



*Dekompozicija programskog rješenja*

## Analiza zahtjeva

Iz priloženog intervjua i bilješki sa posmatranja može se zaključiti da je u klinici potrebno povezati one koji pružaju usluge sa onima koji te usluge koriste. Dakle, vrši se evidencija uposlenika (od kojih je akceat na doktorima), pacijenata i njihovih međuodnosa.

Prije definisanja tih međuodnosa, potrebno je prvo sagledati sastavne dijelove klinike. Kliniku čine prije svega njeni uposlenici, zatim usluge koje pruža, koje se odvijaju kroz ordinacije, te aparati koji omogućavaju izvršavanje određenih usluga.

Pacijenti i doktori su povezani pregledima koje doktori obavljaju nad pacijentima, a pregledi se opet sastoje od nekoliko sekvenci od kojih su istaknute anamneza(koja se obavlja prije samog dolaska u ordinaciju) te propisana terapija nakon obavljenog pregleda. Kako bi klinika imala historiju poteškoća (bolesti) s kojima se susretao jedan pacijent, svakom pacijentu se pridružuje i njegov karton, koji se popunjava održanim pregledima.

Cilj programskog rješenja jeste da omogući bržu i precizniju komunikaciju uposlenika u klinici, veću i bolju preglednost dostupnih podataka te upravljanje resursima u stvarnom vremenu što kao krajnji rezultat treba omogućiti klinici da pruža bolje i kvalitetnije usluge svojim korisnicima.

**1)**

U priloženom programskom rješenju primjenjeni su objektno orijentisani principi, pri čemu su enkapsulacija i sakrivanje podataka primjenjeni tako što su svi atributi klase definisani kao privatni, i jedini način da im korisnik klase pristupi je preko odgovarajućih property-a. Nasljeđivanje je prezentovano kroz klase Osoba, koja je ujedno i apstraktna klasa jer ne želimo instancirati objekte tipa Osoba i služi nam samo kao bazna klasa za klase Uposlenik, Doktor, Pacijent i SmrtniSlučaj, te nam omogućava i eventualno dodavanje novih lica sa različitim specifikacijama (recimo VanjskiSaradnik). Uposlenik, Doktor, Pacijent i SmrtniSlučaj svi predstavljaju osobe koje moraju imati određene lične podatke. Tako su klase Uposlenik i i Pacijent izvedene iz klase Osoba, a kao dodatni atributi dodane su im određene specifikacije koje ih čine pacijentom (npr. Karton), odnosno uposlenikom (npr. Pozicija na kojoj radi i plata). Dalje, klasa SmrtniSlučaj je izvedena iz klase Pacijent, s obzirom da pacijent koji je preminuo je također bio korisnik usluga klinike, ali kao dodatne attribute ima informacije o samoj smrti. Također Doktor je klasa izvedena iz klase Uposlenik, s tim da ima dodatne specifikacije koje ga čine doktorom, a to su ordinacija u kojoj radi ili broj pacijenata koje je pregledao. Ostale uposlenike nije bilo potrebe posebno izvoditi kao klase, s obzirom da se oni u zahtjevima nigdje ne pojavljuju tako da bi bilo potrebno konkretno nešto referencirati na njih. Gore navedene klase nalaze se u fajlovima Osoba.cs, Uposlenik.cs, Doktor.cs, SmrtniSlučaj.cs, Pacijent\_1.cs, Pacijent\_2.cs.

Polimorfizam je ostvaren u klasi Klinika (fajl Klinika.cs) tako što je kao njen atribut definisana lista Pacijenata, u koju se smještaju i redovni i novi pacijenti, ali i pacijenti koji su preminuli.

Klasa Pacijent implementirana je u dva različita fajla kao partial klasa (Pacijent\_1.cs, Pacijent\_2.cs). U prvom fajlu su navedeni atributi klase te odgovarajući property's, dok su u drugom fajlu implementirane metode za upravljanje klasom, odnosno prikaz određenih podatak u određenom formatu. Ovo doprinosi lakšem snalaženju u slučaju dodavanja novih atributa i property-a, odnosno novih metoda za upravljanje klasom.

Klasa Aparat (Aparat.cs) je označena sa sealed.

**2)**

Interfejs je implementiran u fajlu Osoba.cs i sadrži jednu metodu koja služi ispisu jednog objekta tipa klase koja implementira interfejs.

