

Programsko inženjerstvo

Ak. god. 2021./2022.

True Blood

Dokumentacija, Rev. 1

Grupa: *Illidimus Digitus*

Voditelj: *David Kerman*

Datum predaje: *19.11.2021.*

Nastavnik: *Ivan Lovrić*

Sadržaj

1 Dnevnik promjena dokumentacije	2
2 Opis projektnog zadatka	4
3 Specifikacija programske potpore	8
3.1 Funkcionalni zahtjevi	8
3.1.1 Obrasci uporabe	10
3.1.2 Sekvencijski dijagrami	24
3.2 Ostali zahtjevi	28
4 Arhitektura i dizajn sustava	29
4.1 Baza podataka	32
4.1.1 Opis tablica	33
4.1.2 Dijagram baze podataka	37
4.2 Dijagram razreda	38
4.3 Dijagram stanja	42
4.4 Dijagram aktivnosti	43
4.5 Dijagram komponenti	44
5 Implementacija i korisničko sučelje	45
5.1 Korištene tehnologije i alati	45
5.2 Ispitivanje programskog rješenja	46
5.2.1 Ispitivanje komponenti	46
5.2.2 Ispitivanje sustava	46
5.3 Dijagram razmještaja	47
5.4 Upute za puštanje u pogon	48
6 Zaključak i budući rad	49
7 Popis literature	50
Popis literature	50

Indeks slika i dijagrama	51
Dodatak: Prikaz aktivnosti grupe	52

1. Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Napravljen predložak.	Kerman	28.10.2021.
0.2	Dodani funkcionalni zahtjevi	Kerman, Jurinić	28.10.2021.
0.3	Dodani opisi UC11 - UC19	Kerman	25.10.2021.
0.4	Dodani opisi UC1 - UC10	Jurinić	29.10.2021
0.5	Sekvencijski dijagrami	Hudiček	31.10.2021
0.6	Opis projektnog zadatka	Šlezak	01.11.2021
0.7	Dodan opis baze	Vugrinec	03.11.2021
0.6	Dodani dijagrami obrazaca uporabe	Kerman	03.11.2021
0.7	Dodani nefunkcionalni zahtjevi	Okreša	04.11.2021
0.8	Dodani opisi varijabli u opisu baze	Vugrinec	05.11.2021
0.9	Definirana arhitektura sustava	Kerman	10.11.2021.
0.10	Osvježeni obrasci upotrebe	Kerman	12.11.2021.
0.11	Popravljen opis baze	Vugrinec	15.11.2021.
0.12	Popravljen pravopis	Kerman	15.11.2021.
0.13	Popravljeni obrasci uporabe	Kerman, Jurinić	17.11.2021.
0.14	Dodani opisi i dijagrami razreda	Kerman, Hudiček	17.11.2021.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodatka	Autori	Datum
1.0	Verzija samo s bitnim dijelovima za 1. ciklus	*	17.11.2021.

2. Opis projektnog zadatka

Cilj ovog projekta je razviti programsku podršku za stvaranje responzivne web aplikacije "True Blood" koja omogućuje prikupljanje i objavljivanje podataka o prikupljenim dozama darivane krvi te općenito vođenje evidencije podataka za banke krvi. Ovaj projekt omogućio bi smanjenje vremena i znatno olakšanje obavljanje administracijskih poslova pri samoj djelatnosti prikupljanja krvi. Vođenje evidencije 'klasičnim' načinom u današnje vrijeme je skupo, sporo i neefikasno, a ovim projektom nestaje potreba za standardnom hrpom papira, što ujedno i smanjuje kompliciranost, potrebu za dodatnom radnom snagom te dodatne troškove, a da ne govorimo o mogućoj dodatnoj redundanciji podataka i njihovoj mogućoj većoj opsežnosti.

Neregistriranom/neprijavljenom korisniku se na javim web stranicama prikazuje trenutno stanje zaliha različitih krvnih grupa te mogućnost logina i registracije. U sustavu postoje 3 vrste korisnika:

- administrator
- djelatnik banke
- donor

Djelatnik banke i donor na svoju email adresu dobivaju link za aktivaciju korisničkog računa, privremenu lozinku te djelatnici dobivaju svoje korisničko ime, a donor svoj donorId koji će koristiti kao korisničko ime. Prilikom aktivacije korisničkog računa korisnik odabire lozinku koju će koristiti. Administrator sustava administrira korisničke račune. On kreira nove korisničke račune za ulogu djelatnika banke te može u bilo kojem trenutku deaktivirati bilo koji korisnički račun. Administrator isto tako definira gornju i donju granicu optimalne količine krvi za svaku krvnu grupu, kako bi sustav dojavio upozorenja u slučaju prekoračenja gornje ili donje granice. Djelatnik banke krvi evidentira podatke o donoru kada potencijalni donor pristupa darivanju krvi. Ukoliko donor još nije evidentiran u sustavu, djelatnik banke kreira njegov korisnički profil te popunjava sve potrebne podatke:

- matični podaci
- kontakt podaci

- zdravstveni podaci

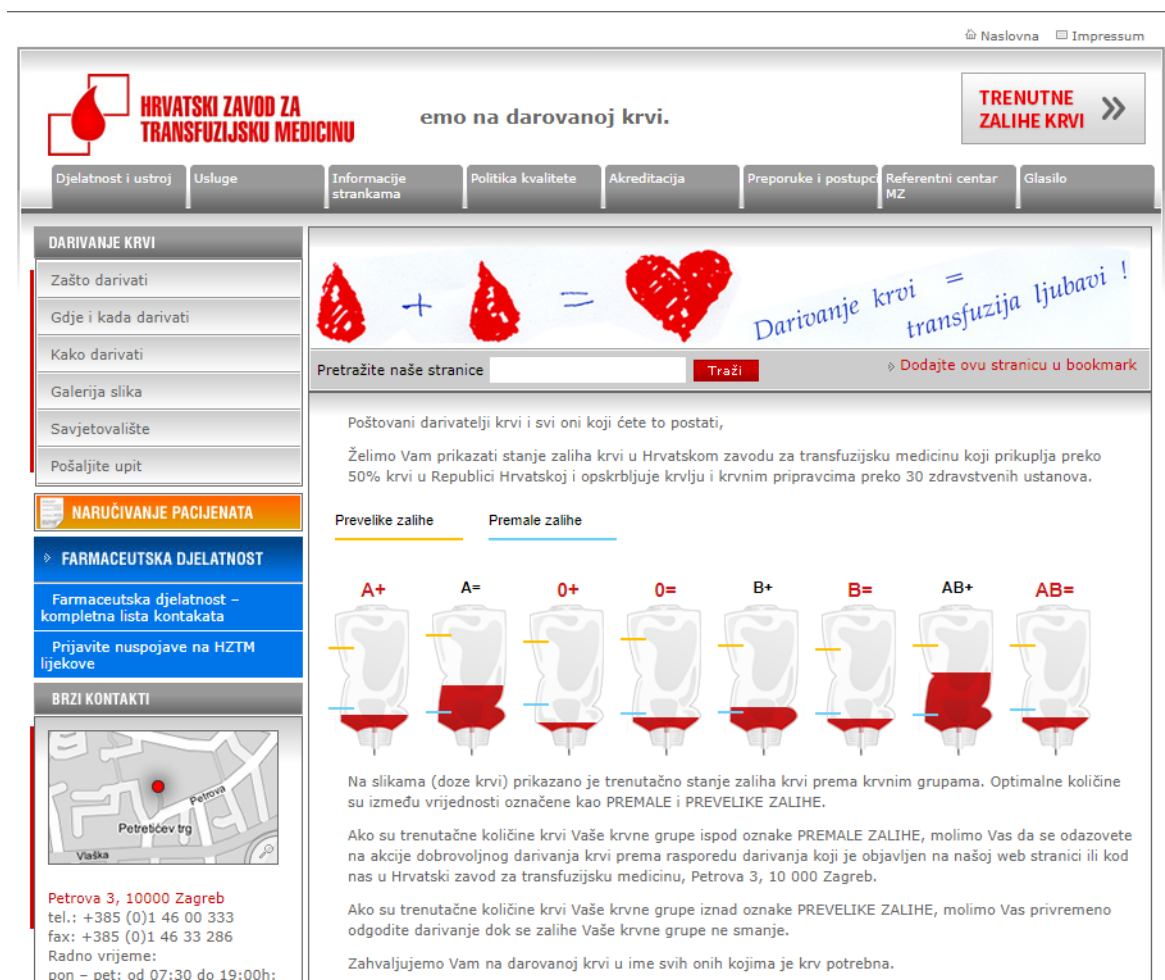
Ukoliko je donor već koristio usluge ustanove, povlače se zadnji aktualni podaci koje djelatnik banke po potrebi nadopunjuje. Djelatnik banke u sustav evidentira svaki pokušaj doniranja. Prije nego donor pristupi darivanju krvi, djelatnik banke provjerava njegovo zdravstveno stanje, te prema tome može prihvatiti te privremeno ili trajno odbiti donora. Djelatnik banke evidentira uspješno doniranje krvi, ali i potrošnju (tj. slanje određenog broja jedinica krvi van banke). Time se povećava tj. smanjuje zaliha određene krvne grupe što je odmah vidljivo na javim web stranicama, a ukoliko se prekorači gornja ili donja granica zalihe za neku krvnu grupu, djelatnici dobivaju notifikaciju putem emaila. Donori imaju mogućnost sami se registrirati na web stranici te ažurirati svoje matične i kontakt podatke te pregledavati zapisane zdravstvene podatke i povijest svojih (uspješnih i odbijenih) doniranja. Svakom evidencijom uspješnog darivanja krvi, donor na svoj email dobiva poruku s potvrdom o pristupanju darivanju krvi u PDF formatu koju može i sam podići naknadno iz aplikacije. Prilikom svakog logiranja donora u sustav (koji nema trajnu zabranu darivanja krvi), aplikacija će donoru prikazati poruku u ovisnosti o trenutnom stanju zaliha krvi za njegovu krvnu grupu:

- stanje zaliha je ispod optimalne granice
- stanje zaliha je optimalno
- stanje zaliha je iznad gornje optimalne granice

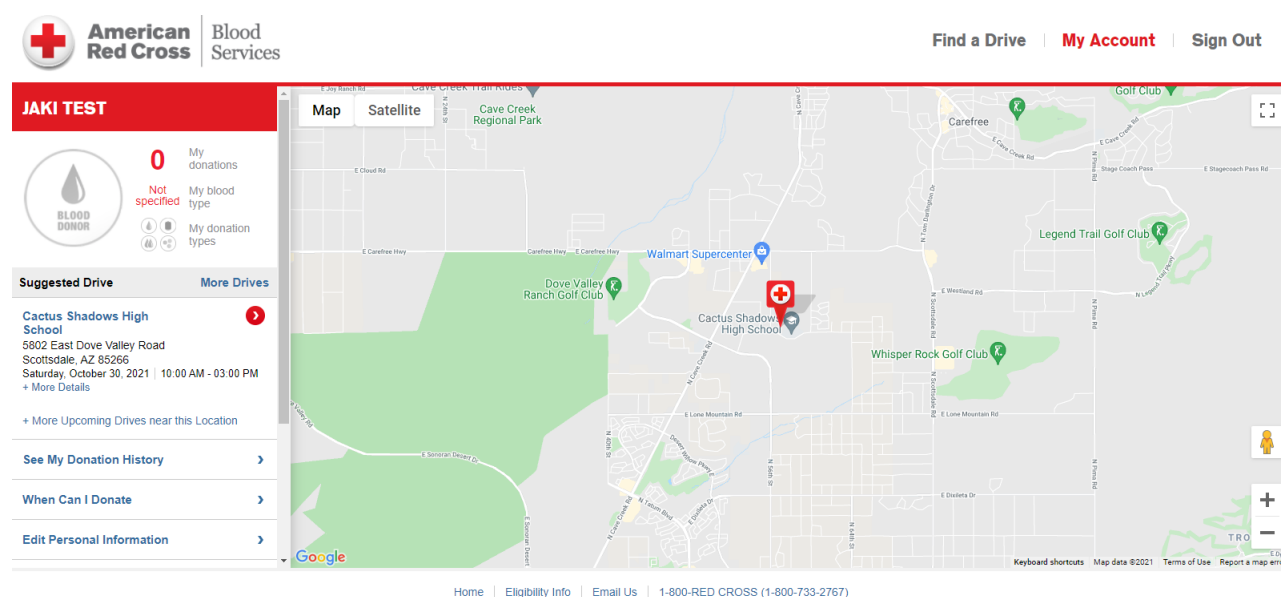
Donori dobivaju notifikacije od sustava nakon što istekne dopušteni period od zadnjeg darivanja i ukoliko zaliha njihove krvne grupe padne ispod minimalne granice.

