Programsko rješenje web sustava na strani korisnika za personalizirano praćenje pacijenata oboljelih od zloćudnih bolesti

Ivezić, Maja

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://urn.nsk.hr/urn:nbn:hr:200:119343

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-05-07**

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 studij

PROGRAMSKO RJEŠENJE WEB SUSTAVA NA STRANI KORISNIKA ZA PERSONALIZIRANO PRAĆENJE PACIJENATA OBOLJELIH OD ZLOĆUDNIH BOLESTI

Diplomski rad

Maja Ivezić

Osijek, 2020.

SADRŽAJ

1. UVOD 1
1.1. Zadatak diplomskog rada 1
2. Pregled stanja u području 2
2.1. Računalne tehnologije i rak pluća 2
2.2. Provedena istraživanja i postojeća programska rješenja 2
2.2.1. Moovcare 2
2.2.2. Lung Cancer Navigator 3
2.2.3. Cancer.Net 3
2.2.4. Lung Cancer Prognosis System 3
2.2.5. Classification of Lung Cancer Based on Deep Learning Strategy 3
3. DIŠNI SUSTAV I BOLESTI DIŠNOG SUSTAVA 4
3.1. Rak pluća 4
3.2. Tipovi raka pluća 5
3.3. Rizični faktori 5
3.4. Simptomi raka pluća 6
3.5. Stadiji raka pluća 7
3.5.1. Sustav za određivanje stadija raka 7
3.5.2. Numeracija stadija 8
3.6. Dijagnosticiranje raka pluća 9
3.6.1. Plućni čvorovi 10
3.7. Tretmani liječenja 10
3.7.1. Kirurško odstranjivanje 11
3.7.2. Radioterapija 11
3.7.3. Kemoterapija 11
3.8. Smrtnost oboljelih od raka pluća 11
4. RAZRADA MODELA APLIKACIJE 13
4.1. Zahtjevi na web aplikaciju 13
4.2. Idejno rješenje web aplikacije 14
4.3. Struktura web aplikacije 15
4.4. Upravljanje podacima i baza podataka 16
5. PROGRAMSKO RJEŠENJE 17
5.1. Opis platformi, alata i tehnologija 17
5.1.1. HTML 17
5.1.2. CSS 17
5.1.3. Razvojni okvir ReactJS 17
5.1.4. Biblioteka Redux 17
5.1.5. Biblioteka Material UI 18

5.1.6. TypeScript 18
5.2. Razvoj i način rada web aplikacije 18
5.2.1. Prijava i registracija korisnika 18
5.2.2. Dodavanje i prikaz pacijenata 25
5.2.3. Lijekovi i liječnički pregledi 30
5.2.4. Prikaz informacija o raku 32
5.2.5. Bilješke pacijenta i liječnika 34
6. KORIŠTENJE I TESTIRANJE 36
6.1. Pokretanje web aplikacije 36
6.2. Izgled web aplikacije 36
6.2.1. Registracija korisnika 37
6.2.2. Prijava korisnika 38
6.2.3. Prikaz i uređivanje pacijenata 38
6.2.4. Zaslon profila u pogledu pacijenta i liječnika 44
6.2.5. Zaslon kalendara liječničkih termina 45
6.2.6. Zaslon bilješki 45
6.2.7. Zaslon edukacijskog sadržaja o raku 46
6.3.Testiranje i analiza rada sustava 47
6.3.1. Prvi testni skup 47
6.3.2. Drugi testni skup 48
6.3.3. Treći testni skup 49
6.3.4.Usporedba rezultata testnih skupova 50
7. ZAKLJUČAK 52
LITERATURA 53
SAŽETAK 56
ABSTRACT 57
ŽIVOTOPIS 58
PRILOZI 59

1. UVOD

Rak je drugi vodeći uzrok smrtnosti u svijetu, a najčešći oblik oboljenja od raka je rak pluća, stoga je vrlo bitna prevencija, poboljšanje oporavka i kvalitete života oboljelih od zloćudnih bolesti, kao i onih u remisiji. Sastavni dio poboljšanja kvalitete života oboljelih je kvalitetno i detaljno praćenje stanja, edukacija i redovno testiranje. Redovita testiranja i rana dijagnoza su ključni parametri u ranom otkrivanju oboljenja što znatno povećava šanse za učinkovitosti liječničkog tretmana. Za optimizaciju liječničkih pregleda, praćenje stanja i prevenciju danas su razvijeni brojni sustavi i alati koji značajno pomažu oboljelima i zdravstvenom osoblju.

Zadatak ovog diplomskog rada je izraditi web sustav na strani korisnika za personalizirano praćenje pacijenata oboljelih od zloćudnih bolesti. U praktičnom dijelu rada ostvareno je programsko rješenje za praćenje stanja oboljelih koje obrađuje podatke unesene od stane liječnika, te olakšava praćenje napretka liječničkih tretmana. Također, postoji edukacijski sadržaj o raku pluća dostupan pacijentima te mogućnost ostavljanja bilješki od strane pacijenata i liječnika.

Drugo poglavlje opisuje pregled stanja u području te postojeća rješenja za personalizirano praćenje oboljelih od zloćudnih bolesti. Treće poglavlje opisuje teorijsku podlogu zloćudnih bolesti i raka pluća. U četvrtom poglavlju razrađen je model, idejno rješenje i struktura web aplikacije. Peto poglavlje opisuje korištene tehnologije, alate i platforme. Šesto poglavlje opisuje testiranje, analizu rada i izgled web aplikacije.

1.1. Zadatak diplomskog rada

U diplomskom radu potrebno je s medicinskog gledišta opisati probleme i izazove s kojima se susreću pacijenti oboljeli od zloćudnih bolesti kao i liječnici u tretiranju istih. Nadalje, treba opisati postojeća web i mobilna rješenja koja omogućuju potporu pacijentu i liječniku, te razraditi model i programsku arhitekturu web sustava na strani korisnika i opisati potrebne programske metodologije, tehnologije, okoline i jezike. Za programsko rješenje na strani korisnika potrebno je predložiti dizajn korisničkog sučelja koji omogućuje zadovoljavajuće korisničko iskustvo pacijenta i liječnika, dohvaćanje podataka iz baze podataka, te prilagodljivi prikaz podataka uključujući generiranje upozorenja i preporuka za liječenje i potporu koristeći načela obradbe i generiranja prirodnih jezika. Za određenu skupinu zloćudnih bolesti, u praktičnom dijelu rada na strani korisnika potrebno je programski ostvariti navedeno korisničko sučelje koristeći razvojni okvir React, te dohvat i prikaz podataka s generiranjem upozorenja i preporuka. Za programsko rješenje treba provesti nužno testiranje i ispitati ga prikladnom skupu ulaznih podataka.

2. PREGLED STANJA U PODRUČJU

Rak je drugi vodeći uzrok smrtnosti u svijetu, a rak pluća najčešći je oblik raka te najčešći uzrok smrtnosti od raka. Smrtnost od raka može biti znatno smanjena u slučaju ranog otkrivanja i početka odgovarajućeg tretmana. Za rano otkrivanje raka i prevenciju značajne su dvije komponente, rana dijagnoza i testiranje. Ukoliko je rak dijagnosticiran u ranom stadiju, šanse za učinkovitim tretmanom se znatno povećavaju. Za ranu dijagnozu ključni su povećanje svijesti i pristupa kliničkoj skrbi i tretmanu te odgovarajuća dijagnoza i tretman. Redovno provođenje testova znatno poboljšava ranu dijagnozu i otkrivanje abnormalnosti kod pacijenata koji ne pokazuju klasične simptome vezane uz oboljenje od raka. S ciljem poboljšanja stope preživljavanja oboljelih od raka i poboljšanja kvalitete života, razvijaju se brojna ICT rješenja koja pridonose olakšanoj organizaciji i praćenju tretmana oboljelih [19].

2.1. Računalne tehnologije i rak pluća

Potražnja i potreba za programskim rješenjima i alatima u personaliziranoj medicini je u naglom porastu. Cilj alata i sustava personalizirane medicine je pomoći oboljelima pomoću praćenja simptoma, razvitka ili poboljšanja zdravstvenog stanja, kao i pomoć stručnom zdravstvenom osoblju kako bi imali bolji uvid u zdravlje i stanje oboljelih. Istraživanje provedeno od lipnja 2014. godine do studenog 2016. godine temeljeno je na usporedbi praćenja simptoma i stope preživljavanja 121 sudionika oboljelog od raka pluća tijekom kemoterapije [20]. Cilj istraživanja bio je usporediti rezultate sudionika čiji su simptomi i napredak praćeni pomoću web programskog rješenje i sudionika čiji su simptomi praćeni pomoću rutinskih pregleda. Rezultat istraživanja pokazao je poboljšanje u stopi preživljavanja kod sudionika koji su koristili web programsko rješenje za praćenje simptoma. Također, korištenje web praćenja simptoma omogućilo je sudionicima aktivno sudjelovanje u vlastitom zdravstvenom stanju i liječenju, a doktorima optimizirane liječničke preglede te detaljnije informacije koje su dovele do superiornog zdravstvenog tretmana.