**3)**

Delegat i lambda iskazi iskorišteni su u fajlu GlavniProgram.cs, a poslužili su prilikom optimizacije rasporeda pregleda. Kako je pacijentu potrebno kreirati raspored pregleda takav da se upućuje prvo u one ordinacije gdje trenutno najmanje pacijenata čeka, to je pokazatelj da je upravu tu listu pregleda koju imamo potrebno sortirati. Međutim kako ugrađena Sort funkcija za List-e nema definisan kriteriji poređenja za objekte klase koje sami kreiramo, potrebno je funkciji sort proslijediti adekvatnu funkciju kriterija. U ovom slučaju kriterij sortiranja pregleda će biti takav da pregledi čije ordinacije imaju manji broj pacijenata na čekanju, dođu nakon onih koji imaju veći, što u konačnici dovodi do optimalnog rasporeda pregleda.

**4)**

U klasi Klinika u fajlu Klinika.cs sastavni dijelovi jedne klinike su prikazani unutar kolekcija podataka. Tako se klinika sastoji od kolekcije ordinacija, kolekcije aparata, kolekcije doktora, kolekcije ostalih uposlenika, kolekcije evidentiranih pacijenata i kolekcije pregleda (usluga) koje klinika nudi. Također, prilikom rješavanja određenih problema korišteni su i Dictionary koji su služili najčešće za prikaz trenutnog stanja ordinacija i pregleda.

**5)**

Klasa Anamneza je kreirana kao ClassLibrary, i njena implementacija se ne nalazi direktno u projektu, već je dodana referenca na odgovarajući dll fajl, nakon čega je klasa Anamneza postala dostupna koristeći direktivu AnamnezaLibrary.

**6)**

Pravila programske dekompozicije ispoštovana su tako što se implementacija svake klase odvojila u zaseban cs fajl, dok je GlavniProgram.cs fajl u kojem je kreiran glavni izbornik. Dekompozicija je omogućila jednostavnije i brže kretanje kroz projekat prilikom dodavanja novih funkcionalnosti.

**7)**

1. Registruj/Brisi pacijenta
  - a) Zakazi pregled pacijentu koji je registrovan od ranije
  - b) Dodaj novog pacijenta
  - c) Dodaj smrtni slučaj
  - d) Obrisi pacijenta
  - e) Izlaz
2. Prikazi raspored pregleda pacijenta
3. Kreiranje kartona pacijenta
4. Pretraga kartona pacijenta
5. Registruj novi pregled
  - a) Završiti zakazani pregled
  - b) Zakazati novi pregled
6. Analiza sadržaja
  - a) Prikazi ordinacije sortirane po posjećenosti
  - b) Prikazi najposjećenijeg doktora
  - c) Prikazi pacijenta sa najvećim dugom
  - d) Prikazi ukupan dug svih pacijenata
  - e) Prikazi stanje racuna klinike
  - f) Obracunaj mjesec
  - g) Izlaz
7. Naplata
  - a) Gotovinsko
  - b) Na rate
8. Izlaz
9. Dodatne opcije
  - a) Prikazi listu cekanja za ordinaciju
  - b) Prijavi pokvaren aparat
  - c) Prijavi popravljen aparat

- d) Oznaci zatvorenu ordinaciju
- e) Oznaci otvorenu ordinaciju
- f) Izlaz

Pod broj 1 u izborniku osobi na portirnici se nudi mogućnost da registruje ili obriše pacijenta. Ulazom u taj izbornik otvara se podizbornik koji nudi mogućnost da se registruje potpuno novi pacijent, tj. Onaj pacijent koji je prvi put u klinici. Uzimaju se njegovi lični podaci, a zatim se zakazuju pregledi koje želi obaviti. U slučaju da na kliniku dolazi pacijent koji je od ranije registrovan osoba na portirnici ima mogućnost da ga odabere iz postojeće kolekcije i samo registruje preglede. Pregledi se registruju tako što se ponudi lista pregleda koje klinika pruža i odabire se onoliko koliko pacijent zahtjeva. Pregledi su unaprijed postavljeni zajedno sa cijenama i ordinacijama, a datum se postavlja automatski na trenutni. Nakon registrovanih pregleda dešava se nekoliko stvari. Prva je generisanje rasporeda pregleda. Zatim se pacijent označava kao aktivan (tačnije to znači da se nalazi na klinici). Također odgovarajućoj ordinaciji u kojoj se pregled obavlja dodjeljuje se pacijent na čekanju i pregled koji čeka, a dictionary pregleda je konstruisan tako da prilikom prikaza rasporeda vidljivo je i da li je pacijent završio s tim pregledom ili još uvijek čeka na pregled.

Također, u slučaju smrti, dodaje se smrtni slučaj sa svim potrebnim podacima. Ovaj segment je realizovan pod pretpostavkom da je klinika samo ustanova za pružanje usluga u vidu pregleda, a ne i ustanova za liječenje u smislu hospitalizacije.

Brisanje pacijenta znači potpuno brisanje iz kolekcije pacijenata.

