

# Web sustav za upravljanje zalihama krvi postupcima višekriterijskog odlučivanja

---

**Dijanešić, Mihaela**

**Master's thesis / Diplomski rad**

**2020**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:200:220435>

*Rights / Prava:* [In copyright](#)/[Zaštićeno autorskim pravom.](#)

*Download date / Datum preuzimanja:* **2025-02-17**

*Repository / Repozitorij:*

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU**  
**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I**  
**INFORMACIJSKIH TEHNOLOGIJA**

**Sveučilišni diplomski studij**

**WEB SUSTAV ZA UPRAVLJANJE ZALIHAMA KRVI**  
**POSTUPCIMA VIŠEKRITERIJSKOG ODLUČIVANJA**

**Diplomski rad**

**Mihaela Dijanešić**

**Osijek, 2020.**

**FERIT**FAKULTET ELEKTROTEHNIKE RAČUNARSTVA  
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Obrazac D1: Obrazac za imenovanje Povjerenstva za diplomski ispit

Osijek, 19.09.2020.

Odboru za završne i diplomske ispite

**Imenovanje Povjerenstva za diplomski ispit**

<b>Ime i prezime studenta:</b>	Mihaela Dijanešić
<b>Studij, smjer:</b>	Diplomski sveučilišni studij Računarstvo
<b>Mat. br. studenta, godina upisa:</b>	D-978R, 23.09.2019.
<b>OIB studenta:</b>	45741389919
<b>Mentor:</b>	Prof.dr.sc. Goran Martinović
<b>Sumentor:</b>	
<b>Sumentor iz tvrtke:</b>	Marko Petric
<b>Predsjednik Povjerenstva:</b>	Izv. prof. dr. sc. Alfonzo Baumgartner
<b>Član Povjerenstva 1:</b>	Prof.dr.sc. Goran Martinović
<b>Član Povjerenstva 2:</b>	Izv. prof. dr. sc. Ivan Aleksić
<b>Naslov diplomskog rada:</b>	Web sustav za upravljanje zalihama krvi postupcima višekriterijskog odlučivanja
<b>Znanstvena grana rada:</b>	<b>Programsko inženjerstvo (zn. polje računarstvo)</b>
<b>Zadatak diplomskog rada:</b>	U diplomskom radu treba na temelju analize postojećih rješenja definirati funkcionalne i nefunkcionalne zahtjeve za razvoj web sustava za upravljanje zalihama krvi, uključujući stranu davatelja i stranu odjela transfuzije krvi. Uz to, treba predložiti prostorne i vremenske kriterije, te odgovarajući višekriterijski hijerarhijski postupak za upravljanje zalihama i njihovo optimiranje. Također, treba predložiti arhitekturu web sustava na strani poslužitelja i klijenta, te opisati potrebne tehnologije za izradu programskog rješenja (Vue.js, Node.js, MongoDB i drugi). U praktičnom dijelu rada navedeni web sustav treba programski ostvariti, ispitati i analizirati za prikladne ulazne podatke i načine korištenja. Tema rezervirana za: Mihaela Dijanešić Sumentor iz tvrtke: Marko Petric, Plava tvornica
<b>Prijedlog ocjene pismenog dijela ispita (diplomskog rada):</b>	Izvrstan (5)
<b>Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:</b>	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 2 bod/boda Jasnoća pismenog izražavanja: 3 bod/boda Razina samostalnosti: 3 razina
<b>Datum prijedloga ocjene mentora:</b>	19.09.2020.
Potpis mentora za predaju konačne verzije rada u Studentsku službu pri završetku studija:	Potpis:
	Datum:

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA  
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK**IZJAVA O ORIGINALNOSTI RADA**

Osijek, 30.09.2020.

Ime i prezime studenta:

Mihaela Dijanešić

Studij:

Diplomski sveučilišni studij Računarstvo

Mat. br. studenta, godina upisa:

D-978R, 23.09.2019.

Turnitin podudaranje [%]:

8

Ovom izjavom izjavljujem da je rad pod nazivom: **Web sustav za upravljanje zalihama krvi postupcima višekriterijskog odlučivanja**

izrađen pod vodstvom mentora Prof.dr.sc. Goran Martinović

i sumentora

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.

Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

# SADRŽAJ

1. UVOD .....	1
1.1. Zadatak diplomskog rada.....	2
2. PREGLED PODRUČJA UPRAVLJANJA ZALIHAMA KRVI S MEDICINSKOG I RAČUNALNOG GLEDIŠTA.....	3
2.1. Transfuzijska medicina.....	3
2.1.1. Općenito o transfuziji.....	3
2.1.2. Odjel za transfuzijsku medicinu.....	3
2.2. Postupak darivanja krvi .....	5
2.2.1. Dobrovoljni darivatelji krvi .....	5
2.2.2. Kriteriji za darivanje krvi.....	6
2.3. Glavni izazovi u području upravljanja zalihama krvi.....	7
2.3.1. Trenutno korišteni sustav upravljanja zalihama krvi .....	7
2.3.2 Prikaz i analiza postojećih rješenja za upravljanje zalihama krvi.....	7
2.3.2 Trenutno stanje u području postupaka višekriterijskog odlučivanja.....	8
3. IDEJNO RJEŠENJE I MODEL WEB SUSTAVA ZA UPRAVLJANJE ZALIHAMA KRVI10	
3.1. Idejno rješenje web sustava aplikacije.....	10
3.2. Funkcionalni i nefunkcionalni zahtjevi na web aplikaciju .....	10
3.3. Prikladni postupci višekriterijskog odlučivanja .....	12
3.4. Građa i komponente web sustava za upravljanje zalihama krvi.....	13
4. PROGRAMSKO RJEŠENJE WEB APLIKACIJE.....	14
4.1. Korišteni programski jezici, tehnologije i okoline .....	14
4.1.1. Korisnička strana.....	14
4.1.2. Poslužiteljska strana.....	15
4.1.3. Baza podataka .....	16
4.2. Programsko rješenje web sustava na strani klijenta.....	18
4.2.1. Registriranje i prijava.....	22
4.2.2. Profil korisnika.....	24
4.2.3. Stranica <i>Darivanja krvi</i> .....	26
4.2.4. Provjera mogućnosti darivanja krvi .....	28
4.3. Programsko rješenje web sustava na strani poslužitelja .....	29
4.3.1. Stranica <i>Svi donori</i> .....	32
4.3.2. Stranica <i>Nadži donore</i> .....	33

4.3.3. Stranica <i>Statistika</i> .....	38
5. PRIKAZ NAČINA RADA I ISPITIVANJE WEB SUSTAVA ZA UPRAVLJANJE ZALIHAMA KRVI.....	39
5.1. Opis načina rada web sustava za upravljanje zalihama krvi.....	39
5.2. Postavke ispitivanja, korisnički i ispitni slučajevi.....	39
5.3. Rezultati ispitivanja web sustava za upravljanje zalihama krvi s analizom .....	41
5.4. Ispitivanje i analiza korištenih postupaka višekriterijskog odlučivanja .....	43
5.4.1. Ispitivanje potraživanja krvi za pacijenta.....	44
5.4.2. Ispitivanje potraživanja krvi za zalihe.....	45
5.4.3. Analiza rezultata ispitivanja višekriterijskog postupka odlučivanja.....	45
6. ZAKLJUČAK .....	47
LITERATURA.....	48
SAŽETAK.....	50
ABSTRACT .....	51
PRILOZI (na CD-u) .....	53

# 1. UVOD

Transfuziologija je vrlo važna grana medicine čiji je cilj spašavanje života. Oslanja se na dobrovoljne darivatelje krvi zbog čega je potrebno pravilno upravljanje zalihama krvi i krvnih pripravaka kako se bi, po potrebi, brzo i učinkovito mogli pripremiti potrebni uzorci. Praćenje postojećih zaliha krvi, upravljanje njima i dobrovoljnim darivateljima najvažniji je dio aktivnosti odjela za transfuziju, a održavanje barem minimalnih zaliha krvi olakšava procese transfuzije. Kako bi se održavale zalihe krvi, potrebno je kontinuirano nadopunjavanje zaliha, odnosno potrebne su redovite donacije dobrovoljnih darivatelja krvi.

Cilj ovog diplomskog rada je modeliranje i programsko ostvarenje aplikacije za upravljanje zalihama krvi postupcima višekriterijskog odlučivanja. Višekriterijsko odlučivanje hijerarhijski je postupak na temelju čijih se kriterija dobivaju različiti rezultati, a ovise o vremenskim i prostornim kriterijima, kriterijima kompatibilnosti krvnih grupa i tražene količine i o namjeni, odnosno jesu li rezultati namijenjeni pacijentu ili zalihama krvi. Praktični dio rada programsko je rješenje aplikacije koja je razvijena pomoću Vue.js-a, Node.js-a i MongoDB baze podataka. Izrađena aplikacija namijenjena je odjelu za transfuzijsku medicinu i dobrovoljnim darivateljima krvi kako bi se pojednostavili poslovi vezani za transfuziologiju te se zbog toga sastoji od klijentske i administratorske strane.

U drugom poglavlju rada objašnjen je zadatak transfuzijske medicine, odjela transfuzijske medicine, postupak darivanja krvi te su navedeni kriteriji koje dobrovoljni darivatelji krvi moraju ispunjavati. Navedeni su izazovi u upravljanju zalihama krvi i prikazana postojeća rješenja upravljanja zalihama krvi. Treće poglavlje odnosi se na idejno rješenje u kojem je opisan rad aplikacije i dijelovi od kojih se treba sastojati, opisani su korisnički zahtjevi s klijentske i s administratorske strane te je detaljno objašnjen postupak višekriterijskog odlučivanja koji je bilo potrebno implementirati u aplikaciju. Također, u trećem poglavlju opisana je struktura aplikacije i navedene su i opisane komponente od kojih se aplikacija sastoji. U četvrtom poglavlju opisano je programsko rješenje klijentske i administratorske strane aplikacije, odnosno svih dijelova od kojih se navedene strane sastoje. Navedene su i korištene tehnologije, programski jezici, biblioteke i dodaci. Također, za svaki dio aplikacije prikazano je sučelje te je prikazan i objašnjen dio programskog koda koji se koristi u tom dijelu. U ovom poglavlju se također navode zahtjevi koje aplikacija šalje, pozadinski procesi i uloge korištenih biblioteka. Peto poglavlje sadrži opis načina rada aplikacije kao i upute za korištenje, testiranje i analizu rada aplikacije. Navedeni su testni kriteriji i slučajevi, opisano je izvršeno testiranje te su zatim prikazani rezultati postignuti navedenim testiranjem. Ovo poglavlje također sadrži detaljan prikaz i analizu rada višekriterijskog

odlučivanja prilikom traženja odgovarajućih darivatelja krvi. Posljednje poglavlje rada odnosi se na zaključak rada.

### **1.1. Zadatak diplomskog rada**

U diplomskom radu treba na temelju analize postojećih rješenja definirati funkcionalne i nefunkcionalne zahtjeve za razvoj web sustava za upravljanje zalihama krvi, uključujući stranu davatelja i stranu odjela transfuzije krvi. Uz to, treba predložiti prostorne i vremenske kriterije, te odgovarajući višekriterijski hijerarhijski postupak za upravljanje zalihama i njihovo optimiranje. Također, treba predložiti arhitekturu web sustava na strani poslužitelja i klijenta, te opisati potrebne tehnologije za izradu programskog rješenja (Vue.js, Node.js, MongoDB i drugi). U praktičnom dijelu rada navedeni web sustav treba programski ostvariti, ispitati i analizirati za prikladne ulazne podatka i načine korištenja.



## **2. PREGLED PODRUČJA UPRAVLJANJA ZALIHAMA KRVI S MEDICINSKOG I RAČUNALNOG GLEDIŠTA**

U ovom poglavlju objašnjen je rad transfuzijske medicine i odjela za transfuziju, odnosno način na koji se upravlja zalihama krvi i kako se koriste krvni pripravci. Objašnjeno je značenje darivatelja krvi te su navedeni kriteriji koje darivatelji moraju ispunjavati kao i razlozi zbog kojih se oni privremeno ili trajno odbijaju. Prikazana su neka postojeća rješenja i navedeni glavni izazovi kod upravljanja zalihama krvi.

### **2.1. Transfuzijska medicina**

Transfuzijska medicina (ili transfuziologija) grana je medicine koja obuhvaća sve aspekte transfuzije krvi i komponenti krvi. U sljedećim poglavljima navedene su opće informacije o transfuziji i radu odjela za transfuzijsku medicinu.

#### **2.1.1. Općenito o transfuziji**

Transfuzija, prema [1], uključuje aspekte povezane s hemovigilansom [2] koja uključuje izvještavanje i procjenu štetnih događaja i incidenta koji mogu utjecati na transfuzijske postupke kao i nadzorne aktivnosti obuhvaćajući cijeli transfuzijski lanac od prikupljanja krvi i komponenti krvi do praćenja primatelja. Ti postupci su, prema [3], namijenjeni za prikupljanje i pristup informacijama o neočekivanim ili nepoželjnim učincima koji proizlaze iz terapijske upotrebe labilnih krvnih proizvoda i za sprječavanje njihovog pojavljivanja i ponavljanja. Transfuziologija uključuje pitanja darivanja krvi, imunohematologije i drugih laboratorijskih ispitivanja na bolesti koje se prenose transfuzijom, proizvodnje krvnih pripravaka od ljudske krvi, upravljanje i praćenje kliničkih praksi transfuzije i upravljanje krvlju pacijenata, dijagnostike poremećaja hemostaze, liječenje bolesnika lijekovima proizvedenim od ljudske krvi, prikupljanje matičnih stanica, stanične terapije kao i velik broj potprocesa kao što su motiviranje darivatelja krvi, testiranja, distribucija krvnih pripravaka i slično.

#### **2.1.2. Odjel za transfuzijsku medicinu**

Kao što je navedeno u [4], odjel za transfuzijsku medicinu odgovoran je za prikupljanje i testiranje krvi koja će se davati pacijentima. Odjel je također odgovoran za prikupljanje i obradu hematopoetskih matičnih stanica za transplantaciju krvi i koštane srži, kao i ispitivanja koja su potrebna za transplantaciju organa. Odjel za transfuzijsku medicinu obavlja transfuzijsku djelatnost, a ona obuhvaća promidžbu i organizaciju darivanja, prikupljanja krvi i krvne plazme, obrađivanje plazme, osiguranje krvnih pripravaka i potreba za njima kao i lijekovima proizvedenim iz krvi, osiguranje kvalitete, distribuciju i nadzor transfuzijskog liječenja.

### **2.1.2.1. Upravljanje zalihama krvi**

Hrvatski zavod za transfuzijsku medicinu, kao i svi transfuzijski centri, prema [5], trebaju sadržavati, uz ostale, prostore za čuvanje krvnih pripravaka, za elektroničku obradu podataka i za raspodjelu krvnih pripravaka. Krvni pripravci prije korištenja prolaze razna testiranja i na kraju se čuvaju u hladnjacima, ledenicama i slično kako bi se osiguralo kvalitetno čuvanje dnevno uzetih doza krvi i pripravaka. Zalihama krvi i krvnim pripravcima upravlja se sa središnjim računalnim sustavom koji je zaštićen od neovlaštenih pristupa, a služi za vođenje dokumentacije, naručivanje i izdavanje krvi i krvnih pripravaka. Sastoji se od terminala za unos proizvodnje krvnih pripravaka, terminala za finalizaciju i obilježavanje krvnih pripravaka, terminala za prijem narudžbi za krvne pripravke i terminala za izdavanje krvnih pripravaka. Svaki darivatelj krvi evidentiran je i svaka jedinica krvi obilježena je serijskim brojem kako bi se mogla voditi evidencija i klasifikacija. Ovlaštena zdravstvena ustanova ima dokumentaciju ili informacijski sustav koja osigurava da se svaka doza uzete krvi ili krvnog sastojka može povezati sa sustavom prikupljanja i prerade u kojem je prikupljena i prerađena.

### **2.1.2.2. Korištenje krvnih pripravaka**

Transfuzije krvnih pripravaka bitan su sastavni dio prakse hitne medicine [4]. Zbog situacija kao što su akutno traumatsko krvarenje sve do kroničnog gubitka krvi zahtijevaju se transfuzije za simptomatsku anemiju, pa su krvni pripravci i upravljanje njima važan alat svakog liječnika i bolničkih ustanova. Nakon prikupljanja, važno je pravilno klasificirati krv, a klasificira se po krvnim grupama i Rh-faktoru.

Dijelovi koji se izoliraju iz pune krvi, kao što je opisano u [6] i [7], navedeni su u nastavku teksta. To su crvene krvne stanice, trombociti i svježa krvna plazma.

Crvene krvne stanice koje se koriste klinički za povećanje hemoglobina i kapaciteta prenošenja kisika krvlju kod anemičnih pacijenata. Nabavljaju se iz uzoraka pune krvi, a iz jednog darovanog uzorka mogu se pribaviti dvije prenosive jedinice crvenih krvnih stanica.

Trombociti koji se također izoliraju iz uzoraka pune krvi, a obično je potrebno više darivatelja kako bi se skupio dovoljan broj trombocita za transfuziju.

Svježa krvna plazma ima stabilan faktor zgrušavanja, albumin, krioprecipitatu i imunogloblin. Plazma se zamrzava i koristi po potrebi u nekom od navedenih oblika.

## **2.2. Postupak darivanja krvi**

Kao što je opisano u [8], darivanje krvi nastaje kada osoba dobrovoljno daruje krv koja se koristi za transfuziju ili za proizvodnju biofarmaceutskih lijekova postupkom koji se naziva frakcioniranje što je razdvajanje komponenti cjelovite krvi. Darivanje može biti darivanje pune krvi ili darivanje određenih komponenti izravno.

Postupak darivanja krvi, prema [9] i [10], sastoji se od tri koraka provjere zdravstvenog stanja darivatelja i provjere zadovoljavanja kriterija za darivanje krvi te od postupka darivanja krvi. Prvi korak je provjera količine hemoglobina u krvi iz kapljice krvi dobivene ubodom u jagodicu prsta. Provjera je važna kako darivatelj ne bi postao anemičan jer se darivanjem krvi smanjuje količina hemoglobina koji prenosi kisik u krvi što je glavni uzrok anemije. Drugi korak je kratak razgovor s liječnikom kako bi se provjerilo dosadašnje i trenutno zdravstveno stanje darivatelja nakon kojeg slijedi treći korak odnosno liječnički pregled. Liječnički pregled uključuje provjeru rada srca i provjeru krvnog tlaka. Nakon što darivatelj prođe navedena tri koraka, može započeti s postupkom darivanja krvi. Tijekom postupka darivanja krvi zdravstveni tehničar smješta darivatelja na krevet te u venu darivatelja uvodi iglu koja je povezana s plastičnom vrećicom u kojoj se prikuplja krv. Svaki darivatelj daruje 450 ml pune krvi, a darivanje traje od 8 do 12 minuta. Nakon darivanja krvi slijedi kratak odmor uz tekućinu i lagani obrok.