2.2. Provedena istraživanja i postojeća programska rješenja

Postoje brojna istraživanja, alati i sustavi na području personalizirane medicine za potporu oboljelima od raka pluća.

2.2.1. Moovcare

Moovcare je web i mobilna aplikacija za rutinsko praćenje bolesnika oboljelih od raka pluća. Kliničke rutinske procjene pacijenata oboljelih od raka pluća u remisiji odvijaju se svakih 3 – 6 mjeseci. Aplikacija je razvijena s ciljem poboljšanja rutinskih pregleda i praćenjem pacijenta

između termina zakazanih pregleda. Moovcare pomaže onkolozima primijetiti nepravilnosti i prve znakove ponovne pojave raka kod oboljelih. Temelji se na tjednim anketama koje oboljeli ispunjavaju, zatim algoritam procesira simptome te ukoliko se pojave nepravilnosti doktor je pravovremeno obaviješten. Rezultati godišnje analize pokazuju poboljšanje stope preživljavanja za 78%, te smanjenja nepotrebnog provođenja pretraga za 49% [21].

2.2.2. Lung Cancer Navigator

Lung Cancer Navigation je mobilna aplikacija za potporu i praćenje bolesti i oporavka oboljelih od raka pluća. Mogu ju koristiti oboljeli, članovi obitelji ili prijatelji oboljelih i doktori. Glavne značajke Lung Cancer Navigator aplikacije su obavijesti za nadolazeće preglede, podsjetnici za uzimanje prepisanih lijekova, edukacijski sadržaj te bilješke koje je moguće podijeliti sa doktorom [22].

2.2.3. Cancer.Net

Cancer.Net je mobilna aplikacija razvijena od strane Američkog saveza za kliničku onkologiju. Aplikacija sadrži raznolik broj alata za poboljšanje i praćenje stanja oboljelih od raka, kao i onkološki odobren informativni sadržaj. Neki od tih alata su praćenje simptoma i nuspojava, praćenje i postavljanje podsjetnika za lijekove, postavljanje podsjetnika za termine liječničkih pregleda te dodavanje bilješki i pitanja za doktora [23].

2.2.4. Lung Cancer Prognosis System

Lung Cancer Prognosis System je web aplikacija sa SQL bazom podataka i korisničkim sučeljem napravljenim pomoću HTML-a i CSS-a. Razvijen je pristupačan, dosljedan i točan algoritam za postavljanje zdravstvene dijagnoze pomoću umjetne inteligencije i rudarenja podataka. Naglasak je na korištenju velikih postojećih baza pacijenata oboljelih od raka pluća ili u remisiji, kako bi se izradio što točniji model prema kojem se razvija i unaprjeđuje algoritam za dijagnosticiranje [24].

2.2.5. Classification of Lung Cancer Based on Deep Learning Strategy

Classification of Lung Cancer Based on Deep Learning Strategy je web sustav za klasifikaciju CT snimki limfnih čvorova. Korisničko sučelje je jednostavno, podijeljeno u 3 sekcije, a to su uvod, učitavanje slika te posljednja sekcija, prikaz rezultata. Na poslužiteljskoj strani koristi se *Keras* biblioteka za strojno učenje s naprednim neuronskim mrežama [28].

3. DIŠNI SUSTAV I BOLESTI DIŠNOG SUSTAVA

Dišni sustav je biološki sustav organa i struktura koji se koriste za izmjenu plinova. Dišni sustav dijelimo na gornji respiracijski trakt i donji respiracijski trakt, a sastoji se od dišnih puteva, pluća i dišnih mišića. Osnovna funkcija pluća je prijenos kisika, koji udišemo iz atmosfere, u krvotok, te odvajanje ugljikovog dioksida, kojeg izdišemo, iz krvotoka. Kada udahnemo, zrak putuje kroz nos ili usta u ždrijelo, zatim grkljan te u dušnik, koji se dijeli na lijevu i desnu dušnicu, a zatim u bronhiole i alveole kroz koje se vrši izmjena plinova [11].

3.1. Rak pluća

Tumori mogu biti benigni i maligni, a kada govorimo o raku, govorimo u malignim tumorima koji agresivno rastu, napadaju okolna tkiva i organe i ulaze u krvotok ili limfni sustav. Rak je naziv za skupinu bolesti kojima je svojstvena nekontrolirana dioba stanica [7]. Svaki tip raka nazvan je prema stanicama u kojima se razvio. Skladno tome, rak pluća nastaje iz stanica pluća čiji je rast abnormalan i koje su se razmnožile te stvorile tumor. Stanice raka se razlikuju od normalnih stanica u tri ključne stvari [1]:

1. Pretjerana proizvodnja stanica
   Stanice raka proizvode nove stanice koje nisu potrebne te koje ne propadaju dovoljno brzo kada su stare ili oštećene. Vremenom, stanice raka stvore nakupinu koju nazivamo tumorom.
2. Invazija
   Stanice raka mogu migrirati u okolno tkivo i strukture i izazvati probleme pri disanju.
3. Metastaza
   Za razliku od normalnih stanica, stanice raka mogu metastazirati izvan pluća. Odvajaju se od tumora i spajaju se sa krvi ili limfom te se na taj način prenose kroz tijelo krvnim žilama. Metastaza predstavlja problem jer prijenosom stanica raka može doći do razvoja sekundarnog tumora.

Najveći broj slučajeva raka pluća su karcinomi. Karcinom je naziv za zloćudnu novotvorinu čije su tumorske stanice podrijetla epitelnih stanica tj. stanica sluznice dišnih puteva. Rak pluća je jedan od najteže lječivih oblika raka obzirom da često metastazira u kratkom periodu nakon formiranja. Najčešća područja metastaze su mozak, nadbubrežna žlijezda, jetra i kosti. Također, pluća su često mjesto metastaziranja drugih malignih tumora [3].

3.2. Tipovi raka pluća

Razlikujemo dva osnovna tipa primanog oblika raka pluća koji se temelje na mikroskopskom izgledu tumorskih stanica, a to su rak pluća malih stanica i rak pluća nemalih stanica [1]. Rak pluća malih stanica je najagresivnijih i najbrže raste. Usko je povezan sa pušenjem duhanskih proizvoda te brzo metastazira na okolne organe i tkiva, a u većini slučajeva se otkrije nakon što se znatno proširio. Rak pluća nemalih stanica obuhvaća 80-90% slučajeva raka pluća. Tri su osnovna histološka podtipa raka pluća nemalih stanica [6]:

1. Adenokarcinom
   Ishodište se nalazi u stanicama koje proizvode sluz, a kojima je obložen dišni put
2. Karcinom pločastih stanica
   Najčešće se razvijaju u bronhima
3. Karcinom velikih stanica
   Najrjeđi tim raka pluća nemalih stanica

3.3. Rizični faktori

Rizične faktori su bilo koja pojava ili aktivnost koja znatno povećava šansu za razvojem raka pluća. Bilo koji od rizičnih faktora ne potvrđuje prisutnost raka pluća. Također, često oboljele osobe nisu imale nijedan rizični faktor [2].

- Duhanski dim
  Konzumacija duhanski proizvoda je najveći rizični faktor za oboljenje od raka pluća. Duhan sadrži više od 50 spojeva za koje je dokazano da uzrokuju rak. Također, dokazano je da su pasivni pušači skloniji oboljenju od raka pluća.
- Starija dob
- Povijest oboljenja od raka
  Rizik oboljenja od raka pluća je povećan ukoliko je osoba prethodno imala određeni tip raka. To uključuje limfomu, rak bubrega, mokraćnog mjehura, gušterače, grlića maternice i leukemiju.
- Genetska predispozicija za oboljenje od raka
  Rizik za oboljenje od raka pluća je veći ukoliko postoji obiteljska povijest oboljenja od raka, jedne ili više osoba, pogotovo u ranijoj životnoj dobi.
- Uzročnici raka
  Postoji 11 poznatih uzročnika raka pluća koji se razvija uslijed dugoročnog kontakta. Također rizik od kontakta je veći za pušače. Najpoznatiji uzročnici su:

- Azbest

  Azbest je prirodno mineral iz skupine silikata te ima mogućnost razdvajanja na čestice koje mogu biti udahnute ili progutane.
- Uran i Radon

  Uran je supstanca koju pronalazimo u stijenama i tlu, koji se pri raspadu pretvara u radon, plin koji pronalazimo u zraku i vodi.

