



- ZAVRŠNI PROJEKAT -

OBJEKTNO-ORIJENTISANO PROGRAMIRANJE

DOKUMENTACIJA -BOLNIČKI SISTEM-

Studenti: **Delaida Muminović 334, Nedim Spahić 341, Amir Abaz 386, Nejra Smajlović 328**

Profesor: **Adnan Dželihodžić**

Asistenti: **Edin Tabak i Narcisa Hadžajlić**

Akadska godina: 2023 - 2024

Osnovni opis projekta:

Tema našega projekta je Bolnički sistem. Izbor ove teme za projekat iz Objektno-orijentisanog programiranja proizilazi iz stvarnih izazova sa kojima se susreću uposlenici zdravstvenih ustanova. Razgovori sa našim prijateljima i porodicom koji su zaposleni u takvoj vrsti ustanovama potakli su nas da razmislimo o postojećim nedostacima u trenutnom informacionom sistemu, te smo odlučili da kroz ovaj projekat ponudimo skromnu, ali ne i manje vrijednu viziju inovacija u bolničkim sistemima, s fokusom na poboljšanje efikasnosti, smanjenje administrativnih poteškoća i unapređenje kvaliteta zdravstvenih usluga. Projekat je koncipiran na način da je prilagođen medicinskom osoblju, konkretno doktorima. Moguće je unositi nove informacije o doktorima, tj. njihove lične podatke, specijalizaciju, broj ordinacije, naknadu po pacijentu i sl. Pored toga, moguće je pregledati i popis svih doktora u sistemu uključujući i informacije o njima. Omogućen je i unos pacijenata zajedno sa njihovim dijagnozama, vremenom prijema, otpusta, ličnim informacijama, cijenom pregleda, te brojem sobe u kojoj borave za vrijeme liječenja. Istovremeno, opcije koje se nude su pregled svih pacijenata, traženje pacijenta u sistemu na osnovu imena i adrese, pregled slobodnih i zauzetih soba, kao i mogućnost otpuštanja pacijenata i samim tim njihovog uklanjanja iz sistema. S obzirom na to da u bolnici postoji potreba za ljudima raznih profesija, postoji opcija za dodavanje osoblja i navođenje njihove pozicije koju obavljaju u bolnici. Kako bismo doktorima olakšali donošenje odluke koju terapiju propisati, omogućili smo im da prilikom unosa određene dijagnoze sistem sam nudi terapiju koja je u skladu sa bolešću njihovih pacijenata. Na taj način doktori će efikasnije propisati lijekove svojim pacijentima i omogućiti im uspješan oporavak. Da bi boravak u bolnici i uslovi rada bili što bolji, pored mjesečnog budžeta za nabavku lijekova unutar apoteke koja je dostupna u sistemu, doktori taj novac mogu iskoristiti da sebi priušte hranu i osvježanje u toku radnog dana. Kako bi bolnica bila sigurna da su doktori zadovoljni uslovima u kojim rade, doktori su dužni na mjesečnoj bazi ocijeniti svoje zadovoljstvo i ostaviti željeni komentar. Obzirom da su sve ocjene i komentari spremljeni u datotekama, posebno za svakog doktora, na kraju godine moguće je znati da li je potrebno raditi na poboljšanju radnog okruženja ili su ipak doktori koji su zaposleni zadovoljni načinom na koji se tretiraju.

Projekat je zasnovan na osnovnim principima objektno-orijentisanog programiranja u C++ te sadrži klase, enumeracije, virtuelne funkcije, pametne pokazivače, iznimke, preopterećene operatore, nasljeđivanje, ali i svo gradivo rađeno na prethodnim predmetima vezanim za programiranje. Također, korišteni su konstruktori koji su odigrali izuzetno značajnu ulogu kada je u pitanju implementacija ovog projekta

Funkcionalnosti

Sve klase su implemetirane u posebnim cpp fajlovima. Klasa koja se poziva prilikom pokretanja programa jeste **Account** sa atributima username, password, role, u kojoj su implementirane metode za kreiranje računa, te čuvanje podataka pomoću datoteka. U implementaciji njenih metoda korištene su iznimke radi definisanja grešaka, te njihovog eventualnog ispravljanja prilikom registracije / prijavljivanja u glavni sistem. U glavnom programu, prva je implementirana, omogućava login ili registraciju, te nakon uspješne prijave imamo pregled glavnog menija čiji nam pregled omogućava klasa **Hospital**. Ključni atributi ove klase uključuju naziv bolnice ('name'), adresu ('address'), telefonski broj ('phone_num'), izbor korisnika ('choose'), apoteku ('pharmacy'), sobu ('room'), informacije o trenutnom korisniku ('person'), i vektor soba ('rooms'). Metoda 'setRooms()' koristi pametne pokazivače za dinamičku alokaciju objekata tipa 'Room' i inicijalizaciju vektora 'rooms' kopiranjem jedinstvenih kopija.

