

# Projekat za niže ocene

## Osnovna ideja

Projektni zadatak je namenjen studentima koji žele da rade sažetiji projekat za ocene 6 i 7. Projekat se radi **individualno**.

**Obavezno je dodati asistentski nalog na repozitorijum gde se smešta izvorni kod bilo da je to GitLab ili GitHub (psw-ftn), kao i na Trello board (psw\_ftn).**

## Zahtevi

1. Veb aplikacija implementirana upotrebom .NET Core radnog okvira koja omogućuje pacijentima registraciju i rad sa sistemom bolnice
2. Integracija sa eksternim servisima - sistemom apoteka radi poboljšanja funkcionisanja bolnice
3. Implementirano rešenje treba da poštuje *Clean code* principe
4. Potrebno je implementirati rešenje po principu *Scrum* metodologije, koja se izučava na predmetu.
5. Kako se kreira realan sistem, potrebno je podatke trajno perzistirati u relacionoj bazi podataka (npr. *MySQL*)
6. Prilikom implementacije rešenja poštovati *TDD* princip koji se obrađuje na predmetu

Ukoliko bilo koja od prethodnih stavki nije zadovoljena smatra se da student nije položio projekat, gde se podrazumeva da će student za obe ocene implementirati sve zahteve date u nastavku specifikacije projekta.

## Specifikacija

### Veb aplikacija

Ova aplikacija je dostupna pacijentu i koristeći istu pacijent ima dostupne funkcionalnosti date u nastavku. Pored pacijenata, ova aplikacija je dostupna i administratorima radi održavanja iste.

### Registracija

Novi korisnici usluga bolnice mogu da se registruju nakon čega postaju poznati sistemu kao novi pacijenti. Nakon registracije, pacijenti mogu da se prijave i koriste sve dostupne funkcionalnosti.

Administratori se ne registruju preko aplikacije, već se zadaju preko baze podataka.


### Prijava na sistem

Omogućiti korisnicima sistema prijavu na sistem i dalji rad unutar sistema. Autentifikovati korisnika prilikom prijave i definisati permisije unutar sistema.

### Zakazivanje pregleda

Zakazivanje pregleda se odvija preko preporuke termina od strane sistema. U sklopu preporuke termina korisnik treba da izabere lekara kod kojeg se zakazuje pregled, vremenski period unutar kojeg želi da zakaže pregled i odredi koji od ova dva načina je prioritetniji. Nakon što korisnik unese podatke sistem treba da nađe neki termin i da ga preporuči korisniku. U idealnom slučaju će se odmah naći termin unutar zadatog intervala za zadatog lekara, u kom slučaju će se samo preporučiti jedan od pronađenih slobodnih termina. Kada se ne pronađe termin za zadatog lekara u zadanom vremenu, primenjuje jedna od dve strategija:

1. ako je pacijent postavio prioritet na lekaru, onda se lekar ne sme menjati i sistem traži termine i van inicijalnog intervala (+/- 7 dana).
2. ako je prioritetniji vremenski interval onda se zadati vremenski interval ne menja, ali sistem dalje proverava rasporedi ostalih lekara sve dok se ne nađe lekar koji pripada istoj specijalističkoj grani koji je slobodan za zadato vreme.



U oba slučaja bi rezultat preporuke bio neki slobodan termin koji sistem preporučuje pacijentu i pacijent ima izbor da prihvati ili odbije taj termin. U slučaju da sistem ne pronađe nikakav termin, korisnik biva obavešten o tome.

Zakazani pregled može biti otkazan najkasnije 48h pre definisanog termina. Nakon otkazivanja, neki drugi pacijent može da zakaže sebi pregled u oslobođenom terminu.

Ako pacijent prvi put dolazi na pregled, on može da zakaže pregled isključivo kod svog izabranog lekara opšte prakse. Ako pacijentu izabrani lekar da uput za odlazak kod nekog specijaliste koji pripada određenoj specijalističkoj grani, tek tada pacijent može da zakaže pregled kod lekara koji pripada toj grani. Uput se može iskoristiti samo jednom i na osnovu jednog uputa pacijent ne može ići više puta kod specijaliste.

## Istorija poseta bolnici

Pacijent ima uvid u prethodne preglede koje je ranije obavio i buduće preglede koje je zakazao.

## Davanje mišljenja (feedback)

Pacijent ima mogućnost da napiše mišljenje (feedback). Administrator sistema može da čita objavljena mišljenja i podeli neka od njih tako da svi posetioци veb stranice mogu da ih pročitaju. Objavljena mišljenja prikazuju se na početnoj stranici veb aplikacije.


## Administracija sistema

Administrator može da blokira pacijente koje sistem identifikuje kao zlonamerne. Pacijent je zlonameran ako u toku mesec dana otkaže bar 3 pregleda. Ako se administrator odluči da blokira pacijenta, sistem o tome obaveštava pacijenta i ne dozvoljava njegovu prijavu na sistem.

Administrator ima uvid u sva mišljenja koja pacijenti ostavljaju o radu bolnice i zaposlenima i ima mogućnost da neka od tih mišljenja objavi na glavnoj stranici aplikacije. Takođe, ima mogućnost da neko od objavljenih mišljenja skloni sa glavne stranice.

## Integracija sa eksternim servisima

Ukoliko student ne polaže ISA projekat potrebno je da implementira dodatnu aplikaciju koristeći npr. *Spring Boot* radni okvir kako bi podržali ponašanje aplikacije koja omogućuje integraciju sistema bolnice sa eksternim sistemima. **Spring Boot aplikacija u tom slučaju**



ne treba da implementira potpuno ponašanje koje se zahteva na ISA predmetu, već samo one delove koji će omogućiti adekvatno ponašanje dela sistema traženom na PSW predmetu u okviru aplikacije za integraciju.

## Rad sa receptima

Radi olakšavanja procedure izdavanja recepata, za pacijente bi bilo korisno ukoliko bi se recept automatski po propisivanju slao u apoteku koja sarađuje sa bolnicom i poseduje lekove propisane terapijom. Jednom kada lekar kreira recept za pacijenta smešta ga na deljenu lokaciju odakle apotekar može da ga preuzme i izda pacijentu traženu terapiju. *Ukoliko ne postoji ISA deo rešenja, potrebno sa Spring strane omogućiti pregled podeljenog recepta sa deljene lokacije.*

## Nabavka lekova

Nabavka lekova za bolnicu je jedna od bitnijih stavki u saradnji između bolnice i apoteke. Ukoliko nam je potrebna hitna nabavka lekova iz apoteke, koja ne bi išla redovnim principom nabavke (a koji je opisan u nastavku specifikacije) potrebno je ostvariti komunikaciju sa apotekom koja poseduje potrebni lek i izvršiti nabavku istog. *Implementirati nabavku preko gRPC protokola.*