Ovo rješenje bi mogle koristiti jednako manje ali i veće banke krvi koje još nemaju informacijski sustav za prikupljanje i objavljivanje podataka o prikupljenim dozama krvi. Pošto se radi o web aplikaciji, ona se lako može implementirati u postojeći sustav web stranica koju pojedini potencijalni korisnik rješenja ima, te se, ukoliko je to zbilja potrebno, i stil same web aplikacije može prilagoditi da korespondira sa stilom postojećih web stranica. Isto tako, ukoliko potencijalni korisnik već posjeduje neku vlastitu bazu podataka koju koristi za neku funkcionalnost logiranja i/ili evidenciju nekih relativnih podataka te bi ju htio nastaviti koristiti, rješenje je moguće prilagoditi da se koristi zahtijevanim podacima.

Trenutno u hrvatskoj postoje implementirani dijelovi ovih rješenja, točnije, postoje javne stranice gdje se prikazuje trenutno stanje zaliha različitih krvnih grupa (npr. slika 2.1). Ako pogledamo malo šire po svijetu, možemo uočiti da ima očito i nekih razrađenih sistema za evidenciju sa loginom donora (npr. slika 2.2), no sva ta rješenja nemaju kompaktno razrađen sustav kod kojih bi prijašnji donori dobivali notifikacije s obzirom na stanje banke krvi i pojedinačnih krvnih grupa, koje je javno i na jednostavan (grafički) način dostupno svima.



Slika 2.1: Primjer javno dostupnih podataka o stanju banke krvi Hrvatskog zavoda za transfuzijsku medicinu



Slika 2.2: Primjer web sučelja logiranog korisnika Američkog crvenog križa

Ovo rješenje ima i nešto mjesta za kasniju nadogradnju. Jedna od mogućnosti je ugradnja podrške za notifikacije putem SMS poruka. Moguće je kasnije dodavanje i nešto poput nagradnog sistema za donore gdje bi donori svakim darivanjem krvi skupljali bodove koje bi mogli iskoristiti npr. za neke popuste kod nekih sponzora, besplatan ručak, ulaznicu u kino itd. Naravno, kasnijih nadogradnji (i samih prilagodbi) može biti još, ovisno o željama i kasnijim potrebama korisnika rješenja, ukoliko su izvediva u sklopu ovog projekta.

3. Specifikacija programske potpore

3.1 Funkcionalni zahtjevi

Dionici:

1. Korisnici
2. Javnost
3. Razvojni tim
4. Naručitelj

Aktori i njihovi funkcionalni zahtjevi:

1. Neregistrirani/neprijavljeni korisnik(inicijator) može:
 - (a) pregledati trenutno stanje zaliha
 - (b) se registrirati u sustav, stvoriti korisnički račun za koji su mu potrebni matični i kontakt podaci
2. Donor (inicijator) može:
 - (a) pregledavati i mijenjati osobne podatke
 - (b) pregledavati povijest svojih doniranja
 - (c) iz aplikacije dobiti PDF potvrdu
 - (d) pregledati poruku u ovisnosti o trenutnom stanju zaliha krvi
 - (e) aktivirati račun aktivacijskim linkom i odabrati lozinku
3. Djelatnik banke (inicijator) može:
 - (a) kreirati korisnički profil donora
 - (b) evidentirati svaki pokušaj doniranja (uspješan / neuspješan)
 - (c) evidentirati privremeno ili trajno odbijanje
 - (d) evidentirati potrošnju krvi
 - (e) aktivirati račun aktivacijskim linkom i odabrati lozinku
 - (f) vidjeti popis registriranih donora

4. Administrator (inicijator) može:

- (a) definirati gornju i donju granicu optimalne količine krvi
- (b) kreirati nove korisničke račune za ulogu djelatnika banke
- (c) deaktivirati korisnički račun djelatnika banke ili donora
- (d) vidjeti popis registriranih svih korisnika i njihovih osobnih podataka

5. Baza podataka (sudionik):

- (a) pohranjuje sve podatke o korisnicima i njihovim ovlastima
- (b) pohranjuje trenutno stanje količine krvi, te donju i gornju granicu optimalne količine krvi
- (c) pohranjuje sve podatke o donacijama krvi

6. Sustav za automatske poslove(inicijator) može:

- (a) slati email notifikacije donoru nakon 3 mjeseca od uspješnog darivanja ako je muškarac ili nakon 4 mjeseca ako je žensko

3.1.1 Obrasci uporabe

Opis obrazaca uporabe

UC1 - Pregledaj količine krvi

- **Glavni sudionik:** Neregistrirani/neprijavljeni korisnik
- **Cilj:** Prikazati trenutnu količinu krvi u banci
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Količine krvi su prikazane otvaranjem aplikacije, te njihove gornje i donje granice

UC2 - Registriraj korisnika

- **Glavni sudionik:** Neregistrirani/Neprijavljeni korisnik, djelatnik banke
- **Cilj:** Stvoriti korisnički račun donora za pristup sustavu
- **Sudionici:** Baza podataka
- **Preduvjet:** Djelatnik banke prijavljen
- **Opis osnovnog tijeka:**
 1. Neregistriran/neprijavljen korisnik/djelatnik banke odabire opciju za registraciju novog donora
 2. Sustav otvara obrazac za registraciju
 3. Neregistriran/neprijavljen korisnik/djelatnik banke unosi potrebne podatke i odabire akciju "Kreiraj račun"
 4. Sustav validira unesene podatke i sprema ih u bazu podataka
 5. Sustav generira email poruku s podacima o korisničkom računu i aktivacijskim linkom i šalje na mail adresu korisnika
- **Opis mogućih odstupanja:**
 - 2.a Odabir već korištenog e-maila, unos podataka u nedozvoljenom formatu ili unos neispravnog e-maila
 1. Korisnik dobiva obavijest o neuspjelom upisu i vraća ga na korak 2 osnovnog tijeka

UC3 - Prijavi se u sustav

- **Glavni sudionik:** Donor
- **Cilj:** Prijava u sustav i dobivanje dodatnih mogućnosti aplikacije
- **Sudionici:** Baza podataka
- **Preduvjet:** Registracija
- **Opis osnovnog tijeka:**
 1. Donor/djelatnik/administrator unosi e-mail i lozinku u login formu te odabire akciju "Login"
 2. Sustav validira unesene podatke, provodi autentikaciju i autorizaciju korisnika
 3. Sustav otvara početni ekran aplikacije te korisniku ispisuje obavijest da pogleda pretnac svojih poruka
 4. Pregledavanje ispisane obavijesti na ekranu da pogleda u pretnac svojih poruka ako su prijedene optimalne granice ili ako dobiva poziv na doniranje
- **Opis mogućih odstupanja:**
 - 2.a Neispravan e-mail i/ili lozinka
 1. Sustav obavještava korisnika o neuspjeloj prijavi i vraća ga u korak 1 osnovnog tijeka

UC4 - Pregledaj osobne podatke

- **Glavni sudionik:** Donor
- **Cilj:** Prikazati osobne podatke prijavljenog donora
- **Sudionici:** Baza podataka
- **Preduvjet:** Donor je prijavljen
- **Opis osnovnog tijeka:**
 1. Donor odabire opciju "Osobni podaci"
 2. Sustav dohvaća osobne podatke donora iz baze podataka te ih prikazuje

UC5 - Promjeni osobne podatke

- **Glavni sudionik:** Donor
- **Cilj:** Promjeniti matične i/ili kontakt podatke
- **Sudionici:** Baza podataka
- **Preduvjet:** Donor je prijavljen
- **Opis osnovnog tijeka:**
 1. Donor odabire opciju "Promijeni osobne podatke"
 2. Sustav otvara obrazac za promjenu osobnih podataka
 3. Donor mijenja svoje podatke i odabire opciju "Spremi promjene"
 4. Sustav ažurira bazu podataka
- **Opis mogućih odstupanja:**
 - 3.a Korisnik ne spremi promjenu
 1. Sustav obavještava korisnika da nije spremio podatke prilikom izlaska iz prozora i vraća ga u korak 3 osnovnog tijeka

UC6 - Pregledaj povijest darivanja

- **Glavni sudionik:** Donor
- **Cilj:** Pregledati svoju povijest darivanja krvi
- **Sudionici:** Baza podataka
- **Preduvjet:** Donor je prijavljen
- **Opis osnovnog tijeka:**
 1. Donor odabire opciju "Povijest darivanja"
 2. Sustav dohvaća povijest darivanja s baze podataka te ju prikazuje

UC7 - Izvadi PDF potvrde

- **Glavni sudionik:** Donor
- **Cilj:** Dobiti PDF potvrdu o doniranju
- **Sudionici:** Baza podataka
- **Preduvjet:** Donor je prijavljen i barem jedanput uspješno darovao krv
- **Opis osnovnog tijeka:**
 1. Donor odabire opciju "Povijest darivanja"
 2. Sustav dohvaća povijest darivanja s baze podataka te ju prikazuje
 3. Donor odabire opciju "Želim PDF potvrdu" koja se nalazi uz svaki zapis darivanja krvi
 4. Sustav donoru šalje potvrdu na mail

UC8 - Pregledaj poruke

- **Glavni sudionik:** Donor
- **Cilj:** pregledati dospjelu poruku
- **Sudionici:** Baza podataka
- **Preduvjet:** Donor je prijavljen i nema trajnu/privremenu zabranu darivanja krvi
- **Opis osnovnog tijeka:**
 1. Donor odabire opciju "Pretinac"
 2. Sustav dohvaća poruke iz baze podataka i prikazuje ih
 3. Donor pregledava prikazane poruke

UC9 - Smanji količinu krvi

- **Glavni sudionik:** Djelatnik banke
- **Cilj:** slanje određenog broja jedinica krvi u vanjsku instituciju
- **Sudionici:** Baza podataka
- **Preduvjet:** Djelatnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Djelatnik odabire krvnu grupu čiju zalihu želi smanjiti
 2. Djelatnik upisuje količinu i naziv institucije u koju je potrebno poslati krv
 3. Djelatnik odabire akciju "Smanji zalihu"
 4. Sustav provjerava je li moguće provesti smanjivanje
 5. Sustav provodi smanjivanje u bazi podataka

- **Opis mogućih odstupanja:**

4.a Djelatnik je upisao ili preveliku količinu krvi ili manju od 1

1. Sustav obavještava djelatnika o neispravnom unosu te ga vraća na korak 2 osnovnog tijeka

5.a Smanjivanjem se prešla donja granica optimalne količine krvi

1. Sustav šalje notifikaciju na mail adresu djelatniku banke i donoru(donoru se također šalje poruka u pretinac poruke)

UC10 - Pregledaj popis donora

- **Glavni sudionik:** Djelatnik banke, administrator
- **Cilj:** Prikazati popis svih registriranih donora
- **Sudionici:** Baza podataka
- **Preduvjet:** Djelatnik je prijavljen ili administrator je prijavljen
- **Opis osnovnog tijeka:**
 1. Djelatnik/administrator odabire opciju "Popis donora"
 2. Sustav djelatniku/administratoru dohvaća iz baze popis registriranih donora te ga prikazuje

UC11 -Aktiviraj račun

- **Glavni sudionik:** Neregistrirani/neprijavljeni korisnik, djelatnik banke
- **Cilj:** Aktivirati račun
- **Sudionici:**Baza podataka
- **Preduvjet:**Registracija
- **Opis osnovnog tijeka:**
 1. Neregistrirani/neprijavljeni korisnik ili djelatnik banke nakon klika na link u mailu vodi u aplikaciju
 2. Neregistrirani/neprijavljeni korisnik ili djelatnik banke se prijavljuje dodijeljenom lozinkom
 3. Sustav ponovno renderira login stranicu gdje se traži da se unese nova proizvoljna lozinka
 4. Neregistrirani/neprijavljeni korisnik ili djelatnik banke se prijavljuje odabranom lozinkom