### **2.2.1. Dobrovoljni darivatelji krvi**

Dobrovoljni darivatelj krvi je, prema [11], osoba koja daruje krv, plazmu ili stanične dijelove krvi po svojoj slobodnoj volji i, u Hrvatskoj, ne prima za to nikakvu nadoknadu, ni novčanu niti na način koji se može smatrati nadomjestkom novca. Dobrovoljni darivatelj krvi je posredan, aktivan sudionik u liječenju bolesnika, to jest on je aktivna veza između zdravog dijela društva i bolesnika. Darivatelj krvi može biti svaki čovjek dobrog općeg zdravstvenog stanja koji zadovoljava kriterije za darivanje krvi. Kriteriji su navedeni u poglavlju 2.2.2.

Danas u svijetu većina darivatelja krvi su neplaćeni dobrovoljci koji doniraju krv za opskrbu u zajednici. U nekim je zemljama uspostavljena zaliha ograničena, a darivatelji obično doniraju krv kada je obitelji ili prijateljima potrebna transfuzija što nazivamo usmjereno darivanje. Mnogi darivatelji doniraju kao dobrotvorni čin, ali u zemljama koje dopuštaju plaćeno darivanje neki ljudi doniraju zbog novca ili pogodnosti poput plaćenog izostanka s posla.

### 2.2.2. Kriteriji za darivanje krvi

Prilikom svakog darivanja krvi, važno je identificirati osobu i provjeriti zadovoljava li sve potrebne kriterije. Osnovni kriteriji, kao što je navedeno u [12], su dobro opće zdravstveno stanje i kriteriji navedeni u nastavku (tablica 2.1).

Osobe mogu biti i trajno ili privremeno odbijene kod pokušaja darivanja krvi, a razlog tome je opasnost od ugrožavanja zdravlja darivatelja i/ili primatelja krvi.

*Tablica 2.1. Osnovni kriteriji za darivanje krvi*

<b>Dob</b>	- 18 do 65 godina - do 60 ako prvi put daruje krv - do 70 uz potvrdu liječnika
<b>Tjelesna težina</b>	- iznad 55 kg, proporcionalno visini
<b>Tjelesna temperatura</b>	- do 37°C
<b>Krvni tlak</b>	- sistolični od 100 do 180 mm Hg - dijastolični od 60 do 110 mm Hg
<b>Puls</b>	- od 50 do 100 otkucaja u minuti
<b>Hemoglobin</b>	- muškarci: minimalno 135 g/L - žene: minimalno 125 g/L
<b>Vremenski razmak od posljednjeg darivanja</b>	- muškarci: 3 mjeseca - žene: 4 mjeseca

Trajno se odbijaju osobe koje boluju ili su bolovala od teških kroničnih bolesti probavnog ili dišnog sustava, bolesti krvnih žila i srca, bolesti jetre, zloćudnih bolesti, AIDS-a, šećerne bolesti kao i duševni i živčani bolesnici. Kao što je opisano u [13], osobe koje boluju od raka, aritmije, epilepsije i nekih autoimunih bolesti trajno se odbijaju. Također, trajno se odbijaju osobe rizičnog ponašanja koje su izložene riziku dobivanja zaraznih bolesti koje se mogu prenijeti krvlju, a to su:

- ovisnici o alkoholu ili drogama
- muškarci koji su u životu imali spolne odnose s drugim muškarcima
- žene i muškarci koji su imali spolni odnos s prostitutkama
- promiskuitetne osobe
- osobe koje su uzele drogu intravenskim putem
- osobe koje su liječene zbog spolno prenosivih bolesti
- HIV-pozitivne osobe
- seksualni partneri ranije navedenih osoba

Privremeno se odbijaju osobe koje uzimaju lijekove jer se oni mogu prenijeti krvlju i kod primatelja izazvati neželjene posljedice, osobe koje u trenutku darivanja imaju simptome alergije, osobe koje su se u zadnjih 12 mjeseci tetovirale ili u zadnjih 6 mjeseci bušile uši ili druge dijelove tijela. Također, navedeno u [13], privremeno se odbijaju:

- osobe koje su konzumirale alkoholna pića unutar 8 sati prije darivanja krvi
- osobe s lakšim akutnim bolestima poput smanjenog željeza u krvi, prehlade, smetnje u probavnom sustavu i slično
- osobe koje su nedugo prije darivanja imale operativne zahvate, transfuziju ili primile cjepiva
- žene za vrijeme menstruacije, trudnoće i dojenja
- osobe koje obavljaju poslove u kojima se mogu ugroziti na dan darivanja (rad na visini, dubini i slično)

## **2.3. Glavni izazovi u području upravljanja zalihama krvi**

U ovom poglavlju opisan je trenutno korišteni sustav upravljanja zalihama krvi, navedena su i prikazana neka rješenja i navedeno je trenutno stanje u postupcima višekriterijskog odlučivanja.

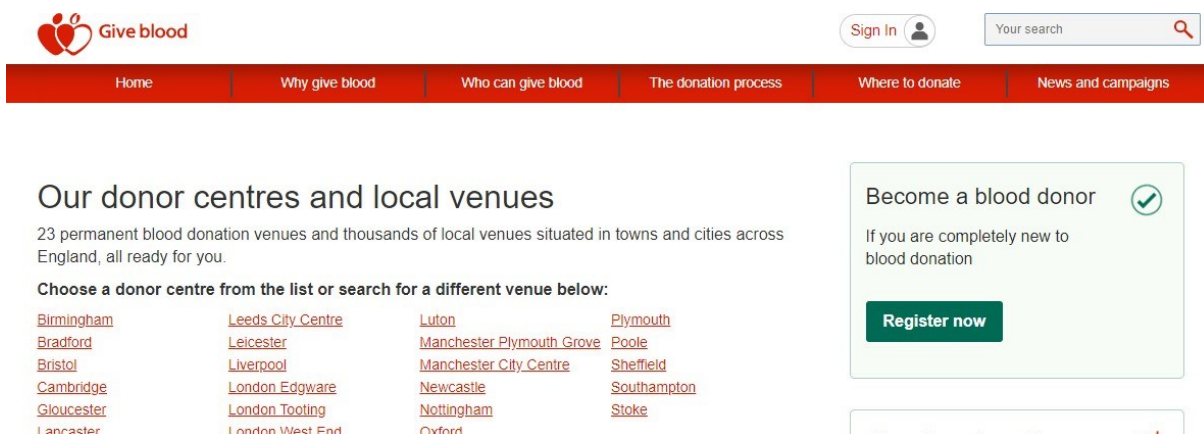
### **2.3.1. Trenutno korišteni sustav upravljanja zalihama krvi**

Prema [5], postojeći sustav koji koristi banka krvi je ručni sustav. Uz ručni sustav, postoje problemi u upravljanju evidencijama darivatelja. Ne može se sa sigurnošću voditi evidencija darivatelja jer je moguća ljudska pogreška. Osim toga što se mogu pojaviti pogreške kada osoblje vodi više od jedne evidencije za istog davatelja, ne postoji centralizirana baza podataka darivatelja volontera. Dakle, u slučaju nužde jedina je mogućnost ručno pretraživanje darivatelja i zatim upućivanje telefonskih poziva ili slanje pisma zamolbe svakom darivatelju. Svaka banka krvi ima svoju vlastitu bazu darivatelja pa ako darivatelj donira u drugoj bolnici, prethodna evidencija ne može se pratiti osim ako darivatelj donese knjižicu doniranja. Stoga se, ako daruje krv na novom mjestu, za darivatelja smatra da krv daruje prvi put. Bez automatiziranog sustava upravljanja postoje i problemi u praćenju stvarnih količina svake krvne grupe u banci krvi. Pojedinačni bolnički centri prate svoje zalihe i njih se može provjeriti na internetskim stranicama pojedinačnih bolničkih centara.

### **2.3.2 Prikaz i analiza postojećih rješenja za upravljanje zalihama krvi**

Trenutno ne postoje aplikacije s mogućnosti registriranja i prijave korisnika, praćenja darivanja, ugovaranja termina darivanja i slično na hrvatskom jeziku.

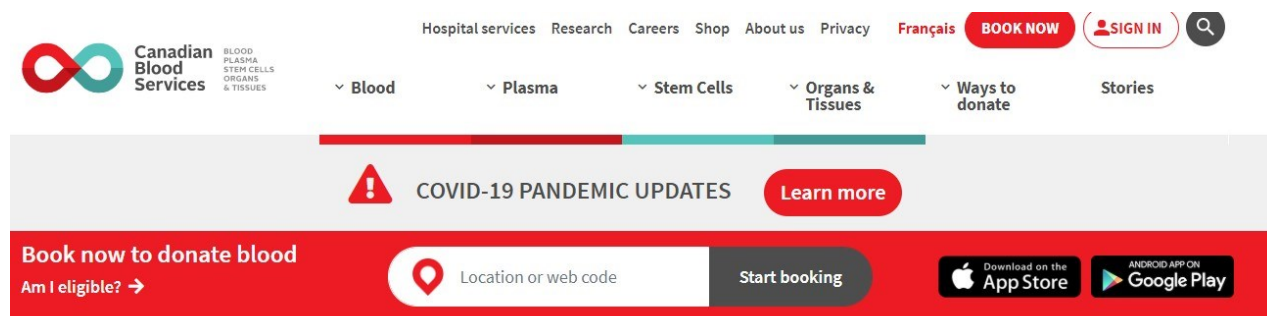
Na slici 2.1 prikazana je aplikacija Give Blood koja služi darivateljima krvi iz Velike Britanije [14]. Aplikacija se sastoji od naslovne stranice i stranice s informativnim sadržajem o darivanju krvi te postoji mogućnost registriranja i prijave korisnika. Nakon registriranja i prijave aplikacija nudi mogućnost praćenja organiziranih akcija darivanja krvi i naručivanja korisnika za darivanje.



*Slika 2.1. Prikaz sučelja web aplikacije Give blood*

Slika 2.2 prikazuje internet aplikaciju Canadian Blood Services [15] koja se sastoji od naslovne stranice, informativnih stranica, mogućnosti traženja centra za darivanje, registriranja i prijave i kviza za određivanje mogućnosti darivanja krvi. Namijenjena je darivateljima na području Kanade. Nakon registriranja korisnik ima mogućnost ugovaranja termina za darivanje i praćenja vlastitih donacija.

Za razliku od navedenih aplikacija, aplikacija izrađena u sklopu ovog diplomskog rada namijenjena je za hrvatsko tržište, prikazuje statistiku o donacijama i darivateljima te je izrađena kako bi bila interaktivna, to jest zanimljiva korisnicima i sadrži više funkcionalnosti usmjerenih na korisnika i sadrži administratorsku stranu. Također, za razliku od navedenih aplikacija, glavna svrha web aplikacije ovog rada je upravljanje zalihama krvi i darivateljima.



*Slika 2.2. Prikaz sučelja web aplikacije Canadian Blood Services*

### 2.3.2 Trenutno stanje u području postupaka višekriterijskog odlučivanja

Svaka metoda analize višestrukih kriterija (MCDA - *Multiple Criteria Decision Analysis*) zahtijeva specifikaciju nekih tehničkih parametara ili parametara koji se temelje na preferencijama. Ovi se parametri koriste tijekom stvaranja preporuka i traženja rezultata koji se odnose na određeni problem s više relevantnih čimbenika [16].

Računala su omogućila programsku potporu odlučivanju i generiranju rezultata korištenjem metoda višekriterijskih analiza, a jedni od sustava koji koriste navedene metode nazivaju se sustavi za potporu odlučivanju (DSS – *decision support system*). Pomoću ovakvih i sličnih sustava problemi se rješavaju odgovarajućim algoritmima koristeći kriterije koje navodi korisnik, a s obzirom na trenutak u kojem korisnik navodi kriterije, postoje tri vrste algoritma [17]:

- a priori – korisnik određuje kriterije prije samog procesa odlučivanja
- a posteriori – korisnik nakon pronalaženja zadovoljavajućih rješenja bira koje mu najbolje odgovara
- interaktivne metode – proces se odvija po iteracijama, a korisnik usmjerava iteracije dodavanjem kriterija

Višekriterijsko odlučivanje korišteno u aplikaciji izrađenoj u sklopu ovog diplomskog rada koristi „a priori“ vrstu algoritma i pripada leksikografskim metodama. Leksikografske metode koriste hijerarhijski pristup, odnosno utvrđuje se koji kriterij ima najveći prioritet te se traže rezultati koji zadovoljavaju taj kriterij. Ako je rezultata više, razvrstavaju se po sljedećem kriteriju po prioritetu i tako dalje.

### **3. IDEJNO RJEŠENJE I MODEL WEB SUSTAVA ZA UPRAVLJANJE ZALIHAMA KRVI**

U ovom poglavlju opisana je ideja rješenja internet aplikacije, navedeni su funkcionalni i nefunkcionalni zahtjevi koje aplikacija treba zadovoljiti, opisan je postupak višekriterijskog odlučivanja koji će biti implementiran u aplikaciji i objašnjena je struktura aplikacije te su navedene komponente koje aplikacija sadrži.

#### **3.1. Idejno rješenje web sustava aplikacije**

Cilj ovog diplomskog rada izrada je internet aplikacije za upravljanje zalihama krvi odnosno darivateljima krvi i za praćenje donacija i izdavanje zahtjeva za određenom krvnom grupom. Aplikacija treba biti podijeljena na dva dijela – klijentski dio i administratorski dio.

Klijentski dio treba obuhvaćati profil korisnika - darivatelja krvi, a treba sadržavati osobne podatke koji su važni kod darivanja krvi kao i svojevrsni kalendar donacija korisnika. Praćenjem donacija korisniku se prikazuju informacije o darivanju krvi kao što su broj dosadašnjih darivanja i datum sljedeće moguće donacije.

Administratorski dio aplikacije treba obuhvaćati dio u kojem administrator može upravljati korisnicima, pretraživati ih i brisati iz baze podataka. Također, glavna funkcionalnost administratorskog dijela aplikacije treba biti pronalazak darivatelja po određenim vremenskim i prostornim kriterijima. Navedeni dio aplikacije služiti će pri nedostatku određene krvne grupe za zalihe krvi ili za pacijenta te će se pretraživanjem i hijerarhijskim višekriterijskim odlučivanjem pozivati darivatelje iz rezultata na donacije. U administratorskom dijelu aplikacije također trebaju biti prikazane informacije o darivateljima, donacijama i statistika darivanja krvi u određenom vremenu kako bi se mogao pratiti i uređivati eventualni nedostatak darivatelja, smanjen broj donacija i slično.

Navedenu aplikaciju također treba upotpuniti s osnovnim informacijama o darivanju, testom kojim korisnik provjerava, bez registriranja, može li trenutno donirati krv i sličnim interaktivnim sadržajem.

#### **3.2. Funkcionalni i nefunkcionalni zahtjevi na web aplikaciju**

Tijekom izrade aplikacije važno je obratiti pažnju na jednostavnost korištenja aplikacije, preglednost podataka i na ispunjavanje korisničkih zahtjeva. Također je potrebno prilagoditi dizajn aplikacije uređaju na kojem se koristi, učiniti aplikaciju interaktivnom kako bi bila zanimljiva za korištenje i izraditi aplikaciju tako da odziv bude što brži. Aplikacija će se sastojati od više stranica



za odgovarajuće zahtjeve kako bi bila preglednija. Potrebno je i osigurati zaštitu korisnika uz pomoć registriranja i prijave jedinstvenim podacima i omogućiti komuniciranje s bazom podataka MongoDB.

U tablici 3.1 navedeni su korisnički zahtjevi klijenta aplikacije odnosno darivatelja krvi, a u tablici 3.2 navedeni su korisnički zahtjevi administratora aplikacije odnosno odjela za transfuziju.

S klijentske strane aplikacije važno je tijekom registriranja spremati unesene podatke u bazu podataka te ih nakon prijave dohvatiti i prikazati korisniku na profilu i omogućiti korisniku dodavanje datuma darivanja u bazu podataka kao i omogućiti prikaz istih. Korisnik ima pristup osobnim podacima te određene može izmijeniti i ima mogućnost uređivanja i sortiranja datuma darivanja.

*Tablica 3.1 Popis korisničkih zahtjeva aplikacije s klijentske strane*

REDNI BROJ	PRIORITET	OPIS
1	1	Prikaz naslovne stranice, zaglavlja, podnožja i osnovnih informacija o darivanju krvi
2	1	Mogućnost rješavanja testa za provjeru trenutne mogućnosti darivanja krvi
3	1	Prikaz interaktivne karte s mjestima na kojima se može donirati krv
4	1	Mogućnosti registriranja i prijave – izrade korisničkog profila i odjave
5	1	Nakon prijave, prikaz profila s podacima iz baze podataka koje je korisnik unio tijekom registriranja
6	1	Mogućnost uređivanja profila odnosno određenih korisničkih podataka
7	1	Izračun i prikaz sljedećeg datuma darivanja na osnovi posljednjeg datuma prema spolu korisnika, za muškarce tri mjeseca, za žene četiri mjeseca
8	1	Na osnovi izračuna sljedećeg datuma darivanja, prikaz trenutne mogućnosti darivanja – „Trenutno možete darivati“ ili „Trenutno ne možete darivati“
9	1	Mogućnost unošenja, brisanja i sortiranja datuma darivanja krvi
10	1	Ograničenje datuma darivanja koji se unosi, ovisno o spolu, na tri ili četiri mjeseca od ostalih datuma
11	1	Na osnovi unesenih datuma darivanja, izračun i prikaz broja dosadašnjih darivanja na profilu
12	1	Mogućnost označavanja trenutne dostupnosti korisnika, na profilu, u slučaju poziva na doniranje, „Nisam dostupan/na za donacije“ ili „Dostupan/na sam za donacije“

S administratorske strane aplikacije važno je dohvatiti podatke o korisnicima iz baze podataka i vršiti izračun o trenutnoj mogućnosti darivanja krvi prema posljednjem datumu darivanja. Važno je dohvatiti podatke o darivateljima kako bi se kroz višekriterijsko odlučivanje, koje uz navedene vremenske kriterije uključuje i prostorne kriterije, moglo odrediti koji korisnik zadovoljava tražene

kriterije i kako bi ga se moglo kontaktirati. Administrator ima mogućnost uvida u bazu podataka kao i statistiku izrađenu prema podacima iz baze podataka o darivanju i darivateljima i ima mogućnost uklanjanja korisnika.

*Tablica 3.2 Popis korisničkih zahtjeva aplikacije s administratorske strane*

REDNI BROJ	PRIORITET	OPIS
1	1	Prikaz broja darivatelja i donacija u bazi podataka po svakoj krvnoj grupi i ukupno
2	1	Prikaz darivatelja na karti prema adresi uz podatak krvne grupe i imena
3	1	Prikaz statistike o darivateljima i donacijama na grafovima
4	1	Prikaz svih darivatelja iz baze podataka s osobnim informacijama - ime, adresa, krvna grupa, kontakt broj i datum posljednje donacije
5	1	Mogućnost pretraživanja svih darivatelja prema osobnim podacima
6	1	Mogućnost uklanjanja darivatelja iz baze podataka
7	1	Prikaz darivatelja koji su trenutno u mogućnosti darivati, prema izračunu po posljednjem datumu darivanja i po tome što su označili da su dostupni za darivanje
8	1	Mogućnost pretraživanja i kontaktiranja iznad navedenih darivatelja u svrhu nadopune zaliha krvi po kriterijima krvne grupe koja se traži, količine krvi i udaljenosti od lokacije na kojoj su potrebni
9	1	Mogućnost pretraživanja i kontaktiranja darivatelja navedenih u zahtjevu 8 u svrhu hitnog liječenja pacijenta po kriterijima svih kompatibilnih krvnih grupa traženoj, količine krvi i udaljenosti od lokacije na kojoj su potrebni

### 3.3. Prikadni postupci višekriterijskog odlučivanja

U administratorskom dijelu aplikacije administrator treba imati mogućnost tražiti korisnike kako bi se nadopunile zalihe krvi ili zadovoljile potrebe krvi za pacijenta po određenim kriterijima.

