

Refactoring aplikacije JAWS

Tokom implementiranja same aplikacije došli smo u mnogo situacija kada je bilo potrebno izvršiti značajne promjene na već postojećem kodu, kao razlog tome zasigurno je zaslužno naše neiskustvo, te određene manjkavosti u samom konceptu aplikacije, što opet direktno proizilazi iz našeg neiskustva.

Slijede neke promjene koje smo izvršili.

Dodana klasa CjenovnikItem.

U klasi Cjenovnik dodat atribut kolekcija CjenovnikItem, dok smo prije imali listu tuple-a string i double, koja bi nam otežala rad sa bazom.

Dodati Id-ovi u gotovo sve klase uz get/set metode. Dodati foreign key stereotipi za lakše kreiranje baze po code-first principu.

Promjenjen konstruktor u klasi Pacijent, za input u atribut Spol koji je tipa enum.

Tokom izrade frontend dijela aplikacije shvatili smo da nam je potreban niz novih view-ova koje nismo pokrili kada smo iste simulirali u mockup design-u, neki od novih view-ova su:

- TerminPacijent

Koji služi za svrhe zakazivanja termina iz perspektive pacijenta.

- TerminPacijentGreska

View koji pokriva funkcionalnost zakazivanja termina, u slučaju da pacijent zakaže termin koji je već zauzet, ovaj view prikazuje određenu ispisnu grešku uz button koji vodi na povratak na formu za zakazivanje termina.

- PacijentHome

View koji je usko vezan za samog pacijenta, što se može zaključiti iz imena samog View-a. Svrha navednog view-a je da odmah nakon login-a pacijenta prikaže sve moguće funkcionalnosti u vidu button-a koji poštuju predodređene role, to jeste, da npr. pacijent ne može kreirati nove kartone i pregledati već postojeće pacijente, što je funkcionalnost rezervisana isključivo za doktora ili stomatološkog tehničara.

- IndexPacijent

Svrha ovog view-a je da pacijent vidi sve zakazane termine koji su već zauzeti, uz onemogućen pristup editovanju i brisanju postojećih, jer su te funkcionalnosti rezervisane za doktora.

- CreatePacijent

Svrha ovog view-a je da pacijentu prikaže mogućnost zakazivanja termina