UC12 -Evidentiraj pokušaj doniranja

- **Glavni sudionik:** Djelatnik banke
- **Cilj:** Evidentirati pokušaj doniranja i osvježiti podatke po potrebi
- **Sudionici:**Baza podataka
- **Preduvjet:**Djelatnik banke je prijavljen i donor je registriran
- **Opis osnovnog tijeka:**
 1. Djelatnik odabere opciju za evidentiranje pokušaja doniranja određenog donora
 2. Sustav prikazuje podatke o donoru koje može mijenjati po potrebi
 3. Sustav ispod osobnih podataka prikazuje upitnik kojeg ispunjava djelatnik banke
 4. Djelatnik evidentira s uspješan ili neuspješan pokušaj
 5. Djelatnik banke sprema promjene
 6. Baza podataka se ažurira, povećava se razina određene vrste krvi i zapisuje se pokušaj doniranja krvi
 7. Sustav šalje email poruku s PDF potvrdom
 8. Ako su se prešle gornje granice šalje se notifikacija na mail djelatniku banke
- **Opis mogućih odstupanja:**
 - 4.a Donor je bolovao ili boluje od teških kroničnih bolesti dišnog i probavnog sustava ili donor boluje od belosti srca i krvnih žila, zloćunih bolesti, bolesti jestre AIDS-a, šećerne bolesti ili je osoba ovisnik o alkoholu ili drogama ili donor ima spolne odnose s drugim muškarcima ili donor često mijenja seksualne partnere ili je donor uzimao drogu intravenskim putem ili je donor liječen zbog spolno prenosivih bolesti ili je donor HIV pozitivan ili su donori seksualni partneri gore navedenih osoba ili je dob iznad 65 godina ili je dob iznad 60 godina ako prvi put daje krv
 1. Donor se trajno odbija
 2. Sustav vraća na home stranicu
 3. U bazu se zapisuje trajno odbijanje za određenog donora

- 4.b Tjelesna težina je ispod 55kg ili Tjelesna temperatura je iznad 37°C ili krvi tlak sistolički je ispod 100, a dijastolički ispod 60 ili je puls ispod 50 ili iznad 100 otkucaja u minuti ili je hemoglobin muškaraca ispod 135g/L, a žena ispod 125g/L ili donor trenutno uzima antibiotike ili druge lijekove ili je donor konzumirao alkohol unutar 8 sati ili donor ima akutno bolesno stanje ili je žena za vrijeme menstruacije ili trudnoće ili je donor obavljao opasne poslove tog dana ili je dob ispod 18 godina ili je muškarac već u tekućoj godini darivao 4 puta (ili nije prošlo 3 mjeseca od zadnjeg) ili je žena koja je već darivala 3 puta (ili nije prošlo 4 mjeseca od zadnjeg)
1. Donor se privremeno odbija
 2. Sustav vraća na home stranicu
 3. U bazu se zapisuje privremeno odbijanje za određenog donora
- 5.a Djelatnik promijeni podatke o donoru i ne spremni promjene
1. Sustav obavještava djelatnika banke da nije spremio podatke i vraća ga na korak 5.
- 8.a Prešle su se gornje optimalne granice
1. šalju se notifikacije na mail djelatnika

UC13 -Obriši donora

- **Glavni sudionik:** Administrator
- **Cilj:** izbrisati kor. račun donora
- **Sudionici:**Baza podataka
- **Preduvjet:**Administrator prijavljen
- **Opis osnovnog tijeka:**
 1. Administrator odabire opciju "Prikaz donora"
 2. Sustav prikazuje popis registriranih donora
 3. Administrator odabire opciju "Obriši donora"
 4. Sustav briše donora
 5. Baza se ažurira

UC14 -Definiraj optimalne granice

- **Glavni sudionik:** Administrator
- **Cilj:** Definirati gornju i donju granicu optimalnih količina krvi za pojedinu grupu
- **Sudionici:**Baza podataka
- **Preduvjet:**Administrator je prijavljen
- **Opis osnovnog tijeka:**
 1. Administrator odabire opciju "Definiraj granice"
 2. Sustav prikazuje trenutno određene granice za pojedinu vrstu krvi
 3. Administrator definira granice za pojedinu vrstu krvi
 4. Administrator sprema promjene
 5. Baza se ažurira
- **Opis mogućih odstupanja:**
 - 4.a Administrator ne spremi promjene
 1. Sustav obavještava administratora da nije spremio promjene i vraća ga na korak 4.
 - 4.b Prešle su se gornje optimalne granice
 1. Sustav šalje notifikacije na mail djelatnika
 - 4.c Prešle su se donje optimalne granice
 1. Sustav šalje notifikacije na mail djelatnika i donora ako se radi o njegovoj krvnoj grupi (također donoru se šalje poruka u pretinac za poruke)

UC15 -Pregledaj popis djelatnika

- **Glavni sudionik:** Administrator
- **Cilj:** Prikazati popis svih djelatnika banke
- **Sudionici:**Baza podataka
- **Preduvjet:**Administrator je prijavljen
- **Opis osnovnog tijeka:**
 1. Administrator odabire opciju "Prikaz popisa djelatnika banke"
 2. Sustav prikazuje popis svih djelatnika banke po abecednom redu

UC16 -Obriši djelatnika banke

- **Glavni sudionik:** Administrator
- **Cilj:** Izbrisati kor. račun djelatnika banke
- **Sudionici:**Baza podataka
- **Preduvjet:**Administrator prijavljen
- **Opis osnovnog tijeka:**
 1. Administrator odabire opciju "Prikaz djelatnika banke"
 2. Sustav vraća popis djelatnika banke
 3. Administrator odabire opciju "Obriši djelatnika banke"
 4. Sustav briše djelatnika
 5. Baza se ažurira

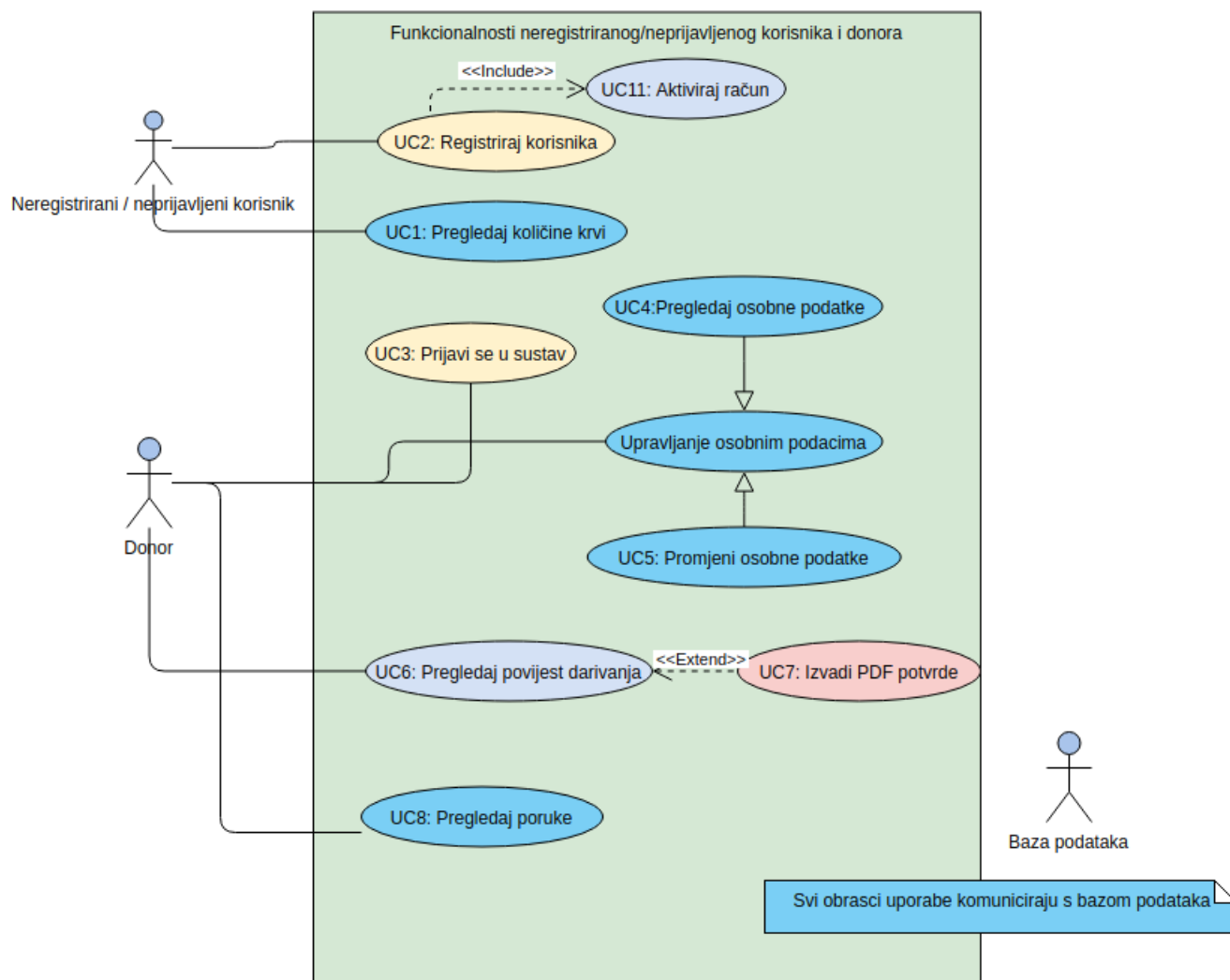
UC17 -Kreiraj kor. računa djelatnika banke

- **Glavni sudionik:** Administrator
- **Cilj:** Kreirati računa djelatnika banke
- **Sudionici:**Baza podataka
- **Preduvjet:**Administrator je prijavljen
- **Opis osnovnog tijeka:**
 1. Administrator odabire opciju "Kreiraj korisnički račun djelatnika banke"
 2. Sustav prikazuje formu za kreiranje računa koju popunjava djelatnik
 3. Odabire gumb "Kreiraj račun"
 4. Baza podataka se ažurira
 5. Sustav šalje aktivacijski link na email djelatnika banke
- **Opis mogućih odstupanja:**
 - 2.a Sva polja nisu ispunjena
 1. Sustav obavještava administratora da unese sve podatke

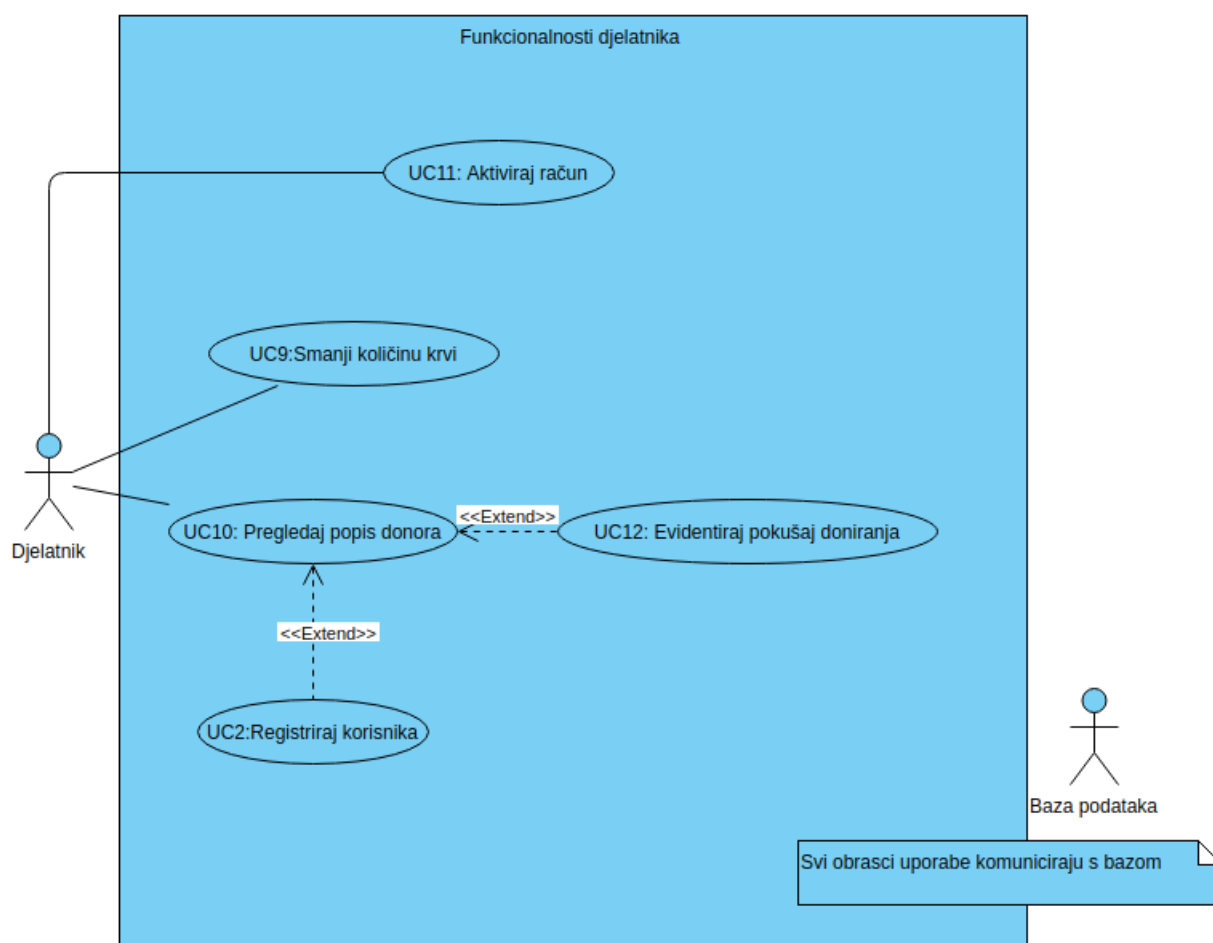
UC18 -Pošalji notifikacije nakon 3/4 mjeseca