- Plućne bolesti

  Dvije plućne bolesti su usko povezane sa rakom pluća. Kronična opstruktivna pluća bolest i plućna fibroza.
- Hormonska nadomjesna terapija

  u nekim slučajevima hormonska nadomjesna terapija kojom se ublažavaju simptomi menopauze može uzrokovati rak.

3.4. Simptomi raka pluća

Simptomi koje vežemo uz rak pluća su različiti i specifični tome gdje se rak nalazi ili odakle se proširio. Najčešće nisu izraženi ili specifični za rano otkrivanje oboljenja te se zbog toga rak pluća najčešće dijagnosticira kada bolest uznapreduje. Niti jedan simptom pojedinačno ne ukazuje nepogrešivo na rak pluća [12].

Najčešći simptomi raka pluća su:

- Česta upala pluća
- Uporan kašalj

  Uporan kašalj može biti vezan uz prehladu ili gripu, ali na rak pluća može ukazivati ako je vremenski konzistentan, suh ili uz iskašljavanje krvi ili sluzi. Također ako je prisutno hroptanje, promuklost ili produbljivanje glasa. Kod pušača koji već imaju prisutan kašalj je bitno pripaziti pogoršava li se u odnosu na uobičajeno stanje.
- Poteškoće u disanju

  Kratkoća daha i zadihanost mogu ukazivati na suženje dišnog sustava ili nakupljanje tekućine u prsima uzrokovano rakom.
- Neprestana bol u predjelu prsa, ramena i leđa

  Bol u gornjem dijelu tijela može biti posljedica povećanih limfnih čvorova ili metastaza na stijenki prsa ili sluznice prsa.
- Bol u kostima

  Mjesta zahvaćena tumorom nakon metastaze mogu biti ishodište jake boli.

Nespecifični simptomi su:

- Gubitak apetita
  Gubitak apetita i neželjen gubitak tjelesne mase može biti uzrokovan promjenama u načinu na koji stanice troše energiju uzrokovano pojavom raka.
- Povišena temperatura i glavobolje
- Konstantan umor

Također, 25% slučajeva oboljenja od raka pluća nema izraženih simptoma i znakova bolesti. Rak se otkriva slučajno na rutinskim pregledima [7].

3.5. Stadiji raka pluća

Stadij raka je informacija koja prvenstveno predstavlja opseg raka pluća, ali također služi za određivanje dijagnoze, tretmana liječenja i nadopunu daljnjih istraživanja. U većini slučajeva određivanje stadija se odvija u dva koraka [1]:

1. Klinički stadij
   Procjena stadija koja se odvija prije testova.
2. Patološki stadij
   Određivanje stadija na temelju biopsije tkiva.

3.5.1. Sustav za određivanje stadija raka

Najčešće korišten sustav za određivanje stadija raka naziva se TNM klasifikacijski sustav. Princip sustava se bazira na dodjeljivanju slova i brojeva kako bi opisao tumor (T), limfne čvorove (N) i napredak metastaze raka (M) [2, 7].

- T vrijednost
  Opisuje rast primarnog tumora. Razlikujemo sedam vrijednosti:
  1. TX
     Označava da se tumor ne može mjeriti.
  2. T0
     Označava da tumor ne može biti pronađen.
  3. To
     Označava da tumor raste samo u najvišem površinskom sloju tkiva.
  4. T1-T4
     Označava veličinu tumora i opseg metastaze.
- N vrijednost

Opisuje koliko daleko se rak proširio na limfne čvorove. Razlikujemo pet vrijednosti:

1. NX

   Označava da se bližnji limfni čvorovi ne mogu procijeniti.

2. N0

   Označava da bližnji limfni čvor nema stanice raka.

3. N1-N3

   Označavaju veličinu, položaj i broj bližnjih limfnih čvorova koje imaju stanice raka.

- M vrijednost

  Označava stadij metastaze raka. Razlikujemo tri vrijednosti:

  1. MX

     Označava da se metastaza ne može mjeriti.

  2. M0

     Označava odsutnost metastaze.

  3. M1

     Označava prisutnost metastaze u udaljenim organima i tkivima.

3.5.2. Numeracija stadija

TNM klasifikacijski sustav kombinacijom parametara određuje numerički stadij raka. Razlikujemo pet stadija raka uključujući i okultni karcinom [2, 7].

- Okultni karcinom

  Predstavlja odsutnog primarnog raka, ali prisutnost stanica raka u plućnoj tekućini ili sluzi. Također predstavlja odsutnost raka u ostalim područjima tijela.

- Stadij 0

  Nulti stadij predstavlja prisutnost stanica raka koje se još nisu razvile u tumor ili proširile izvan pluća.

- Stadij 1 – stadij 3

  Stadiji 1, 2 i 3 predstavljaju prisutnost tumora koji zahvaća obližnje limfne čvorove, ali nije metastazirao u druga područja.

- Stadij 4

  Četvrti stadij predstavlja rak koji je metastazirao daleko od primarnog tumora.

3.6. Dijagnosticiranje raka pluća

Rak pluća može se dijagnosticirati pomoću širokog spektra dijagnostičkih postupaka i testova. To uključuje [5, 13]:

1. Fizikalni pregled i anamneza

   Tijekom pregleda i ispitivanja anamneze mogu se otkriti rizični čimbenici koje povezujemo s razvojem raka pluća, kao i simptomi koji potencijalno ukazuju na razvoj bolesti.
2. Rendgenska snimka pluća

   Najčešće prvi korak nakon otkrivanja prisutnosti simptoma raka pluća. Izrađuju se dvije snimke: sprijeda-straga i snimka iz profila. Snimka može ukazati na potencijalno sumnjiva područja u plućima, ali ne može ukazati jesu li ta područja maligna .
3. Kompjutorizirana tomografija

   Postoji mogućnost skeniranja mozga, abdomena i prsnog koša. Skeniranjem dobivamo informaciju o opsegu i lokalizaciji tumora. Kompjuterizirana tomografija je osjetljivija od rendgenskih snimki pluća. Skeniranje abdomena može ukazati na metastazu raka u jetri i nadbubrežnim žlijezdama, a skeniranje glave na prisutnost metastaze u mozgu.
4. Višeslojna kompjutorizirana tomografija

   Prednost, a ujedno i nedostatak ove metode je izražena osjetljivost pri otkrivanju čvorića koji u većini slučajeva nisu kancerogeni. Unatoč tome, ova metoda se pokazala kao pouzdan alat za otkrivanje malih karcinoma pluća kod pušača [9].
5. Magnetska rezonancija

   Ova metoda se primjenjuje kada želimo precizno lokalizirati tumor. Nedostatak ove metode je nemogućnost korištenja kod osoba s kirurški ugrađenim metalnim strukturama s obzirom da ova tehnika koristi magnetizam.
6. Pozitronska emisijska tomografija

   Za razliku od kompjutorizirane tomografije i magnetske rezonancije koje daju informaciju o anatomiji struktura i tkiva, ova metoda skenira metaboličku aktivnost i funkciju tkiva.
7. Scintigrafija kostiju

   Ova metoda se koristi za otkrivanje je li rak metastazirao u kosti.
8. Citologija iskašljaja

   Dijagnoza raka pluća uvijek zahtjeva potvrdu patologa da postoje maligne stanice. Jedno od rješenja je ispitivanje iskašljaja koje može omogućiti vizualizaciju

tumorskih stanica. Nedostatak je što unatoč tome što je rak prisutan, u iskašljaju ne moraju postojati tumorske stanice.

9. Bronhoskopija

   Pomoću ove metode mogu se otkriti područja tumora vizualizacijom dišnih putova kroz optičku sondu, te se može uzeti uzorak za patološku potvrdu postojanja tumorskih stanica.

10. Iglena biopsija

    Ova metoda je iznimno korisna kada je tumor u plućima smješten periferno te je zbog toga onemogućena bronhoskopija. Također se uzima uzorak za patološki pregled.

11. Torakocenteza

    Postupak uzimanja tekućine u prostoru između pluća i grudnog koša, u slučaju kada rak zahvati pluća i tanku ovojnicu.

12. Kirurški zahvati

    Kirurškim zahvatom se vrši biopsija ako nijedna od gore navedenih metoda nije moguća ili uspješna. Ova metoda ima znatne rizike i moguće komplikacije poput krvarenja i infekcija.

13. Krvni testovi

    krvni testovi mogu pomoći u otkrivanju biokemijskim abnormalnosti, ali ne mogu samostalno dijagnosticirati oboljenje od raka.

