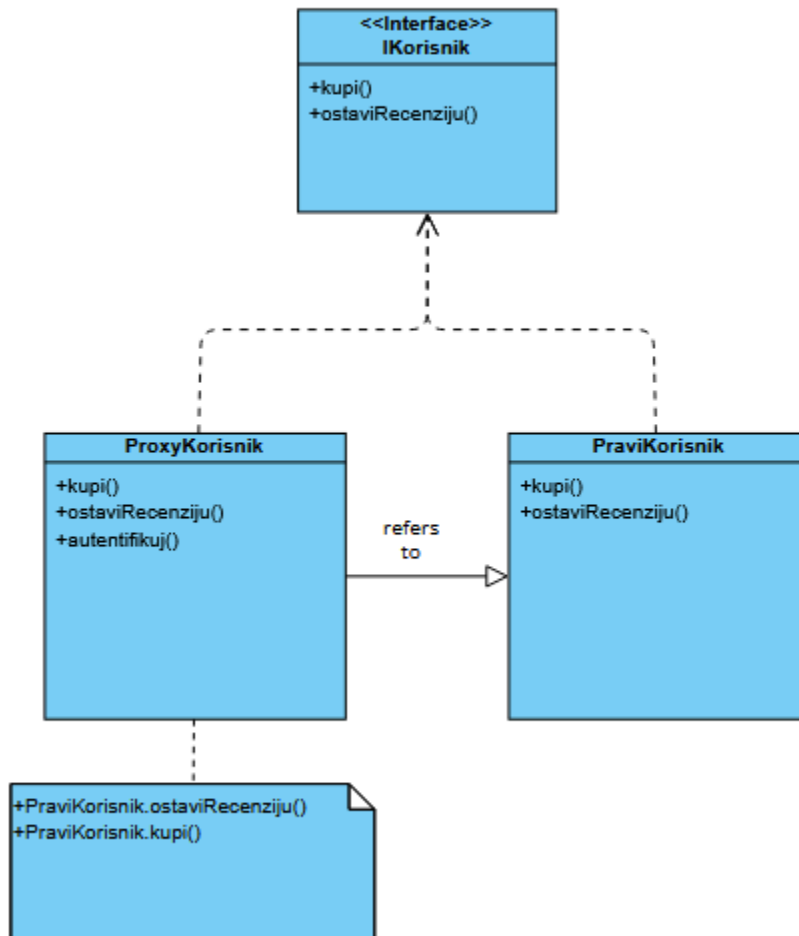


Proxy

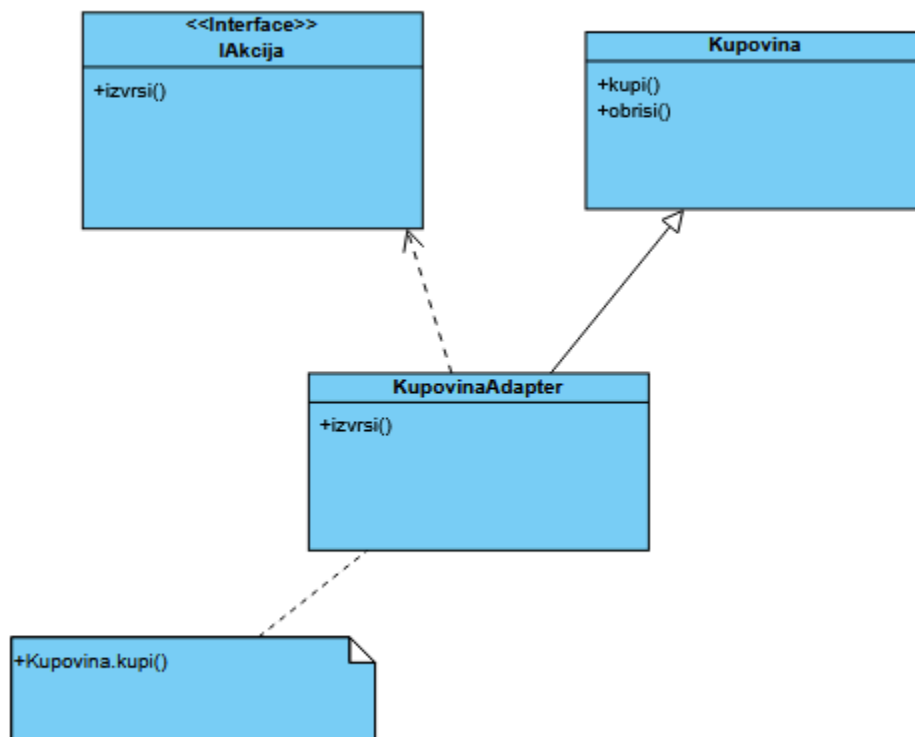
U postojeći sistem možemo dodati proxy strukturalni pattern, kako bismo zaštitili metode koje mogu koristiti samo prijavljeni korisnici (Protected Proxy) i centralizovali provjeru autentifikacije i autorizacije

Korisnik nakon login-a dobija pristup nekim operacijama — recenzijama, kupovinama, bookmarkovima itd.



Adapter

U postojeći sistem možemo dodati adapter strukturalni pattern, kako bismo napravili razliku između kupovine ulaznica za događaj u kojem potrebna i rezervacija mjesta i događaj na kojem se samo kupuje ulaznica.



Bridge

Bridge pattern omogućava odvajanje apstrakcije od njene implementacije tako da obje mogu nezavisno evoluirati. Na ovaj način se smanjuje međuzavisnost klasa i povećava fleksibilnost sistema.

U našem sistemu, Bridge pattern je primijenjen tako da se ponašanje različitih vrsta korisnika može odvojiti od osnovnih operacija nad događajima. Na primjer, klase koje predstavljaju različite vrste korisnika (npr. PremiumKorisnik, StandardniKorisnik) mogu koristiti zajednički interfejs za kupovinu ulaznica, recenziranje događaja i bookmarkovanje događaja, dok su konkretne implementacije ponašanja za ove operacije odvojene u posebne klase. Na ovaj način omogućava se proširivanje ponašanja korisnika bez izmjene osnovne strukture sistema.

Decorator

Decorator pattern omogućava dinamičko dodavanje novih funkcionalnosti objektima bez mijenjanja njihove osnovne strukture.

U sistemu, Decorator pattern može biti korišten za proširivanje funkcionalnosti događaja (Event) ili recenzija (Recenzija). Na primjer, možemo imati osnovnu klasu Recenzija kojoj možemo dinamički dodavati dodatne informacije kao što su "oznaka korisnosti recenzije" ili "broj lajkova", bez izmjene postojeće klase Recenzija. Na ovaj način proširujemo funkcionalnosti bez da remetimo osnovnu implementaciju.

Facade

Facade pattern pruža pojednostavljeni interfejs prema kompleksnom podsistemu, skrivajući njegovu unutrašnju kompleksnost od korisnika sistema.

U sistemu je Facade pattern primijenjen kreiranjem klase EventService koja nudi jednostavne metode za najčešće operacije poput registracije korisnika, kupovine ulaznica, pisanja recenzija i dodavanja događaja u bookmark. Ova klasa koristi interne komponente (Kupovina, Recenzija, Bookmark) i omogućava da se složene operacije izvrše kroz jednostavne metode, bez potrebe da korisnik zna detalje implementacije ili međusobne odnose među klasama.