- **Glavni sudionik:** Sustav za automatske poslove
- **Cilj:** Poslati notifikaciju donoru
- **Sudionici:**Baza podataka
- **Preduvjet:**Prošlo 3 mjeseca od uspješnog darivanja
- **Opis osnovnog tijeka:**
 1. Sustav svakodnevno provjerava uspješna doniranja od prije 3 mjeseca za muškarce i od prije 4 mjeseca za žene
 2. Svima koji zadovoljavaju uvjet šalju se email notifikacije da mogu doći ponovno donirati krv
 3. Svima koji zadovoljavaju uvjet šalju se poruke u njihov pretinac poruka da mogu doći ponovno donirati krv

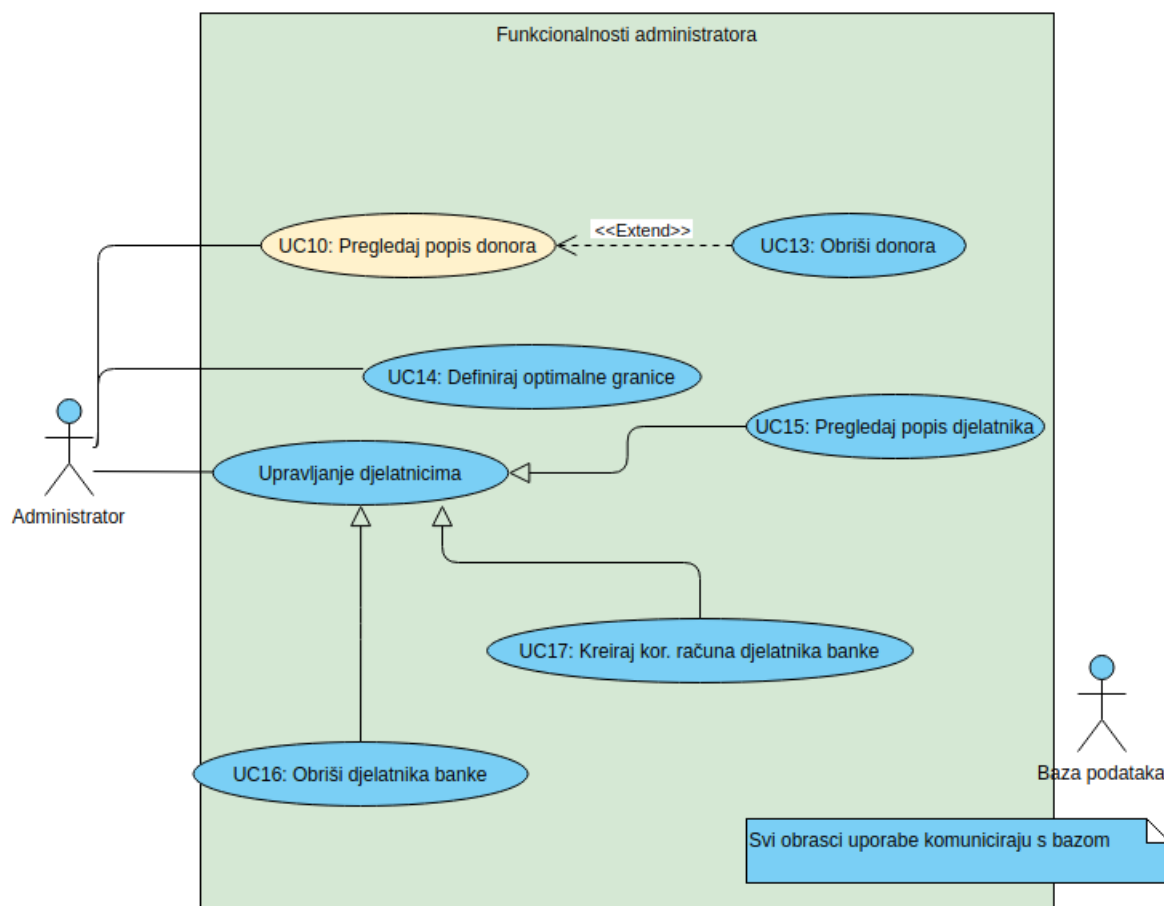
Dijagrami obrazaca uporabe



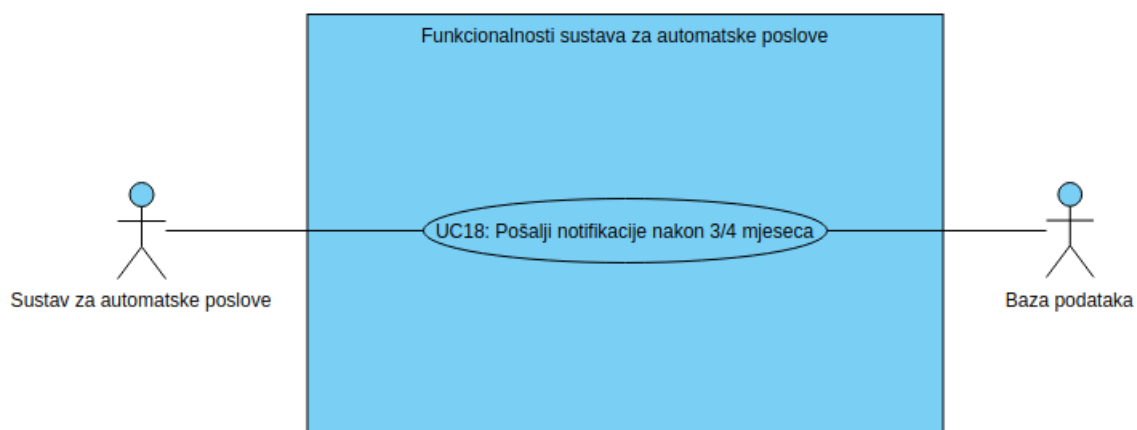
Slika 3.1: Dijagram obrasca uporabe, funkcionalnost neregistriranog / neprijavljenog korisnika i donora



Slika 3.2: Dijagram obrasca uporabe, funkcionalnost djelatnika



Slika 3.3: Dijagram obrasca uporabe, funkcionalnost administratora

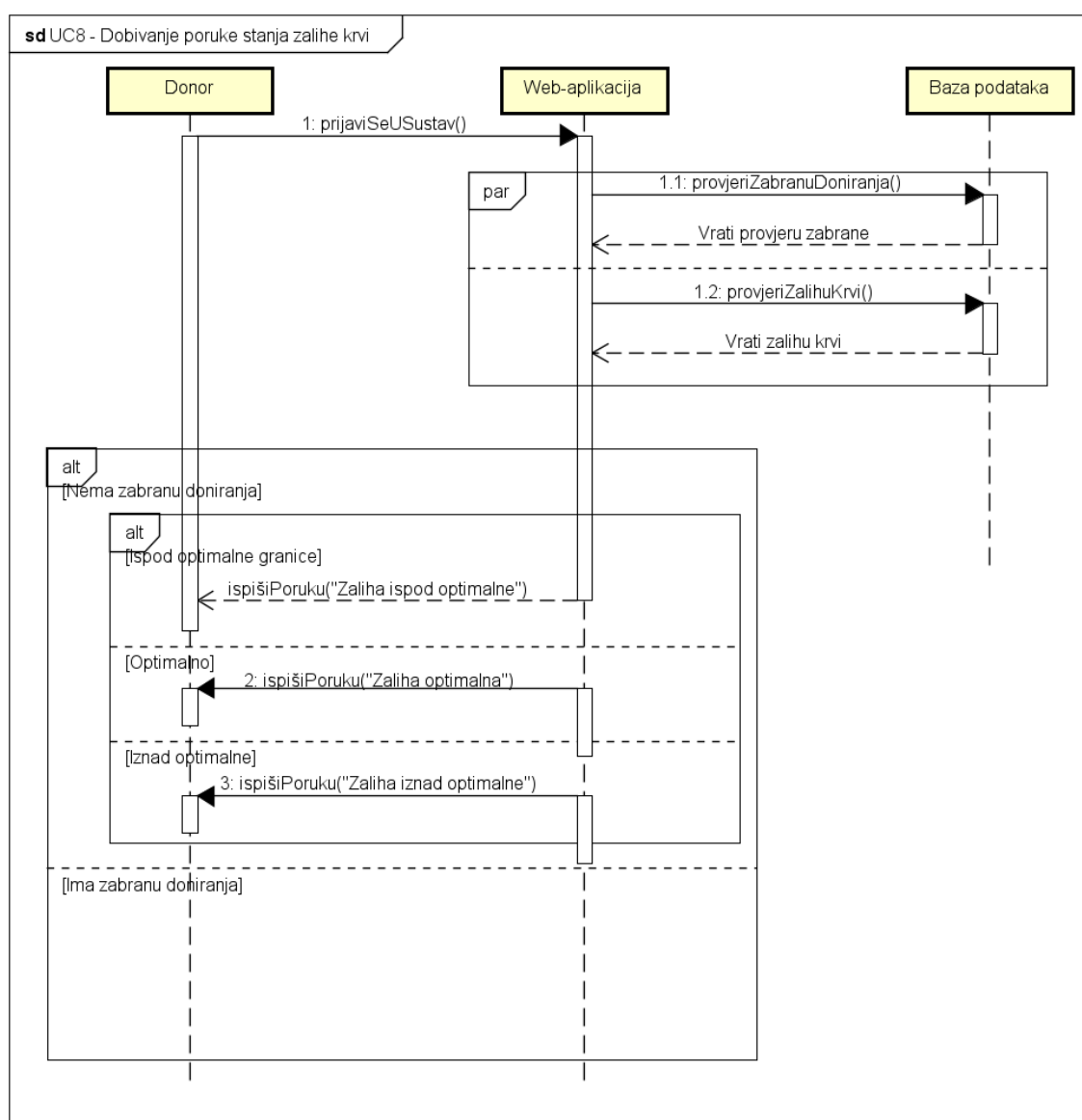


Slika 3.4: Dijagram obrasca uporabe, funkcionalnost sustava za automatske poslove

