Kreacijski paterni

1. Singelton pattern
   Singleton pattern omogućava da se klasa instacira samo jednom i da osigura globalni pristup kreiranoj instanci klase. Moguće je da postoji više objekata koje je potrebno samo jednom instancirati i nad kojim je potrebna jedinstvena kontrola pristupa.

   U našem projektu, Singelton pattern možemo koristiti za servise za slanje notifikacija kako bi se uspostavila i djelila jedna veza kroz aplikaciju, također za praćenje broja klikova, prijava i pregleda oglasa može se koristiti Singleton koji centralizuje prikupljanje statistike.

2. Prototype pattern
   Prototype pattern pruža mogućnost kreiranja objekata koristeći neki postojeći prototip postojećeg objekta. Prednost ovog paterna je što se mogu kreirati prilagođeni objekti bez poznavanja detalja kako je objekat kreiran ili njihove klase.

   Klijent može objaviti novi oglas koji je skoro identičan prethodnom. Umjesto da popunjava sve ispočetka, aplikacija može klonirati postojeći oglas (Prototype), a korisnik izmijeni samo ono što je različito.

3. Factory method pattern
   Factory method Pattern pruža mogućnost kreiranja objekata tako da podklase odluče koju klasu instancirati.Patern instancira odgovarajuću izvedenu klasu,tj podklasu na osnovu tekućeg stanja ili od strane klijenta.

   Factory Method pattern možemo koristiti na mjestima gdje imamo potrebu za kreiranjem objekata na osnovu određenih kriterija. Na primjer, prilikom kreiranja oglasa, možemo imati različite tipove – standardni, hitni ili promovisani – i svaki od njih može imati specifično ponašanje. Umjesto da direktno instanciramo konkretne klase, koristimo Factory metodu koja na osnovu tipa oglasa vrati odgovarajuću instancu.

   Slično tome, kod slanja obavještenja korisnicima, Factory metoda može odlučiti da li se notifikacija šalje putem emaila ili SMS-a u zavisnosti od korisničkih postavki.

4. Abstract factory pattern
   Abstract Patern pruža mogućnost da se kreira familija povezanih objekata i na osnovu toga se kreiraju konkretne fabrike različitih kombinacija i tipova.

   Ako aplikacija podržava više tema (npr. svijetla i tamna), Abstract Factory može proizvoditi odgovarajuće UI komponente (dugmad, forme, menije) za svaku temu. Umjesto da klasa sama odlučuje koju komponentu koristi, cijeli „paket“ dolazi iz odgovarajuće konkretne fabrike.

5. Builder pattern
   Builder patern odvaja specifikacije kompleksnih objekata od njihove stvarne konstrukcije.

   Kreiranje oglasa gdje poslodavac može postaviti osnovne informacije (naslov, opis, lokaciju), ali i dodatne opcije poput roka prijave, budžeta, promocije, testnih pitanja i sl.

   Može se i koristiti za kreiranje složenih filtera za pretragu oglasa. Radnici mogu filtrirati oglase po više kriterija: kategoriji, lokaciji, budžetu, roku prijave, tipu posla (remote ili na lokaciji), itd. Umjesto da koristimo konstruktor s puno parametara, koristimo Builder koji omogućava postepeno dodavanje filtera.