

PROCES UZIMANJA ZAHTEVA I ANALIZA ZAHTEVA

Upravljanje zahtevima je proces uzimanja zahteva i upravljanja promenama u zahtevima sistema. Zahtevi sistema se uvek menjaju kako bi oslikali promene potreba zainteresovanih korisnika - stejkholdera, promene u okruženju u kojem je sistem instaliran, promene u preduzeću koje planira da instalira sistem, promene u zakonskoj regulativi, itd. Tim promenama mora da se upravlja, kako bi one imale ekonomski smisao i doprinele poslovnim potrebama organizacije koja kupuje sistem. Mora se oceniti tehnička izvodljivost predloženih promena i omogućiti da se one odvijaju u okviru predviđenog budžeta i vremena. Osnovne aktivnosti upravljanja zahtevima su kontrola promena i ocena uticaja promena. Kontrola promena se odnosi na utvrđivanje i izvršavanje formalnih procedura za sakupljanje, verifikovanje i ocenjivanje promena, dok se ocena uticaja promena odnosi na procenu uticaja predloženih promena na sistem kao celinu. Kada se promene odnose na specifične zahteve, važno je proveriti na koje će druge zahteve verovatno te promene uticati. Upravljanje zahtevima zahteva beleženje informacija o npr. vezama između zahteva, izvoru zahteva i dizajnu sistema.

U procese uzimanja zahteva i u njihovom upravljanju učestvuju sledeći procesi:

Domen aplikacije

Treba da se definiše šta sve spada u aplikaciju, odnosno koje zahteve treba uzimati i analizirati, a šta ne pripada samoj aplikaciji. Definisanje domena aplikacije je jako bitan korak i treba da predstoji bilo kakvom postupku prikupljanja zahteva. Domen aplikacije u mnogome definiše šta i kako će biti urađeno u procesu evidencije i analize zahteva. Ovo predstavlja početnu tačku.

Razbijanje sistema na manje module

Nije moguće obuhvatiti odjednom zahteve za složeni sistem. Sistem treba razbiti na odgovarajuće manje celine, module. Zatim treba definisati međusobne odnose između tih modula, pa proces evidencije zahteva voditi po modulima. Kasnije, prilikom analize treba uzeti u obzir veze između modula i proveriti da li je sve evidentirano i razrađeno na vezama između modula.

Zainteresovani korisnici - stejkholderi

Da bi postupak evidencije zahteva bio potpun i adekvatan neophodno je da se napravi spisak svih stejkholdera. Potrebno je indentifikovati sve koji na bilo koji način utiču na budući sistem.

Treba napraviti kategorizaciju stejkholdera, jer zahtevi direktora sektora moraju da se tretiraju na različit način od zahteva operativca koji unosi podatake. Stejkholderi nisu samo osobe koje utiču na evidenciju zahteva već i svi ostali koji su na neki način uključeni u proces, tako da spisak stejkholdera treba proširiti i sa spoljnim korisnicima budućeg sistema, timom koji će raditi razvoj i implementaciju sistema, osobama koje odobravaju investiciju i nikad neće biti direktno uključene u rad sistema. Bitna je i tehnička osposobljenost svakog od evidentiranih stejkholdera.

Nivo detalja

Pre nego što se pređe na konkretnu evidenciju zahteva neophodno je definisati nivo detalja koji se očekuje od procesa evidencije i analize zahteva. Ponekad je neophodno napraviti samo grubu sliku sistema kako bi kasnije implementacioni tim u direktnoj komunikaciji sa klijentom saznao sve potrebne detalje ili u slučaju da je sistem jako veliki, potrebno je napraviti prvu grubu sliku zahteva. Druga situacija je kada tim koji razvija i implementira sistem neće biti u mogućnosti da direktno komunicira sa stejkholderima i krajnjim korisnicima, pa je neophodno da se prilikom evidencije zahteva i njihovom analizom definišu svi detalji i varijacije kako bi kasnije na osnovu toga mogao da se kreira konkretan sistem. Nivo detalja zavisi i od dinamike razvoja, često se ne zna da li će biti dovoljno sredstava za kompletan projekat i da li će neki delovi sistema biti menjani pre nego što bude urađena njihova implementacija. Od toga takođe zavisi nivo detalja sa kojim će se raditi evidencija i analiza zahteva.