3.1.2 Sekvencijski dijagrami

Obrazac uporabe UC8 - Dobivanje poruke stanja zalihe krvi

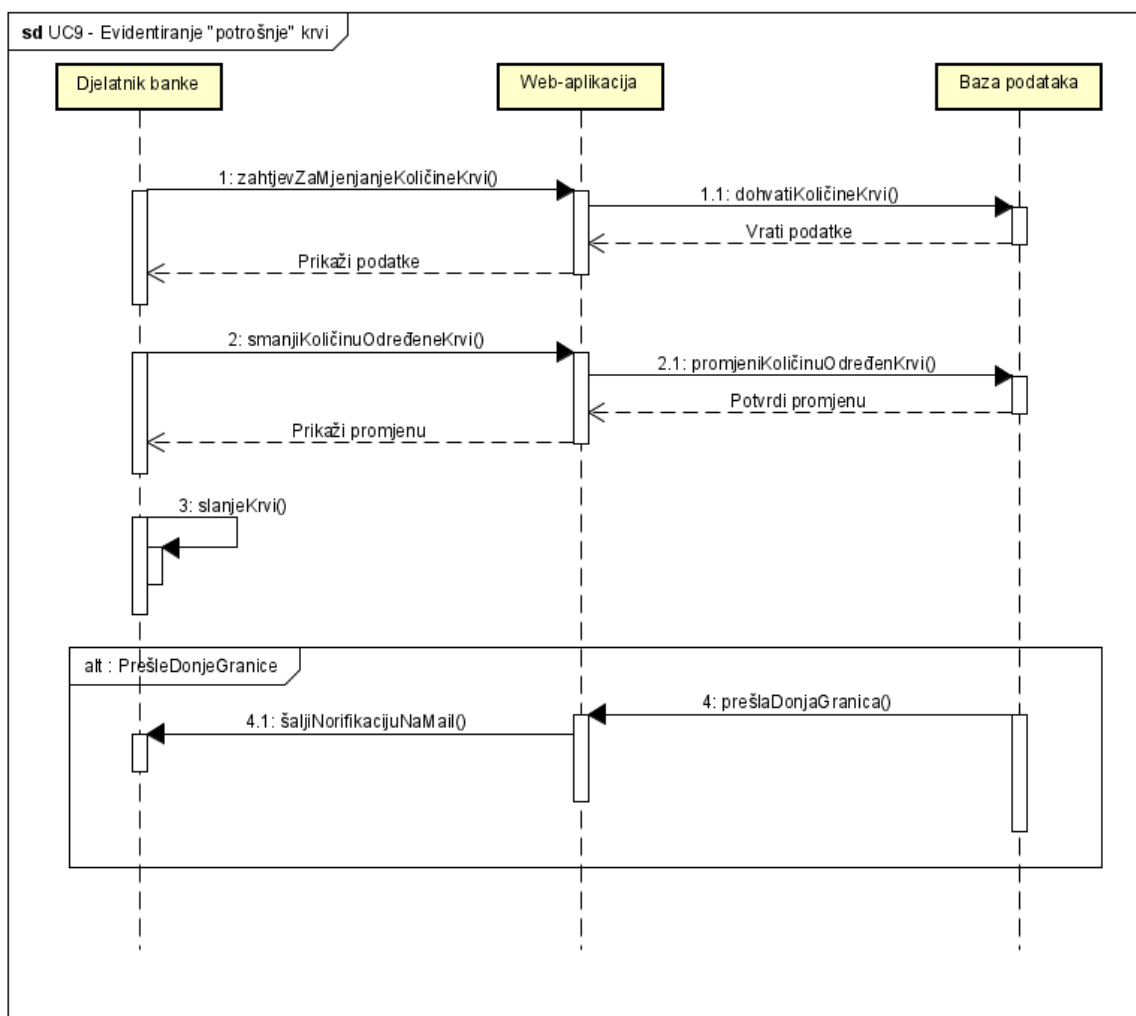
Donor, koji nema trajnu zabranu darivanja krvi, kod svakog spajanja u sustav dobiva poruku u ovisnosti o trenutnom stanju zaliha krvi. Poslužitelj dohvaća podatke iz baze podataka i prema zadanim granicama vraća jednu od tri poruke. Poruke mogu biti: stanje zaliha ispod optimalne granice; stanje zaliha je optimalno; stanje zaliha je iznad gornje optimalne granice.



Slika 3.5: Sekvencijski dijagram za UC8

Obrazac uporabe UC9 - Evidentiranje "potrošnje" krvi

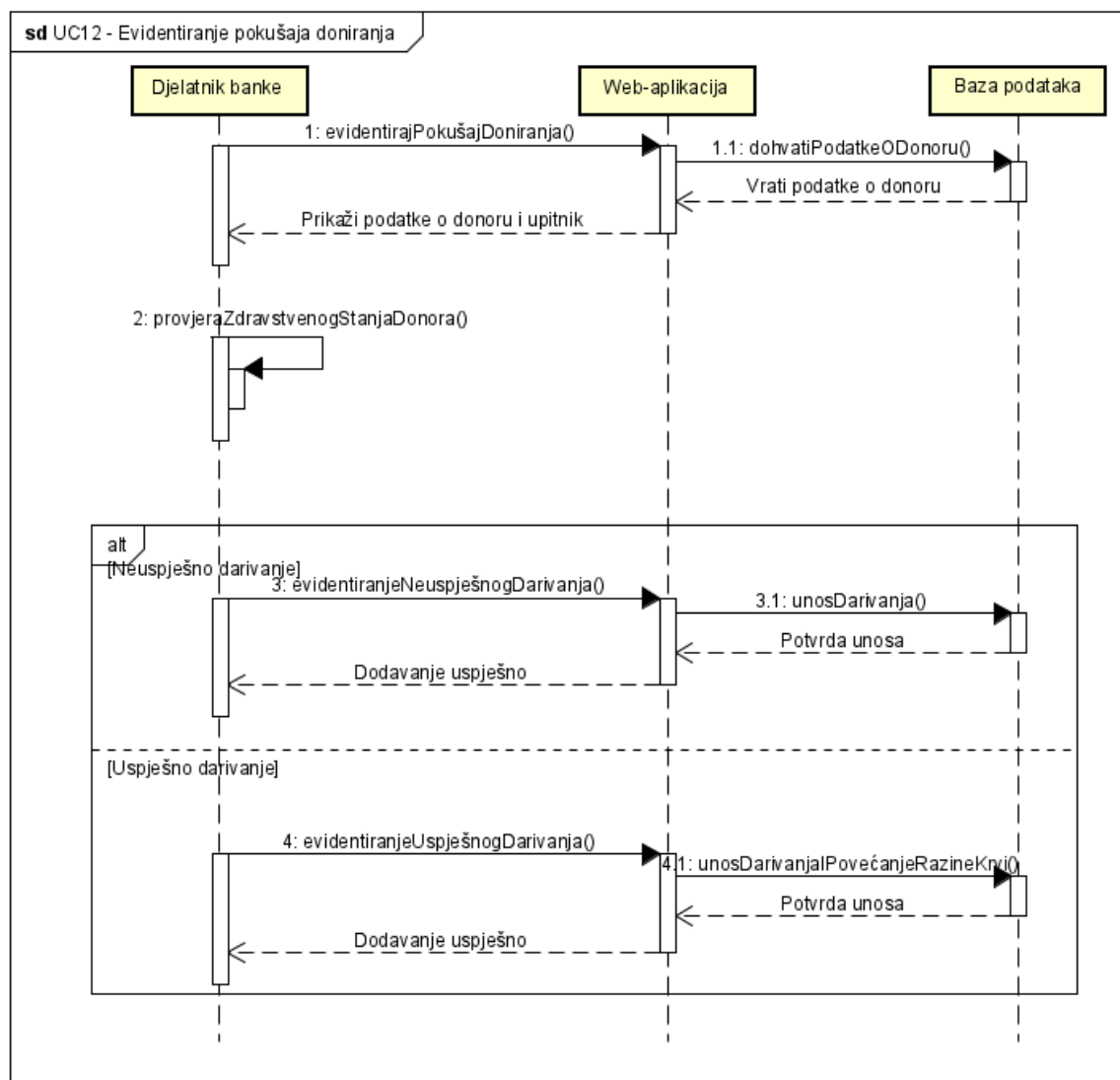
Djelatnik banke može evidentirati "potrošnju" krvi odnosno slanje određenog broja jedinica krvi u vanjsku instituciju. U bazi podataka se smanjuje količina određene vrste krvi i ako se prešla donja granica šalje se mail djelatniku.



Slika 3.6: Sekvencijski dijagram za UC9

Obrazac uporabe UC12 - Evidentiranje pokušaja doniranja

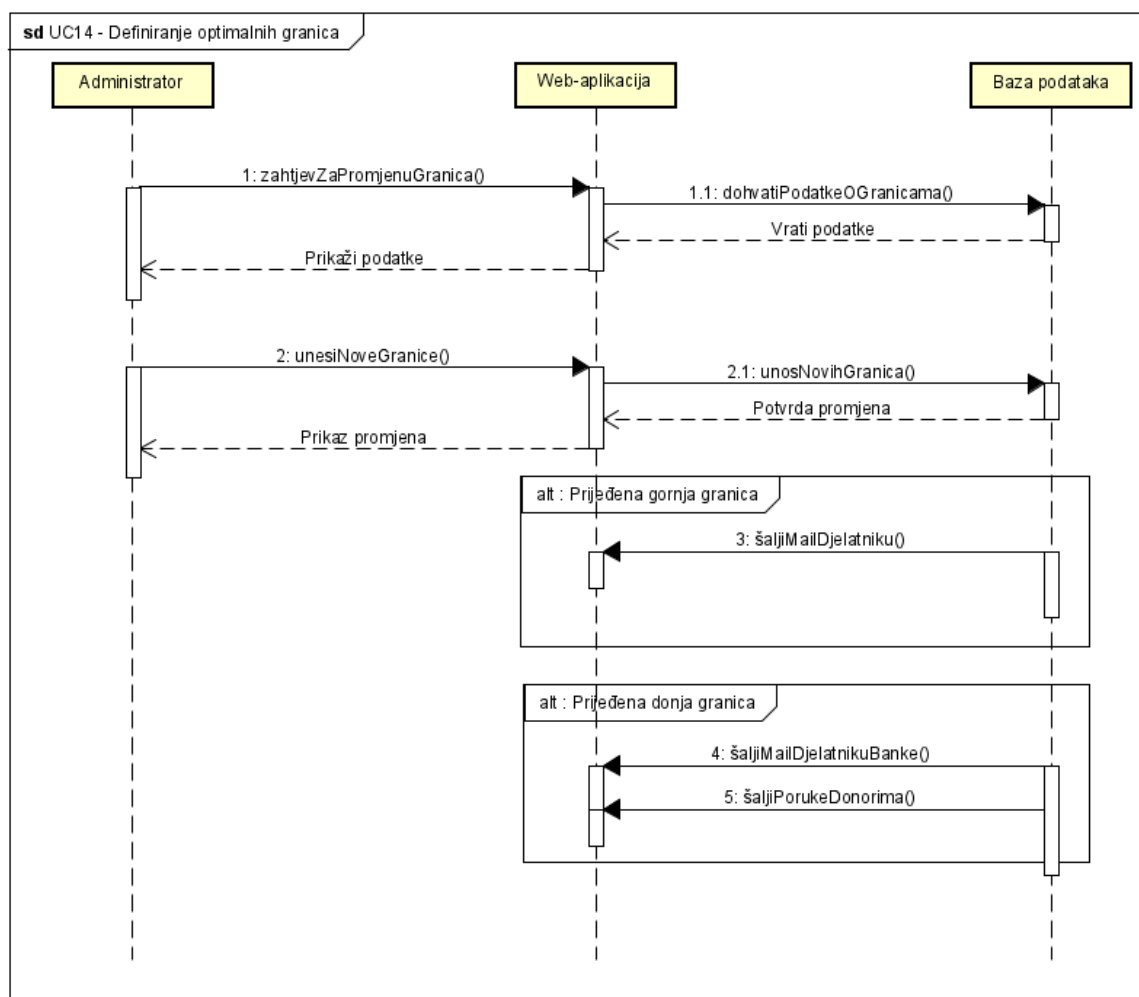
Djelatnik banke evidentira i uspješne i neuspješne pokušaje doniranja. Na temelju zdravstvenog stanja donor može pristupiti doniranju, ali može i biti privremeno ili trajno odbijen. Baza podataka se ažurira, zapisuje se pokušaj doniranja krvi i ako je uspješan poveća se razina određene vrste krvi.



Slika 3.7: Sekvencijski dijagram za UC12

Obrazac uporabe UC14 - Definiranje optimalnih granica

Za svaku krvnu grupu administrator sustava definira gornju i donju granicu optimalne količine. Administratoru se prikazuju trenutne granice za pojedinu vrstu krvi. Zatim on ih može mijenjati i promjene se spremaju i bazu podataka. Ako je prijeđena gornja granica šalje se mail djelatniku. Ako je prijeđena donja granica šalje se mail djelatniku i poruke donorima.