3.6.1. Plućni čvorovi

Čvorovi uzrokovani rakom najčešće nemaju nakupljen kalcija te su nepravilnog oblika i oštrih rubova. Dvije najznačajnije karakteristike čvorova uzrokovanih rakom su [2]

- Veličina

  Čvorovi uzrokovani rakom su veći i brže rastu od normalnih čvorova.

- Gustoća

  Razlikujemo čvorove niske, visoke i podijeljene gustoće. Najčešće čvorovi visoke i podijeljene gustoće su uzrokovani rakom.

3.7. Tretmani liječenja

Nekoliko faktora se uzima u obzir pri određivanju tretmana liječenja raka pluća, a jedan on njih je broj primarnih tumora. Za tretman liječenja može se odrediti primarni ili sekundarni tretman. Cilj primarnog tretmana je otkloniti rak iz tijela, a kada pričamo o raku pluća to može biti kemoterapija,

radioterapija ili kirurško odstranjivanje. Cilj sekundarnog tretmana je ublažiti posljedice raka, kada ne postoji mogućnost za izlječenjem [10].

3.7.1. Kirurško odstranjivanje

Kirurško odstranjivanje tumora se izvodi u slučajevima prvog i drugog stadija raka nemalih stanica i kada se tumor nije proširio izvan pluća. Rijetko se izvodi ako je dijagnosticiran rak malih stanica s obzirom da se u većini slučajeva on dijagnosticira kada se tumor proširio na okolna tkiva i organe. Kirurško odstranjivanje je veliki zahvat, a kirurška metoda se određuje na temelju veličine i mjesta tumora. Kirurg može izvesti klinastu resekciju pluća, lobektomiju ili pulmektomiju. Bitno je naglasiti da uklanjanje kirurško uklanjanje tumora ne rezultira uvijek potpunim izlječenjem [1].

3.7.2. Radioterapija

Radioterapija se može izvoditi kada je dijagnosticira rak malih stanica ili rak nemalih stanica. Također može biti primijenjena u sklopu primarnog ili sekundarnog tretmana liječenja. Radioterapija se može vršiti izvana pomoću uređaja koji usmjerava zrake na područje zahvaćeno tumorom, ili iznutra pomoću doziranja radioaktivnih tvari. Malih broj slučajeva liječenja radioterapijom dovodi do dugoročne remisije, zbog toga se radioterapija u većini slučajeva koristi za ograničavanje ili usporavanje rasta tumora. Nedostatci ove metode su nuspojave poput podložnosti infekcijama, razvijanje anemije, mučnine, umor, nedostatak energije i probavni problemi [8].

3.7.3. Kemoterapija

Kemoterapija se također kao i radioterapija može koristiti za liječenje raka malih i raka nemalih stanica. Ova metoda se najčešće koristi uz radioterapiju ili kirurško odstranjivanje, a provodi se konzumiranjem lijekova ili intravenoznim doziranjem lijekova koji sprječavaju dijeljenje stanica raka ili ih ubijaju. Nedostatci ove metode su što osim uništavanja stanica raka uništava i obične stanice. Također, javljaju se neugodne nuspojave slične nuspojavama radioterapije, ali postoje dostupni lijekovi na tržištu za ublažavanje nuspojava [8].

3.8. Smrtnost oboljelih od raka pluća

Rak pluća je vodeći uzrok smrti kod muškaraca, a s vremenom sve veći urok smrti kod žena. Prognoza raka pluća je statistički lošija u usporedbi s drugim tipovima raka, stopa preživljavanja je znatno niža, a iznosi 16%. Rak malih stanica raka pluća je najagresivniji oblika raka. Statistički, polovina oboljelih preživi 2-4 mjeseca nakon dijagnoze bez liječenja. Petogodišnja stopa preživljavanja nakon dijagnoze iznosi 5-10%. Rak nemalih stanica ima petogodišnju stopu preživljavanja od 75% ako moguće izvesti kirurško uklanjanje. U ostalim slučajevima, stopa preživljavanja je znatno manja [4].

U sljedećem poglavlju bit će definirano idejno rješenje, programska arhitektura i zahtjevi na web aplikaciju za pomoć pri praćenju stanja oboljelih na strani liječnika i pacijenta.

4. RAZRADA MODELA APLIKACIJE

Ovo poglavlje opisuje idejno rješenje sustava na strani korisnika za personalizirano praćenje pacijenata oboljelih od zloćudnih bolesti koje se temelji na potrebama i funkcionalnostima spomenutim i opisanim u prethodnim poglavljima koja su se pokazala obavezna za izradu kvalitetnog programskog rješenja za praćenje zdravstvenog stanja i poboljšanja života oboljelih od raka pluća. Obuhvaća definirane zahtjeve za web aplikaciju, razradu idejnog rješenja i strukturu aplikacije.

4.1. Zahtjevi na web aplikaciju

Prije izrade web aplikacije potrebno je definirati zahtjeve korisnika te pripadajuće scenarije koje aplikacija mora ispunjavati. Web aplikacija je podijeljena na dva pogleda, pogled pacijenta i pogled liječnika. Definirani su zahtjevi za oba pogleda:

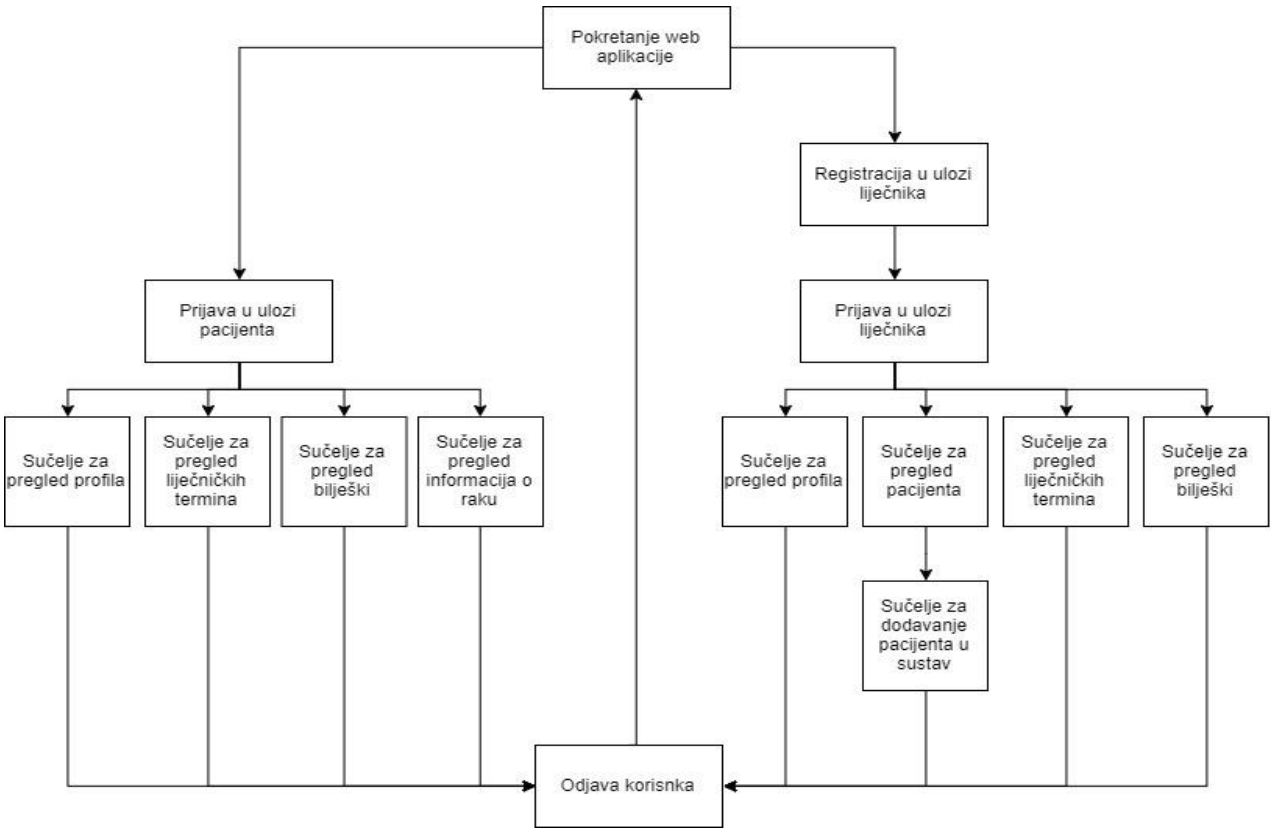
1. Sučelje liječničkog pogleda:
    - Korisnik se može registrirati kao liječnik
    - Korisnik se može prijaviti u aplikaciju u ulozi liječnika
    - Korisnik se može odjaviti iz aplikacije
    - Korisnik u ulozi liječnika ima mogućnost uređivanja vlastitog profila
    - Korisnik u ulozi liječnika ima mogućnost uređivanja podataka o pacijentu
    - Korisnik u ulozi liječnika ima mogućnost dodavanja pacijenta
    - Korisnik u ulozi liječnika ima mogućnost pregleda popisa pacijenta
    - Korisnik u ulozi liječnika ima mogućnost detaljnog pregleda pacijenta
    - Korisnik u ulozi liječnika ima mogućnost dodavanja propisanih zdravstvenih tretmana i lijekova pacijentu
    - Korisnik u ulozi liječnika ima mogućnost pregleda i dodavanja elektroničkih nalaza pacijenta
    - Korisnik u ulozi liječnika ima mogućnost zakazivanja liječničkog pregleda
    - Korisnik u ulozi liječnika ima mogućnost pregleda zakazanih termina liječničkih pregleda
    - Korisnik u ulozi liječnika ima mogućnost dodavanja bilješki pacijentu i pregleda pacijentovih bilješki