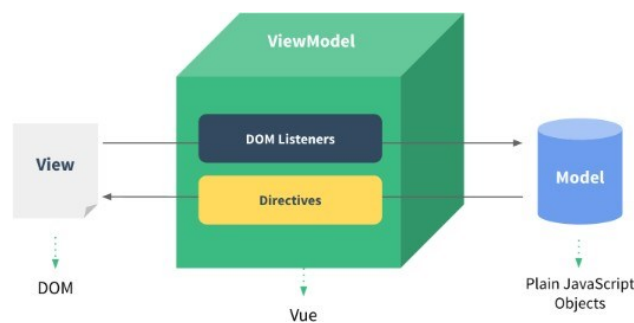
Prvi kriterij koji korisnik treba zadovoljavati kako bi ušao u izbor kao potencijalni darivatelj je trenutna mogućnost darivanja odnosno vremenski kriterij. Izračun mogućnosti darivanja treba se vršiti prema spolu darivatelja – od posljednjeg datuma darivanja korisnika računa se je li do datuma na koji se traži darivatelj prošao, za muškarce tri mjeseca i za žene četiri mjeseca. Ako je prošao potreban period, sljedeći kriterij koji se treba pregledavati je je li korisnik označio, na svom profilu, da je trenutno dostupan za pozivanje na darivanja ili nije dostupan. Nakon što su darivatelji filtrirani po navedenim kriterijima, trebaju se gledati traženi kriteriji koje navodi administrator, a ti kriteriji su:

1. krvna grupa – najvažniji kriterij
2. krv je potrebna za:

- a. pacijenta – ako administrator označi da je tražena krv za pacijenta, prema krvnoj grupi se trebaju gledati sve kompatibilne krvne grupe jer se pacijent smatra hitnim slučajem i nije nužno da krv bude direktno tražene krvne grupe, već bilo koja koja može što brže pomoći
  - b. zalihe krvi – ako administrator označi da je tražena krv za zalihe, trebaju se filtrirati darivatelji samo tražene krvne grupe jer je kod zaliha važno održavati barem minimalne količine određene krvne grupe
3. lokacija:
- administrator treba unositi lokaciju na kojoj su potrebne jedinice krvi kako bi put darivatelja do mjesta darivanja bio što kraći, to jest kako bi aplikacija efikasno birala darivatelje
  - aplikacija najprije treba tražiti korisnike unutar deset kilometara od navedene lokacije, ako postoje, filtrira ih i prikazuje, ako ne postoje, proširuje krug pretrage za po deset kilometara sve dok ne pronađe najbliže potencijalne darivatelje krvi
4. količina – označava koliko jedinica krvi je potrebno, to jest svaki darivatelj može donirati jednu jedinicu krvi pa se količina odnosi na broj darivatelja koji se traži

### 3.4. Građa i komponente web sustava za upravljanje zalihama krvi

Web aplikacija strukturnog je uzorka MVVM. *Model-view-viewmodel* (MVVM) prikazan na slici 3.1, prema [18], strukturni je uzorak programske podrške koji olakšava odvajanje razvoja grafičkog korisničkog sučelja (pogled) od razvoja pozadinske logike (model) tako da pogled ne ovisi o određenoj platformi modela. Model prikaza MVVM pretvarač je vrijednosti, što znači da je model prikaza odgovoran za izlaganje, to jest pretvaranje podatkovnih objekata iz modela na takav način da se objektima lako upravlja i prezentira. U tom pogledu, model prikaza je više model nego pogled i obrađuje većinu, ako ne i svu logiku prikaza. Model prikaza može implementirati obrazac posrednika koji organizira pristup pozadinskoj logici oko skupa slučajeva upotrebe koji su podržani tim prikazom.



**Slika 3.1.** Struktura Vue.js aplikacije – MVVM [18]

## 4. PROGRAMSKO RJEŠENJE WEB APLIKACIJE

Kao praktični dio ovog diplomskog rada izrađen je internet sustav za upravljanje zalihama krvi. Aplikacija je izrađena u programu Visual Studio Code, a u ovom poglavlju navedene su i opisane tehnologije, okoline i korišteni programski jezici, opisana je klijentska i administratorska strana aplikacije uz prikaze sučelja i dijelova programskih kodova i objašnjeno je programsko rješenje hijerarhijskog višekriterijskog odlučivanja.

### 4.1. Korišteni programski jezici, tehnologije i okoline

U ovom poglavlju navedene su korištene tehnologije, okoline i programski jezici kao i korištene biblioteke i dodaci i objašnjena je njihova uloga. Tijekom izrade aplikacije bilo je potrebno izraditi dva dijela – *frontend* i *backend* aplikacije i spojiti aplikaciju s bazom podataka.

#### 4.1.1. Korisnička strana

Kod izrade korisničke strane aplikacije (*frontend-a*) korišten je Vue.js, a glavne korištene biblioteke su Vuex biblioteka za upravljanje stanjima, Moment.js biblioteka za računanje vremena i formatiranje prikaza vremena, vue-chart.js za izradu grafičkih prikaza i izradu statistike u aplikaciji i dodatak vue2-google-maps za prikaz karata i podataka na kartama kao što su markeri te za izračun udaljenosti lokacija preko zemljopisne širine i dužine.

Kao što je opisano u [19], Vue.js je Javascript progresivni okvir otvorenog koda koji se koristi za izgradnju korisničkih sučelja i aplikacija na jednoj stranici. Za razliku od drugih monolitnih okvira, Vue.js dizajniran je kako bi se mogao inkrementalno usvojiti, odnosno ima inkrementalno prilagodljivu arhitekturu koja koristi deklarativno prikazivanje i sastavljanje komponenata. Glavna knjižnica fokusirana je samo na sloj prikaza i lako ju je usvojiti i integrirati s drugim knjižnicama ili postojećim projektima. Također, pomoću Vue.js-a mogu se izraditi sofisticirane aplikacije kada se koristi u kombinaciji s modernim bibliotekama koje pružaju razne alate. Vue.js omogućuje proširivanje HTML-a atributima koji se nazivaju direktivama, a nude razne funkcionalnosti HTML aplikacijama i mogu biti korisnički definirane ili ugrađene.

Neke od značajki Vue.js-a, prema [20], su:

- virtualni DOM
- vezanje podataka
- rad s komponentama
- rukovanje događajima
- dodavanje animacija ili tranzicija

- predlošci
- direktive
- navigacija rutama

Vuex, prema [21], služi kao centralizirano spremište za sve komponente u aplikaciji, s pravilima koja osiguravaju da stanja mogu mutirati samo na predvidljiv način. Također se integrira s Vue-ovim službenim proširenjem za razvojne programe kako bi pružio neke napredne značajke. Koristi se kada ista stanja koristimo više puta ili kada iz različitih komponenti mijenjamo ista stanja kako bi osigurali pregledan kod i mogućnost održavanja koda bez nepotrebnih komplikacija.

Moment.js, prema [22], biblioteka je koja služi za manipuliranje datumima, to jest koristimo ju za analiziranje vremena, provjere i validacije vremena i datuma te formatiranje prikaza datuma i vremena. Pomoću navedene biblioteke možemo računati protekle periode između vremena, vršiti validacije kao u slučaju dodavanja datuma rođenja, to jest ograničiti samo na prošle datume ili starije od određenog broja godina i slično. Pomoću Moment.js-a možemo vršiti sva potrebna računanja s vremenom na jednostavan način i prilagoditi prikaz datuma kako bi bio čitljiv korisnicima.

Vue-chart.js je biblioteka Chart.js prilagođena korištenju u Vue.js-u i, kao što je navedeno u [23], omogućuje lako stvaranje komponenti grafova za višekratnu upotrebu. Ideja vue-chart.js-a je pružiti korisniku izradu komponenti s maksimalnom fleksibilnošću i proširivošću. Korisnik treba stvoriti vlastitu komponentu i zatim ima mogućnost proširiti ju s komponentama iz navedene biblioteke.

Vue2-google-maps dodatak je koji omogućuje korištenje Google karata, prikaza markera na kartama preko zemljopisne širine i dužine, potvrđivanje lokacije kod unosa adrese i slično. Za korištenje navedenog dodatka potrebno je koristiti Google Maps API i generirati Google Maps API jedinstveni ključ.

Također, kod dizajniranja određenih dijelova aplikacije korišten je Bootstrap. Bootstrap je, prema [24], CSS okvir otvorenog koda usmjeren na responzivni dizajn koji sadrži CSS predloške dizajna za razne komponente sučelja kao što su gumbi, navigacijska traka, forme i slično.

#### **4.1.2. Poslužiteljska strana**

Tijekom izrade poslužiteljske strane aplikacije (*backend-a*) korišteno je Node.js okruženje uz Express.js, a za autentifikaciju korisnika korištena je posrednička programska podrška Passport.js

i bcrypt funkcija za obradu, to jest kriptiranje lozinki kako bi one bile zaštićene i poznate samo korisniku.

Node.js je, prema [25], JavaScript okruženje, a izvršava JavaScript kod izvan Internet preglednika. Otvorenog je koda i cross-platforme, a programerima omogućuje korištenje JavaScripta za skriptiranje na strani poslužitelja za izradu dinamičnog sadržaja Internet stranice i za pisanje alata naredbenog retka. Node.js spaja razvoj internet aplikacija oko jednog programskog jezika umjesto različitih za strane klijenta i poslužitelja. Koristi module koji su predviđeni za umrežavanje, kriptografske funkcije, protoke podataka i ostale funkcije, a koriste API dizajniran za smanjenje složenosti koda poslužiteljskih aplikacija.

Neke od značajki Node.js-a, kao što je navedeno u [26], su:

- asinkron je i pokreće se događajima
- vrlo je brz
- korisni jednu nit, ali je vrlo skalabilan
- prikazuje podatke u „komadima“ bez prethodnog skidanja i učitavanja

Express.js je, prema [27], besplatan okvir internet aplikacija, otvorenog koda, namijenjen Node.js-u. Dizajniran je za izradu internet aplikacija i API-ja i olakšava nam izradu pozadinskih zahtjeva i funkcija u aplikaciji.

Passport.js, kao što je opisano u [28], posrednička je programska podrška za provjeru autentičnosti kod korištenja Node.js-a. Navedena programska podrška fleksibilna je i modularna i može se ubaciti u bilo koju aplikaciju koja koristi Express.js. Koristi se za provjeru autentičnosti pomoću korisničkog imena i lozinke ili uz račune na društvenim mrežama kao što su Facebook, Twitter i slično.

#### **4.1.3. Baza podataka**

Internet sustav izrađen u sklopu ovog diplomskog rada za bazu podataka koristi MongoDB i biblioteku Mongoose. Također, za pregled baze podataka korišten je program MongoDB Compass Community.

MongoDB je, kao što je opisano u [29], višeplatformski program za baze podataka orijentiran na dokumente. Koristi JSON dokumente s izbornim shemama, a klasificiran je kao program baze podataka NoSQL. Razvila ga je organizacija 10gen koja je danas poznata kao MongoDB Inc. Može se koristiti besplatno, a ciljevi razvijanja ove baze su skalabilnost, dostupnost, brže razvijanje, skaliranje s jednog poslužitelja na više i slično.

Neke od značajki MongoDB baze podataka, prema [30], su:

- Ad-hoc upiti
- indeksiranje
- repliciranje
- može se koristiti kao datotečni sustav
- sjedinjavanje podataka
- balansiranje raspodjele podataka

Mongoose je, prema [31], biblioteka objektnog modeliranja podataka (engl. *Object Data Modeling*) koja se koristi uz MongoDB i Node.js. Upravlja odnosima između podataka, pruža provjeru valjanosti sheme i koristi se za prevođenje između objekata u kodu i predstavljanja tih objekata u MongoDB-u.

MongoDB Compass Community program, kao što je opisano u [32], omogućuje povezivanje s bazom podataka kako bi se mogli vršiti upiti i kako bi se moglo upravljati podacima – stvarati i uklanjati zbirke i određene podatke. Također, pomoću navedenog programa baza podataka je pregledna i lako se mogu pratiti promjene koje se događaju u njoj.

Korištenjem biblioteke Mongoose izrađena je shema korisnika koja sadrži sve potrebne podatke o korisniku, a može se vidjeti na slici 4.1. Za primjer, shema sadrži varijablu *name* tipa *String* za koju je označeno da je obavezna tijekom registriranja – *required: true*.

```
const UserSchema = new Schema({
  name: {
    type: String,
    required: true
  },
  username: {
    type: String,
    required: true
  },
  email: {
    type: String,
    required: true
  },
  password: {
    type: String,
    required: true
  },
  role: {
    type: String,
    default: 'basic',
    required: false,
    enum: ['basic', 'admin']
  },
});
```

**Slika 4.1.** Prikaz dijela programskog koda sheme *User*

Svaka se Mongoose shema preslikava na MongoDB zbirku i definira oblik dokumenata unutar te zbirke. Kako bi se shema mogla koristiti, potrebno ju je konvertirati u model, a to se izvršava prosljeđivanjem u kod na slici 4.2.

```
module.exports = User = mongoose.model('users', UserSchema);
```

*Slika 4.2. Programski kod za konvertiranje Mongoose sheme u model*

Nakon pretvorbe sheme u model mogu se izvršavati razne metode kao što su kreiranje novih dokumenata, pretraživanje po identifikacijskom nizu, dohvaćanje određenih podataka, dohvaćanje svih podataka i slično. Na slici 4.3 prikazan je dokument iz MongoDB baze podataka kreiran pomoću modela *User* sa slike 4.1.

```
_id: ObjectId("5f4804f6cc773929d8387fa6")
  residence: Array
    0: "Ulica kralja Držiislava"
    1: "11"
    2: "34000"
    3: "Požega"
    4: "Hrvatska"
    5: "45.3442401"
    6: "17.6762924"
  donationDate: Array
    0: 2020-09-01T00:00:00.000+00:00
    1: 2020-02-13T00:00:00.000+00:00
    2: 2019-10-26T00:00:00.000+00:00
    3: 2019-04-09T00:00:00.000+00:00
    4: 2019-01-01T00:00:00.000+00:00
    5: 2014-12-30T00:00:00.000+00:00
  role: "basic"
  name: "Marko Jurić"
  username: "Marko"
  password: "$2a$10$izMdIawoq3JwFvJCKFM.Qe2za6qWESQxghyQPQSIM1ChAkva90uK"
  email: "marko.juric@gmail.com"
  birthdate: 1996-08-11T00:00:00.000+00:00
  bloodgroup: "0+"
  phonenumber: 997112366
  sex: "Muškarac"
  __v: 47
```

*Slika 4.3. Prikaz strukture modela User u bazi podataka*

## 4.2. Programsko rješenje web sustava na strani klijenta

U ovom poglavlju objašnjena je klijentska strana aplikacije, prikazan je dijagram toka i prikazi dijelova aplikacije kojima korisnik ima pristup. Klijentska strana aplikacije sastoji se od:

- naslovne stranice (slika 4.4) koja sadrži činjenice o darivanju krvi i interaktivnu kartu na kojoj su postavljene markeri s adresama mjesta na kojima korisnik može darivati krv

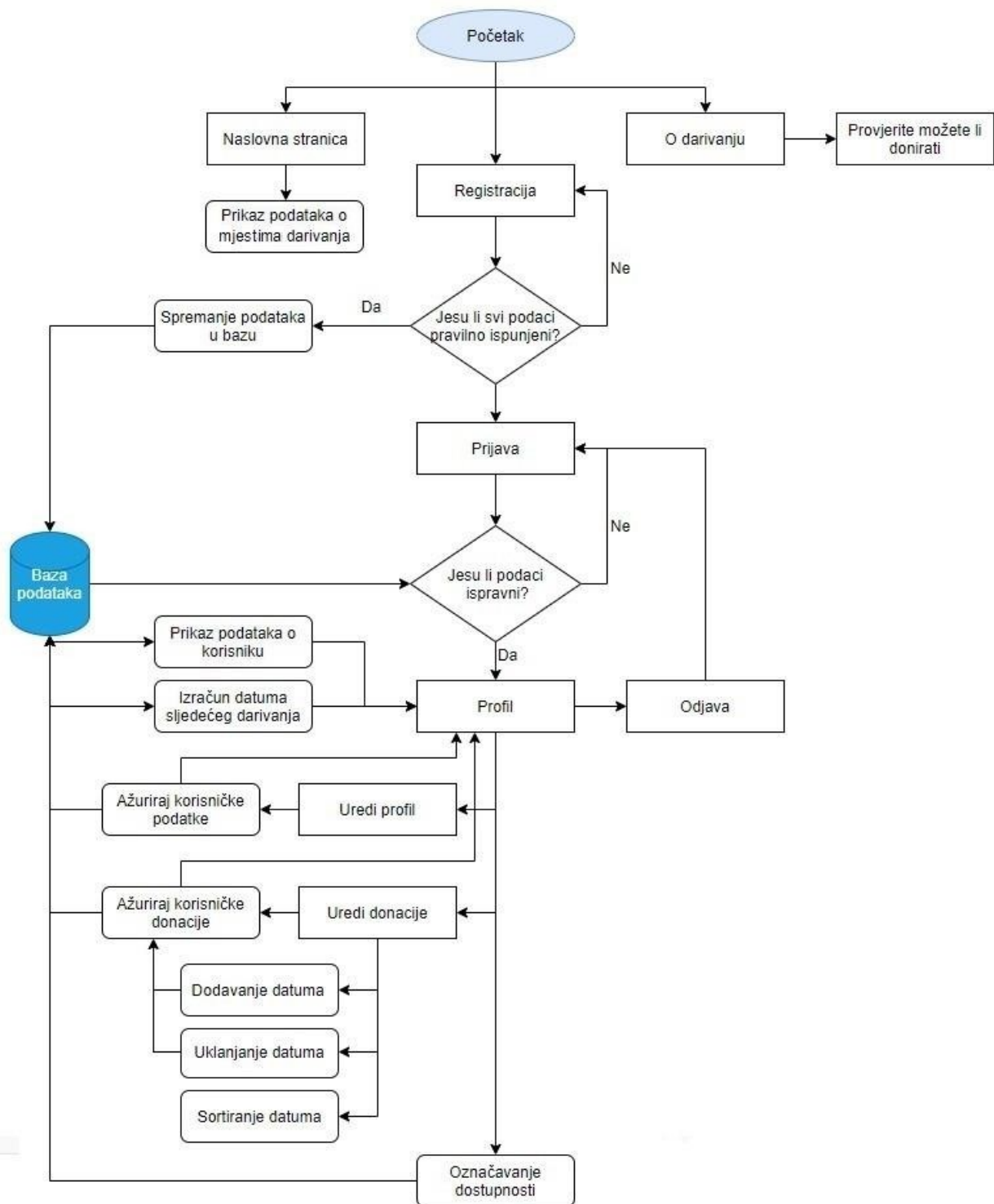


- stranice „O darivanju“ gdje korisnik može pročitati osnovne informacije o darivanju, mogućnosti darivanja, biranju darivatelja, postupku darivanja i slično te poveznicom na kratki test u kojem korisnik može provjeriti može li trenutno darivati krv
- stranice registriranja na kojoj korisnik kreira korisnički račun
- stranice prijave gdje se korisničkim imenom i lozinkom korisnik prijavljuje u aplikaciju
- profila na kojem su prikazane informacije o korisniku, datumi darivanja koje je korisnik unio, sljedeći datum darivanja i trenutna mogućnost darivanja
- stranice za uređivanje korisničkih podataka na kojoj korisnik može promijeniti ime, adresu, kontakt broj i lozinku
- stranice za dodavanje i brisanje datuma donacija kako bi korisnik mogao pratiti svoju aktivnost