Klasa '**Room**' poseduje sledeće attribute: 'int **room_arr': Dvodimenzionalni niz cijelih brojeva koji predstavlja raspored soba, 'const int total_rooms': konstanta koja označava ukupan broj soba u bolnici, 'int free_r': broj slobodnih soba, 'string room_type' - iskorištena prilikom kreiranja operatora unosa i ispisa za ovu klasu, da bi omogućila ispis tipa sobe (da li je slobodna ili zauzeta). Za to je iskorištena dvodimenzionalna matrica koja vrijednosti broja sobe mijenja u 0 ukoliko je unesen novi pacijent, tj. ukoliko je soba zauzeta.

Klasa '**Patient**' predstavlja entitet pacijenta u bolnici sa atributima poput opisa zdravstvenog problema, vremena prijema, vremena otpusta, naplaćene naknade te informacija o dodijeljenoj sobi, povezanoj sobi i osnovnim ličnim podacima pacijenta, a koristi se za praćenje, pretragu i ažuriranje informacija o pacijentima. Klasa 'Patient' je izvedena iz klase '**Person**', čime nasljeđuje osnovne attribute i funkcionalnosti koje se odnose na lične podatke osobe, uključujući ime, adresu, godine, spol, kontakt i informacije o kontaktu. Nasljeđivanjem od klase 'Person', klasa 'Patient' stvara hijerarhiju koja omogućava zajedničko upravljanje osnovnim ličnim podacima za sve entitete u sistemu bolnice.

Klasa '**Doctor**' je također izvedena iz apstraktne klase 'Person', stvarajući sličnu hijerarhiju kao i klasa 'Patient'. Ova hijerarhija omogućava zajedničko upravljanje osnovnim ličnim podacima između različitih entiteta u bolnici, kao što su pacijenti i doktori. Klasa 'Doctor' dodatno sadrži attribute specifične za doktora, poput radnog vremena, broja ordinacije i naknade, a virtualna funkcija 'display' omogućava prilagođeno prikazivanje informacija o doktoru.

Klasa 'Doctor' je osnovna klasa koja je povezana s tri specifična tipa doktora: '**Gynecologist**', '**Child_Specialist**', i '**Surgeon**'. Svaki od ovih specifičnih tipova doktora nasljeđuje od klase 'Doctor' i proširuje je dodatnim karakteristikama. Na primjer, 'Gynecologist' sadrži atribut koji označava oblast specijalizacije (specijalizacija za ginekologiju), 'Child_Specialist' ima atribut za tip pedijatra, dok 'Surgeon' ima atribut za tip hirurga.

Sve ove klase dijele zajedničke karakteristike kao što su radno vrijeme, ordinacija, naknada, lični podaci (ime, adresa, dob, spol, broj telefona), te se koriste zajedničkom apstraktnom klasom `Person` koja sadrži te osnovne informacije. Ovim nasljeđivanjem i proširenjem hijerarhije, omogućava se organizacija i zajedničko upravljanje podacima o različitim tipovima doktora u okviru bolničkog sistema.

Klase `**Staff**`, `**Worker**`, i `**Receptionist**` čine hijerarhiju osoblja u bolničkom sistemu. Osnovna klasa `Staff` predstavlja radnike uopšteno, a od nje su izvedene klase `Worker` i `Receptionist`.

Klasa `Worker` predstavlja radnika sa dodatnim atributom koji označava tip uposlenika. Povezana je sa klasom `Staff` nasljeđivanjem, proširujući funkcionalnosti osnovne klase. Ova klasa također omogućava ažuriranje informacija o radnicima u datoteci "Staff.txt".

Klasa `Receptionist` također nasljeđuje od klase `Staff` i dodaje svoj specifičan atribut za označavanje obrazovanja. Ova klasa omogućava prikazivanje podataka o radnicima na recepciji, kao i ažuriranje datoteke "Staff.txt" s informacijama o portirima. Sve tri klase koriste zajedničku apstraktnu klasu `Person`.

Sljedeće klase čine dodatne funkcionalnosti u bolničkom sistemu, uključujući upravljanje nabavkom i korišćenjem grickalica (`**Snacks**`), lijekova (`**Medicine**`) te apotekom (`**Pharmacy**`), za čiju su implementaciju također iskorištene iznimke.

Klasa `Medicine` predstavlja lijekove dostupne u sistemu apoteke u bolnici. Pruža funkcionalnosti za kupovinu lijekova, provjeru dostupnosti i upravljanje budžetom.