2. Sučelje pogleda pacijenta:

- Korisnik se može prijaviti u aplikaciju u ulozi pacijenta
- Korisnik se može odjaviti iz aplikacije
- Korisnik u ulozi pacijenta ima mogućnost pregleda vlastitih propisanih zdravstvenih tretmana i lijekova
- Korisnik u ulozi pacijenta ima mogućnost pregleda vlastitih elektroničkih nalaza
- Korisnik u ulozi pacijenta ima mogućnost pregleda vlastitih zakazanih termina liječničkih pregleda
- Korisnik u ulozi pacijenta ima mogućnost pregleda edukacijskog sadržaja o raku i raku pluća
- Korisnik u ulozi pacijenta ima mogućnost dodavanja bilješki i pregleda liječnikovih bilješki

4.2. Idejno rješenje web aplikacije

Cilj ovog rada bila je izrada web aplikacije za personalizirano praćenje, edukaciju i podršku pacijentima oboljelim od raka pluća te samim time i potporu prevenciji oboljenja od raka i ranog dijagnosticiranja izlaska oboljelog iz remisije. Model web aplikacije prikazan je na slici 4.1.



Sl. 4.1. *Model web aplikacije*

4.3. Struktura web aplikacije

Za programsko rješenje bilo je potrebno razviti klijentski dio web aplikacije. Struktura klijentskog dijela aplikacije sastoji se od izgleda web aplikacije, prikaza podataka na korisničkom sučelju te komunikacije s poslužiteljem. Struktura klijentskog dijela je podijeljena na:

- assets – direktorij koji sadrži dodatne sadržaje
- state – direktorij koji sadrži postavke za Redux store aplikacije
  - ducks – direktorij koji sadrži zahtjeve na poslužitelja, reduktore i selektore
    - auth
    - patients
  - reducers.ts – glavna reducer datoteka aplikacije
  - store.ts – glavna store datoteka aplikacije
- theme – direktorija koji sadrži definirane prilagođene boje i stilove koje se koriste u web aplikaciji
  - colors.ts
- ui – direktorij gdje se nalaze sve komponente za korisničko sučelje
  - components – direktorij koji sadrži sve prilagođene kopomonente za višekratnu upotrebu
  - app – direktorij koji sadrži sve poglede korisničkog sučelja podijeljene u poddirektorije zbog preglednosti
    - about.cancer
    - notes
    - dashboard
    - navigation
    - patients
    - users
  - constants – direktorij koji sadrži sve datoteke konstanti
  - hooks – direktorij koji sadrži prilagođene React hooks funkcije
  - routes – direktorij koji sadrži logiku preusmjeravanja pogleda
  - index.tsx – glavna ulazna točka web aplikacije
  - Layout.tsx – datoteka koja sadrži glavnu strukturu projekta
  - Routes.tsx – glavna datoteka za logiku preusmjeravanja pogleda

Glavne značajke aplikacije su Redux store i višenamjenske React komponente. Redux arhitekturu čini *actions.ts, reducer.ts, selectros.ts* i *constants.ts* datoteke. Datoteka *actions.ts* sadrži sve API zahtjeve na poslužitelja, svu poslovnu logiku te funkcije za ažuriranje podataka koje spremamo u Redux store. Funkcije za ažuriranje podataka govore reduktorima u *reducer.ts* datoteci koje podatke treba ažurirati. Reduktori upravljaju radnjama i ažuriraju podatke, ali nemaju implementiranu logiku. Datoteka *selecotrs.ts* sadrži selektore za dohvaćanje specifičnih podataka iz Redux stanja koji mogu koristiti logiku poput filtriranja. Posljednja datoteka je *constants.ts* koja sadrži sve konstante za radnje reduktora i API zahtjeve.

Druga glavna značajka aplikacije su višenamjenske React komponente. Korisničko sučelje web aplikacije je podijeljeno na manje komponente, od kojih svaka obavlja određeni posao. Svaka komponenta ima *container* dio gdje je implementirana logika poput pozivanja zahtjeva na poslužitelja ili obrade podataka, te *presenter* dio koji služi za prikaz elementa i podataka obrađenih logikom iz *container* dijela. Korištenjem višenamjenskih komponenti sprječavamo nepotrebno pisanje iste logike na više različitih mjesta.

4.4. Upravljanje podacima i baza podataka

Poslužiteljski dio web aplikacije izrađen je u Python programskom jeziku i Flask razvojnom okviru, a obrađen je u diplomskom radu Programsko rješenje web sustava na strani poslužitelja za personalizirano praćenje pacijenata oboljelih od zloćudnih bolesti zasnovano na funkcijskom programiranju [26], kolege Mate Antunovića. Korištena je PostgreSQL baza koja s poslužiteljem komunicira pomoću TCP/IP protokola, a za komunikaciju poslužitelja i baze koristi se SQLAlchemy.

Arhitektura poslužiteljskog dijela se sastoji od 2 glavne komponente, a to su baza i poslužiteljska aplikacija. Baza služi za pohranu i pretragu podataka, a poslužiteljska aplikacija odgovara na upite korisničkog dijela aplikacije. Za komunikaciju baze i poslužiteljske aplikacije koristi se TCP/IP protokol, a za komunikaciju poslužiteljske aplikacije i korisničke aplikacije koristi se HTTP protokol.

Prilikom obrade podataka na poslužiteljskoj strani, poslužiteljska aplikacija nakon primanja zahtjeva od strane korisničke aplikacije nakon primanja jedinstvenog identifikatora resursa izrađuje HTML dokument te ga šalje korisničkoj aplikaciji. Prednosti obrade podataka na poslužiteljskoj strani su mogućnost raspoređivanja posla na više računala, te više klijenata može udaljeno pristupiti poslužitelju.

5. PROGRAMSKO RJEŠENJE

Prilikom izrade sustava za personalizirano praćenje oboljelih od zloćudnih bolesti, korišten je TypeScript programski jezik i ReactJS razvojni okvir te Redux biblioteka za čuvanje stanja aplikacije.

5.1. Opis platformi, alata i tehnologija

5.1.1. HTML

HTML, kratica za *HyperText Markup Language*, je prezentacijski jezik za označavanje hipertekstualnih dokumenta na kojem se zasniva izrada korisničkog sučelja internetskih stranica. HTML prosljeđuje podatke Internetskom pregledniku o sadržaju i strukturi internetske stranice. HTML dokument sastoji se od HTML elemenata koji se sastoje od oznaka (engl. *tag*). Razlikujemo elemente sa početnim i završnim oznakama, koje obuhvaćaju sadržaj i označavaju početak i kraj elementa, te elemente sa samostalnim oznakama koji ne opisuju sadržaj. Svaki element možemo dodatno opisati dodavanjem atributa (engl. *atribute*) koji su specifični za element kojem se dodaju. Pravilo je da svaki HTML dokument započinje sa <html> oznakom, a završava sa </html> oznakom [14].

5.1.2. CSS

CSS, kratica za *Cascading Style Sheets*, je stilski jezik te se koristi za opisivanje izgleda HTML dokumenta. Razlikuje se od HTML-a jer se koristi isključivo za dizajn i raspored internetske stranice, ne za prikaz sadržaja [15].

5.1.3. Razvojni okvir ReactJS

Razvojni okvir ReactJS je JavaScript biblioteka za izgradnju korisničkog sučelja te alat za izgradnju komponenti korisničkog sučelja jednostraničnih internetskih ili mobilnih aplikacija. ReactJS biblioteka je deklarativna, bazirana na komponentama te se zasniva na višestrukoj upotrebi komponenti. Omogućava izradu jednostavnih pogleda za sva stanja u aplikaciji, te učinkovito ažuriranje i prikazivanje komponenti u slučaju promjene podataka. Komponente upravljaju vlastitim stanjem, te se kombiniraju kako bi se izgradilo kompleksno korisničko sučelje [16].