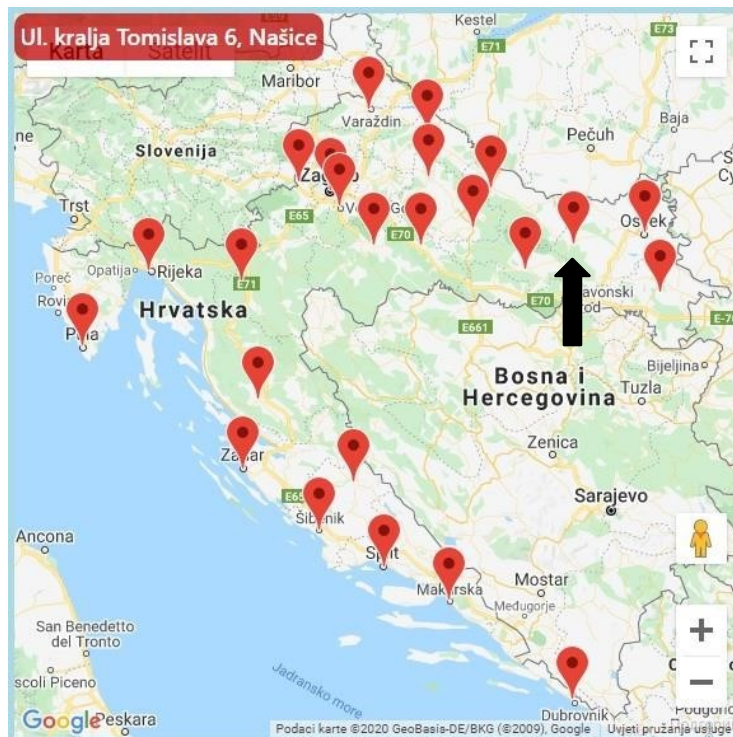
**Slika 4.4.** Prikaz sučelja naslovne stranice aplikacije

Dijagram toka klijentske strane aplikacije prikazan je na slici 4.5, a postupak registriranja, prijave, prikaza profila i informacija o korisniku detaljnije je objašnjen u sljedećim poglavljima.



*Slika 4.5. Dijagram toka klijentske strane internet aplikacije*

Karta na naslovnoj stranici koja prikazuje mjesta darivanja koristi Google Maps API i podatke izvlači iz polja u kojem su navedene adrese ustanove i zemljopisna širina i dužina. Adresa je navedena kako bi se prikazala korisniku, a zemljopisna širina i dužina se koriste za postavljanje markera na karti. Prikaz karte može se vidjeti na slici 4.6.



*Slika 4.6. Prikaz interaktivne karte mjesta za darivanje krvi*

Na slici 4.7 prikazan je HTML dio programskog koda karte na kojoj su prikazani markeri mjesta bolnica. Kako bi se karta prikazala, postavlja se oznaka GmapMap kojoj se određuje veličina, središte karte i faktor zumiranja. Za prikaz markera na karti koristi se polje *markers* u koje su spremljene zemljopisne širine i dužine mjesta za darivanje, ti podaci se koriste za određivanje varijable pozicije markera. Također su određene i funkcije za prijelaz kursorom preko markera, a to su *showHospital()* i *hideHospital()*, to jest prikazivanje adrese kada je kursor na markeru i sakrivanje adrese kada kursor nije na markeru.

```

<GmapMap id="map" style="width: 570px; height: 570px" :zoom="7"
:center="{lat: 44.555519, lng: 16.369490}">
  <GmapMarker v-for="(marker, index) in markers"
    label = ""
    :key="index"
    :position="marker.position"
    @mouseover="showHospital(index)"
    @mouseout="hideHospital()"
  />
</GmapMap>

```

*Slika 4.7. Programski kod karte prikaza mjesta darivanja*

#### 4.2.1. Registriranje i prijava

Kako bi korisnik mogao koristiti aplikaciju, potrebno je izraditi korisnički račun, to jest registrirati se. Prilikom registriranja korisnik unosi podatke koji trebaju biti jedinstveni - korisničko ime i adresu elektroničke pošte i osobne podatke:

- ime i prezime – važno je zbog prikaza na administratorskoj strani
- datum rođenja – provjerava se je li osoba punoljetna
- spol – važno je odabrati spol zbog računanja sljedećih datuma darivanja krvi jer se razlikuje kod muškaraca i kod žena
- adresa – služi za prikaz donora na administratorskoj strani i kod višekriterijskog odlučivanja, to jest kod računanja udaljenosti od traženog mjesta, a validacija unesene adrese se vrši korištenjem vue2-google-maps biblioteke
- krvna grupa – važan kriterij kod višekriterijskog odlučivanja zbog kompatibilnosti s pacijentima i kao kriterij pretrage na administratorskoj strani
- datum posljednje donacije – koristi se kod izračuna sljedećeg datuma darivanja i trenutne mogućnosti darivanja krvi
- kontakt broj – služi administratorskoj strani kako bi se korisnik mogao pozvati na darivanje krvi
- lozinka i potvrda lozinke – uz korisničko ime, daje sigurnost korisniku kako bi samo korisnik mogao pristupiti svom profilu

Sučelje stranice za registriranje prikazano je na slici 4.8.

**REGISTRACIJA**

**Korisničko ime**  
Unesite korisničko ime

**Ime i prezime**  
Unesite ime i prezime

**E-mail**  
Unesite e-mail

**Adresa**  
Unesite lokaciju

**Datum rođenja**  
dd.mm.gggg.

**Datum donacije**  
dd.mm.gggg.

**Spol**  
 Muškarac  Žena

**Krvna grupa**  
 A+  B+  AB+  0+  
 A-  B-  AB-  0-

**Kontakt**  
Unesite kontakt

**Lozinka**  
Unesite lozinku

**Potvrdite lozinku**  
Unesite lozinku

[Registracija](#) [Već imate račun?](#)

*Slika 4.8. Sučelje stranice registriranja*

Tijek registriranja je prvo na *frontend-u* aplikacija provjera je li korisnik ispunio sva tražena polja, je li unio ispravan datum rođenja i datum posljednje donacije. Ako su podaci ispravno uneseni i korisnik je pritisnuo gumb „Registracija“, korištenjem biblioteke axios aplikacija šalje zahtjev na API s unesenim podacima. Zatim se na *backend* dijelu aplikacije provjerava podudaraju li se lozinka i potvrđena lozinka, postoji li korisničko ime u bazi podataka i koristi li se već adresa elektroničke pošte. Nakon navedenih provjera, ako su podaci ispravni i jedinstveni, spremaju se u model koji je kreiran korištenjem Mongoose sheme i vrši se *POST* zahtjev na API. Lozinka se kriptira uz pomoć biblioteke bcrypt kako u bazi podataka ne bi bila čitljiva i otkrivena te se podaci spremaju u bazu podataka. Nakon spremanja u bazu, aplikacija prebacuje na stranicu „Prijava“ kako bi se korisnik mogao prijaviti s ranije odabranim korisničkim imenom i lozinkom.

Slika 4.9 prikazuje kreiranje varijable *user* koja se sastoji od svih potrebnih podataka za ispunjenje modela prikazanog u potpoglavlju 4.1.3 i sadrži pozivanje metode *register()* s tim podacima. Ako je metoda uspješno izvršena, kao što je ranije navedeno, aplikacija prebacuje korisnika na stranicu „Prijava“ – *router.push(„login“)*.

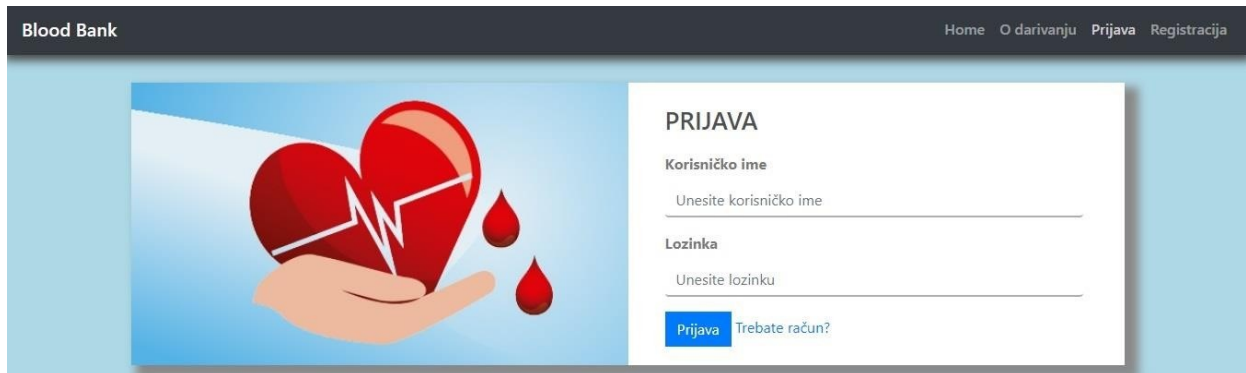
```
let user = {
  username: this.username,
  password: this.password,
  confirm_password: this.confirm_password,
  email: this.email,
  name: this.name,
  birthdate: this.birthdate,
  residence: this.residence,
  bloodgroup: this.bloodgroup,
  phonenumber: this.phonenumber,
  sex: this.sex,
  donationDate: this.donationDate
};
this.register(user).then(res => {
  if (res.data.success) {
    this.$router.push("login");
  }
});
```

**Slika 4.9.** Prikaz dijela programskog koda registriranja

Prilikom registriranja korisniku se određuje uloga – „*admin*“ ili „*basic*“. Zadana uloga koja se dodjeljuje prilikom registriranja je „*basic*“, a administratoru je unaprijed određena uloga „*admin*“. Također se dodaje dostupnost korisnika, zadana dostupnost je „Dostupan“, a važna je kod pretrage potencijalnih donora na administratorskoj strani.

Tijekom prijave, prikazane na slici 4.10, u aplikaciji se prvo vrši provjera uloge, ako je uloga određena kao „*basic*“, pomoću axios biblioteke se vrši *POST* zahtjev s unesenim korisničkim imenom i lozinkom. U *backend* dijelu aplikacije, također se vrši *POST* zahtjev i pretražuje se

postoji li korisničko ime u bazi podataka funkcijom *findOne()* (slika 4.11) te ako postoji, unesena lozinka se pomoću biblioteke *bcrypt* uspoređuje s onom u bazi. Ako je prijava uspješna, aplikacija prebacuje na stranicu „Profil“, a ako nije, korisniku se prikazuju upozorenja o pogreškama koje su se pojavile.



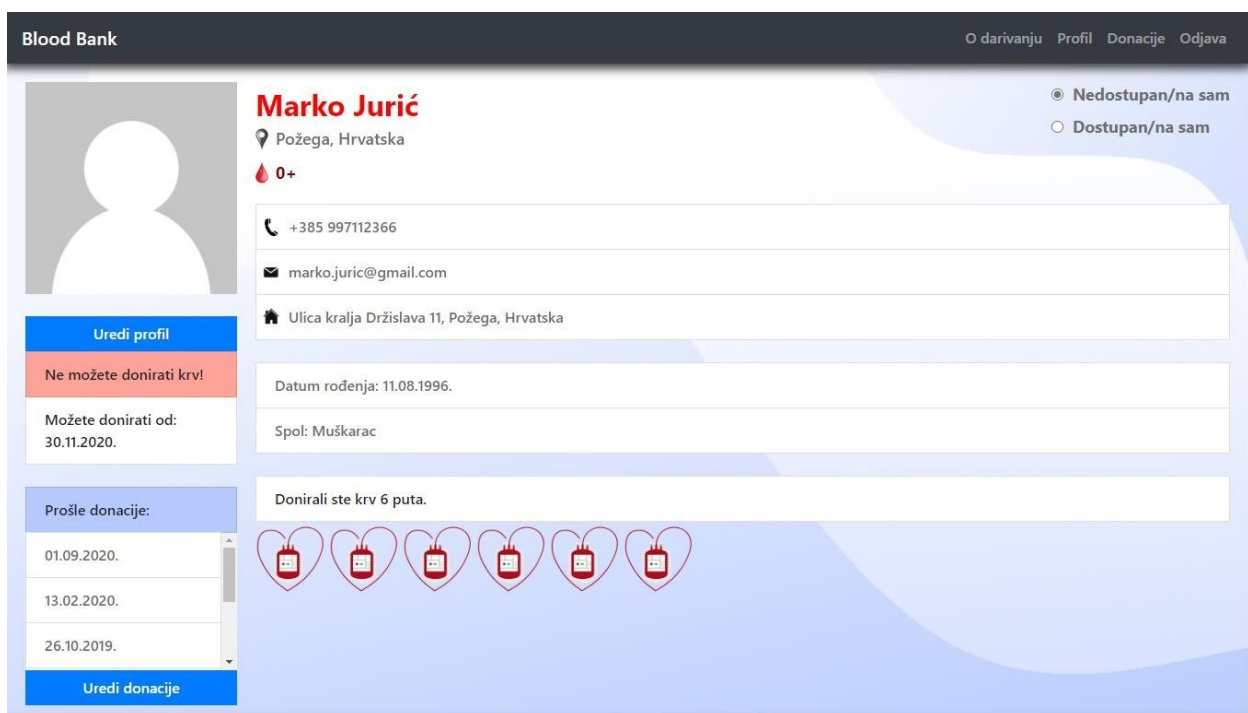
*Slika 4.10. Sučelje stranice prijave*

```
User.findOne({
  username: req.body.username
}).then(user => {
  if (!user) {
    return res.status(404).json({
      msg: "Korisničko ime nije pronađeno.",
      success: false
    });
  }
})
```

*Slika 4.11. Programski kod za traženje korisničkog imena u bazi podataka*

#### 4.2.2. Profil korisnika

Nakon prijave, korisnik je usmjeren na stranicu „Profil“ na kojoj može vidjeti svoje podatke koji su uneseni tijekom registriranja. Podaci se dohvaćaju iz baze podataka korištenjem *GET* zahtjeva i uz pomoć jedinstvenog identifikacijskog niza koji se dodjeljuje svakom korisniku. Nakon prijave dohvaća se identifikacijski niz trenutno prijavljenog korisnika, niz se traži u bazi i dohvaćaju se svi podaci koji se nalaze u njoj. Prikaz sučelja korisničkog profila nalazi se na slici 4.12.



*Slika 4.12. Sučelje korisničkog profila*

Profil je dizajniran kako bi svi podaci bili pregledni pa su podijeljeni u određene skupine funkcionalnosti. Kao što je prikazano na slici 4.12, s desne strane profila prikazane su osobne informacije o korisniku i prikazano je koliko puta je korisnik do sada darivao krv. Izračun se vrši brojanjem unesenih datuma doniranja i za svaki put se dodaje slika vrećice krvi kako bi pregled bio zanimljiviji. Također, u desnom kutu aplikacije nalazi se potvrdni okvir u kojem korisnik bira je li trenutno dostupan ili nije. Zadana vrijednost prilikom registriranja je „Dostupan“, korisnik ju može promijeniti na navedenom mjestu, a prikazuje se bojom imena korisnika, ako je ime crvene boje, korisnik je označio da je nedostupan, a u suprotnom je ime zelene boje. Lijeva strana profila podijeljena je u tri skupine funkcionalnosti.

Prva skupina funkcionalnosti se odnosi na sliku profila i poveznicu za uređivanje podataka koja vodi na stranicu na kojoj korisnik može ažurirati ime, adresu, kontakt broj i lozinku. Ažuriranje se vrši slanjem *PUT* zahtjeva na API s unesenim podacima koje korisnik želi ažurirati. Slika 4.13 prikazuje sučelje navedene stranice. Na stranici se nalaze trake s trenutnim podacima i gumbom „Uredi“. Klikom na gumb, traka se proširuje i prikazuju se okviri za unos novih podataka i gumb za ažuriranje.

**Slika 4.13.** Sučelje stranice za uređivanje podataka

Druga skupina funkcionalnosti odnosi se na obavještanje korisnika o mogućnosti doniranja i prikazom datuma darivanja od kojeg korisnik može ponovno darivati krv. Sljedeći datum darivanja računa se uz pomoć biblioteke moment.js tako što se iz baze podataka povlači posljednji datum darivanja i spol korisnika po kojem se posljednjem datumu darivanja dodaje tri ili četiri mjeseca i provjerava se je li izračunati datum u prošlosti ili budućnosti. Datum se zatim prikazuje i ako je datum u prošlosti, odnosno ako zadovoljava uvjet u if petlji na slici 4.14, korisniku se prikazuje obavijest „Možete donirati krv“, ako je datum u budućnosti prikazuje se obavijest na slici 4.12. Dio funkcije računanja u kojoj se prikazuje obavijest prikazan je na slici 4.14.

```

if (this.nextDonation < now) {
  this.$refs.notifyDonation.innerText = "Možete donirati krv!"
  this.$refs.notifyDonation.style.backgroundColor = 'lightgreen'
} else {
  this.$refs.notifyDonation.innerText = "Ne možete donirati krv!"
  this.$refs.notifyDonation.style.backgroundColor = '#ffa399'
}

```

**Slika 4.14.** Prikaz dijela programskog koda za obavještanje korisnika o mogućnosti darivanja

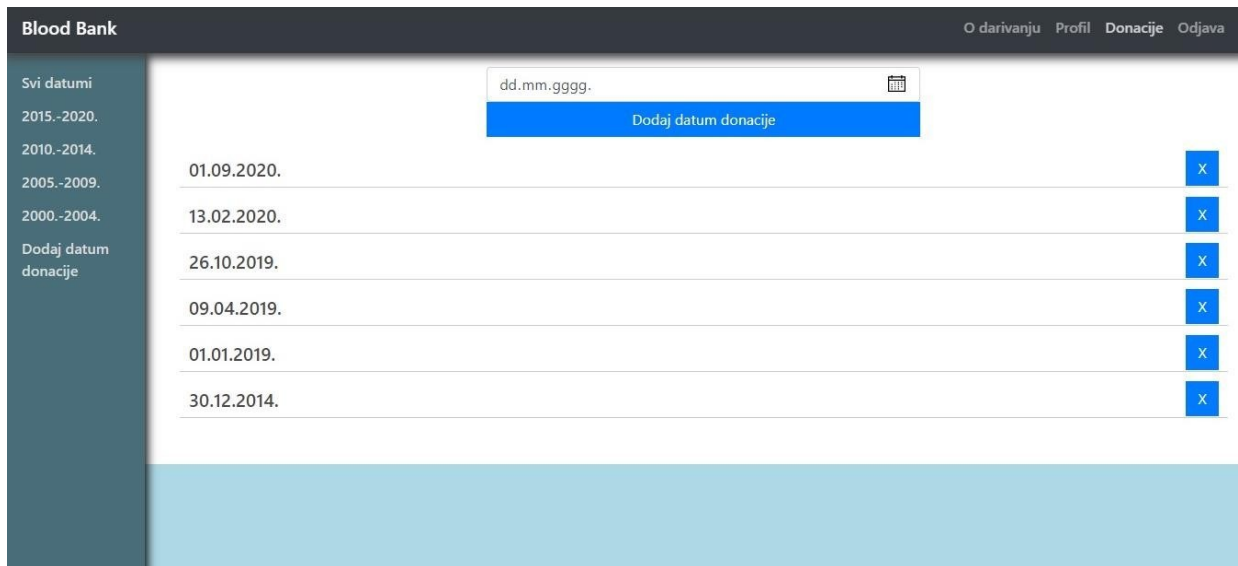
Treća skupina funkcionalnosti odnosi se na datume darivanja krvi i poveznicu na stranicu za uređivanje datuma donacija. Prikazani datumi donacija dohvaćaju se iz baze podataka i prikazuju se sortirani od najnovijeg do najstarijeg datuma.