Definisanje metodologije

Sledeći korak je definisanje metodologije uzimanja zahteva. Ovde se definiše šta će da se radi odnosno na koji način će se evidentirati zahtevi i kako će ti zahtevi biti dokumentovani (intervjui, postojeće aplikacije, postojeća dokumentacija, ISO ili drugi standardi koji su u primeni,...).

Definisanje izlaznih dokumenata

U zavisnosti od tehničke obrazovanosti stejkholdera i od potreba, definiše se kako će izgledati rezultujući dokumenti, tj. da li će to biti samo Word dokument sa spisakom opisnih zahteva, ili će biti neki PowerDesigner modeli sa svim detaljima i analizama. Naravno, uvek je moguće kombinovati ove rezultate ili prilagoditi izlaze za različite stejkholdere.

Evidencija zahteva

- Pravljenje spiska rola
- Evidencija tačaka posmatranja

- Definisane potencijalnih rizika
- Pravljenje spiskova pitanja za intervju
- Pravljenje templejta za zahteve
- Izvlačenje podataka iz dostavljene dokumentacije
- Kreiranje scenarija
- Evidencija nefunkcionalnih zahteva

Modeliranje zahteva

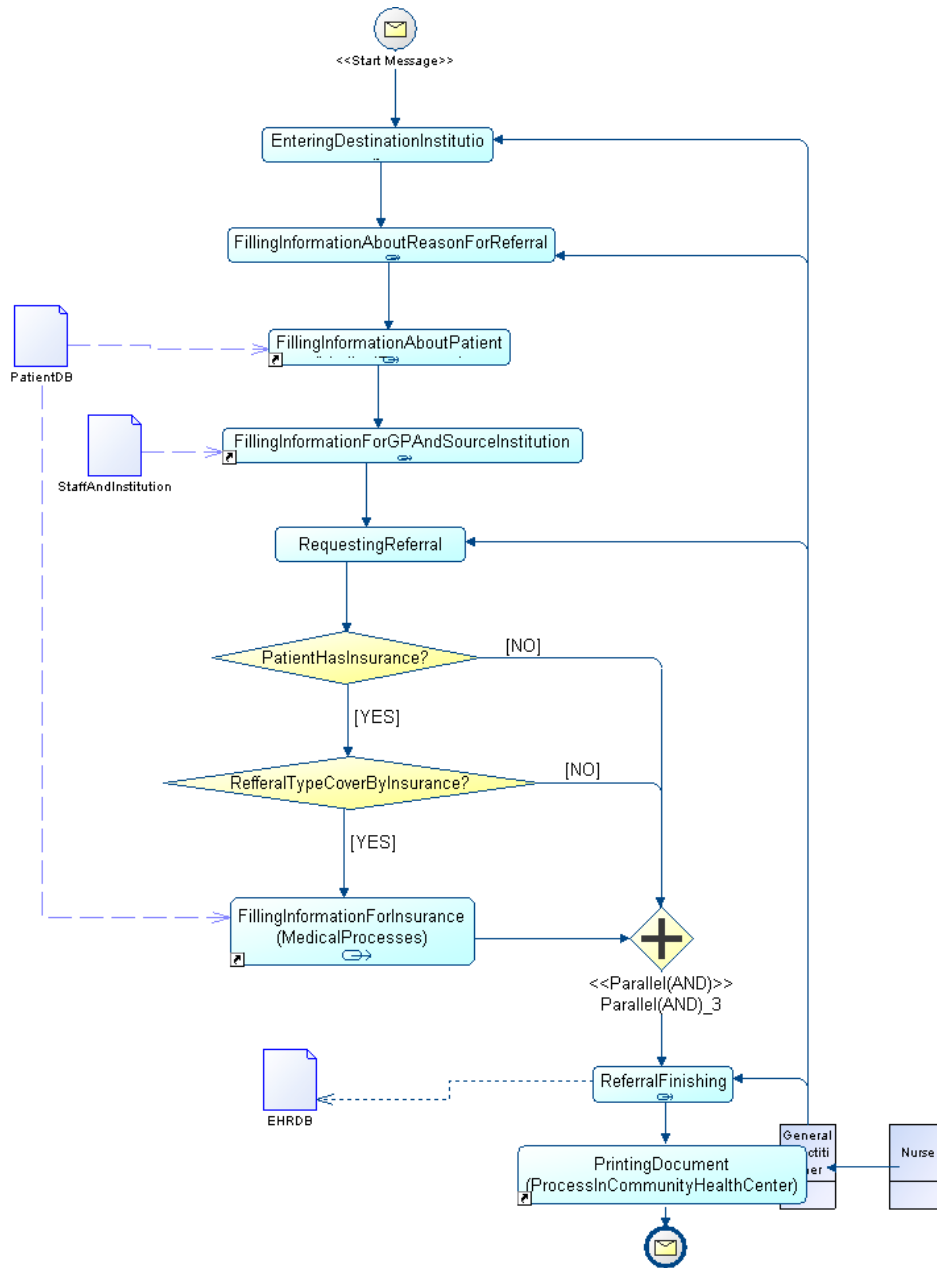
- Kreiranje modela zahteva u nekom alatu (PowerDesigner)
- Model obezbeđuje lako upravljanje izmenama nad zahtevima, kao i praćenje zahteva čak i tokom implementacije
- Mogućnost istovremenog pristupa i rada nad zahtevima
- Definisane prioriteta zahteva, rizika i statusa svakog pojedinačnog zahteva
- Statusi verifikacije za pojedinačne zahteve
- Matrica alokacije resursa, odnosi između zahteva