Klasa `Snacks` modelira grickalice koje se mogu kupiti u bolničkom okruženju. Obraduje operacije kao što su odabir i kupovina grickalica, provjera količina i upravljanje budžetom bolnice.

Klasa `Pharmacy` predstavlja cjelokupni sistem apoteke unutar bolnice. Enkapsulira instance klase `Medicine` i `Snacks` te pruža mogućnost korisnicima za odabir između kupovine lijekova, grickalica ili izlaska iz apoteke. Omogućava interakciju i upravljanje ukupnim tokom sistema apoteke.

Klasa `**Feedback**` omogućava pacijentima da daju ocjene i komentare o određenom doktoru. Pruža funkcionalnosti za unos ocjena, komentara, prikazivanje povratnih informacija i pohranu u datoteke.

Klasa `**Medic**` predstavlja medicinsko osoblje i pridružene dijagnoze koje su im povezane. Omogućava preporuke lijekova za određene dijagnoze i podržava povezivanje medicinskog osoblja s dijagnozama.

Dodatno

Pored ispunjenih zahtjeva projekta, neizostavan dio čine kreativne ideje koje smo implementirali u svoj kod. Prije svega, na samom početku moguće je izvršiti registraciju i prijavu u sistem, te ukoliko je korisnik zaboravio svoju lozinku, u datoteku Reset_Password šalje se kod za promjenu lozinke te nakon njegovog unosa u sistem moguće je kreirati novu lozinku za račun korisnika.

```
Unesite kod iz Reset_Password file-a
Code: 71162

Kod se poklapa!
Promijenite lozinku

Unesite novu lozinku
****
Potvrdite lozinku
****
█
```

Pored navedenog, ideja je bila omogućiti osoblju predah uz grickalice i osvježenje zbog čega smo za te svrhe čak stavili na raspolaganje i bolnički budžet. Iz datoteke čita se spisak proizvoda koje je moguće kupiti zajedno sa dostupnom količinom i cijenom, te se nakon kupovine, obračunava novi iznos budžeta. Tu je i apoteka za koju je također predviđen bolnički budžet i pomoću koje je moguće kupiti razne vrste lijekova.

```
*****
          SNACKS          Kolicina      Cijena
*****
          Cips            10             3
          Voda            10             2
          Cola            10             3
          Keks            10             4
          Caj             10             2
          Sok             10             3
*****

Trenutni budzet bolnice: 10000
```

Lijek	Bolest	Kolicina	Cijena
Promethazine	Alergija	25	60
Methadone	Bolovi	19	75
Acefyl	Kasalj	23	45
Midazolam	Anksioznost	12	112
Trazodone	Depresija	14	152
Phenytoin	Epilepsija	7	250
Actos	Diabetes	33	220
Disprin	Glavobolja	50	20
Flagyl	Bolovi	45	70

Dostupan budzet 9992

Još jedna od prednosti našeg projekta je to što smo unutar main programa omogućili funkciju koja na osnovu unesene dijagnoze ispisuje sve lijekove koji se koriste za njeno liječenje. Na taj način doktorima smo dali veći izbor i na taj način ih potakli da preciznije biraju lijekove za svoje pacijente što rezultuje uspješnijim liječenjem i oporavkom, te većim ugledom bolnice. Podaci o lijekovima i dijagnozama smješteni su unutar datoteke.

```
Unesi dijagnozu: Alergija
- Loratadin
- Cetirizin
- Cetirizin
- Fexofenadin
- Montelukast
- Diphenhydramine
```

Da bi se na neki način dobila povratna informacija doktora kada je u pitanju zadovoljstvo uslovima rada, implementirana je opcija za ostavljanje ocjena i komentara koju su doktori dužni popunjavati svaki mjesec kako bi se vodila detaljna evidencija. Ta mogućnost je implementirana tako da prilikom unosa svoga imena, kreira se datoteka (ime.txt) koja je jedinstvena za ime doktora, te se sve njegove ocjene i komentari upisuju isključivo u tu istu datoteku. Za tu svrhu korištene su enumeracije te je moguće dati ocjenu od 1-5 od kojih svaka nosi određeni stepen zadovoljstva/nezadovoljstva doktora kao uposlenika bolnice.

```
Ocjene i komentari za doktora Nejra:
Ocjena: ODLICNO, Komentar: Uslovi rada su perfektni, kolege su divni ljudi!
Ocjena: PROSJEKNO, Komentar: Naglo smanjenje budzeta, nadam se poboljsanju
Ocjena: ODLICNO, Komentar: Uslovi rada su ponovo idealni:)
Ocjena: ODLICNO, Komentar: Uslovi su idealni, kolege su ljubazne!
```

GitHub repozitorij

https://github.com/delaidam/OOP_PROJEKAT