#### 4.2.3. Stranica *Darivanja krvi*

Na stranici za uređivanje donacija (slika 4.15) korisnik može dodavati datume u bazu podataka zahtjevom *PUT*, brisati ih iz baze podataka zahtjevom *DELETE* te ih sortirati na *frontend-u*



aplikacije. Prilikom dodavanja datuma, vrši se provjera kako korisnik ne bi dodao datum iz budućnosti i kako bi razmak između dviju donacija bio pravilan, kao što je ranije navedeno, prema spolu korisnika tri ili četiri mjeseca. Navedeno računanje vrši se korištenjem biblioteke moment.js te se korisniku prikazuju upozorenja ako je datum neispravan. Stranica se sastoji od okvira za unos datuma i gumba za dodavanje datuma u bazu podataka, od popisa datuma gdje svaki element sadrži gumb za brisanje iz baze podataka i od bočne navigacijske trake na kojoj su ponuđene mogućnosti sortiranja datuma po periodu godina.



**Slika 4.15.** Sučelje stranice donacija

Na slici 4.16 prikazan je HTML programski kod u kojem se koristi biblioteka Moment.js za formatiranje i ispisivanje datuma. Funkciji *moment()* se prosleđuje *item*, odnosno datum koji se želi formatirati i ispisati i u funkciji *format()* se navodi oblik u koji je potrebno formatirati. U ovom slučaju formatira se u oblik „dd.mm.yyyy“, to jest dan, mjesec pa godina.

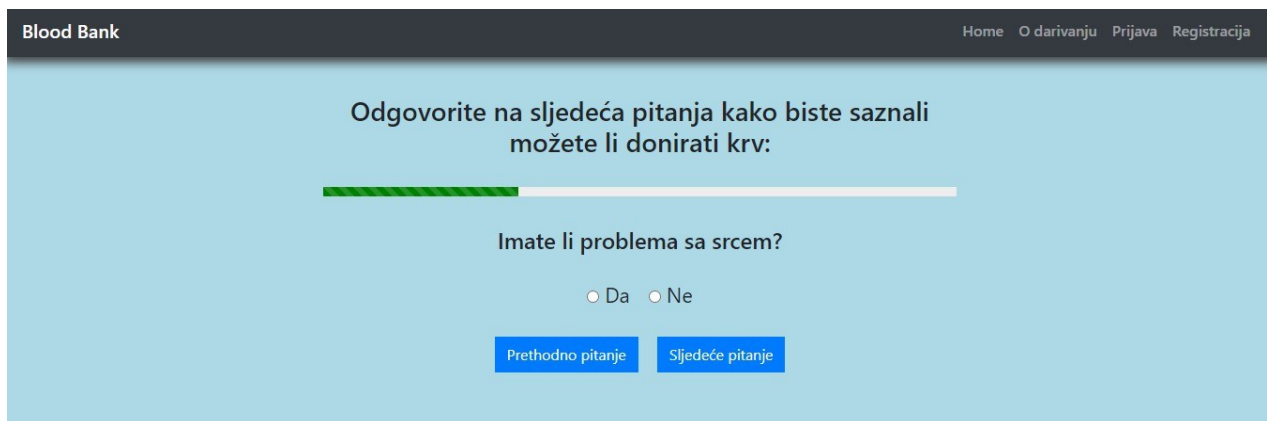
```
<div class="dateList">
  <li class="dateItem" v-for="(item,index) in lista" :key="item.id">
    {{ moment(item).format('DD.MM.YYYY.') }}
    <button class="btn btn-primary dateItemDelete" v-on:click="removeElement(index)">
      X
    </button>
  </li>
</div>
```

**Slika 4.16.** Prikaz HTML programskog koda za formatiranje i ispisivanje datuma

Dodavanje datuma na stranici donacija utječe na prikaz podataka na korisničkom profilu. Ako je dodani datum najnoviji, mijenja obavijest korisnika o mogućnosti darivanja, sljedeći datum darivanja i broj puta koliko je korisnik darivao.

#### 4.2.4. Provjera mogućnosti darivanja krvi

Provjera mogućnosti darivanja krvi sastavljena je kao kviz u kojem korisnik mora odgovoriti na trinaest pitanja nakon čega aplikacija ispisuje poruku u kojoj savjetuje korisniku da pokuša donirati krv ili da ne trenutno ne pokuša. Na slici 4.17 prikazano je sučelje provjere mogućnosti darivanja krvi. Sastoji se od naslova, trake koja prikazuje napredak, pitanja, gumbova s odgovorima i gumbova za navigaciju između pitanja. Pitanja i točni odgovori su spremljeni u polje i aplikacija prolazi kroz to polje te prebrojava koliko je točnih odgovora korisnik označio.



*Slika 4.17. Prikaz sučelja provjere mogućnosti darivanja*

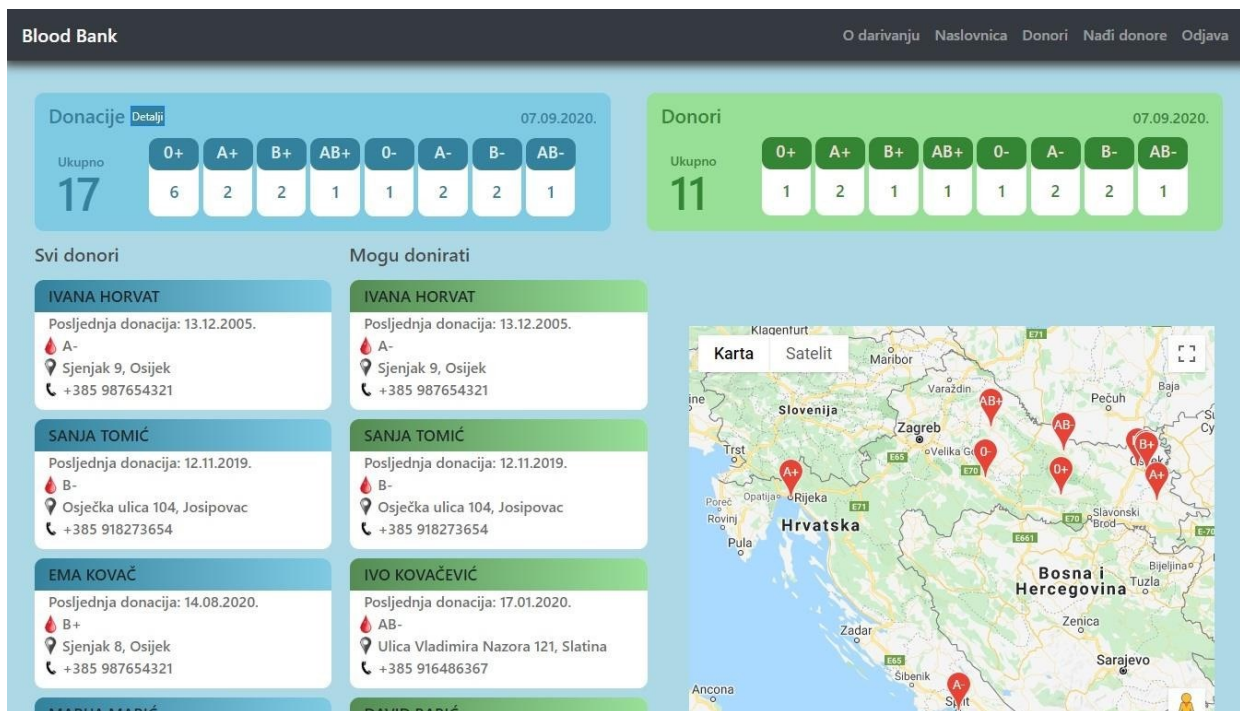
Na slici 4.18 prikazana je funkcija bodovanja provjere `score()`. Ako je korisnik označio sve točne odgovore (u funkciji je navedeno 12, jer brojanje počinje od 0), ispisuje mu se odgovarajuća poruka i prikazuje odgovarajuća slika da može donirati kao i u slučaju kada je označio manje točnih odgovora kada mu se prikazuje da ne može donirati.

```
score: function() {
  if(this.questionIndex == this.quiz.questions.length) {
    if(this.userResponses.filter(function(val) { return val }).length >= 12) {
      document.getElementById("message").innerText = this.message
      document.getElementById("bloodDrop").src= require('../assets/can-donate.png');
    } else {
      document.getElementById("bloodDrop").src= require('../assets/cant-donate.png');
    }
  }
  return this.userResponses.filter(function(val) { return val }).length;
}
```

*Slika 4.18. Prikaz dijela programskog koda provjere mogućnosti darivanja*

### 4.3. Programsko rješenje web sustava na strani poslužitelja

Ovo poglavlje odnosi se na administratorsku stranu internet sustava. Navedena strana sastoji se od naslovnice (slika 4.19), stranice sa svim darivateljima, stranice za traženje darivatelja i statistike o donacijama i darivateljima.



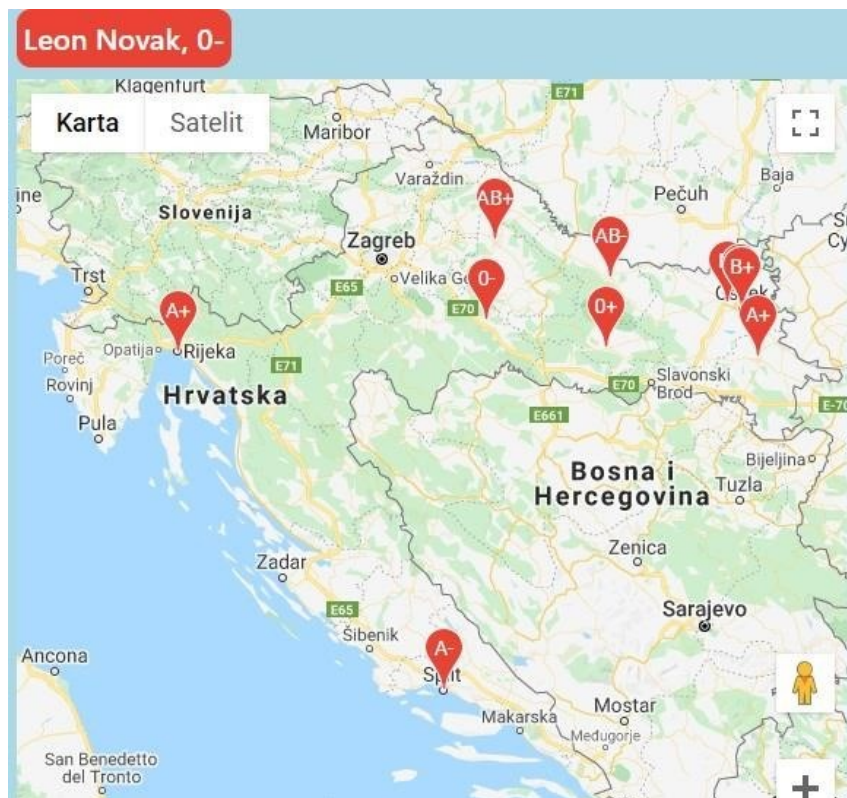
Slika 4.19. Prikaz sučelja naslovne stranice administratorske strane

Naslovnica administratorske strane sadrži opće podatke o donacijama i darivateljima i poveznicu na ostale navedene stranice. Iz baze podataka se *GET* zahtjevom dohvaćaju podaci o korisnicima u bazi podataka, spremaju se u polja iz kojih se izvlače i prikazuju. Prikazan je ukupan broj donacija kao i broj donacija svake krvne grupe posebno, ukupan broj darivatelja kao i broj darivatelja svake krvne grupe o čemu podatke dobivamo filtriranjem podataka iz baze po krvnoj grupi *filter()* funkcijom i zatim prebrojavanjem - *donationDate.length* (slika 4.20).

```
countDonations(bgroup, groupId) {
  this.temp22 = 0;
  this.temp2 = this.donors.filter(function(result) {
    return result.bloodgroup == bgroup;
  });
  for(var l=0;l<this.temp2.length;l++) {
    this.temp22 = this.temp22 + this.temp2[l].donationDate.length
  }
  document.getElementById(groupId).innerHTML = this.temp22
}
```

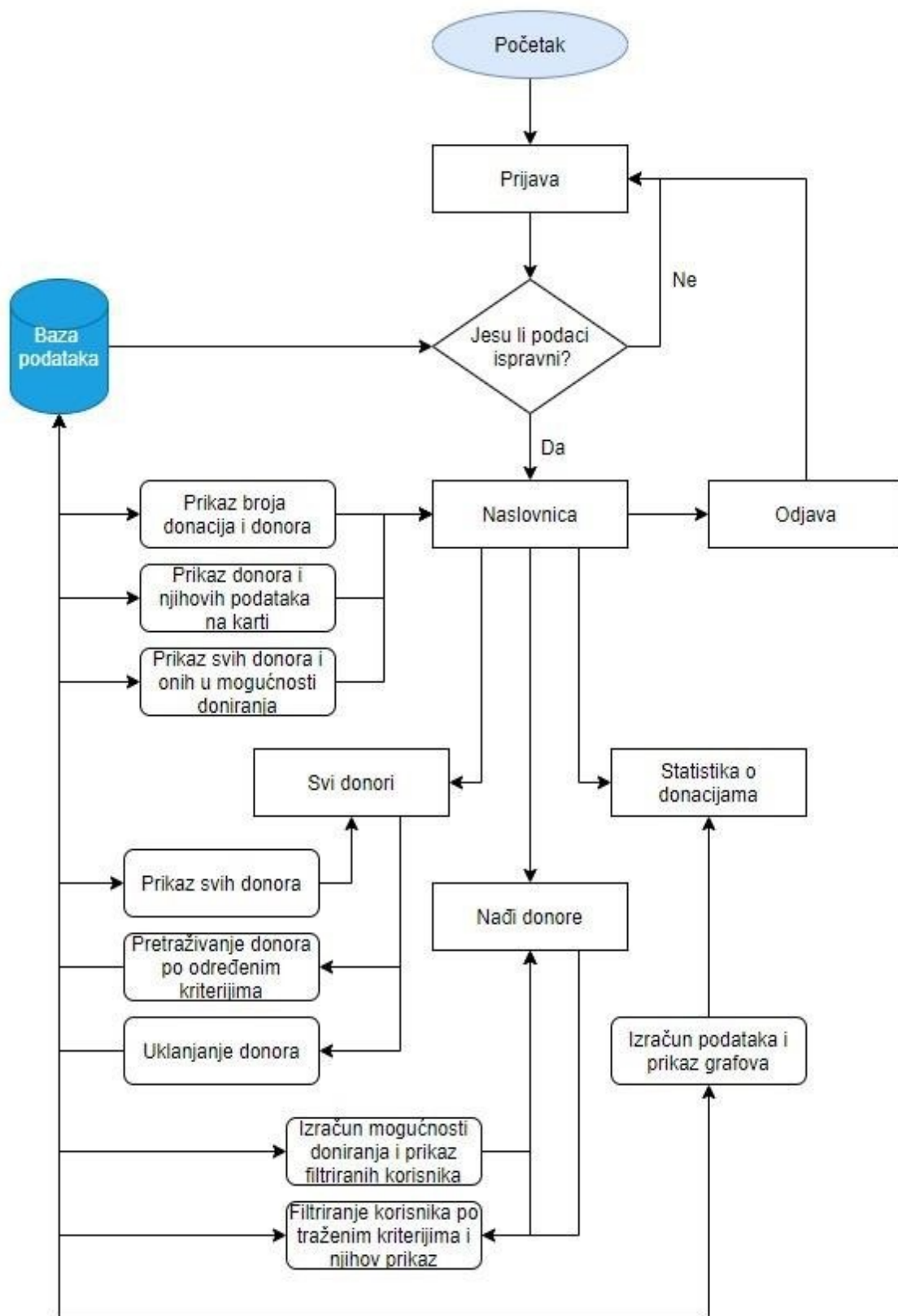
Slika 4.20. Prikaz funkcije *countDonations()* za prebrojavanje donacija

Također, na naslovnoj stranici prikazano je nekoliko korisnika iz baze podataka i nekoliko korisnika koji su trenutno u mogućnosti darivati krv. Kako bi se prikazali korisnici koji trenutno mogu darivati krv, vrše se izračuni po spolu i koristeći datum posljednje donacije korisnika uz pomoć biblioteke moment.js. Uz navedene prikaze korisnika nalaze se poveznice na odgovarajuće stranice. Također, naslovnica sadrži interaktivnu kartu (slika 4.21) na kojoj se nalaze markeri s oznakom krvne grupe i mjestom na kojem se nalazi ta krvna grupa te se prijelazom preko markera ispisuje ime, prezime i krvna grupa darivatelja na tom mjestu. Za prikaz karte korištena je biblioteka vue2-google-maps. Na naslovnici se nalazi i poveznica za stranicu na kojoj je prikazana statistika o donacijama i darivateljima.