Slika 3.8: Sekvencijski dijagram za UC14

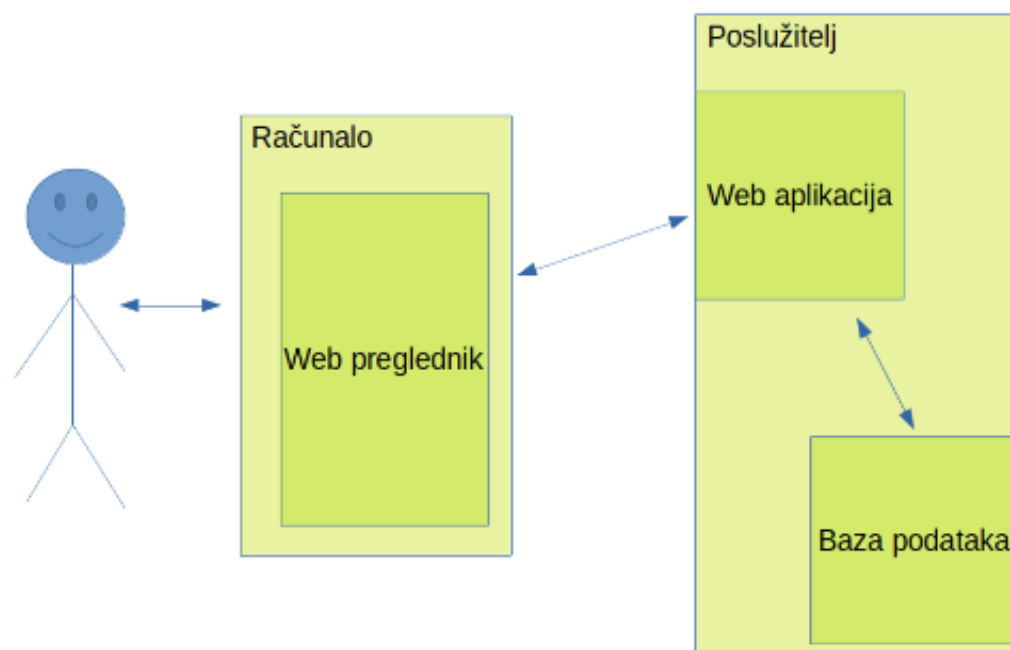
3.2 Ostali zahtjevi

- Sustav treba biti implementiran kao web aplikacija koja je prilagođena različitim veličinama ekrana. (responsive web design)
- Sustavu trebaju moći pristupati tri vrste korisnika (administrator, djelatnik banke i donor). Svaki korisnik se treba ovjeravati korisničkim imenom i lozinkom.
- Donori trebaju imati mogućnost sami kreirati svoj korisnički račun kao i djelatnici banke u slučaju da donor sam nije napravio korisnički račun prije prvog darivanja.
- Administratori trebaju imati mogućnost kreiranja korisničkog računa za djelatnike banke.
- Novi korisnici trebaju dobiti aktivacijski link na svoj mail, donori uz aktivacijski link trebaju dobiti i donorId koji će koristiti kao korisničko ime.
- Prilikom aktivacije korisničkog računa, korisnici trebaju imati mogućnost odabrati svoju lozinku.
- Sustav treba omogućiti rad više korisnika u stvarnom vremenu.
- Neispravno korištenje sustava ne smije narušiti rad sustava.
- Korisničko sučelje treba biti intuitivno, tako da se korisnici mogu koristiti sustavom bez dodatnih uputa.
- Sustav treba podržavati hrvatsku abecedu pri unosu i prikazu tekstualnog sadržaja.
- Veza s bazom podataka mora biti zaštićena i brza.

4. Arhitektura i dizajn sustava

Arhitekturu našeg sustava možemo podijeliti na tri podsustava:

- *Web preglednik*
- *Baza podataka*
- *Web poslužitelj*



Slika 4.1: Arhitektura sustava

Web preglednik je program koji korisniku omogućuje pregled web-stranica i multimedijalnih sadržaja vezanih uz njih. Korisnik putem web preglednika šalje zahtjeve na obradu poslužitelju.

Baza podataka je zbirka zapisa pohranjenih u računalu na sustavan način, tako da joj se računalni program može obratiti prilikom odgovaranja na problem. Web poslužitelj komunicira s bazom podataka te povlači potrebne zapise iz nje.

Web poslužitelj je srce našeg sustava. Njegova zadaća je komunikacija klijenta s aplikacijom te bazom podataka. Pri komunikaciji koristi HTTP protokol. Korisnik i aplikacija razmjenjuju HTTP zahtjeve (eng. HTTP request) i HTTP odgovore (HTTP response). Radi jednostavnosti, baza podataka je također smještena na poslužitelju.

Korisnik kroz grafičko sučelje, odnosno frontend, šalje zahtjeve na REST pristupne točke backenda. Tada stražnji kraj procesira zahtjev i ako je potrebno komunicira s bazom podataka. Nakon konstrukcije, stražnji kraj šalje odgovor prednjem kraju u obliku JSON objekta, a prednji kraj procesira odgovor i promjene prikazuje korisniku u obliku HTML stranice. Za aplikaciju je odabrana višeslojna arhitektura temeljena na **MVC** (Model - View - Controller) arhitekturnom stilu te uslužnoj arhitekturi. Podjela slojeva možemo napraviti na idući način:



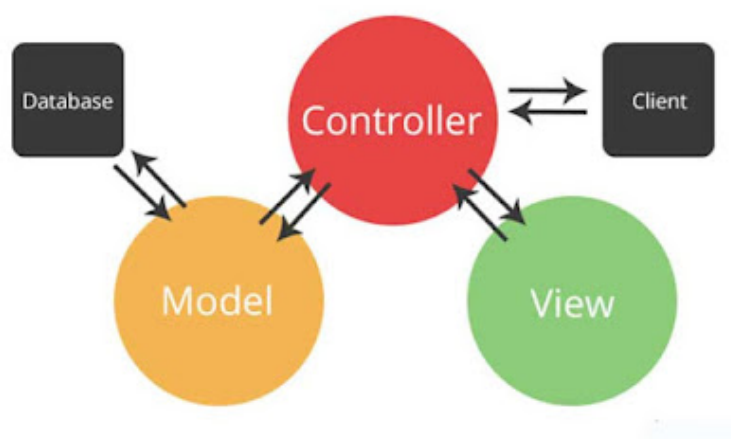
Slika 4.2: Podjela slojeva

- *sloj korisničke strane* - korisničko sučelje
- *sloj nadglednika* - REST nadglednici
- *sloj domene* - model podataka iz domene primjene
- *sloj za pristup podacima* - posrednik između sloja domene i baze podataka
- *sloj baze podataka* - pohrana podataka

Ovakva arhitektura odabrana je zbog poželjnih svojstava MVC arhitekturnog stila i višeslojne arhitekture: razvoj pojedinih slojeva je jednostavniji i u velikom stupnju nezavisan od razvoja drugih slojeva. Također komunikacija prednjeg i stražnjeg kraja ostvarena je primjenom REST arhitekturnog stila. Zbog toga su i prednji i stražnji kraj neovisno o jeziku implementacije, što potiče ponovnu uporabu.

Model-View-Controller se sastoji od:

- **Model** je centralni dio aplikacije, koja obuhvaća promjenljivu (dinamičku) strukturu podataka, direktno upravljanje podacima, logikom i pravilima aplikacije
- **View** je bilo koji izlazni prikaz podataka u korisničkom okruženju, pri čemu se isti podaci mogu prikazati na više načina
- **Controller** ulazne podatke pretvara u komande koje upravljaju modelom ili prikazom podataka u korisničkom okruženju



Slika 4.3: MVC model

4.1 Baza podataka

Za potrebe našeg sustava koristit ćemo relacijsku bazu podataka koja svojom strukturom olakšava modeliranje stvarnog svijeta. Gradivna jedinka baze je relacija, odnosno tablica koja je definirana svojim imenom i skupom atributa. Zadaća baze podataka je brza i jednostavna pohrana, izmjena i dohvat podataka za daljnju obradu. Baza podataka ove aplikacije sastoji se od sljedećih entiteta:

- korisnikAplikacije
- uloge
- pokusajDoniranja
- krvnaVrsta
- potrosnjaKrv
- zdravstveniPodaci
- doniranjeZdravljeOdgovori

4.1.1 Opis tablica

korisnikAplikacije • Ovaj entitet predstavlja sve korisnike aplikacije (admin, djelatnik, donor) i diferencira ih pomoću ulogaId što je referenca na tablicu uloga. Za admina i djelatnika postoje atributi : ime, prezime, oib, email dok su ostale vrijednosti null. Kod stvaranja atributa trajnoOdbijanjeDarivanja je false, a razlogOdbijanja je null ,ako se korisniku odbije darivanje zauvijek onda se ti atributi mijenjaju. One-to-many veza s pokusajDoniranja preko korisnikID dvaput, One-to-many veza s potrosnjaKrv preko korisnikId, many-to-one veza s uloge preko ulogaId i many-to-one veza s krvnaVrsta preko krvId.

korisnikAplikacije		
korisnikId	VARCHAR	id pomoću kojeg se korisnik prijavljuje u sustav (jednako donorId za donore)
lozinka	VARCHAR	hash korisničke lozinke
ime	VARCHAR	ime korisnika
prezime	VARCHAR	prezime korisnika
mjestoRođenja	VARCHAR	mjesto rođenja (nullable)
oib	VARCHAR	oib korisnika
adresaStanovanja	VARCHAR	adresa stanovanja (nullable)
mjestoZaposlenja	VARCHAR	firma u kojoj je korisnik zaposlen (nullable)
email	VARCHAR	email na koji korisniku dolaze korisne informacije (nullable)
brojMobitelaPrivatni	VARCHAR	privatni broj korisnika (nullable)
brojMobitelaPoslovni	VARCHAR	broj na koji korisniku dolaze korisne infomacije poslovni(nullable)
datumRođenja	DATE	datum rođenja
krvId	INT	vrsta krvi donora

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

korisnikAplikacije		
trajnoOdbijenoDarivanje	BOOLEAN	true ako je korisniku zabranjeno trajno darivanje, false ako nije
ulogaId	BIGINT	oznaka uloge korisnika
aktivacijskiKljuc	VARCHAR(10)	aktivacijski ključ koji služi za aktivaciju računa

uloge • entitet koji sadrži dva atributa, id za oznaku rednog broja uloge i ulogaName za ime uloge One-to-many veza s korisnikAplikacije preko ulogaId.

Uloge		
ulogaId	BIGINT	označava id uloge
ulogaName	VARCHAR	naziv uloge

pokusajDonacije• -entitet koji predstavlja pokušaj doniranja, sadrži podatke o datumu, mjestu, darivatelju, djelatniku, uspješnosti i razlogu odbijanja ako je uspješnost false. Many-to-one veza preko korisnikIdDjelatnika s korisnikAplikacije, Many-to-one veza preko korisnikIdDonora s korisnikAplikacije, one-to-many veza s doniranjeZdravljeOdgovori reko brDoniranja.

pokusajDonacije		
brDoniranja	BIGINT	redni broj donacije
datum	DATE	datum donacije
mjestoDarivanja	VARCHAR	opisno mjesto donacije
korisnikIdDjelatnika	VARCHAR	identifikator djelatnika
korisnikId	VARCHAR	identifikator donora
uspjeh	BOOLEAN	oznaka uspješnosti darivanja

krvnaVrsta•

- entitet čuva podatke o zalihi i predviđenim gornjim i donjim granicama za sve krvne grupe. one-to-many veza s potrosnjaKrvu preko krvId i one-to-many veza s korisnikAplikacije preko krvId.

krvnaVrsta		
krvId	SERIAL	identifikator krvne grupe
imeKrvneGrupe	VARCHAR	naziv krvne grupe
gornjaGranica	INT	gornja granica dopuštene količine krvi u jedinicama
donjaGranica	INT	donja granica dopuštene količine krvi u jedinicama
trenutnaZaliha	INT	trenutna zaliha konkretne krvne grupe u jedinicama

potrosnjaKrv

entitet koji prati isporuke krvi . Many-to-one veza s korisnikAplikacije preko korisnikIdDjelatnika i Many-to-one veza s krvnaVrsta preko krvId.

potrosnjaKrv		
idPotrosnje	SERIAL	redni broj isporuke krvi
timestampPotrosnje	TIMESTAMP	timestamp potrošnje
krvId	INT	id krvne grupe
količinaJedinica	INT	broj jedinica koje su se potrošili
korisnikIdDjelatnika	VARCHAR	identifikator djelatnika koji je inicirao slanje krvi bolnici
lokacijaPotrosnje	VARCHAR	opisna lokacija kamo ide isporuka krvi

zdravstveniPodaci

entitet koji sprema sve moguće zdravstvene podatke koji se pitaju korisnika u upitniku. Oni sadrže svoj id, koji se sam generira u tablici, opis zdravstvenog podatka i težinu kriterija odnosno, 0 ako se na temelju toga podatka trajno odbija darivanje i 1 ako se privremeno odbija, inače null. One-to-many veza s doniranjeZdravljeOdgovori preko idZdravstvenih.