5.1.4. Biblioteka Redux

Redux je biblioteka namijenjena za čuvanje stanja JavaScript aplikacije. Stanje aplikacije se sastoji od odgovora poslužitelja, pred memoriranih podataka i podataka koji još nisu zadržani na poslužitelju. Koristi se u kombinaciji sa ReactJS bibliotekom za izradu korisničkog sučelja. Redux

je dizajniran kao predvidljivi spremnik za stanje aplikacije s ograničenim aplikacijskim programskim sučeljem [17].

5.1.5. Biblioteka Material UI

Material UI je biblioteka interaktivnih React komponenti koja omogućava bržu i lakšu izgradnju korisničkog sučelja. Osnova biblioteke je tablični prikaz, prilagodljive animacije i prijelazi te razni dubinski efekti poput osvjetljenja i sjene [18].

5.1.6. TypeScript

TypeScript programski jezik zasnovan je na JavaScript programskom jeziku sa svim njegovim značajkama. Ključna razlika je što je JavaScript skriptni jezik na strani klijenta, a TypeScript objektno orijentirani prevedeni jezik koji se temelji na apstrakciji podataka. Glavna prednost TypeScript programskog jezika je podržavanje nasljeđivanja i enkapsulacije jer se temelji na klasama [25].

5.2. Razvoj i način rada web aplikacije

Programsko rješenje sustava za personalizirano praćenje pacijenata oboljelih od zloćudnih bolesti temelji se na web aplikaciji čija realizacija je prikazana ključnim dijelovima sustava u sklopu izdvojenih dijelova programskog koda sa adekvatnim rješenjem. Detaljno su opisane sve funkcionalnosti dostupne korisniku, iz perspektive liječnika i pacijenta, za efikasno korištenje. Zahtjevi na poslužitelja ostvareni su pomoću Axios API kontrolera, temeljenog na HTTP metodama. Za dohvaćanje podataka se koristi GET metoda, a za slanje podataka POST metoda.

5.2.1. Prijava i registracija korisnika

Registracija novih korisnika sastoji se od unosa podataka korisnika koje čine email, lozinka, potvrđena lozinka, ime i prezime, te slanja zahtjeva na poslužitelja. Povezivanje na poslužitelja ostvareno je pomoću Axios API kontrolera. Potrebni podaci za registraciju i njihova pravila za provjeru ispravnosti podataka prikazani su programskim kodom 5.1. Programski kod 5.2. prikazuje formu za registraciju. Prilikom provjere ispravnosti provjerava se jesu li svi podaci uneseni, jesu li email i lozinka u ispravnom formatu, te jesu li lozinka i potvrđena lozinka iste vrijednosti. U slučaju da bilo koja vrijednost ne ispunjava uvjet, korisnik je o tome obaviješten.

```
const initialValues: RegisterFormValues = {
  email: "",
  password: "",
  confirm_password: "",
  first_name: "",
  last_name: "",
};

const schema = Yup.object().shape({
  first_name: Yup.string()
    .required("First Name is required"),
  last_name: Yup.string()
    .required("Last Name is required"),
  email: Yup.string()
    .email("Email is invalid")
    .required("Email is required"),
  password: Yup.string()
    .min(6, "Password must be at least 6 characters")
    .required("Password is required"),
  confirm_password: Yup.string()
    .oneOf([Yup.ref("password"), null], "Passwords must match")
    .required("Confirm Password is required"),
});
```

Programski kod 5.1. *Podaci za registraciju korisnika i pravila za validaciju*

Sljedeći korak je slanje zahtjeva za registraciju prikazanog programskim kodom 5.3. Zahtjev za registraciju korisnika ostvaren je pomoću HTTP POST metode koja šalje korisnikove podatke na */register* API. Nakon uspješnog izvršenja zahtjeva korisnika je preusmjeren na formu za prijavu pomoću *redirectTo* funkcije prikazane programskim kodom 5.3. *RedirectTo* funkcija za parametar prima *url* te postavlja lokaciju korisničkog prozora na danu vrijednost, u našem slučaju */login.*

Na poslužiteljskoj strani, prilikom registracije, kreira se korisnik u ulozi liječnika. Korisnik u ulozi liječnika mora biti registriran i prijavljen u sustav kako bi u sustav mogao dodati korisnika u ulozi pacijenta [26].

```
export const SignUpPresenter: React.FunctionComponent<FormikProps<RegisterFormValue
s>> = ({ errors, touched, ...props }) => {
  const classes = styles();
  return (
    <Grid container className={classes.wrap} >
      <Grid item xs={12} className={classes.titleWrap}>
        <Box mb={5} className={classes.title}> Sign Up </Box>
      </Grid>
      <Grid item xs={12}>
       <Form>
          <Field name="first_name" label="First Name" as={CustomInputComponent}
            {...{ errors, touched }}
          />
          <Field name="last_name" label="Last Name" as={CustomInputComponent}
            {...{ errors, touched }}
          />
          <Field name="email" label="Email" as={CustomInputComponent}
            {...{ errors, touched }}
          />
          <Field name="password" label="Password" type="password"
            as={CustomInputComponent} {...{ errors, touched }}
          />
          <Field name="confirm_password" label="Confirm Password" type="password"
            as={CustomInputComponent} {...{ errors, touched }}
          />
          <Box width={500}>
            <CustomButton label="Sign Up" />
          </Box>
          {loginLinkRenderer()}
        </Form>
      </Grid>
    </Grid>
  ); };
```

Programski kod 5.2. *Forma za registraciju korisnika*

```
export const redirectTo = (url: string): (() => any) => () => {
  window.location.href = url;
  return null;
};

function register(registerBody: RegisterFormValues,) {
  return function(dispatch) {
    axios.post("/register", registerBody)
      .then(() => {
        redirectTo("/login")();
      });
  };
}
```

Programski kod 5.3. *Zahtjev za registraciju korisnika i redirectTo funkcija*

Podaci za prijavu korisnika i pravila za validaciju prikazana su programskim kodom 5.4. Prijava korisnika ostvarena je pomoću API kontrolera, HTTP POST metodom i zahtjevom na */login* API.

Prijava s korisničke strane se odvija na način da korisnik unese podatke potrebne za prijavu koje čine email i lozinka, zatim se vrši provjera ispravnosti podataka, te ukoliko su svi podaci ispravni šalje se zahtjev za prijavu korisnika prikazan programskim kodom 5.5. nakon čijeg uspješnog izvršenja se postavljaju podaci o trenutno prijavljenom korisniku u Redux store aplikacije, čije je stanje prikazano programskim kodom 5.6. Podaci se postavljaju predajom podataka funkciji *dispatch(setUser(data))* prikazanoj u programskom kodu 5.5. Pozivanjem spomenute funkcije, dodajemo *user* objekt u Redux store, te spremamo proslijeđene podatke o prijavljenom korisniku koje čine *id, firstName, lastName, email* i *role*. Role parametar označava je li prijavljeni korisnik liječnik ili pacijent, te je vrlo bitan parametar za daljnje prikazivanje komponenti koje ovise o ulozi prijavljenog korisnika.

Prijava s poslužiteljske strane, u svrhu jednostavnijeg upravljanja sesijama i standardiziranoj funkcionalnosti, koristi *login manager*. *Login manager* upravlja korisničkim funkcijama i objektima što omogućava sigurnost i autentifikaciju tokena pri slanju svakog zahtjeva s korisnika na poslužitelja [26]. Autentifikacijski token kreiran je pomoću email računa, lozinke i identifikacijske vrijednosti. Korištenje tokena i login managera sprječava istovremenu prijavu dva različita korisnika u web aplikaciji.

```
const initialValues: LoginFormValues = {
  email: "",
  password: "",
};

const schema = Yup.object().shape({
  email: Yup.string()
    .email("Email is invalid")
    .required("Email is required"),
  password: Yup.string()
    .required("Password is required"),
});
```

Programski kod 5.4. *Podaci za prijavu korisnika i pravila za validaciju*

Forma za prijavu korisnika prikazana je programskim kodom 5.7., čine ju dva polja, email polje i polje za lozinku, tipka za predaju popunjene forme i link koji preusmjerava korisnika na formu za registraciju ukoliko nema izrađen korisnički račun. Ukoliko uneseni podaci nisu ispravno uneseni, korisnik je obaviješten ispisivanjem pogreške ispod polja.

