

- 1. Što se tiče novih funkcionalnosti što moramo ubaciti u zadaću, user name i password za uposlenike, da li ti podaci smiju biti predefinisani (kao hardkodirani), da svaki doktor odmah pri kreiranju ima te podatke, ili moramo omogućiti unos user name i password koji on zeli?**

Potrebno je username i password dodijeliti kao atribut klasama, tako da bi trebalo omogućiti i registraciju. Međutim, radi lakšeg testiranja, dobro bi bilo i hardkodirati (instancirati po jedan objekat) pristupne podatke za svaku od uloga. Drugi način implementacije bi bio da se kreira rola rolu administrator aplikacije koji će moći registrovati, ažurirati podatke i brisati korisnike aplikacije.

- 2. Da li je potrebno čuvati samo atribut MD5 za password za uposlenike i ako da, da li to znači da svaki put kada uposlenik želi da se prijavi, mora čitav hash kucati?
Ili možemo imati privatni atribut tipa string i javni atribut tipa MD5 pa kada uposlenik unese password koji je odabrao prilikom registracije(string), da se samo provjeri sa hashom(MD5) kada se prijavljuje?**

Kao atribut je potrebno čuvati hash šifre, a pri registraciji i loginu se unosi šifra (ne hash).

- 3. Zanima me da li je sada potrebno napraviti "Kreirati jednu funkcionalnost rješenja u sklopu odvojenog dll-a?**

Ne, DLL se neće opet bodovati. Potrebno je ispraviti greške koje su vezane za principe objektno orijentisanog programiranja kao što su ispisi u konzolu, public atributi klasa, korištenje polimorfizma na mjestima gdje je logično i tako dalje.

- 4. Da li mi možemo pojedine stavke koje smo pisali u zadaci 1(neke klase koje nisu iskorištene, interfejs) izbaciti?**

Možete, ako vam nisu potrebne ni u implementaciji zadaće 2.

- 5. Da li se treba pri pokretanju projekta prikazati forma na kojoj će korisnik odabrati da li pristupa kao pacijent ili kao uposlenik, ili se kao početna forma prikazuje registracija uposlenika? Ukoliko se prvo prikazuje forma RegistracijaUposlenika, odakle pacijenti pristupaju meniju?**

Inicijalna forma može biti login forma, a da se iz nje može odabrati opcije registracije korisnika ili obrnuto (kako je praksa inače u aplikacijama koje zahtijevaju login i registraciju).

6. Zašto je potrebno kreirati izuzetke u pozadinskoj logici ako pravilnim kreiranjem GUI validacije onemogućavamo bilo kakve greške pozadinske logike (koje bi pored toga i krahirale program)?

Šta ako prenesemo klasu u drugu aplikaciju ili dodamo novu funkcionalnost na GUI koja radi sa tom klasom? Dobra prakse je imati validaciju i na nivou klase i na nivou korisničkog interfejsa zato što time smanjujemo prostor za slučajne ili namjerne greške, a i olakšavamo samom korisniku rad u aplikaciji.

7. Da li klasu klinika smijemo proglasiti static klasom radi lakše manipulacije kroz čitavu aplikaciju?

Da, budući da je aplikacije namijenjena za konkretno jednu kliniku i nema smisla instancirati više njih, takav način implementacije ovdje ima smisla.