	Title ID	Full Description	Code	Priority	Workload	Risk	Status
→	1.	Zakazivanje termina Zakazivanje termina pacijentima koji dodju zbog intervencije	REQ_0001	5	500	Medium	Verified
2	1.1	Forma za zakazivanje Forma za zakazivanje sa svim neophodnim podacima potrebnim za unos pacijenta u bazu.	REQ_0007	5	500	Medium	Verified
3	1.1.1	Unos imena i prezimena Unos imena i prezimena pacijenta u bazu nakon poziva forme.	REQ_0008	5	500	Low	Verified
4	1.1.1.	Pretrazivanje Pretrazivanje baze po nekim od implementiranih parametara za pretragu.	REQ_0009	5	500	Low	Verified
5	2.	Evidencije pregleda Forma za evidenciju pregleda nad pacijentima, sadrzi njihove osnovne podatke kao i podatke o potrošenom materijalu.	REQ_0005	4	300	Medium	Verified
6	2.1	Unos podataka o pregledu Unos podataka nakon intervencije doktora kao i utrošeni materijal prilikom intervencije	REQ_0010	4	300	Medium	Verified
7	3.	Pristup kartonu Pristup zdravstvenom kartonu pacijenta sa prethodno unesenim podacima kao i pretrazivanje istih u bazi	REQ_0003	5	400	Low	Verified
8	3.1	Unos imena i prezime Unos imena i prezimena pacijenta u bazu ja jedinstvenim identifikatorima.	REQ_0011	5	400	Low	Verified
9	3.1.1	Prikaz trazanih podataka Prikaz trazanih podataka nakon zahteva doktora.	p	5	400	Low	Verified
10	4.	Pristup finansijama Forma za unos troskova intervencije kao i prikaze vec postojećih ili ne završenih intervencija.	REQ_0004	5	400	High	Verified
11	4.1	Forma za unos datuma Unos početnog i krajnjeg datuma od strane doktora.	REQ_0012	5	400	High	Verified
12	4.1.1	Prikaz plaćanja Sistem prikazuje listu plaćanja za preglede kao i cene nabavljenih lekova i materijala od početnog do krajnjeg datuma i totale.	REQ_0013	5	400	High	Verified

Slika 1: Model zahteva, popis zahteva

Modelovanje biznis procesa

Savremeno poslovanje se uglavnom bavi poslovnim procesima. Većina stejkholdera razume poslovne procese i njihove odnose i zato je zgodno da se zahtevi evidentiraju i predstave u obliku odgovarajućih biznis procesa. Alati poput PowerDesigner-a imaju mogućnost kreiranja modela procesa sa svim detaljima i vezom ka ostalim modelima poput objektnog, modela entiteta, pa i modela zahteva.