Forme u web aplikaciji izrađene su pomoću *Formik* biblioteke koja omogućava praćenje stanja forme, pogreške, posjećenosti polja, provjere ispravnosti i rukovanje predavanjem ispunjene i ispravne forme. Svakom polju potrebno je navesti *name* parametar koji će se podudarati sa *initialValues* parametrom pomoću kojeg formi predajemo željene početne vrijednosti polja. Također, pravila validacije *scehma* prikazane programskim kodom 5.4. predajemo formi, na temelju kojih provjerava ispravnost i prikazuje pogreške pojedinih polja.

```
function login(loginBody: LoginFormValues,) {
  return function(dispatch) {
    axios.post("/login", loginBody)
      .then((data) => {
        dispatch(setUser(data));
      });
  };
}
```

Programski kod 5.5. *Zahtjev za prijavu korisnika*

```
export const user = (
  state: User = USER_DEFAULT_STATE,
  action: any,
): User => {
  const data = action.payload?.data;
  switch (action.type) {
    case SET_USER:
      return {
        ...state,
        id: data.user_id,
        firstName: data.first_name,
        lastName: data.last_name,
        email: data.email,
        role: data.role,
      };
    case LOGOUT_USER:
      return { ...USER_DEFAULT_STATE };
    default:
      return state;
  }
};
```

Programski kod 5.6 *Postavljanje podataka o prijavljenom korisniku u Redux*

```
export const LoginPresenter: React.FunctionComponent<FormikProps<LoginFormValues>>
 = ({ ...props }) => {
  const classes = styles();
  const signUpLinkRenderer = () =>
    <Box mb={4} mt={2} width={500} display="flex" justifyContent="center" >
      <Box component="span" mr={0.5} color={Colors.light_gray}>
        {`Don't have an account?`}
      </Box>
      <Box component="span" color={Colors.dark_blue}>
        <Link href={SIGN_UP_URL}>
        Sign up
        </Link>
      </Box>
    </Box>;
  return (
    <Grid container className={classes.wrap} >
      <Grid item xs={12} className={classes.titleWrap}>
        <Box mb={5} className={classes.title}> Log In </Box>
      </Grid>
      <Grid item xs={12}>
        <Form>
          <Field name="email" placeholder="Email" label="Email"
            as={CustomInputComponent} />
          <Field name="password" placeholder="Password" label="Password" type="password"
            as={CustomInputComponent} />
          <Box width={500}>
            <CustomButton label="Log In"/>
          </Box>
          {signUpLinkRenderer()}
        </Form>
      </Grid>
    </Grid>
  ); };
```

Programski kod 5.7. *Forma za prijavu korisnika*

5.2.2. Dodavanje i prikaz pacijenata

Dodavanje novog pacijenta u sustav može izvršiti samo korisnik u ulozi liječnika. Postupak dodavanja novog pacijenta zahtjeva unos podataka prikazanih programskim kodom 5.8. Potrebni podaci su opći podaci koje čine *first_name, last_name, email, gender* i *smoking_status*. Zatim *family_history* i *lung_disease history* koji su jedni od najvažnijih rizičnih čimbenika, opisanih u 3. poglavlju. Te posljednji podaci, *cancer_type*, te *t, n* i *m* vrijednosti na temelju kojih se vrši izračun stadija raka.

```
const initialValues: AddPatientValues = {
  first_name: "",
  last_name: "",
  email: "",
  gender: "female",
  smoking_status: false,
  family_history: "",
  lung_disease_history: false,
  cancer_type: "",
  t_value: "",
  n_value: "",
  m_value: "",
};
```

Programski kod 5.8. *Podaci potrebni za dodavanje novog pacijenta u sustav*

Dodavanje pacijenta omogućeno je slanjem HTTP POST zahtjeva te je prikazano programskim kodom 5.9. Korisnička strana osigurava slanje zahtjeva za dodavanje novog pacijenta omogućavanje vidljivosti komponente samo korisnicima u ulozi liječnika, a poslužiteljska strana izvršava provjeru pomoću prije spomenutog *login managera*. Također, na poslužiteljskoj strani za svakog dodanog pacijenta se generira nasumična lozinka, te je pacijent obaviješten putem maila o podacima za prijavu nakon čeka se može prijaviti u sustav preko zaslona za prijavu korisnika.

Nakon što je zahtjev uspješno izvršen i pacijent dodan u bazu podataka, korisnik u ulozi liječnika može pristupiti listi pacijenta koja se dohvaća zahtjevom prikazanim programskim kodom 5.10. Također, HTPP GET metoda, nakon čijeg se uspješnog izvršenja zahtjeva popis pacijenata također sprema u Redux store te mu možemo pristupiti iz bilo koje komponente u aplikaciji.

```
function addPatient(body: any) {
  return function(dispatch) {
    axios.post(ADD_PATIENT_URL, body);
  };
}
```

Programski kod 5.9. *Zahtjev za dodavanje novog pacijenta u sustav*

Nakon slanja podataka na */add_patient* API pomoću *addPatient* metode, prikazane programskim kodom 5.9. poslužiteljska strana na temelju *t*, *n* i *m* vrijednosti, objašnjenih u poglavlju 3, računa stadij raka pluća oboljele osobe pomoću funkcije *calculate_cancer_stage* [26]. Stadij raka pluća računa se prema pravilima priručnika Američkog odbora za rak koja su prikazana tablicom 5.1. gdje se za izračun stadija raka pluća koriste vrijednosti primarnog tumora (T), regionalnih limfnih čvorova (N) i udaljene metastaze (M). Priručnik s pravilima se redovno ispravlja i nadopunjuje rezultatima novih istraživanja kako bi određivanje stadija raka bio optimiziran, efikasan i točan proces.

Tab. 5.1. *Sustav određivanja stadija raka pluća* [27]

STADIJ	T	N	M
Okultni karcinom	T1X	N0	M0
0	Tis	N0	M0
IA	T1a	N0	M0
IA	T1b	N0	M0
IB	T2a	N0	M0
IIA	T2b	N0	M0
IIA	T1a	N1	M0
IIA	T1b	N1	M0
IIA	T2a	N1	M0
IIIB	T2b	N1	M0
IIIB	T3	N0	M0

IIIA	T1a	N2	M0
IIIA	T1b	N2	M0
IIIA	T2a	N2	M0
IIIA	T2b	N2	M0
IIIA	T3	N1	M0
IIIA	T3	N2	M0
IIIA	T4	N0	M0
IIIA	T4	N1	M0
IIIB	T1a	N3	M0
IIIB	T1b	N3	M0
IIIB	T2a	N3	M0
IIIB	T2b	N3	M0
IIIB	T3	N3	M0
IIIB	T4	N2	M0
IIIB	T4	N3	M0
IV	Bilo koji T	Bilo koji N	M1a
IV	Bilo koji T	Bilo koji N	M1b

```
function getPatients() {
  return function(dispatch) {
    axios.get(GET_PATIENTS_URL)
      .then((data) => {
        dispatch(setPatients(data));
      });
  };
}
```

Programski kod 5.10. *Zahtjev za dohvaćanje popisa pacijenta*

Pacijenti su prikazani pomoću tablice, a za svakog pacijenta u tablici je prikazano ime, prezime, email račun, datum zadnjeg i slijedećeg liječničkog termina. Tablični prikaz pacijenata prikazan

je programskim kodom 5.11. Ostvaren je pomoću Grid komponente iz *Material UI* biblioteke, koja omogućuje responzivan tablični prikaz, a temelji se na 12 stupaca tablice. Podaci o pacijentima dohvaćeni iz baze su u obliku niza objekata, gdje svaki objekt predstavlja jednog pacijenta. Kako bi prikazali pojedinog pacijenta, koristimo *map* funkciju na nizu objekata, te za svaki objekti izdvajamo vrijednosti *gender* za prikaz profilne slike, *first_name, last_name, last_appointment* i *next appointment* koji su potrebni za tablični prikaz podatak o pacijentima. Forma za dodavanje pacijenata prikazana je programskim kodom 5.12

```
{patients.map((patient) =><Grid key={patient.id} container  style={{ paddingTop: "40px", c
ursor: "pointer" }} onClick={() => history.push(generatePath(PATIENT_INFO_URL, { id: p
atient.id }))}>
        <Grid item xs={3} style={{ display: "flex", justifyContent: "center" }}>
          {avatarRenderer(patient.gender)}  </Grid>
        <Grid container xs={3} style={{ fontSize: "18px" }}>
          <Grid item xs={12} style={{ color: Colors.dark_blue }}>
            {patient.first_name} {patient.last_name}
          </Grid>
          <Grid item xs={12} style={{ color: Colors.light_gray }}> {patient.email}  </Grid>
        </Grid>
        <Grid item xs={3} style={{ display: "flex", alignSelf: "center",
          color: Colors.dark_blue, fontSize: "18px" }} > {patient.last_appointment} </Grid>
        <Grid item xs={3} style={{ display: "flex", alignSelf: "center",
          color: Colors.dark_blue, fontSize: "18px" }}> {patient.next_appointment} </Grid>
      </Grid>  )}
```

