

| | |
|--|---|
| PREDMET: Razvoj Programskih Rješenja | |
| AK. GOD. : 2017/2018 RESURS: Zadaća 2 DATUM OBJAVE: 08.12.2017. ROK ZA PREDAJU: 23.12.2017 do 23:59:59. Ukoliko je potrebno pomjeriti rok za predaju predstavnik godine treba da javi prijedlog novog roka do 12.12.2017. BODOVI: 8 bodova | STUDENTI samostalno rade zadaću. ! Pitanja oko zadaće (šta treba uraditi ne kako) slati na e-mail khodzic2@etf.unsa.ba do 15.12.2017. Nakon toga će se pitanja i odgovori objaviti na c2. !Prepisivanje će biti kažnjeno-0 bodova i koje prepisao i od koga je prepisao, zavođenje na listu prepisivača. |

ZADAĆA 2

Postavka: Na osnovu strukture klasa i pozadinske logike koja je urađena u sklopu zadaće 1, za rješenja koja pomažu funkcionisanju fakulteta potrebno je implementirati stavke koje su nabrojane ispod.

1. Studenti koji nisu implementirali zadaću 1 mogu to sada uraditi. Neće se bodovati zadaća 1, ali omogućava studentima da nauče propušteno i da rade ostale zadaće. Studenti prije početka rada zadaće 2 trebaju uraditi:
Prilagođavanje pozadinske logike – Prvi zadatak jeste *prilagođavanje* pozadinske logike koju ste kreirali u sklopu prve zadaće. Ova stavka je proizašla iz razloga što je dosta studenata miješalo elemente konzolnih ulaza i izlaza unutar samih metoda klasa što se sada javlja kao jedan veliki problem koji zahtjeva restrukturiranje koda. To se odnosi na poštovanje OOP principa, na izbacivanje elemenata konzolnih ulaza i izlaza iz metoda, ispravno struktuiranje pozadinske logike i ispravljanje zamjerki na koje su vam oduzimani bodovi na prvoj zadaći. Dodatna konsultacija sa asistentima je preporučena. ***Ukoliko restrukturiranje traži puno promjena koda studenti mogu ponovo uraditi zadaću 1. Restruktuiran kod je preduslov za zadaću 2.***
2. ***Proširenje rješenja*** – Nakon što je pozadinska logika restrukturirana potrebno je implementirati dodatne funkcionalnosti. Prvo je potrebno proširite pretrage doktora, pacijenata i kartona. Omogućiti da rješenje funkcioniše na osnovu uloga. To znači da je potrebno Uposlenim dodijeliti pristupne podatke tj. korisnička imena i lozinke kao dodatne attribute. Lozinka se treba spremati zakodirano kao *MD5 hash*. Na osnovu uloga se odlučuje o tome šta koja uloga ima od privilegija tj. koje operacije su im dostupne. Običnom osoblju klinike je potrebno omogućiti kreiranje i popunjavanje kartona za nove pacijente, dok je doktorima potrebno omogućiti evidentiranje pregleda i upisivanje terapija u karton. Za obje uloge je potrebno omogućiti pregled trenutnih redova čekanja po ordinacijama u klinici. U ovom dijelu rješenja je potrebno razviti i dodatni modul koji će pristup aplikaciji dozvoliti i pacijentima. Pacijenti trebaju imati pristup informacijama o svim dostupnim informacijama vezanim za njih, trenutnoj zauzetosti ordinacija, pregled vlastitog kartona i sl. Također je potrebno proširiti da pacijenti imaju sliku (atribut tipa *Image*) kao dodatni atribut. Analitičke metode koje

su bile implementirane u zadaći 1 treba proširiti tako da doktorima daju dobru sliku o stanju njihovih pacijenata i historijatu njihovog zdravstvenog stanja.

[2 boda]

3. **Razvoj grafičkog interfejsa (GUI)** – Na osnovu uočenih funkcionalnosti iz teksta *Zadaće 1* i prethodne stavke potrebno je za iste kreirati grafički interfejs. To podrazumijeva korištenje odgovarajućih kontrola za svaki atribut koji se unosi. Pod uočenim funkcionalnostima se podrazumijeva: unos, ažuriranje, pregled i brisanje svih uposlenih na klinici i pacijenata, sve spomenute vrste pretraga, pregled i mogućnost kreiranja i modificiranja kartona svim ovlaštenim osobama. Prilikom dizajniranja interfejsa pored osnovnih dobro odabranih kontrola obavezno treba na bar jednom mjestu imati smisljeno iskorištene i implementirane sljedeće kontrole: **MenuStrip**, **StatusStrip**, **TabControl**, **ContextMenu** i **TreeView**. Analitičke metode koje su implementirane trebaju biti dostupne i na grafičkom interfejsu. Pri kreiranju korisničkog interfejsa koristiti preporuke za dobar dizajn date na predavanjima.

[2 boda]

4. **Validacija i upravljanje greškama na svim nivoima** – Kao što je to navedeno u tekstu vježbi i predavanjima u kojima je rađena validacija, potrebno je u potpunosti osigurati validaciju na svim nivoima. To se odnosi i na rukovanje izuzecima na strani pozadinske logike, kao i korištenje **errorProvider** kontrola na strani korisničkog interfejsa. Poruke o greškama se ne smiju prenositi preko **MessageBox** klase, već se treba implementirati **statusStrip** sa prikazom i skrivanjem odgovarajućih poruka. Takođe je potrebno prilikom registracije pacijenata i uposlenika osigurati *cross-validaciju* pri unosu i potvrdi korisničke šifre. Dobro obratiti pažnju koje kontrole se trebaju, a koje se ne trebaju provjeravati. Prazna polja, pogrešni tipovi podatka se trebaju smatrati nevalidnim unosom. Voditi računa šta se provjerava mehanizmom validacije, a šta mehanizmom upravljanja greškama (exception handling).

[2 boda]

5. **Korisnička kontrola** – Odabir slike i datuma kada je slika uslikana (**PictureBox**, **Button** i **DateTimePicker** kontrole) zapakovati u jednu korisničku kontrolu koja će se koristiti na svakom mjestu gdje je potrebno unijeti sliku za pacijente. Korisnička kontrola treba biti u mogućnosti da vrati sve informacije koje su potrebne i koje se unose preko nje, te treba validirati datum slike, jer slika koja se unosi ne smije biti starija od 6 mjeseci.

[1 bod]

6. **Grafika** – Na početnoj stranici vašeg rješenja gdje će se nalaziti mjesta za unos korisničkog imena i korisničke šifre treba biti kreiran logo klinike. Taj logo treba da bude kreiran korištenjem bar 3 različita objekta GDI+ biblioteke. Također je za analitičku metodu po želji potrebno nacrtati jedan graf (*pie*, *bar*, *line* itd.) korištenjem GDI+ metoda koji će predstavljati vizuelni prikaz rezultata analize.

[1 bod]