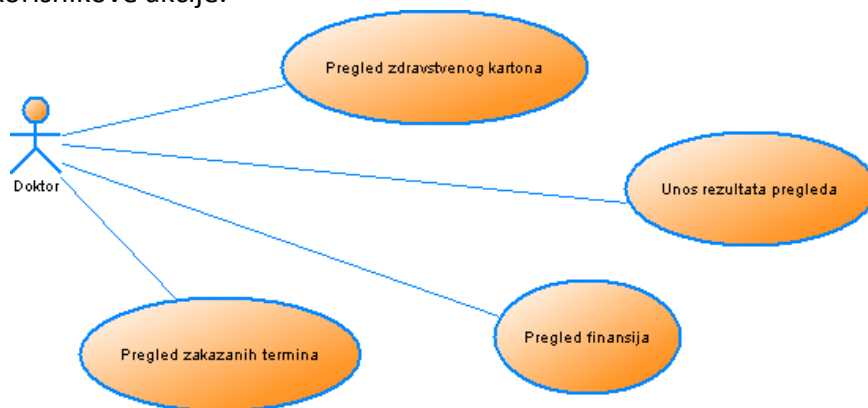


Slika 2: Primer biznis procesa

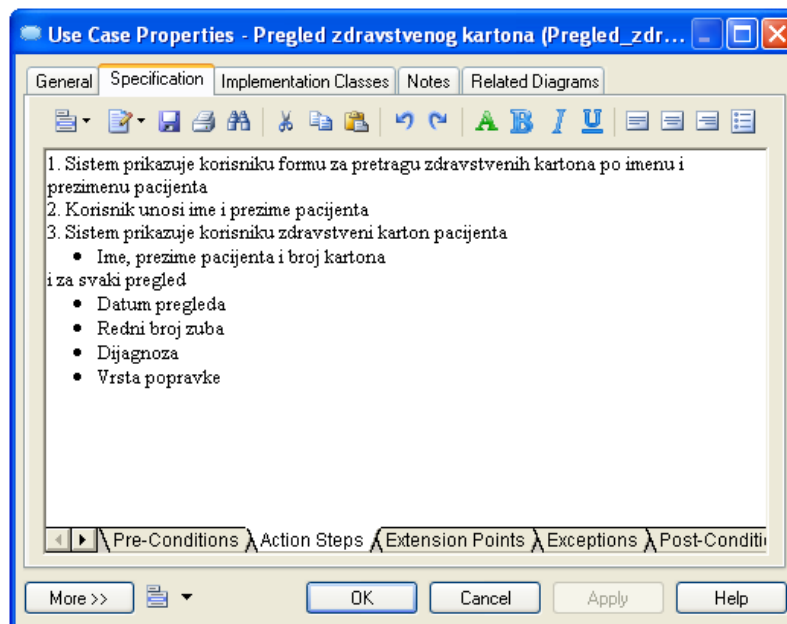
UseCase model

Use case model treba da definiše kako se sistem ponaša u interakciji sa korisnikom. UseCase modeli imaju dva bitna dela, jedan je dijagram u kome može da se vidi koja grupa/korisnik manipuliše sa kojim slučajem korišćenja.

Drugi deo slučaja korišćenja je detaljan opis kako korisnik koristi sistem i kako sistem treba da odgovara na korisnikove akcije.



