

# Dizajn paterni ponašanja

## Strategy

Strategy patern izdvaja algoritme iz matične klase i uključuje ga u posebne klase. Pogodan je kad postoje različiti primjenjivi algoritmi (strategije) za neki problem. U našem sistemu postoji mogućnost pregleda historije poslanih zahtjeva, te možemo ovaj patern primijeniti u filtriranju zahtjeva. Korisnik ima mogućnost da izabere različite načine sortiranja poslanih zahtjeva, prvo najstariji, prvo najmlađi, prvo po alfabetu, i sl. Ovu su različite strategije/algoritmi koje korisnik može izabrati, bez da poznaje njihovu implementaciju.

## State

State patern podrazumijeva da objekat mijenja način ponašanja na osnovu trenutnog stanja. U našem sistemu se korisnik može nalazi u dva stanja: sa validnom ličnom dokumentacijom i bez. Ponašanje unutar sistema je drugačije zavisno od stanja. Ako mu je validna dokumentacija, ima pristup mogućnostima sistema, a ako nije, svaki pokušaj pristupa usluge sistema ga upućuje na to da mora obnoviti dokumentaciju. Primjena ovog paterna može biti pretjerana jer se korisnik većinom nalazi u prvom stanju, te ne dolazi do česte promjene stanja.

## Template Method

Template Method patern omogućava izdvajanje određenih koraka algoritma u odvojene podklase. Struktura algoritma se ne mijenja, mali dijelovi operacija se izdvajaju i mogu se različito implementirati. U našem sistemu bi mogli primijeniti template method sa formiranjem dokumenata, koji bi imao osnovnu strukturu u Dokument klasi, a njene podoperacije bi se obavljale na osnovu zahtjeva koji je poslan. Na taj način bismo mogli primjenjivati isti kod za generisanje dokumenata nezavisno od vrste zahtjeva koji ga traži.

## Observer

Uloga observer paterna je da uspostavi relaciju između objekata tako kada jedan objekat promijeni stanje drugi zainteresirani objekti se obavještavaju. U našem sistemu možemo koristiti ovaj patern prilikom obavještavanja korisnika da su mu lični dokumenti pred istekom validnosti (lična karta, pasoš, vozačka,...), pri čemu bi korisnici bili observeri a subject bi bio naš sistem.

## Iterator

Iterator patern omogućava pristup elementima kolekcije bez poznavanja kako je kolekcija struktuirana. U našem sistemu bi mogli koristiti ovaj patern prilikom prikazivanja historije zahtjeva korisniku. Korisnik ne poznaje kako se ovi zahtjevi interno čuvaju, već ga samo zanima da može pogledati detalje za bilo koji, i da prođe/pogleda sve njih.