*Slika 4.21. Prikaz interaktivne karte s darivateljima*

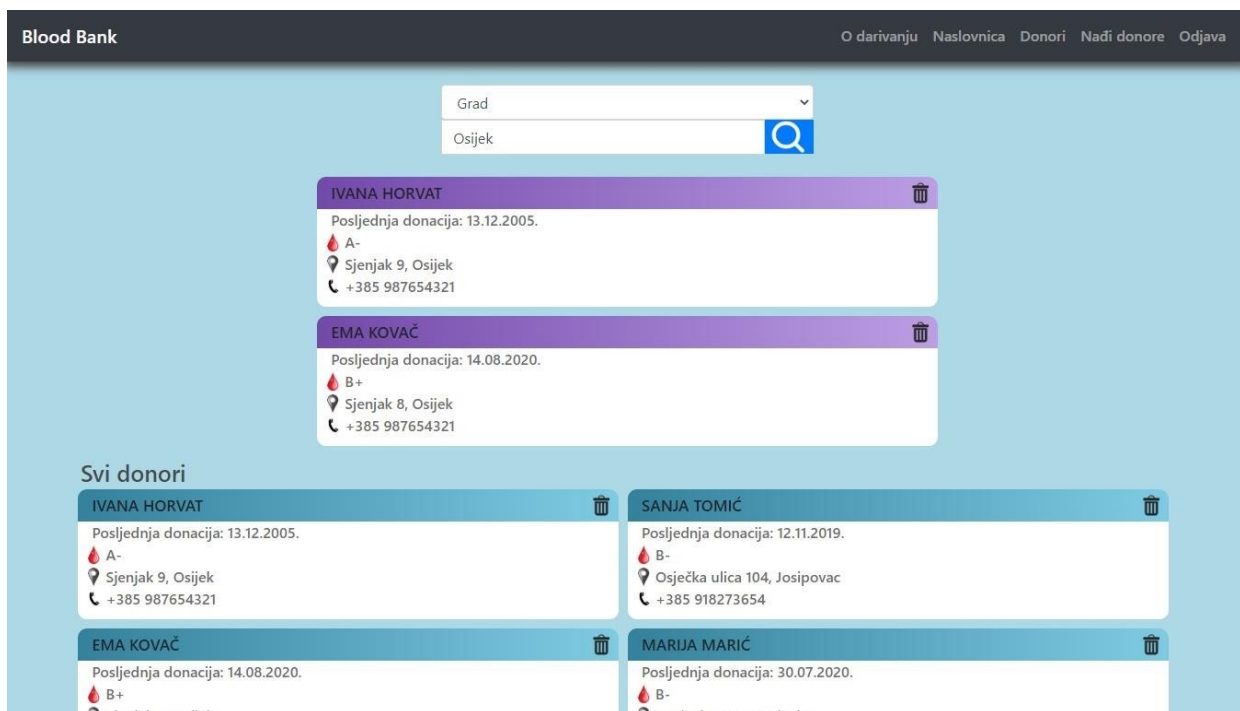
Slika 4.22 prikazuje dijagram toga administratorske strane aplikacije. U dijagramu toka nije navedeno registriranje budući da su podaci za administratora unaprijed određeni i nije potrebno registriranje.



*Slika 4.22. Dijagram toka administratorske strane internet aplikacije*

### 4.3.1. Stranica *Svi donori*

Stranica „Svi donori“ (slika 4.23) prikazuje kartice darivatelja koje sadrže podatke ime i prezime, datum posljednje donacije, krvnu grupu, adresu i kontakt broj. Podaci se dohvaćaju iz baze podataka i njihov prikaz se uređuje kako bi bili pregledni korisniku. Stranica služi za pretraživanje korisnika i brisanje korisnika iz baze podataka. Na stranici se nalazi padajući izbornik u kojem administrator bira po kojem podatku želi pretražiti bazu korisnika, a ponuđeni su mu svi navedeni na početku ovog poglavlja i na stranici se također nalazi okvir za unos podatka koji se želi pronaći.



*Slika 4.23. Prikaz sučelja stranice „Svi donori“*

Na kartici darivatelja nalazi se ikona kante za smeće pomoću koje, klikom na nju, administrator uklanja korisnika iz baze podataka. Nakon što pritisne ikonu, aplikacija traži potvrdu brisanja kako se ne bi dogodilo slučajno brisanje korisnika krivim klikom. Tijekom brisanja iz baze podataka izvršava se funkcija *deleteUser()* koja čita identifikacijsku niz odabranog korisnika i šalje zahtjev *DELETE* (slika 4.24). Identifikacijski niz se zatim traži u bazi podataka i uklanja se korisnik kod kojeg je navedeni niz pronađen.

```

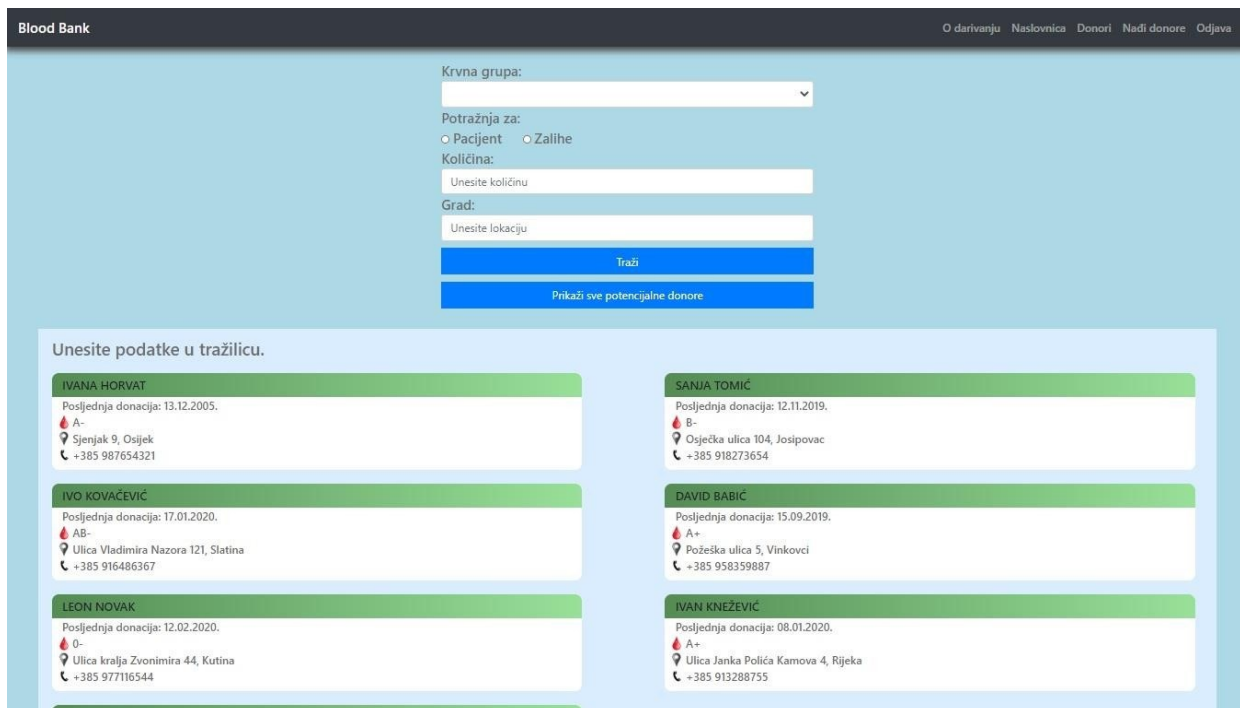
async deleteUser({
  commit
}, userData) {
  try {
    commit('users_update_request');
    let res = await axios.delete(`http://localhost:5000/api/users/${userData._id}`, {
      data: userData
    });
    if (res.data.success !== undefined) {
      commit('users_update_success');
    }
    return res;
  } catch (err) {
    commit('users_update_error', err);
  }
},

```

**Slika 4.24.** Prikaz funkcije za brisanje darivatelja

### 4.3.2. Stranica Nađi donore

Glavni dio administratorske strane aplikacije nalazi se na stranici „Nađi donore“ (slika 4.25), a služi za upravljanje zalihama krvi i potrebama krvi za pacijente.

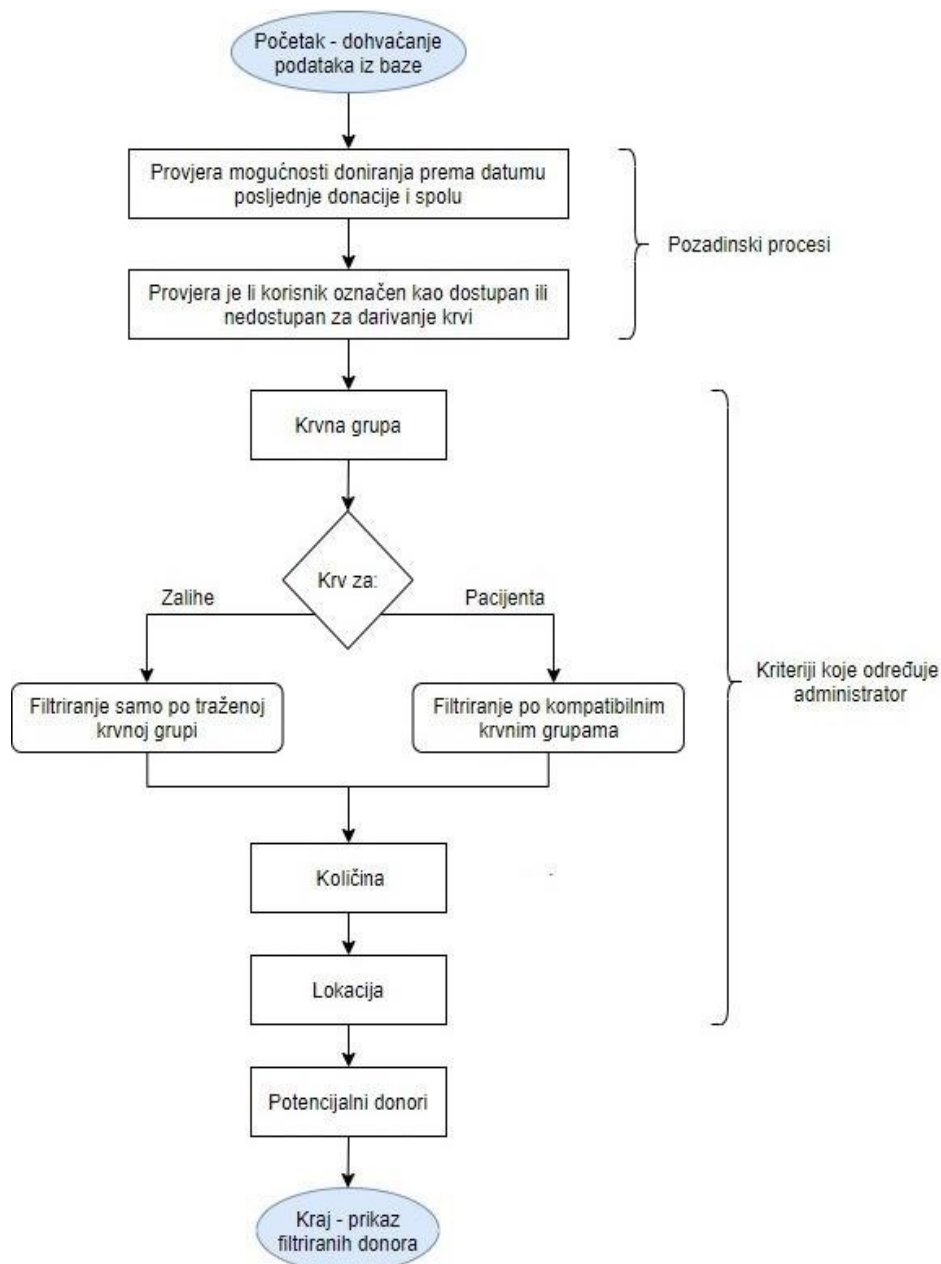


**Slika 4.25.** Prikaz sučelja stranice „Nađi donore“

Kao i na stranici „Svi donori“ opisanoj u poglavlju 4.3.1, stranica „Nađi donore“ prikazuje kartice darivatelja kako bi podaci bili pregledni. Darivatelji prikazani na ovoj stranici filtrirani su po mogućnosti darivanja koja se određuje prvo računanjem je li od posljednje donacije prošao dovoljan period i zatim po korisničkoj dostupnosti koju korisnik označava na profilu. Ovi darivatelji spremaju se u polje koje se zatim pretražuje i filtrira po kriterijima koje navodi

administrator. Administrator ima mogućnost biranja potrebne krvne grupe, označavanja namjene krvi ili za zalihe ili za pacijenta, određivanja potrebne količine i određivanja lokacije na kojoj je krv potrebna. Nakon što administrator odredi sve kriterije, klikom na gumb pretrage prikazuju se kartice filtriranih darivatelja koje zatim administrator kontaktira kako bi se nabavila krv. Također, administrator ima, klikom na gumb „Prikaži sve potencijalne donore“, pregled svih darivatelja koji su trenutno u mogućnosti darivati krv.

Tijek filtriranja po navedenim kriterijima prikazan je na slici 4.26, a programsko rješenje i detaljniji opis filtriranja pomoću pojedinog kriterija objašnjen je u sljedećem poglavlju.



**Slika 4.26.** Prikaz tijeka višekriterijskog odlučivanja



#### 4.3.2.1. Programsko rješenje višekriterijskog odlučivanja

Hijerarhijsko višekriterijsko odlučivanje ove aplikacije sastoji se od šest kriterija koji filtriraju darivatelje redom po važnosti.

Prvi kriterij je vremenski kriterij odnosno računanje je li za korisnika od posljednjeg datuma darivanja do datuma pretraživanja prošao dovoljan period da bi mogao ponovo darivati krv. Filtriranje korisnika po ovom kriteriju odvija se u pozadini aplikacije te administrator ne utječe na ovaj kriterij. Kada se iz baze podataka povlače podaci o korisnicima za prikaz, u aplikaciji se uzima datum pretraživanja i datum posljednje donacije darivatelja te se uspoređuje period između ta dva datuma. Potrebno je da period između ta dva datuma, za muškarce, bude tri mjeseca i, za žene, četiri mjeseca. Na slici 4.27 prikazan je dio programskog koda u kojem se izvodi računanje je li od posljednjeg datuma darivanja *donation1* prošao dovoljan period i je li datum koji korisnik pokušava dodati u budućnosti što je prikazano u if petlji s *nextDonate < now*.

```
for(var i=0; i<this.allDonors.length;i++){
  const donation1 = new Date(this.allDonors[i].donationDate[0])
  var donationMili = donation1.getTime();

  if(this.allDonors[i].sex == "Muškarac"){
    donationMili += 7776000000
  }else if(this.allDonors[i].sex == "Žena") {
    donationMili += 10368000000
  }

  var nextDonate = new Date(donationMili)
  var now = new Date()

  if(nextDonate < now) {
    this.tempCan.push(i)
  }
}

for(var j=0;j<this.tempCan.length;j++){
  this.potentialDonors[j] = this.allDonors[this.tempCan[j]]
}
```

**Slika 4.27.** Prikaz dijela programskog koda za računanje vremenskog kriterija

Sljedeći kriterij po važnosti određuje se na korisničkom profilu. Kao što je objašnjeno u poglavlju 4.2.2, korisnik ima mogućnost označiti je li trenutno dostupan za pozive na darivanje krvi ili nije. Filtriranje po ovom kriteriju, prikazano na slici 4.28, također se odvija u pozadini aplikacije, to jest administrator nema utjecaj na ovaj kriterij, a filtrirani darivatelji koji su vraćeni *result.avaliabile* provjerom se spremaju u polje *potentialDonors* koje se zatim dodatno filtrira prema ostalim kriterijima.

```
this.potentialDonors = this.potentialDonors.filter(function(result) {
  return result.avaliabile === "avaliabile";
});
```

**Slika 4.28.** Prikaz programskog koda filtriranja po dostupnosti

Ostale kriterije određuje administrator, a oni su prvo krvna grupa. Krvna grupa je najvažniji kriterij zbog kompatibilnosti krvnih grupa, to jest zbog toga što određenim krvnim grupama odgovaraju samo određene krvne grupe, a administrator odabire onu za koju je potrebno pronaći kompatibilne darivatelje. Iz polja darivatelja dobivenog prijašnjim filtriranjem, pretražuju se darivatelji preko podataka odabranog u padajućem izborniku, to jest preko krvne grupe. Ovisno o kriteriju je li krv potrebna za pacijenta ili zalihe, određeno je koje će se sve krvne grupe prikazivati. Slika 4.29 prikazuje funkciju filtriranja za krvnu grupu AB+ u slučaju potraživanja krvi za pacijenta kada se pretražuju sve krvne grupe jer je AB+ kompatibilna sa svim krvnim grupama i za lokacije unutar deset kilometara.

```
else if(grupa == "AB+") {
this.filterDonors("0-", "0+", "A-", "A+", "B-", "B+", "AB-", "AB+", 10000)
```

*Slika 4.29. Prikaz dijela programskog koda filtriranja po kriteriju krvne grupe*

Razlog potraživanja krvne grupe – za pacijenta ili zalihe, nakon krvne grupe, sljedeći je najvažniji kriterij. Ako administrator označi da se krv potražuje za zalihe, aplikacija izbacuje korisnike filtrirane samo po traženoj krvnoj grupi, a ako je namjena krvi za pacijenta, aplikacija filtrira sve kompatibilne krvne grupe. Na slici 4.30 prikazana je funkcija pretrage korisnika u kojoj je određeno da svi kriteriji moraju biti ispunjeni ili se prikazuje obavijest da ih treba ispuniti i ako je razlog potraživanja krvi za pacijenta, poziva se funkcija *criticalSearch()*, a ako je krv za zalihe funkcija *supplySearch()*.

```
searchForDonor() {
  if(this.bloodGroup && this.bloodAmount && this.bloodPlace != "" && this.use != "")
  {
    if(this.use == "supply") {
      this.supplySearch()
    } else {
      this.criticalSearch()
    }
  }
  }else {
    alert("Unesite sve podatke!")
  }
},
```

*Slika 4.30. Prikaz programskog koda kriterija razloga potražnje krvi*

Sljedeći kriterij koji administrator treba odrediti je količina krvi koja je potrebna. Određuje se kao broj koji se odnosi na jedinice krvi, odnosno budući da darivatelj donira jednu jedinicu krvi, odnosi

se na broj darivatelja. Iz dosadašnjih filtriranih korisnika spremljenih u polje *list* prikazuje se onoliko kartica korisnika koliko administrator označi da je potrebno količine krvi. Slika 4.31 prikazuje programski kod u kojem se iz polja darivatelja uzima onoliko darivatelja koliko je administrator odabrao funkcijom *slice()* i ako trenutno postoji manji broj od odabranog, prikazuje se obavijest, odnosno *alert()* koliko korisnika koji zadovoljavaju kriterije trenutno postoji.

```
this.list = this.list.slice(0,this.bloodAmount)
if(this.list.length < this.bloodAmount) {
  alert("Trenutno postoji samo "+this.list.length+" rezultata koji zadovoljavaju kriterije.")
}
```

**Slika 4.31.** Prikaz programskog koda za kriterij količine

Posljednji kriterij po kojem se darivatelji filtriraju je lokacija. Administrator unosi lokaciju na kojoj je potrebna krv i aplikacija računa udaljenost korisnika od tražene lokacije. Iz baze podataka izvlače se podaci zemljopisne širine i zemljopisne dužine adrese korisnika (*result.residence[]*), a iz okvira za unos lokacije u aplikaciji uzimaju se zemljopisna širina i dužina tražene lokacije funkcijama *location.lat()* i *location.lng()* i uz pomoć *vue2-google-maps* biblioteke, funkcijom *computeDistanceBetween()*, računa se udaljenost između navedene dvije lokacije. Aplikacija prvo filtrira korisnike koji se nalaze unutar kruga od deset kilometara od tražene lokacije (*dist < 10000*) i zatim, ako u tom krugu nema korisnika, povećava krug pretrage za po deset kilometara sve dok se ne pronađu darivatelji koji zadovoljavaju sve kriterije i najbliži su traženoj lokaciji. Slika 4.32 prikazuje programski kod u kojem se potencijalni darivatelji filtriraju prema traženoj krvnoj grupi, računa se udaljenost između tražene lokacije i lokacije darivatelja i filtriraju se oni u krugu od deset kilometara.

