa prima notifikaciju).  
 **Korist:** Promjene se dešavaju automatski u zavisnim komponentama bez direktnog povezivanja objekata.

### **3. Command Pattern – za administraciju i undo radnje**

**Mjesto u sistemu:** Akcije administratora (brisanje korisnika, slanje opomena)  
 **Zašto:** Potrebno je logovati, izvoditi i po potrebi poništavati komande koje administrator izvrši.  
 **Korist:** Centralizovano rukovanje akcijama, podrška za undo/redo.

### **4. State Pattern – za praćenje stanja meča**

**Mjesto u sistemu:** Mečevi (zakazan, održan, otkazan, prijateljski)  
 **Zašto:** Meč se dinamički mijenja kroz više faza.  
 **Korist:** Svaka faza se može implementirati kao zasebno stanje s vlastitim ponašanjem bez komplikovanih if-else blokova.

### **5. Template Method – za validaciju i kreiranje korisničkih akcija**

**Mjesto u sistemu:** Registracija, prijava, postavljanje oglasa  
 **Zašto:** Mnoge akcije slijede sličan šablon: unos podataka → validacija → akcija → poruka.  
 **Korist:** Centralizovan šablon logike, lako prilagodljiv s redefinisanim koracima.

### **6. Mediator Pattern – za upravljanje interakcijom korisnik–meč–recenzija**

**Mjesto u sistemu:** Komunikacija između klasa: *Korisnik*, *Meč*, *Recenzija*, *Oglas* **Zašto:** Mnogo direktne zavisnosti između entiteta može učiniti sistem krhkim.  
 **Korist:** Decentralizuje zavisnosti i centralizuje komunikaciju u jednom objektu – mediatoru

Na taj način, Matchletic ne samo da ispunjava funkcionalne zahtjeve, nego se i priprema za dugoročno održavanje, razvoj i rast.