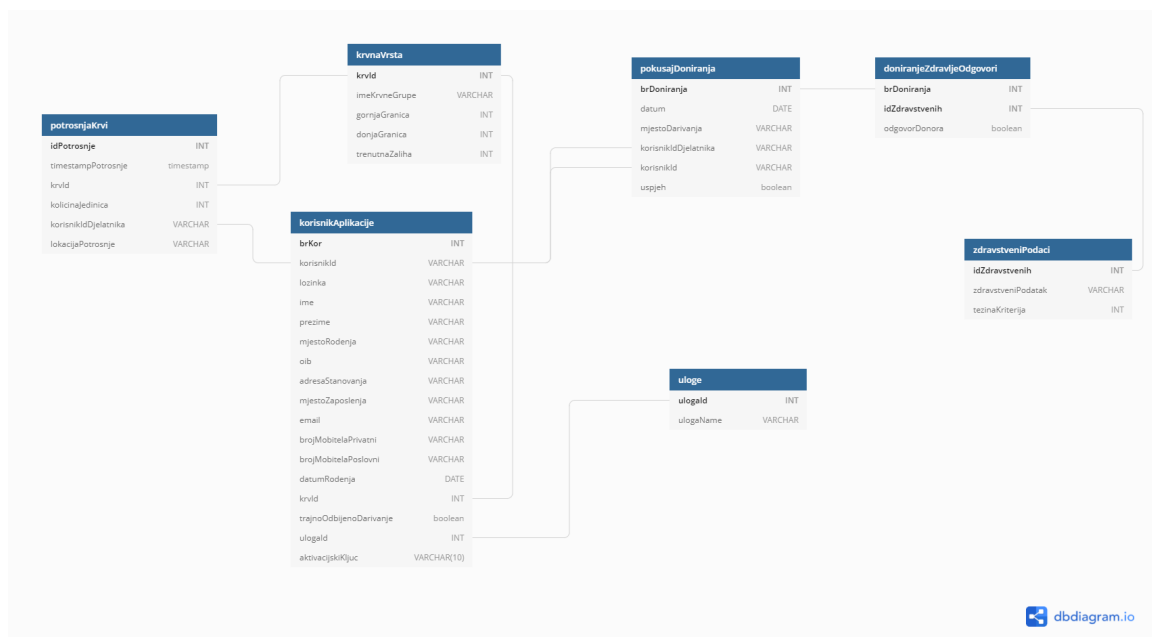
zdravstveniPodaci		
idZdravstvenih	SERIAL	id zdravstvenog podatka
zdravstveniPodatak	VARCHAR	opis zdravstvenog podatka koji se ispituje npr. osoba teži manje od 55 kg
tezinaKriterija	INT	tezina kriterija (0 za trajno, 1 za privremeno)

doniranjeZdravljeOdgovori•

entitet koji služi kao spremište odgovora korisnika koji su došli donirati krv. Many-to-one veza s zdravstveniPodaci preko idZdravstvenih. Many-to-one veza s pokusajDoniranja preko brDoniranja.

doniranjeZdravljeOdgovori		
brDoniranja	INT	brDoniranja iz tablice pokusajDoniranja
idZdravstvenih	INT	idZdravstvenih iz tablice zdravstveni podaci
odgovorDonora	BOOLEAN	odgovor donora na zdravstveni podatak pod idZdravstvenih

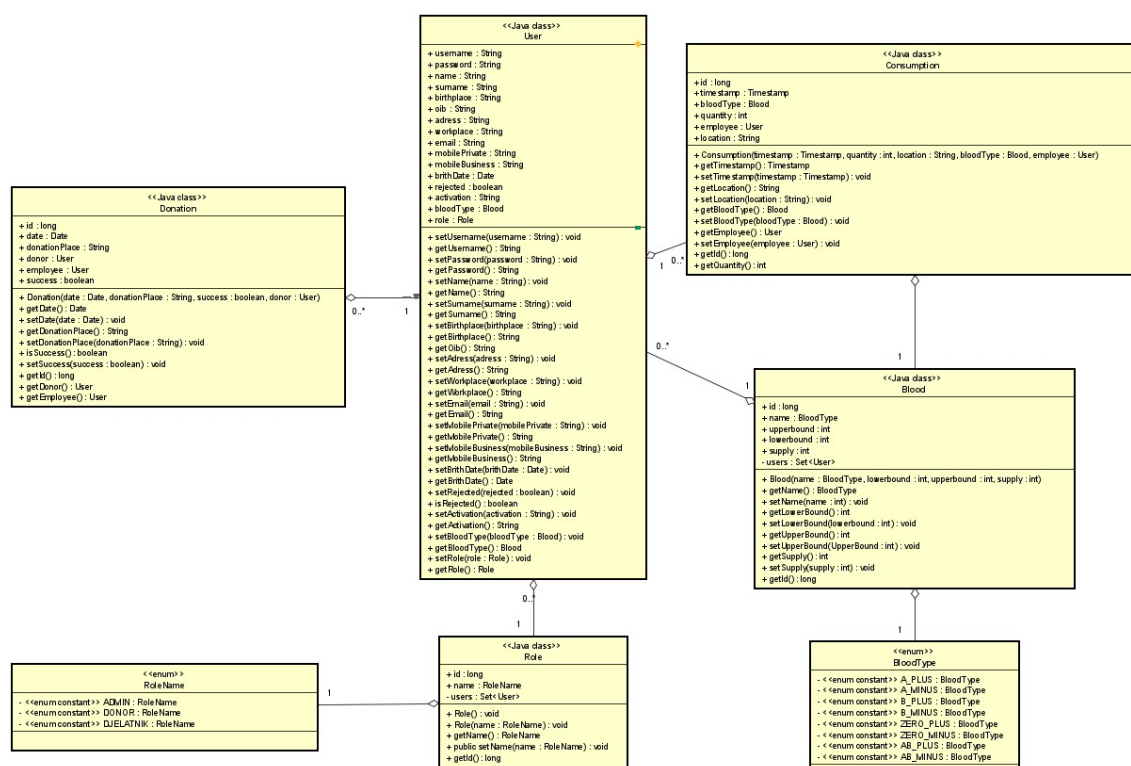
4.1.2 Dijagram baze podataka



Slika 4.4: relacijski model baze podataka

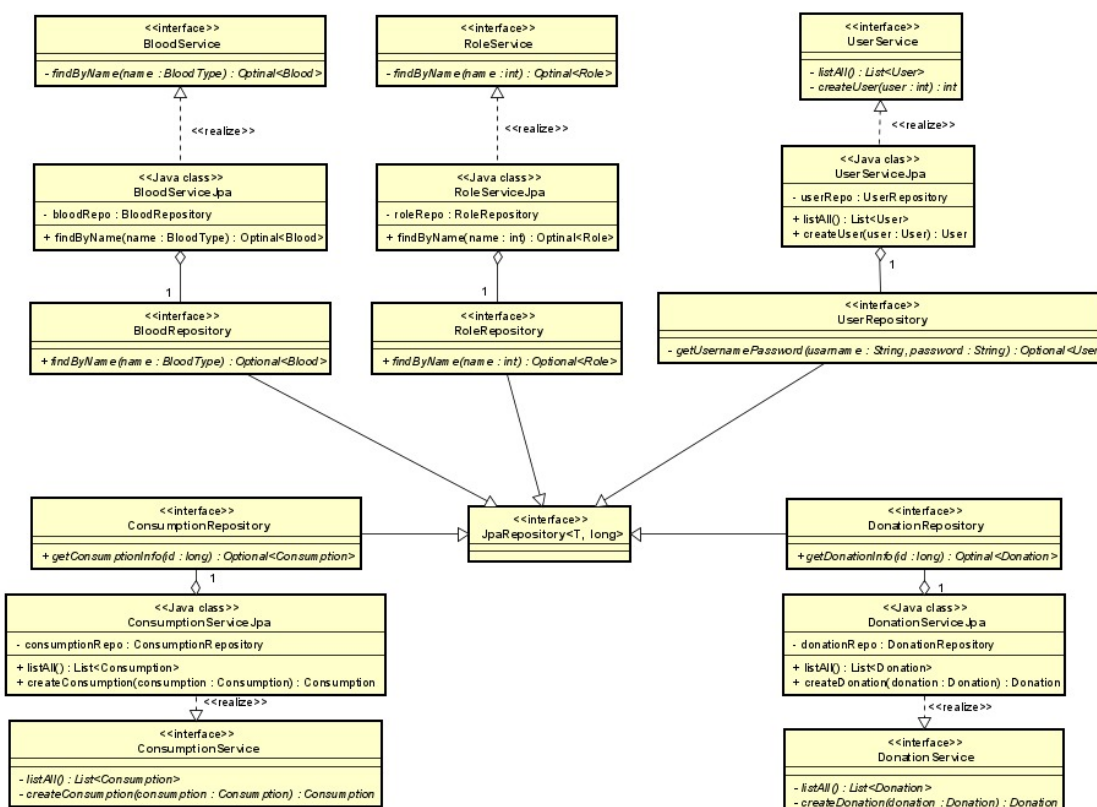
4.2 Dijagram razreda

Dijagram razreda pokazuje odnose između različitih objekata, te njihove atribute i operacije kojima vladaju. Na slikama 4.5., 4.6, 4.7. prikazani su razredi koji pripadaju *backend* dijelu naše arhitekture. Radi jednostavnosti, dijagram razreda je podijeljen u više slika, no bez obzira na to, prikazani razredi na neki način komuniciraju. Na idućoj slici prikazan je model podataka kojima backend rukuje. Korisnik aplikacije (administartor, djelatnik banke, donor) modeliran je razredom *User*. Razred *Blood* modelira krvne grupe. Razred *Consumption* modelira potrošnju krvi. Razred *Donation* modelira pokušaj donacije krvi, dok razred *Role* modelira 3 moguće uloge u aplikaciji. Također navedene su enumeracije za nazive krvnih grupa i nazive uloga.



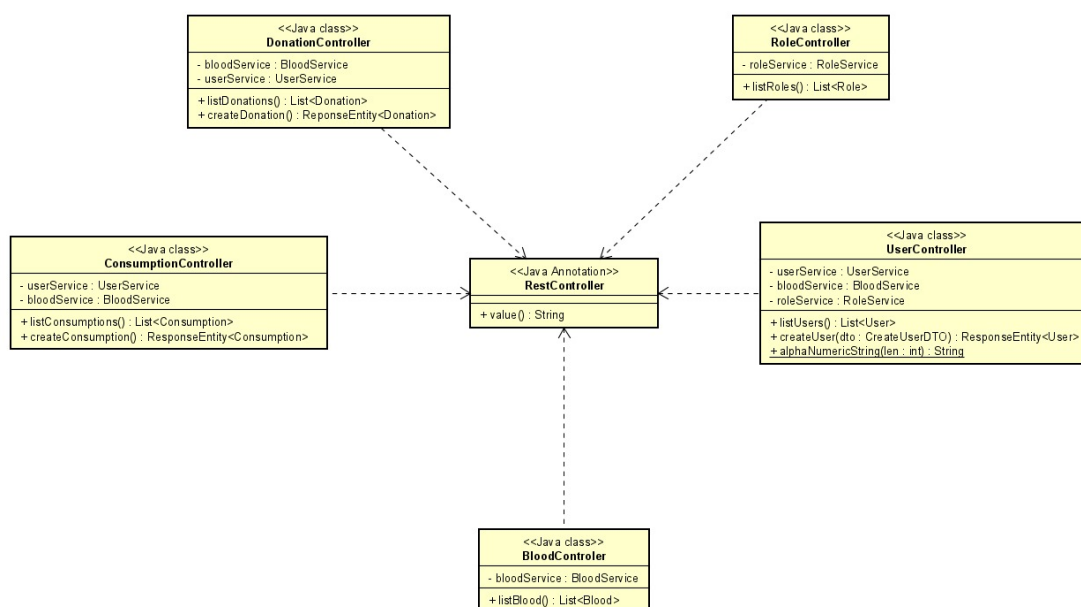
Slika 4.5: Dijagram razreda koji opisuje model

Na idućoj slici prikazana je sloj *backenda* koji je odgovoran za neposrednu komunikaciju s poslužiteljem baze podataka. Kao glavna komponenta na slici prikazano je sučelje `JpaRepository`, koje predstavlja apstraktni repozitorij podataka. Iz tog sučelja, izvedena su sučelja `UserRepository`, `RoleRepository`, `BloodRepository`, `ConsumptionRepository`, `DonationRepository`. Ta sučelja predstavljaju repozitorij podataka za prije navedene razrede modela, tj. oni predstavljaju poveznicu s bazom ili DAO (eng. Data Access Object). Također koriste se različite `ServiceJpa` klase koje koriste te repozitorije, te one implementiraju sučelje `ImeKlaseService`.