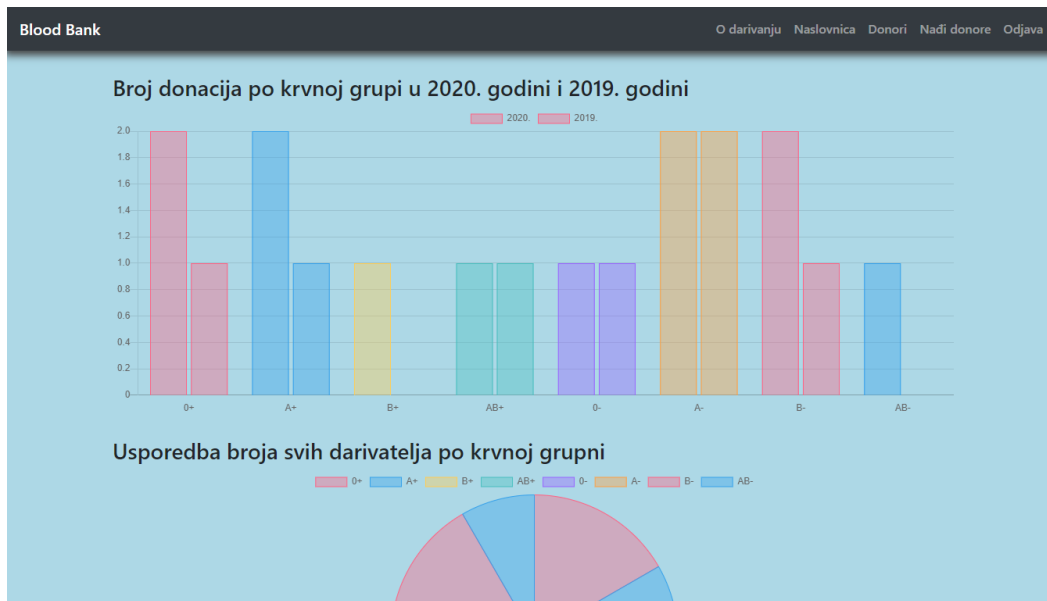
```
if (this.searchPlace) {
  var a = new google.maps.LatLng(this.searchPlace.geometry.location.lat(), this.searchPlace.geometry.location.lng());
}
this.list = this.potentialDonors.filter(function(result) {
  var b = new google.maps.LatLng(result.residence[5], result.residence[6]);
  var dist = google.maps.geometry.spherical.computeDistanceBetween(a,b);
  return result.bloodgroup === grupa && dist < 10000;
});
```

**Slika 4.32.** Prikaz dijela programskog koda za prostorni kriterij

U slučaju da administrator ne unese sve podatke, prikazano mu je upozorenje koje govori da treba unijeti sve podatke.

### 4.3.3. Stranica *Statistika*

Na stranici statistike, obrađuju se i prikazuju podaci o donacijama i darivateljima. Slika 4.33 prikazuju sučelje stranice, a na njoj se nalaze tri grafikona – stupčasti, tortni i linijski grafikon. Stupčasti grafikon prikazuje usporedbu u broju donacija po krvnim grupama iz 2019. i 2020. godine, tortni grafikon prikazuje odnose broja darivatelja i linijski grafikon prikazuje porast ili pad u broju donacija kroz posljednje četiri godine. Svi grafikoni izrađeni su korištenjem vue-chartjs biblioteke i interaktivni su.



**Slika 4.33.** Prikaz sučelja stranice statistike

Slika 4.34 prikazuje programski kod tortnog grafikona, to jest funkciju kojom se prebrojavaju darivatelji i postavlja se vrijednost (*datasets[]*) grafikona. Po krvnoj grupi se filtrira polje s darivateljima, zatim se dužina novog polja (*filters.length*), odnosno broj darivatelja sprema u podatke grafikona.

```
countBloodgroups(group,place) {  
  this.filters = this.lista.filter(function(result) {  
    return result.bloodgroup === group  
  })  
  this.chartData.datasets[0].data[place] = this.filters.length  
}
```

**Slika 4.34.** Prikaz dijela programskog koda tortnog grafikona

## **5. PRIKAZ NAČINA RADA I ISPITIVANJE WEB SUSTAVA ZA UPRAVLJANJE ZALIHAMA KRVI**

U ovom poglavlju opisan je način korištenja web aplikacije, navedeni su ispitni slučajevi i prikazani su njihovi rezultati te je detaljno analiziran postupak višekriterijskog odlučivanja.

### **5.1. Opis načina rada web sustava za upravljanje zalihama krvi**

Web sustav za upravljanje zalihama krvi pokreće se korištenjem cmd-a (*Command prompt*). Potrebno je unutar mape projekta pokrenuti cmd i naredbu „npm install“. Nakon instalacije, potrebno je unutar mape pokrenuti naredbu „npm run server“ kako bi se pokrenula baza podataka, to jest server dio projekta. Također, potrebno je otvoriti još jedan prozor cmd-a i naredbom „cd client“ prijeći u *client* mapu u kojoj je potrebno pokrenuti naredbu „npm run serve“. Nakon izvršenja navedenih koraka, aplikacija se pokreće na adresi <http://localhost:8080/> gdje se može koristiti. Za pokretanje navedenih naredbi potrebno je na računalu imati instaliran Node.js.

Za korištenje cijele aplikacije s korisničke strane, potrebno je registriranje, to jest izrada korisničkog profila jer je aplikacija namijenjena osobama koje su već darivale krv kako bi imale uvid u svoje donacije i kako bi administratorska strana imala podatke o darivateljima. Informativni dio aplikacije i provjera mogućnosti darivanja može se koristiti bez registriranja. Nakon registriranja, korisnik se prijavljuje u aplikaciju i ima mogućnost uređivanja profila, donacija i označavanja dostupnosti za pozive na doniranje te ima uvid u mogućnost darivanja, broj darivanja i datuma sljedećeg darivanja. Tijek korištenja klijentske strane aplikacije prikazan je u poglavlju 4.2 na slici 4.5.

Administrator se u aplikaciju prijavljuje s jedinstvenim korisničkim imenom – admin i lozinkom – admin te ima pristup strani aplikacije koja običnim korisnicima nije dostupna. Administrator ima mogućnost uređivanja baze podataka darivatelja, pretraživanja donora te uvid u osobne podatke darivatelja, mogućnost kontaktiranja darivatelja, uvid u statistiku donacija i darivatelja i slično. Tijek korištenja administratorske strane aplikacije prikazan je u poglavlju 4.3 na slici 4.22.

### **5.2. Postavke ispitivanja, korisnički i ispitni slučajevi**

Ispravnost rada glavnih funkcionalnosti aplikacije s korisničke i s administratorske strane provjerena je testovima čiji se opis i parametri ispitivanja nalaze u nastavku, a rezultati su prikazani u poglavlju 5.3.

Registriranje je ispitano unosom ispravnih i neispravnih podataka. Kod unosa ispravnih podataka, korisnik je usmjeren na stranicu prijava, a kod unosa neispravnih podataka (npr. kada osoba nije punoljetna) dobije obavijest o pogrešnom unosu.

Kod prijave se ispituje postoji li korisničko ime u bazi podataka i, ako postoji, uspoređuje se unesena lozinka s onom iz baze podataka. Ako korisničko ime ne postoji ili lozinka nije ispravna, korisnik dobije obavijest i treba unijeti ispravne podatke, a ako su podaci ispravni, usmjeren je na stranicu profila.

Na korisničkom profilu ispituje se rad postavljanja dostupnosti korisnika – dostupan ili nedostupan za darivanje. Ako je korisnik označen kao dostupan, iz baze podataka se izvlači taj podatak i ime korisnika na profilu će biti zelene boje, a u suprotnom – nedostupan, crvene boje.

Kod donacija korisnika ispituje se mogućnost unošenja budućih datuma darivanja i unošenje datuma između kojih nije prošao dovoljan period za darivanje. Ako korisnik pokuša unijeti neispravne podatke, dobije obavijest, a ako su podaci ispravni, datum se prikazuje na listi.

Kod pretraživanja baze korisnika ako administrator ne unese podatke pretraživanja dobije obavijest, a u suprotnom su mu prikazani darivatelji koji zadovoljavaju kriterije pretrage ili, ako ih nema, dobije obavijest da ne postoje takvi darivatelji. Kod brisanja iz baze podataka, ako administrator obriše korisnika, taj korisnik treba nestati s liste darivatelja što je ujedno i ispitivanje ispravnosti rada ove funkcije.

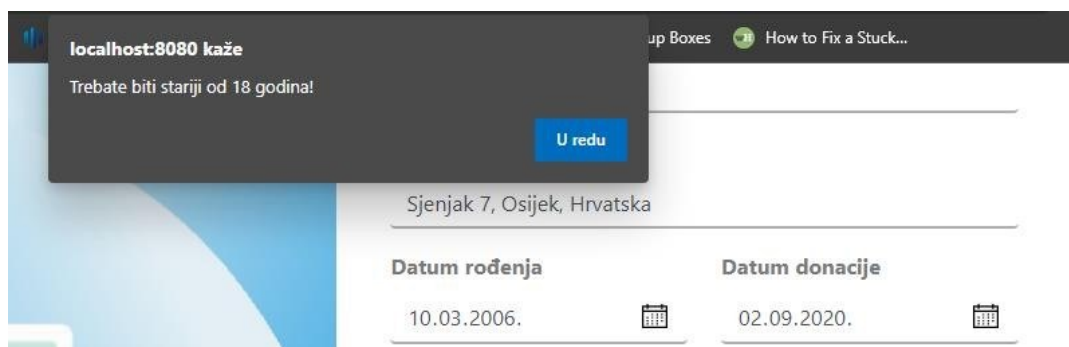
Nazivi i postavke ispitivanja navedeni su u tablici 5.1. Ispitivanje postupka višekriterijskog odlučivanja objašnjeno je i prikazano u poglavlju 5.4.

**Tablica 5.1.** Nazivi, postavke i rezultati testova web aplikacije

Redni broj	Naziv	Postavke	Rezultat (uspješno +/- neuspješno -)
<b>KORISNIK</b>			
1	Test registriranja korisnika	Osoba nije punoljetna	+
2	Test prijave korisnika	Pogrešno korisničko ime	+
3	Test okvira za označavanje dostupnosti korisnika	Korisnik je nedostupan	+
4	Test unosa datuma donacija	Unos budućeg datuma	+
5	Test unosa datuma donacija	Unos datuma s nedovoljnim razmakom	+
<b>ADMIN</b>			
6	Test pretraživanja korisnika po određenim kriterijima	Kriteriji nisu uneseni	+
7	Test brisanja korisnika iz baze podataka	Korisnik se briše iz baze	+

### 5.3. Rezultati ispitivanja web sustava za upravljanje zalihama krvi s analizom

Slika 5.1 prikazuje ishod nakon pokušaja registriranja korisnika koji nije punoljetan. Nakon prikaza obavijesti, korisnik ostaje na stranici registriranja.



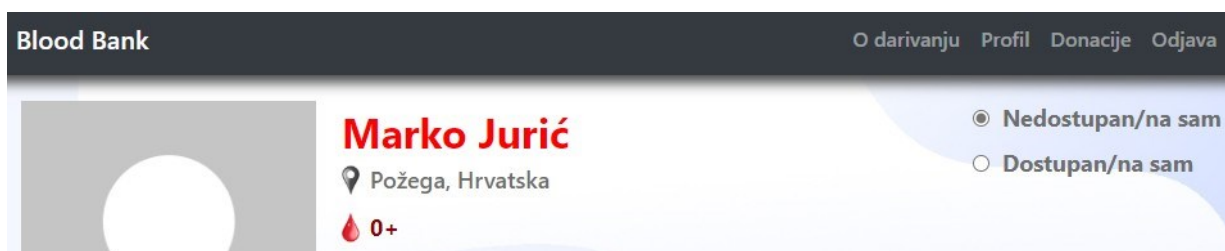
*Slika 5.1. Prikaz obavijesti kod pokušaja registriranja osobe koja nije punoljetna*

Na slici 5.2 prikazana je obavijest koja se prikazuje korisniku u slučaju pokušaja prijave s nepostojećim korisničkim imenom. Nakon prikaza obavijesti, korisnik ima mogućnost ponovnog pokušaja prijave sve dok ne unese ispravno korisničko ime i odgovarajuću lozinku.



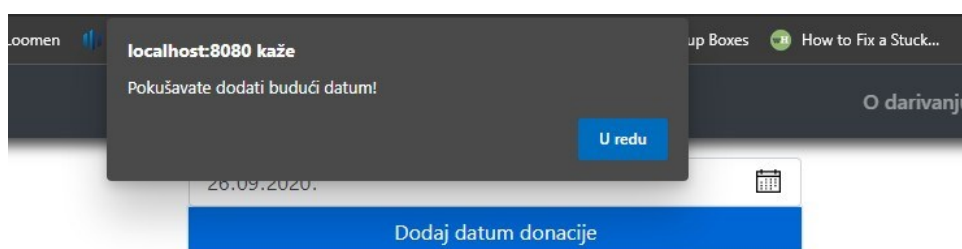
*Slika 5.2. Prikaz obavijesti kod pokušaja prijave s nepostojećim korisničkim imenom*

Rezultat provjere označavanja korisnika nedostupnim prikazan je na slici 5.3. Budući da je ime korisnika crvene boje i označen je kao nedostupan, zaključuje se da okvir za označavanje dostupnosti ispravno funkcionira. Također je provjereno nalazi li se korisnik na administratorskoj strani na popisu potencijalnih darivatelja.



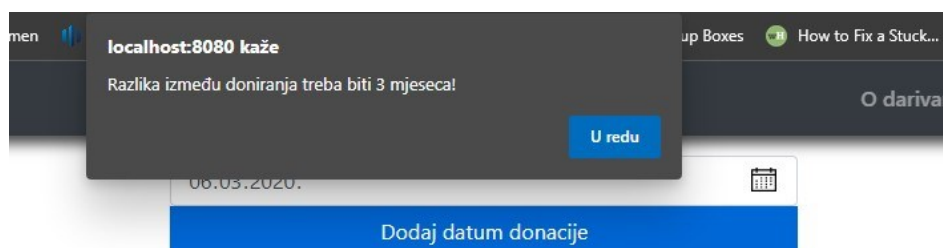
*Slika 5.3. Prikaz korisničkog imena u slučaju kada je korisnik označen kao nedostupan*

Kod dodavanja datuma donacije, u slučaju pokušaja unosa budućeg datuma, korisniku se prikazuje obavijest na slici 5.4, nakon čijeg prikaza korisnik ima mogućnost ponovnog unosa datuma. Obavijest se prikazuje sve dok korisnik ne unese ispravan datum.



*Slika 5.4. Prikaz obavijesti kod pokušaja unosa budućeg datuma donacije*

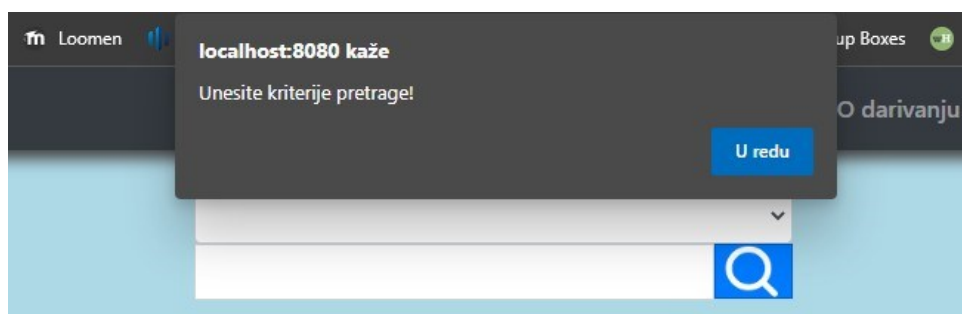
Slika 5.5 prikazuje obavijest kod pokušaja unosa datuma za koji nije prošao dovoljan period od ostalih datuma u bazi podataka. U ovom slučaju obavijest je prikazana za tri mjeseca jer se radi o korisniku čiji je spol muškarac. Nakon prikaza obavijesti, korisnik ima mogućnost ponovnog unosa datuma. Obavijest se prikazuje sve dok korisnik ne unese ispravan datum.



*Slika 5.5. Prikaz obavijesti kod pokušaja unosa datuma s nedovoljnim razmakom od postojećih datuma u bazi*

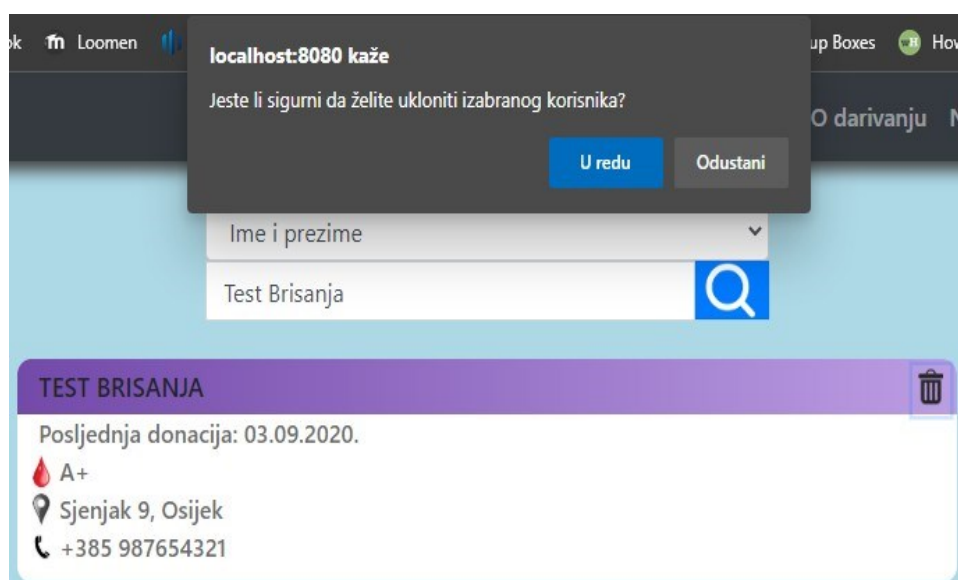
Rezultat ispitivanja pretrage korisnika na administratorskoj strani prikazan je na slici 5.6. Administratoru je prikazana obavijest nakon koje treba unijeti potrebne kriterije pretrage.





*Slika 5.6. Prikaz obavijesti kod pretrage bez unesenih kriterija*

Slika 5.7 prikazuje rezultate pokušaja brisanja korisnika iz baze podataka. Korisniku se prikazuje okvir u kojem treba potvrditi brisanje i nakon toga se korisnik uklanja s liste prikaza i iz baze podataka.



*Slika 5.7. Prikaz brisanja korisnika iz baze podataka*

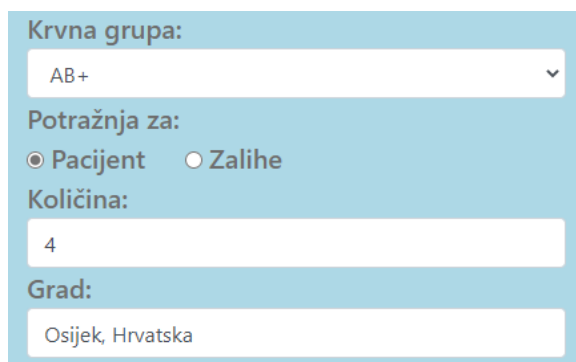
Rezultati provedenih testova, prikazani u tablici 5.1, ukazuju na to da web sustav za upravljanje zalihama krvi ispunjava korisničke zahtjeve i zadane funkcionalnosti izvršava ispravno budući da su svi testovi uspješno izvršeni.