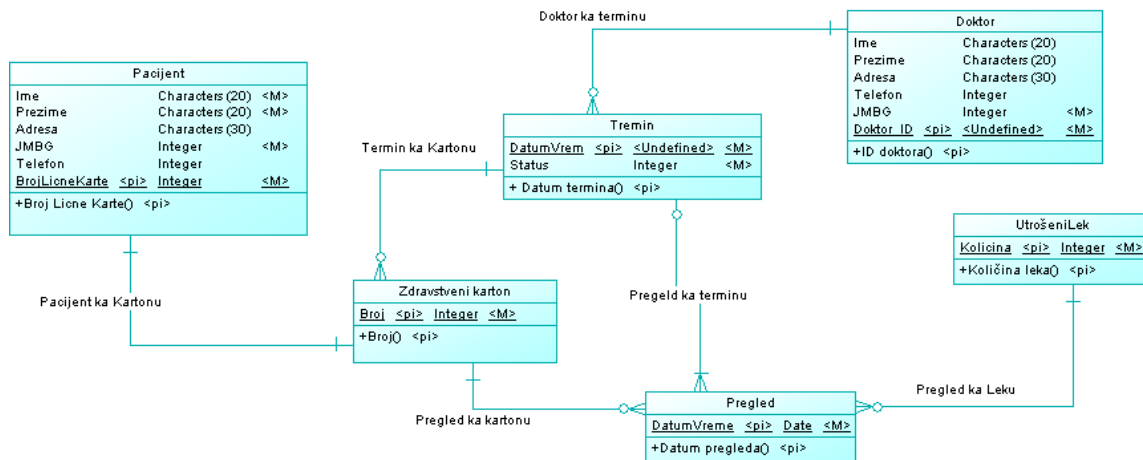
Slika 3: UseCase proces dijagram



Slika 4: Definisanje koraka u okviru slučaja korišćenja

Modeliranje entiteta

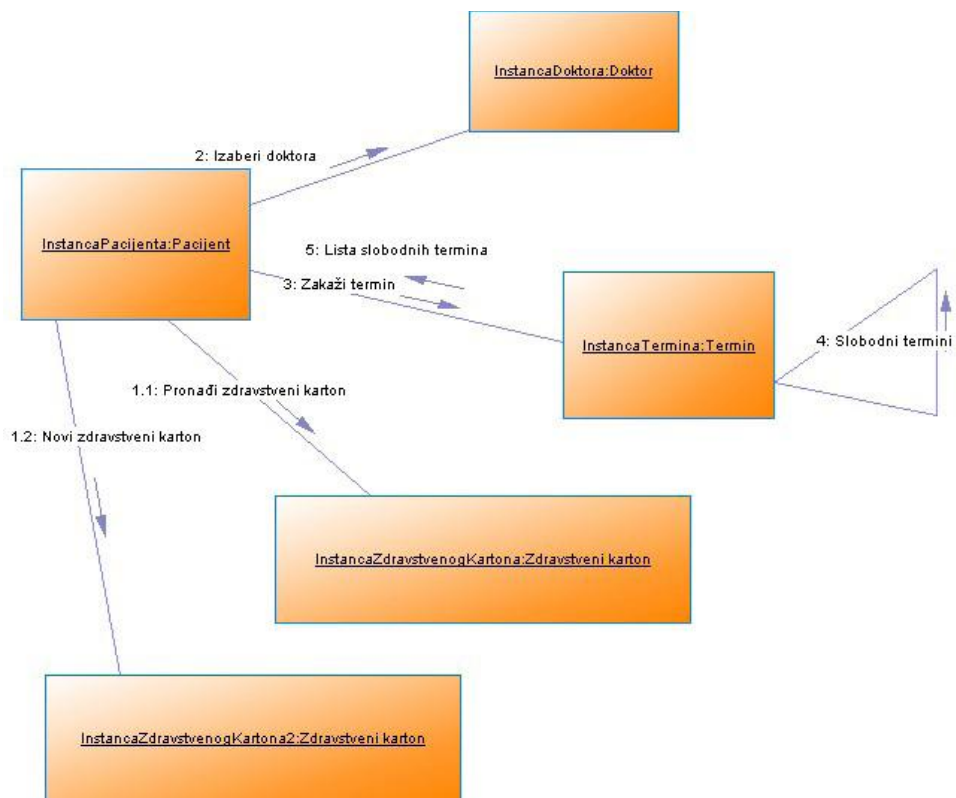
Jedan od načina da se definišu zahtevi je da se kreira konceptualan dijagram podataka kako bi se u njemu definisali svi podaci koje budući sistem treba da pamti, tip tih podataka i veze među podacima.