Slika 4.6: Dijagram razreda koji opisuju repozitorije

Na zadnjoj slici prikazani su *Controlleri* koji komuniciraju s vanjskim svijetom odnosno *frontendom* te upravljaju modelom podataka. Ovdje vidimo razrede *UserController*, *RoleController*, *BloodController*, *ConsumptionController* i *DonationController*. Svi ti razredi koriste Java anotaciju *RestController* koji predstavlja REST endpoint. Ti razredi su oni koji dobivaju zahtjeve iz vanjskog svijeta, a odgovaraju HTTP odgovorima i JSON objektima.



Slika 4.7: Dijagram razreda koji opisuju kontrolere

dio 2. revizije

Prilikom druge predaje projekta dijagram razreda i opisi moraju odgovarati stvarnom stanju implementacije

4.3 Dijagram stanja

dio 2. revizije

*Potrebno je priložiti dijagram stanja i opisati ga. Dovoljan je jedan dijagram stanja koji prikazuje **značajan dio funkcionalnosti** sustava. Na primjer, stanja korisničkog sučelja i tijekom korištenja neke ključne funkcionalnosti jesu značajan dio sustava, a registracija i prijava nisu.*

4.4 Dijagram aktivnosti

dio 2. revizije

Potrebno je priložiti dijagram aktivnosti s pripadajućim opisom. Dijagram aktivnosti treba prikazivati značajan dio sustava.

4.5 Dijagram komponenti

dio 2. revizije

Potrebno je priložiti dijagram komponenti s pripadajućim opisom. Dijagram komponenti treba prikazivati strukturu cijele aplikacije.

5. Implementacija i korisničko sučelje

5.1 Korištene tehnologije i alati

dio 2. revizije

*Detaljno navesti sve tehnologije i alate koji su primijenjeni pri izradi dokumentacije i aplikacije. Ukratko ih opisati, te navesti njihovo značenje i mjesto primjene. Za svaki navedeni alat i tehnologiju je potrebno **navesti internet poveznicu** gdje se mogu preuzeti ili više saznati o njima.*

5.2 Ispitivanje programskog rješenja

dio 2. revizije

U ovom poglavlju je potrebno opisati provedbu ispitivanja implementiranih funkcionalnosti na razini komponenti i na razini cijelog sustava s prikazom odabranih ispitnih slučajeva. Studenti trebaju ispitati temeljnu funkcionalnost i rubne uvjete.

5.2.1 Ispitivanje komponenti

*Potrebno je provesti ispitivanje jedinica (engl. unit testing) nad razredima koji implementiraju temeljne funkcionalnosti. Razraditi **minimalno 6 ispitnih slučajeva** u kojima će se ispitati redovni slučajevi, rubni uvjeti te izazivanje pogreške (engl. exception throwing). Poželjno je stvoriti i ispitni slučaj koji koristi funkcionalnosti koje nisu implementirane. Potrebno je priložiti izvorni kôd svih ispitnih slučajeva te prikaz rezultata izvođenja ispita u razvojnom okruženju (prolaz/pad ispita).*

5.2.2 Ispitivanje sustava

*Potrebno je provesti i opisati ispitivanje sustava koristeći radni okvir Selenium¹. Razraditi **minimalno 4 ispitna slučaja** u kojima će se ispitati redovni slučajevi, rubni uvjeti te poziv funkcionalnosti koja nije implementirana/izaziva pogrešku kako bi se vidjelo na koji način sustav reagira kada nešto nije u potpunosti ostvareno. Ispitni slučaj se treba sastojati od ulaza (npr. korisničko ime i lozinka), očekivanog izlaza ili rezultata, koraka ispitivanja i dobivenog izlaza ili rezultata.*

Izradu ispitnih slučajeva pomoću radnog okvira Selenium moguće je provesti pomoću jednog od sljedeća dva alata:

- *dodatak za preglednik **Selenium IDE** - snimanje korisnikovih akcija radi automatskog ponavljanja ispita*
- ***Selenium WebDriver** - podrška za pisanje ispita u jezicima Java, C#, PHP koristeći posebno programsko sučelje.*

Detalji o korištenju alata Selenium bit će prikazani na posebnom predavanju tijekom semestra.

¹<https://www.seleniumhq.org/>

5.3 Dijagram razmještaja

dio 2. revizije

*Potrebno je umetnuti **specifikacijski** dijagram razmještaja i opisati ga. Moguće je umjesto specifikacijskog dijagrama razmještaja umetnuti dijagram razmještaja instanci, pod uvjetom da taj dijagram bolje opisuje neki važniji dio sustava.*

5.4 Upute za puštanje u pogon

dio 2. revizije

*U ovom poglavlju potrebno je dati upute za puštanje u pogon (engl. deployment) ostvarene aplikacije. Na primjer, za web aplikacije, opisati postupak kojim se od izvornog kôda dolazi do potpuno postavljene baze podataka i poslužitelja koji odgovara na upite korisnika. Za mobilnu aplikaciju, postupak kojim se aplikacija izgradi, te postavi na neku od trgovina. Za stolnu (engl. desktop) aplikaciju, postupak kojim se aplikacija instalira na računalo. Ukoliko mobilne i stolne aplikacije komuniciraju s poslužiteljem i/ili bazom podataka, opisati i postupak njihovog postavljanja. Pri izradi uputa preporučuje se **naglasiti korake instalacije uporabom natuknica** te koristiti što je više moguće **slike ekrana** (engl. screenshots) kako bi upute bile jasne i jednostavne za slijediti.*

Dovršenu aplikaciju potrebno je pokrenuti na javno dostupnom poslužitelju. Studentima se preporuča korištenje neke od sljedećih besplatnih usluga: Amazon AWS, Microsoft Azure ili Heroku. Mobilne aplikacije trebaju biti objavljene na F-Droid, Google Play ili Amazon App trgovini.

6. Zaključak i budući rad

dio 2. revizije

U ovom poglavlju potrebno je napisati osvrt na vrijeme izrade projektnog zadatka, koji su tehnički izazovi prepoznati, jesu li riješeni ili kako bi mogli biti riješeni, koja su znanja stečena pri izradi projekta, koja bi znanja bila posebno potrebna za brže i kvalitetnije ostvarenje projekta i koje bi bile perspektive za nastavak rada u projektnoj grupi.

Potrebno je točno popisati funkcionalnosti koje nisu implementirane u ostvarenoj aplikaciji.

7. Popis literature

Kontinuirano osvježavanje

Popisati sve reference i literaturu koja je pomogla pri ostvarivanju projekta.

1. Programsko inženjerstvo, FER ZEMRIS, <http://www.fer.hr/predmet/proinz>
2. I. Sommerville, "Software engineering", 8th ed, Addison Wesley, 2007.
3. T.C.Lethbridge, R.Langaniere, "Object-Oriented Software Engineering", 2nd ed. McGraw-Hill, 2005.
4. I. Marsic, Software engineering book", Department of Electrical and Computer Engineering, Rutgers University, <http://www.ece.rutgers.edu/~marsic/books/SE>
5. The Unified Modeling Language, <https://www.uml-diagrams.org/>
6. Astah Community, <http://astah.net/editions/uml-new>

Indeks slika i dijagrama

2.1	Primjer javno dostupnih podataka o stanju banke krvi Hrvatskog zavoda za transfuzijsku medicinu	6
2.2	Primjer web sučelja logiranog korisnika Američkog crvenog križa .	7
3.1	Dijagram obrasca uporabe, funkcionalnost neregistriranog / neprijavljenog korisnika i donora	20
3.2	Dijagram obrasca uporabe, funkcionalnost djelatnika	21
3.3	Dijagram obrasca uporabe, funkcionalnost administratora	22
3.4	Dijagram obrasca uporabe, funkcionalnost sustava za automatske poslove	23
3.5	Sekvencijski dijagram za UC8	24
3.6	Sekvencijski dijagram za UC9	25
3.7	Sekvencijski dijagram za UC12	26
3.8	Sekvencijski dijagram za UC14	27
4.1	Arhitektura sustava	29
4.2	Podjela slojeva	30
4.3	MVC model	31
4.4	relacijski model baze podataka	37
4.5	Dijagram razreda koji opisuje model	38
4.6	Dijagram razreda koji opisuju repozitorije	39
4.7	Dijagram razreda koji opisuju kontrolere	40

Dodatak: Prikaz aktivnosti grupe

Dnevnik sastajanja

1. sastanak

- Datum: 14. listopada 2021.
- Prisustvovali: Svi
- Teme sastanka:
 - inicijalni dogovor s CROZ djelatnicima

2. sastanak

- Datum: 21. listopada 2021.
- Prisustvovali: Svi
- Teme sastanka:
 - dogovor oko podjele zadatka
 - podijeljeni zadaci iz dokumentacije na sedmero ljudi

3. sastanak

- Datum: 28. listopada 2021.
- Prisustvovali: Jurinić, Kerman
- Teme sastanka:
 - pripremljeni obrasci uporabe
 - podjela oko crtanja i opisivanja dijagrama

4. sastanak

- Datum: 05. studenoga 2021.
- Prisustvovali: Jurinić, Hudiček, Vugrinec, Šlezak, Kerman
- Teme sastanka:
 - klijentski dio aplikacije
 - dogovor oko poslužitelja

5. sastanak

- Datum: 11. studenoga 2021.
- Prisustvovali: Jurinić, Hudiček, Vugrinec, Šlezak, Okreša, Kerman
- Teme sastanka:
 - poslužiteljski dio aplikacije, dogovor oko dizajna baze podataka

6. sastanak

- Datum: 12. studenoga 2021.
- Prisustvovali: Kerman
- Teme sastanka:
 - konzultacije s asistentom i demosom oko obrazaca uporabe i baze podataka

7. sastanak

- Datum: 17. studenoga 2021.
- Prisustvovali: svi
- Teme sastanka:
 - demonstracija aplikacije asistentu i demosu

Tablica aktivnosti

Napomena: Doprinosi u aktivnostima treba navesti u satima po članovima grupe po aktivnosti.

	David Kerman	Ivan Jurinić	Matija Vugrinec	Jakov Šlezak	Matej Hudiček	Marko Okreša	Josip Pardon
Upravljanje projektom	4						
Opis projektnog zadatka							
Funkcionalni zahtjevi	1	1					
Opis pojedinih obrazaca	4	4					
Dijagram obrazaca	4	2					
Sekvencijski dijagrami					4		
Opis ostalih zahtjeva							
Arhitektura i dizajn sustava	4						
Baza podataka							
Dijagram razreda	1				2		
Dijagram stanja							
Dijagram aktivnosti							
Dijagram komponenti							
Korištene tehnologije i alati							
Ispitivanje programskog rješenja							
Dijagram razmještaja							
Upute za puštanje u pogon							
Dnevnik sastajanja							

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

	David Kerman	Ivan Jurinić	Matija Vugrinec	Jakov Šlezak	Matej Hudiček	Marko Okreša	Josip Pardon
Zaključak i budući rad							
Popis literature							
<i>Dodatne stavke kako ste podijelili izradu aplikacije</i>							
<i>npr. izrada početne stranice</i>							
<i>izrada baze podataka</i>							
<i>spajanje s bazom podataka</i>							
<i>back end</i>							

Dijagrami pregleda promjena

dio 2. revizije

Prenijeti dijagram pregleda promjena nad datotekama projekta. Potrebno je na kraju projekta generirane grafove s gitlaba prenijeti u ovo poglavlje dokumentacije. Dijagrami za vlastiti projekt se mogu preuzeti s gitlab.com stranice, u izborniku Repository, pritiskom na stavku Contributors.