Broj dva korisniku programskog rješenja nudi mogućnost da generiše raspored pregleda pacijentu u svakom trenutku dok je on u klinici. Realizovan je tako što se prvo prikaže spisak aktivnih pacijenata, to su pacijenti koji se trenutno nalaze u klinici (registrovani za preglede), zatim se odabire traženi pacijent nakon čega se generiše prvobitan raspored njegovih pregleda. Također će i pisati koji su pregledi obavljani ukoliko pacijent izgubi raspored negdje na pola puta.

Broj tri je rezervisan za uzimanje anamneze ukoliko se pacijent prvi put nalazi u klinici. U suprotnom pacijent već ima svoj karton, a ova mogućnost će samo generisati novu anamnezu. Naravno, prvo je potrebno sa spiska aktivnih pacijenata odabrati onog za kojeg je potrebno kreirati karton.

Broj četiri omogućava doktoru da sa spiska svih pacijenata odabere onog čiji kompletan karton želi da pregleda. To uključuje sve preglede obavljene do tog trenutka. Ispis uključuje ime pregleda, datum obavljanja, doktora koji je pregled obavio, te rezultat i mišljenje doktora, zajedno sa propisanom terapijom i njenim trajanjem.

Broj pet omogućava doktoru da završi pregled koji je registrovan na portirnici. To podrazumijeva unošenje rezultata pregleda, njegovo mišljenje i propisivanje terapije (ako je potrebno). Kada doktor završi sa pregledom, taj pregled se automatski dodaje u pacijentov karton, a na rasporedu pregleda se označava kao obavljen. Također, broj pregleda na čekanju ordinacije se smanjuje, odnosno ta osoba i taj pregled se brišu sa liste čekanja. Tako da se rasporedi na portirnici mogu generisati prema trenutnom stanju, tj. Ukoliko pregled u jednoj ordinaciji potraje toliko da se u drugoj lista čekanja smanji i postane manja nego u prvoj, pacijent koji se registruje na portirnici u tom trenutku bit će prvo upućen u onu u kojoj je u tom trenutku manja lista čekanja. Druga mogućnost koju doktor ima jeste ukoliko smatra da je potrebno zakazati pacijentu novi pregled sa liste dostupnih pregleda. U tom

trenutku raspored pregleda za tog pacijenta se ažurira tako da opet ide u ordinacije gdje je čekanje najmanje.

Broj šest omogućava analizu sadržaja upravi. Između ostalog moguće je vidjeti najposjećenije ordinacije s brojem posjeta (kako bi se možda otvorila još jedna istog tipa ako je pretrpana), prikaz doktora koji je u tom mjesecu najposjećeniji, prikaz pacijenta s najvećim dugom, i ukupan dug koji svi pacijenti imaju prema klinici, te trenutno stanje računa klinike. Stavka obračunaj mjesec služi za izračunavanje koliko klinika treba da plati doktorima u skladu s postavljenim kriterijima i to se oduzima sa računa klinike, ali i koliko treba taj mjesec da dobije od pacijenata koji plaćaju na rate. Klinici se na račun dodaje po jedna rata od svakog pacijenta koji ima dug, a pacijentima se smanjuje ukupan dug za tu ratu, i broj preostalih rata.

Broj sedam je završna stanica pacijenta u klinici. Svi obavljeni pregledi se sumiraju i prikaže se ukupan račun. Pacijent ima mogućnost odabira vrste plaćanja, gdje se razlikuju redovni i novi, te ukoliko pacijent odabere gotovinsko plaćanje, novac se odmah dodaje na račun klinike, a za plaćanje na rate se dodaje prva rata, a pacijentu se kreira dug i broj preostalih rata.

Broj 9 dodatno omogućava realizaciju nekih od problema zabilježenih tokom posmatranja u klinici. Tako prva opcija prikazuje raspored pacijenata koji su sljedeći na redu za pregled za određenu ordinaciju. To će omogućiti doktoru da prati da li pacijent koji je ušao je zaista na redu, što dovodi do smanjenja ulazaka prekoreda. Opcije dva i tri su tu, da u slučaju kvara aparata u ordinaciji doktor ima mogućnost odmah prijaviti kvar, tako da ukoliko se na portirnici pokuša registrovati pregled bit će ispisana poruka kako to nije moguće. Nakon popravke vrši se ponovna aktivacija aparata. Opcije četiri i pet su tu da ukoliko doktor želi napustiti ordinaciju (radi pauze ili bilo čega drugog) on može prikazati ordinaciju kao zatvorenu i u tom slučaju registracija pacijenata na portirnici za tu ordinaciju će biti onemogućena. Nakon povratna, ordinacija se opet aktivira.

Ova opcija omogućava klinici da bolje upravlja resursima, čime će zadovoljiti cilj iz uvoda, a to je veće zadovoljstvo pacijenata, koje će definitivno nastupiti usljed manjih i nepotrebnih redova čekanja.