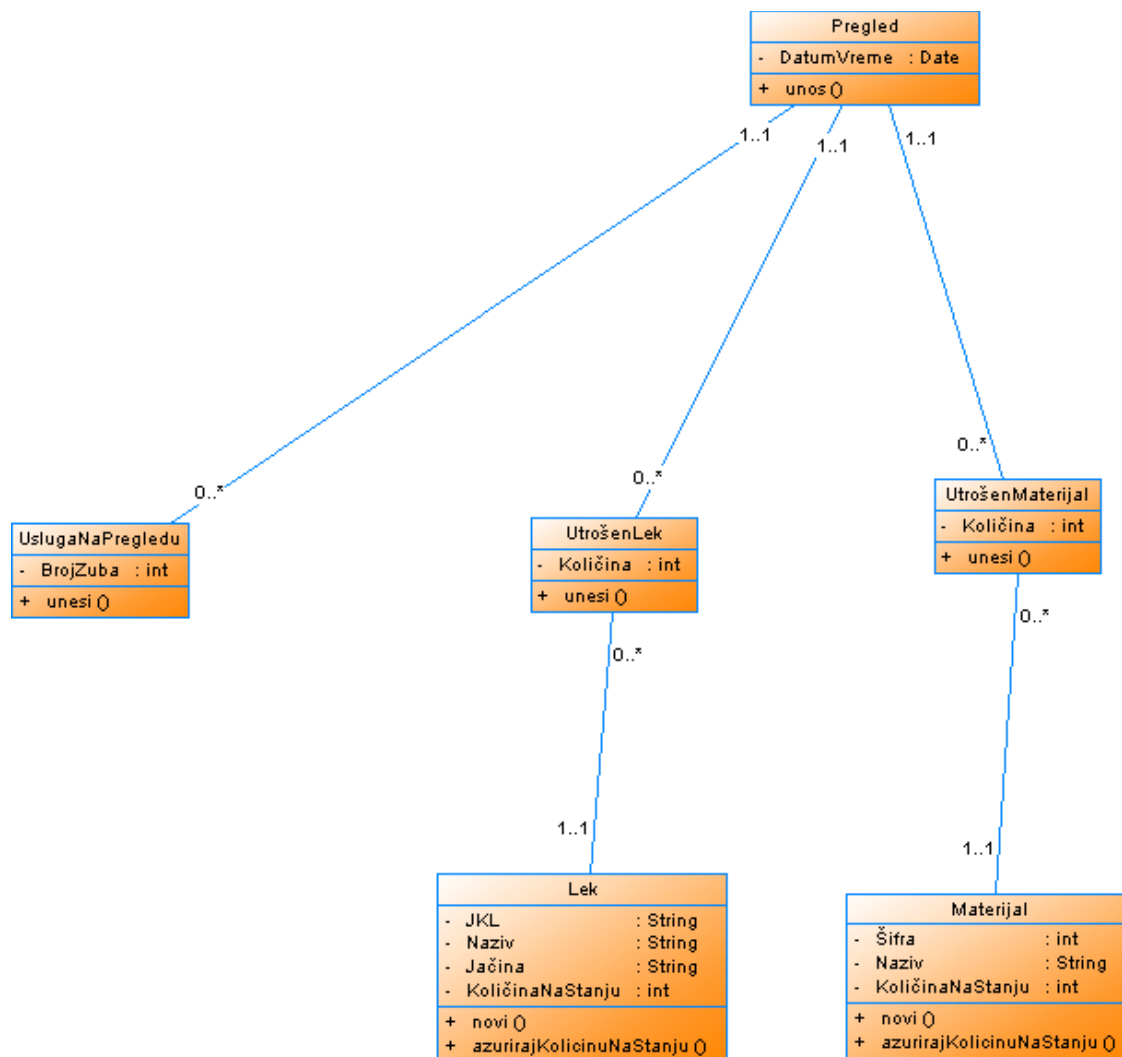
Slika 5: Deo konceptualnog modela

Modelovanje objekata

Konceptualni model je dobar za definisanje podataka koji treba da se čuvaju, ali informacioni sistem osim podataka koji se pamte u bazi mora da ima i razne druge objekte koji učestvuju u kreiranju sistema. Neki od zahteva se odnose na te druge objekte. Da bi to moglo adekvatno da se definiše kreira se i **objektni model**. Objektni model ima više različitih dijagrama (UML) pa ti dijagrami mogu da se koriste da bi demonstrirali različite aspekte zahteva. Najčešće se za potrebe uzimanja zaheva koristi UseCase dijagram i klasni dijagram. Ali, osim ovih ponekad je potrebno koristiti i dijagram paketa, dijagram toka, itd.



Slika 6: Primer kolaboracionog dijagrama



Slika 7: Primer klasnog dijagrama

Prototyping

Ovaj proces podrazumeva kreiranje prototipa u zavisnosti od toga šta se želi demonstrirati prototipom. Prototip može da bude funkcionalni prototip ili GUI prototip. **Funkcionalni prototip** obično ima samo funkcionalnosti koje se direktno odnose na posao aplikacije, a nema funkcionalnosti koje su opšte poput logovanja, backup mehanizma, itd. **GUI prototip** se pravi kao demonstracija kako će buduća aplikacija izgledati, a ujedno služi i kao mehanizam za proveru da li su svi zahtevi dobro shvaćeni i da li su pokriveni GUI-jem.