#### **5.4. Ispitivanje i analiza korištenih postupaka višekriterijskog odlučivanja**

Postupak višekriterijskog odlučivanja ispitan je u slučaju potraživanja krvi za pacijenta i u slučaju potraživanja za zalihe.

### 5.4.1. Ispitivanje potraživanja krvi za pacijenta

U slučaju potraživanja za pacijenta korišteni podaci su bili AB+ krvna grupa, potrebna količina jedinica krvi - četiri i mjesto s obzirom na koje se traže darivatelji – Osijek (Slika 5.8). U ovom slučaju aplikacija prikazuje darivatelje s kompatibilnim krvnim grupama koji se nalaze u krugu od maksimalno 100 kilometara. Darivatelji se pretražuju u ovom krugu budući da u manjem nema dovoljno darivatelja koji zadovoljavaju tražene kriterije, a za veći krug je potrebno previše vremena da bi se krv dostavila.



The screenshot shows a search form with the following fields and options:

- Krvna grupa:** A dropdown menu with "AB+" selected.
- Potražnja za:** Radio buttons for "Pacijent" (selected) and "Zalihe".
- Količina:** A text input field containing the number "4".
- Grad:** A text input field containing "Osijek, Hrvatska".

*Slika 5.8. Kriteriji korišteni u testiranju postupka višekriterijskog odlučivanja za pacijenta*

Slika 5.9 prikazuje rezultate pretraživanja. Budući da su prikazani svi mogući darivatelji koji zadovoljavaju sve kriterije pretrage, zaključuje se da postupak višekriterijskog odlučivanja u slučaju potraživanja za pacijenta radi ispravno.



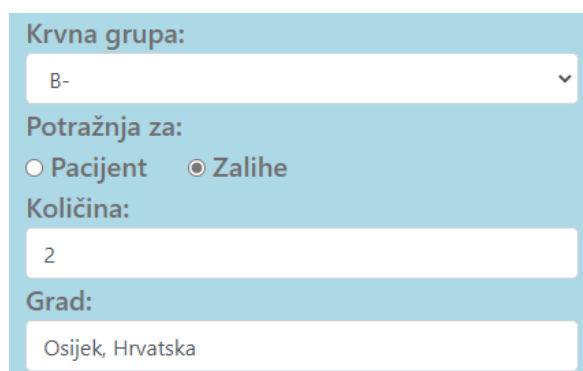
The screenshot displays search results under the heading "Potencijalni donori:" with three donor profiles:

- SANJA TOMIĆ**  
Posljednja donacija: 12.11.2019.  
B-  
Osječka ulica 104, Josipovac  
+385 918273654
- IVO KOVAČEVIĆ**  
Posljednja donacija: 17.01.2020.  
AB-  
Ulica Vladimira Nazora 121, Slatina  
+385 916486367
- DAVID BABIĆ**  
Posljednja donacija: 15.09.2019.  
A+  
Požeška ulica 5, Vinkovci  
+385 958359887

*Slika 5.9. Rezultat pretrage postupkom višekriterijskog odlučivanja za pacijenta*

### 5.4.2. Ispitivanje potraživanja krvi za zalihe

Prilikom potraživanja krvi za zalihe korišteni podaci su krvna grupa B-, tražena količina – dvije jedinice i mjesto potrebe – Osijek (Slika 5.10). U ovom slučaju aplikacija prikazuje samo darivatelje tražene krvne grupe u krugu od maksimalno 200 kilometara budući da u manjem krugu ne postoje korisnici koji mogu zadovoljiti navedene kriterije. Kod potraživanja za zalihe, maksimalan krug pretrage povećan je na duplu vrijednost kruga potraživanja za pacijenta.



Krvna grupa:  
B-

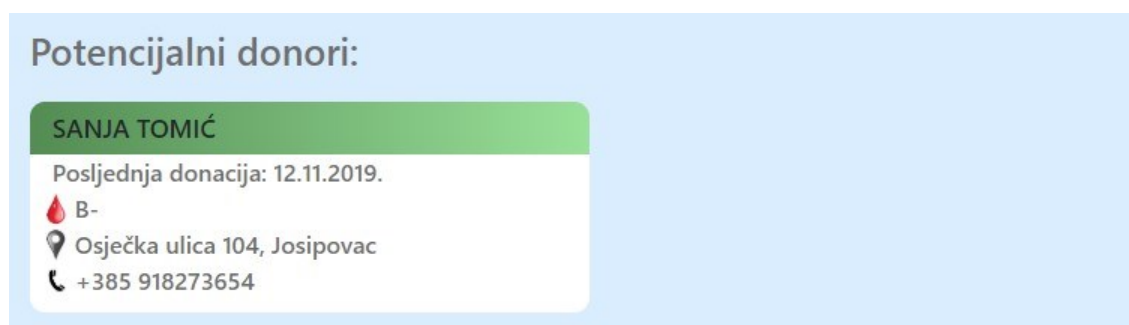
Potražnja za:  
 Pacijent  Zalihe

Količina:  
2

Grad:  
Osijek, Hrvatska

*Slika 5.10. Kriteriji korišteni u testiranju postupka višekriterijskog odlučivanja za zalihe*

Na slici 5.11 prikazani su rezultati pretrage po kriterijima sa slike 5.10. Kao potencijalni darivatelj naveden je samo jedan korisnik jer u bazi podataka ne postoji više darivatelja u zadanom krugu koji zadovoljavaju kriterije. Tijekom prikaza, korisniku se prikazuje obavijest ako je broj postojećih darivatelja manji od traženog broja količine.



Potencijalni donori:

**SANJA TOMIĆ**  
Posljednja donacija: 12.11.2019.  
B-  
Osječka ulica 104, Josipovac  
+385 918273654

*Slika 5.11. Rezultat postupka višekriterijskog odlučivanja za zalihe*

### 5.4.3. Analiza rezultata ispitivanja višekriterijskog postupka odlučivanja

Uspoređivanjem višekriterijskog odlučivanja kod potraživanja krvi za pacijenta i za zalihe utvrđene su razlike prikazane u tablici 5.2.

**Tablica 5.2.** Razlike višekriterijskog odlučivanja kod potraživanja za pacijenta i za zalihe krvi

<b>Potraživanje krvi za pacijenta</b>	<b>Potraživanje krvi za zalihe</b>
Krug pretraživanja maksimalno 100 km	Krug pretraživanja maksimalno 200 km
Rezultati su sve kompatibilne krvne grupe	Rezultati su samo tražena krvna grupa
Kao rezultat prikazuje veći broj darivatelja	Kao rezultat prikazuje manji broj darivatelja

Ispitivanjem višekriterijskog postupka odlučivanja utvrđeno je da aplikacija uspješno dohvaća podatke iz baze podataka te da uspješno vrši izračune mogućnosti darivanja, provjerava dostupnost korisnika i filtrira korisnike po zadanim kriterijima. Također, postignut je brz odziv pozadinskih procesa i sučelja iz čega zaključujemo da je uspješno implementirano hijerarhijsko višekriterijsko odlučivanje. Zbog jednostavnog korištenja, brzog odziva i preglednosti podataka, analizom je utvrđeno da aplikacija olakšava potražnju krvi i upravljanje zalihama krvi.

## 6. ZAKLJUČAK

U ovom diplomskom radu teorijski je opisana transfuziologija, funkcije odjela za transfuziologiju, uloga darivatelja krvi i postupak darivanja krvi te je na osnovi analize postojećih rješenja opisana ideja i prikazan model web sustava za upravljanje zalihama krvi. Aplikacija je izrađena u programu Visual Studio Code korištenjem Vue.js i Node.js tehnologija i MongoDB baze podataka.

Razvijeni web sustav za upravljanje zalihama krvi, namijenjen darivateljima krvi i odjelima za transfuziju, primjenjuje računalne tehnologije u zdravstvu i olakšava upravljanje podacima. Na korisničkoj strani upravlja praćenjem datuma darivanja i obavještavanjem korisnika o terminima darivanja i na administratorskoj strani upravlja darivateljima, pretraživanjem darivatelja i potražnjom zaliha krvi. Navedeni sustav se sastoji od strane poslužitelja koja radi s bazom podataka, ima zadatak pohranjivanja, obrade i upravljanja podacima i strane klijenta koja je zadužena za prikaz sučelja i omogućavanje zadanih funkcionalnosti sustava. Prikupljanje podataka o darivateljima krvi najvažnija je zadaća ovog internet sustava, a služi administratoru kako bi imao uvid u bazu darivatelja krvi i na osnovu tih podataka upravljao darivateljima, zalihama i potražnjom krvi. Glavni dio internet sustava je potražnja zaliha krvi postupkom višekriterijskog odlučivanja koje koristi vremenske i prostorne kriterije i kriterije kompatibilnosti. Ispitivanjem aplikacije utvrđeno je da su zadovoljeni svi korisnički zahtjevi, da funkcionalnosti aplikacije rade ispravno i da postupak višekriterijskog odlučivanja ima brz odziv i olakšava potražnju jedinica krvi. Također, dizajn aplikacije zadovoljava korisničke potrebe. Iz ovih razloga, aplikacija se može koristiti u odjelima za transfuziju i mogu ju koristiti dobrovoljni darivatelji krvi. Ovaj internet sustav može se unaprijediti dodavanjem mogućnosti odabira termina, to jest naručivanja darivatelja u centrima za darivanje krvi.

## LITERATURA

- [1] S. Benazić, A. Silvar, *Transfuziologija*, *Transfuziologija Glasnik pulske bolnice*, br. 1, sv. 4, str. 111-115, 2007.
- [2] P. Tiberghien, *The best blood product and its best use for each patient: An evolving role for hemovigilance?*, *Transfusion Clinique et Biologique*, br. 3, sv. 26, str. 188-191, 2019.
- [3] A. Jain, R. Kaur, *Hemovigilance and blood safety*, *Asian Journal of Transfusion Science*, br. 2, sv. 6, str. 137-138, 2012.
- [4] Hrvatski sabor, *Zakon*, *Narodne novine*, Članak 135., br. 100, 2018., dostupno na: [https://narodne-novine.nn.hr/clanci/sluzbeni/2018\\_11\\_100\\_1929.html](https://narodne-novine.nn.hr/clanci/sluzbeni/2018_11_100_1929.html), 8.9.2020.
- [5] *Zakon o krvi i krvnim pripravcima*, *Narodne novine*, 79/06 i 124/11, 2019., dostupno na: [https://narodne-novine.nn.hr/clanci/sluzbeni/full/2019\\_09\\_91\\_1814.html](https://narodne-novine.nn.hr/clanci/sluzbeni/full/2019_09_91_1814.html), 8.9.2020.
- [6] J. L. Osterman, *Blood Product Transfusions and Reactions*, *Hematology/Oncology Clinics of North America*, br. 6, sv. 31, str. 727-738, 2014.
- [7] R. Kishen, C. Naisbitt, K. Mos, *Crystalloids, colloids, blood products and blood substitutes*, *Anaesthesia & Intensive Care Medicine*, br. 6, sv. 20, str. 353-360, 2019.
- [8] *O darivanju*, HZTM, *Kreativna poslovna rješenja*, 2020., dostupno na: <https://www.hztm.hr/hr/content/2/darivanje-krvi/14/o-darivanju/>, 8.9.2020.
- [9] *Darivanje krvi - opće informacije*, 2014., dostupno na: <http://www.unicath.hr/hks2015/wp-content/uploads/2014/11/DARIVANJE-KRVI-2014.pdf>, 8.9.2020.
- [10] *Darivanje krvi*, *Udruga darivatelja krvi*, 2020., dostupno na: <http://uddk.hr/darivanje-krvi/>, 8.9.2020.
- [11] *O darivanju: Tko je dobrovoljni darivatelj krvi*, HZTM, *Kreativna poslovna rješenja*, 2020., dostupno na: <https://www.hztm.hr/hr/content/2/darivanje-krvi/14/o-darivanju/#16>, 8.9.2020.
- [12] *O darivanju: Tko može darivati krv?*, HZTM, *Kreativna poslovna rješenja*, 2020., dostupno na: <https://www.hztm.hr/hr/content/2/darivanje-krvi/14/o-darivanju/#1>, 8.9.2020.
- [13] L.S.H. Niessen-Mayer, J. Seghatchian, *Donor health assesment – When is blood donation safe?*, *Transfusion and Apheresis Science*, br. 1, sv. 58, str. 113-116, 2019.
- [14] *NHS Blood Donation*, NHS, 2020., dostupno na: <https://www.blood.co.uk/>, 10.9.2020.
- [15] *Canadian Blood Services*, Canadian Blood Services, 2020. dostupno na: <https://www.blood.ca/en>, 10.9.2020.
- [16] J. Baduraa, J. R. Figueira, M. Kadzinski, *Using a segmenting description approach in multiple criteria decision aiding*, *Expert Systems With Applications*, br. 1, sv. 147, str. 113-176, 2020.

- [17]G.W. Evans, An Overview of Techniques for Solving Multiobjective Mathematical Programs, Management Science, br. 11, sv. 30, 1984.
- [18]The MVVM Pattern, Microsoft, 2020., dostupno na: [https://docs.microsoft.com/en-us/previous-versions/msp-n-p/hh848246\(v=pandp.10\)?redirectedfrom=MSDN](https://docs.microsoft.com/en-us/previous-versions/msp-n-p/hh848246(v=pandp.10)?redirectedfrom=MSDN), 8.9.2020.
- [19]Introduction: Vue.js, Evan You, 2014-2020., dostupno na: <https://vuejs.org/v2/guide/>, 8.9.2020.
- [20] Vue.js: Overview, Tutorials Point, 2020., dostupno na: [https://www.tutorialspoint.com/vuejs/vuejs\\_overview.htm](https://www.tutorialspoint.com/vuejs/vuejs_overview.htm), 8.9.2020.
- [21] What is Vuex?, 2020., dostupno na: <https://vuex.vuejs.org/>, 8.9.2020.
- [22] Moment.js, 2020., dostupno na: <https://momentjs.com/>, 8.9.2020.
- [23]Vue-chartjs, Jakub Juszczak, 2018.-2020., dostupno na: <https://vue-chartjs.org/guide/#browser>, 8.9.2020.
- [24]Bootstrap: Introduction, Bootstrap Team, 2020., dostupno na: <https://getbootstrap.com/docs/4.5/getting-started/introduction>, 8.9.2020.
- [25] Node.js, OpenJs Foundation, 2020., dostupno na: <https://nodejs.org/en/>, 8.9.2020.
- [26]Node.js: Introduction, Tutorials Point, 2020., dostupno na: [https://www.tutorialspoint.com/nodejs/nodejs\\_introduction.htm](https://www.tutorialspoint.com/nodejs/nodejs_introduction.htm), 8.9.2020.
- [27] Express, OpenJs Foundation, 2020., dostupno na: <https://expressjs.com/>, 8.9.2020.
- [28] Passport.js, Auth0 Inc., 2020., dostupno na: <http://www.passportjs.org/>, 8.9.2020.
- [29]Learn MongoDB, JavaTpoint, 2011.-2018., dostupno na: <https://www.javatpoint.com/mongodb-tutorial>, 8.9.2020.
- [30]MongoDB features, JavaTpoint, 2011.-2018., dostupno na: <https://www.javatpoint.com/mongodb-features>, 8.9.2020.
- [31]Mongoose: Getting started, LearnBoost, 2011., dostupno na: <https://mongoosejs.com/docs/>, 8.9.2020.
- [32]Compass, MongoDB Inc., 2020., dostupno na: <https://www.mongodb.com/products/compass>, 8.9.2020.

## SAŽETAK

Cilj ovog diplomskog rada je osmisliti, modelirati i izraditi web sustav za upravljanje zalihama krvi. Izrađena aplikacija sastoji se od korisničke i administratorske strane. Na korisničkoj strani omogućeno je praćenje donacija i obavještanje korisnika o mogućnosti darivanja krvi te prikupljanje informacija o darivateljima. Na administratorskoj strani omogućeno je upravljanje darivateljima i zalihama krvi, odnosno osiguravanje dostupnosti krvnih jedinica za transfuzije uz pomoć višekriterijskog odlučivanja. Korišteni postupak višekriterijskog odlučivanja uzima u obzir vremenske i prostorne kriterije, kriterije kompatibilnosti po krvnim grupama, količini i svrsi za koju je krv potrebna. Sigurnost korisničkih podataka postignuta je obaveznom autentifikacijom bez koje korisnik nema mogućnost korištenja internet aplikacije. Aplikacija je ostvarena uz korištenje Vue.js-a za korisničku stranu, Node.js-a za poslužiteljsku stranu, lokalne MongoDB baze podataka i prikladnih dodataka i biblioteka. Na temelju analize rada aplikacije i izvršenih ispitivanja utvrđeno je da aplikacija ispunjava sve korisničke zahtjeve i radi ispravno te da omogućuje brzo i jednostavno pretraživanje i dobavljanje jedinica krvi. Njene prednosti su jednostavnost korištenja, prilagođeni dizajn, to jest preglednost podataka i svrha kojoj aplikacija služi.

**Ključne riječi:** darivanje krvi, MVVM, transfuziologija, upravljanje zalihama krvi, web aplikacija.



## **ABSTRACT**

**Title: Web system for blood stock management by multicriteria decision-making procedures**

The aim of this thesis was to design, model and create a web system for blood supply management. The created application consists of user and administrator sides. On the user's side, it is enabled to monitor donations and to inform users about the possibility of donating blood and to collect information about donors. On the administrator's side, it is enabled to manage blood donors and supplies, that is, to ensure the availability of blood units for transfusions using the multicriteria decision-making procedures. The used multicriteria decision-making procedures take into account temporal and spatial criteria, criteria of blood compatibility by blood groups and quantity and of the criteria of the purpose for which the blood is needed. The protection of user data is achieved by mandatory authentication, without which the user does not have the ability to use the web application. The application development was achieved with the use of Vue.js for frontend, Node.js for backend, local MongoDB database and other suitable plugins and libraries. Based on the analysis of the application operation and performed tests, it was determined that the application meets all user requirements and works properly and that it allows quick and easy searching and getting blood units. The advantages of the application are the ease of use, custom design, i.e. clear interface design and the purpose which the application serves.

**Key words:** blood donation, blood stock management, MVVM, transfusiology, web application.

## **ŽIVOTOPIS**

Mihaela Dijanešić je rođena 19. srpnja 1996. godine u Virovitici. Osnovnoškolsko obrazovanje stječe u Osnovnoj školi Josipa Kozarca u Slatini od 2003. do 2011. godine. Upisuje Srednju školu Marka Marulića 2011. godine, smjer opća gimnazija koji završava 2015. godine kada upisuje Fakultet Elektrotehnike, Računarstva i Informatičkih tehnologija u Osijeku. Upisan preddiplomski studij računarstva završava 2018. godine te upisuje diplomski studij, smjer Programsko inženjerstvo.

---

## **PRILOZI (NA CD-U)**

Prilog 1. Diplomski rad u .docx formatu

Prilog 2. Diplomski rad u .pdf formatu

Prilog 3. Poveznica s programskim rješenjem diplomskog rada