Slika 8: Primer GUI aplikacije

Korisničko uputstvo

Korisničko uputstvo je dokument koji olakšava razumevanje budućeg sistema stejkholderima koji nemaju dovoljnu tehničku obrazovanost da sagledaju budući sistem iz kreiranih modela. Korisničko uputstvo se koristi i kao mehanizam analize jer u uputstvu moraju da budu obuhvaćene sve aktivnosti koje su sprovedene u sistemu.

Verifikacija zahteva

Kontrolne liste

Kontrolne liste predstavljaju spisak svih pitanja na koja mora da se odgovori u okviru evidentiranih zahteva. Ova kontrolna pitanja služe da se ustanovi da li je zahtevima obuhvaćeno sve što je očekivano. Ujedno su ovo i pitanja pomoću kojih stejkholderi kontrolišu kompletnost evidentiranih zahteva i dostavljene dokumentacije.

Proces verifikacije:

- Dostavljanje dokumenata zahteva odgovornoj osobi
- Distribucija dokumenta svim stejkholderima
- Slanje zamerki od strane stejkholdera
- Pravljenje izmena i zakazivanje sastanaka
- Sastanak na kome se razjašnjavaju problematične zamerke
- Dostavljanje nove verzije dokumenta zahteva
- Ponavljanje procesa dok ima zamerki

Dokument zahteva – struktura dokumenta zahteva

1. Uvod

U njemu treba definisati očekivane korisnike (čitaoce) dokumenta, opisati verzije dokumenta uključujući i obrazloženje za kreiranje nove verzije i dati kratak opis izmena koje su napravljene u svakoj verziji.

2. Predstavljanje projekta

Ovde treba definisati proizvod u koji se softver ugrađuje, očekivanja vezana za korišćenje softvera, prezentovati i dati pregled funkcionalnosti softvera.

3. Rečnik

On treba da definiše sve tehničke termine i skraćenice koje se koriste u dokumentu. Ako je dokument namenjen medicinskim radnicima onda ne treba da se objašnjavaju tipične medicinske skraćenice i definicije, ako dokument ima širu upotrebu onda je potrebno u rečnik smestiti i „domain specific“ informacije.

4. Opšti zahtevi korisnika

Ovde treba definisati systemske zahteve iz perspektive korisnika sistema. Oni treba da budu predstavljeni korišćenjem prirodnog jezika i dijagrama. Ovde se najčešće koristi opisni jezik kojim se što vernije predstavlja ono što je traženo. Uz to dijagrami i screenshot-ovi treba da daju dodatne informacije o tome šta se tražilo i na koji način.

5. Arhitektura sistema

U ovom poglavlju treba predstaviti očekivanu arhitekturu sistema na visokom nivou koja pokazuje distribuciju funkcija na module sistema. Komponente arhitekture koje se koriste u okviru modula treba posebno istaći. Za ovo je najbolje koristiti PowerDesigner Enterprise model, pomoću koga ove informacije mogu lako vizuelno da se predstavljaju, tako da i ne-tehničkom osoblju bude jasno kako će izgledati arhitektura sistema.

6. Specifikacija hardvera

Ovo poglavlje je opciono i ono treba detaljno da opiše hardver koji treba da se koristi za projektovani softver. Može se izostaviti ukoliko se koriste usluge provajdera ili je hardver na neki drugi način definisan.

7. Detaljne softverske specifikacije

Ovde se daje detaljan opis očekivane funkcionalnosti softvera sistema. Kada je neophodno, mogu se uključiti specifični algoritmi kojima se detaljno opisuju potrebna izračunavanja. Ukoliko se za razvoj koristi pristup prototipa i platforma standardnih instrumenata, ovo poglavlje može biti izostavljeno. Ovaj deo se obično izostavlja kada su stejkholderi više „domain specific“ (lekari i medicinsko osoblje), a ne tehničko osoblje koje se razume u ovu oblast.

8. Zahtevi koji se odnose na pouzdanost i performanse

U ovom poglavlju treba opisati zahteve koji se odnose na pouzdanost i performanse koje se očekuju od sistema. Oni treba da se odnose na opis korisničkih zahteva iz četvrtog poglavlja.

9. Ukoliko za to ima opravdanja mogu se uključiti sledeći prilozi:

- Specifikacije hardverskog interfejsa
- Softverske komponente koje se pri implementaciji sistema koriste više puta
- Specifikacija strukture podataka
- Model toka podataka softverskog sistema
- Detaljni model objekata softverskog sistema.

10. Indeks