Programski kod 5.11. Tablični prikaz popisa pacijenata

```
const AddPatientPresenter: React.FunctionComponent<Props> = ({ ...props }) => (
  <Form>
    <Field name="first_name" label="First Name" as={CustomInputComponent}{...props}/>
    <Field name="last_name" label="Last Name" as={CustomInputComponent}{...props} />
    <Field name="email" label="Email" as={CustomInputComponent} {...props} />
    <Field name="gender" label="Gender" options={genderOptions}
      as={CustomRadioGroupComponent}{...props} />
    <Field name="smoking_status" label="Smoking Status" options={smokingStatusOptions}
      as={CustomRadioGroupComponent} {...props} />
    <Field
      name="lung_disease_history"
      label="Lung Disease History ( COPD, Pulmonary fibrosis): "
      options={lungDiseaseHistory}
      as={CustomRadioGroupComponent}
      {...props}
    />
    <Field name="family_history" label="Family History" multiline rowsMax={20}
      as={CustomInputComponent} {...props} />
    <Box mb={4} mt={6}>
      <Typography variant="h6" style={{ color: Colors.dark_blue }}>
        TNM staging:
      </Typography>
    </Box>
    <Field name="t_value" label="T - Primary Tumor" options={primaryTumorOptions}
      as={CustomSelectComponent} {...props} />
    <Field name="n_value" label="N - Regional Lymph Nodes"
      options={regionalLymphNodes} as={CustomSelectComponent} {...props}
    />
    <Field name="m_value" label="M - Distant Metastasis " options={distantMetastasis}
      as={CustomSelectComponent} {...props} />
    <Box width={200}> <CustomButton label="Add patient"/> </Box>
  </Form> );
```

Programski kod 5.12. *Forma za dodavanje pacijenta*

5.2.3. Lijekovi i liječnički pregledi

Korisnik u ulozi liječnika ima mogućnost pacijentu dodati propisani lijek, pregledati popis lijekova, te također i ukloniti pojedini lijek. Zahtjevi za dodavanje, brisanje i dohvaćanje propisanih lijekova su prikazani programskim kodom 5.13. Dodavanje lijekova omogućeno je slanjem HTTP POST zahtjeva na */add_medication* API kojem predajemo parametre id pacijenta kojem dodajemo lijekove, te *MedicationValues* koje sadrže parameter *name, dose, frequency* i *route*. *Name* predstavlja ime lijeka, te su opcije postavljene na korisničkoj strani te su iste za svakog liječnika i pacijenta. Zatim, *dose* i *frequency* parametar koji opisuju dozu i frekvenciju uzimanja lijeka, te posljednje *route* parametar koji predstavlja način uzimanja lijeka, te su također opcije postavljenje na korisničkoj strani. Nakon izvršenja zahtjeva za dohvaćanje lijekova, popis lijekova za određenog pacijenta se sprema u Redux store, te tim vrijednostima možemo pristupiti u bilo kojem trenutku iz bilo koje komponente, te tako smanjujemo broj potrebnih zahtjeva na poslužitelja, samim time i optimizirani proces prikazivanja podataka.

```
function addMedication(id: string, body: MedicationValues) {
  return function(dispatch) {
    axios.post(`${ADD_MEDICATION_URL}${id}`, body)
  };
}

function getMedication(id: string) {
  return function(dispatch) {
    axios.get(`${GET_MEDICATION_URL}${id}`)
      .then((data) => {
        dispatch(setMedication(data));
      });
  };
}

function removeMedication(id: string) {
  return function(dispatch) {
    axios.get(`${REMOVE_MEDICATION_URL}${id}`)
  };
}
```

Programski kod 5.13. *Zahtjevi za dodavanje, dohvaćanje i brisanje propisanih lijekova*

Osim dodavanja propisanih lijekova, korisnik u ulozi liječnika ima mogućnost dodavanja liječničkih termina kao i prikaz popisa termina. Korisnik u ulozi pacijenta ima samo mogućnost prikaza popisa liječničkih termina. Postupak dodavanja liječničkog termina sastoji se od unosa potrebnih podataka *appointmentValues* koje čine podaci o pacijentu, početak i kraj termina te opis razloga termina. Zahtjevi za dodavanje i dohvaćanje termina prikazani su programskim kodom 5.14. Oba zahtjeva su ostvarena pomoću HTTP GET metode.

```
function getAppointments() {
  return function(dispatch) {
    axios.get(GET_APPOINTMENTS)
  };
}

function addAppointment(body: appointmentValues) {
  return function(dispatch) {
    axios.get(ADD_APPOINTMENT, body)
  };
}
```

Programski kod 5.14. *Zahtjevi za dodavanje i prikaz popisa liječničkih termina*

5.2.4. Prikaz informacija o raku

Korisniku u ulozi pacijenta omogućen je informativni prikaz korisnih informacija o raku i raku pluća, dio je prikazan programskim kodom 5.15. To uključuje tipove raka pluća, faktore rizika, znakove i simptome. Također, odabirom tipa raka pluća otvara se dijalog sa detaljnijim objašnjenjem, prikazan programskim kodom 5.16. Dijalog je ostvaren pomoću *Material UI* biblioteke.

```
const AboutCancerPresenter: React.FunctionComponent<RouteComponentProps> = ({ matc
h, ...props }) => {
  const nsclcDialog = useDialog();
  const sclcDialog = useDialog();
  const classes = styles();
  return (
    <Grid container>
      <Grid item xs={12} className={classes.centerWrap}>
        <Box className={classes.titleWrap}> Cancer </Box>
      </Grid>
      <Grid item xs={12}>
        <ul className={classes.contentWrap}>…</ul>
      </Grid>
      <Grid item xs={12} className={classes.centerWrap}>
        <Box className={classes.titleWrap}>
          Lung Cancer
        </Box>
      </Grid>
      <Grid item xs={12}> <ul className={classes.contentWrap}>…</ul>
        <Box className={classes.subtitleWrap}>
            Types of Lung Cancer
        </Box> <ol className={classes.contentWrap}>… </ol>
      </Grid>
      <NSCLCDialog dialog={nsclcDialog} />
      <SCLCDialog dialog={sclcDialog} />
    </Grid>
  ); };
```

Programski kod 5.15 *Dio prikaza informacija o raku i raku pluća*

```
const NSCLCDialog: React.FunctionComponent<Props> = ({ dialog, ...props }) => (
  <Dialog open={dialog.isOpen} fullWidth={true} maxWidth="md" >
    <DialogTitle>
      <IconButton onClick={dialog.close} style={{ width: "24px", height: "24px", padding: "1
6px" }}>
        <CloseIcon />
      </IconButton>
      <Box> Non-small cell lung cancer (NSCLC)  </Box>
    </DialogTitle>
    <DialogContent >
      <ul>
        <li> About 80% to 85% of lung cancers are NSCLC.< li>
        <li>The main subtypes of NSCLC are adenocarcinoma, squamous cell carcinoma, and l
arge cell carcinoma </li>
        <li>These subtypes, which start from different types of lung cells are grouped together a
s NSCLC because their treatment and prognoses (outlook) are often similar</li>
        <ol>
          <li><b>Adenocarcinomas</b> start in the cells that would normally secrete substances
 such as mucus. </li>
          <li><b>Squamous cell carcinomas</b> start in squamous cells, which are flat cells that
 line the inside of the airways in the lungs. They are often linked to a history of smoking and t
end to be found in the central part of the lungs, near a main airway (bronchus). </li>
          <li><b>Large cell carcinoma</b> can appear in any part of the lung. It tends to grow a
nd spread quickly, which can make it harder to treat. A subtype of large cell carcinoma, know
n as large cell neuroendocrine carcinoma, is a fast-
growing cancer that is very similar to small cell lung cancer. </li>
        </ol> </ul>
  </DialogContent>
  </Dialog>);
```

Programski kod 5.16. *Dijalog za tip raka pluća*

5.2.5. Bilješke pacijenta i liječnika

Programskim kodom 5.17. prikazana je mogućnost dodavanja bilješki koju imaju korisnici u ulozi pacijenta i korisnici u ulozi liječnika. Prikaz bilješke se dinamički mijenja ovisno o ulozi korisnika. Zahtjev na poslužitelja za dodavanje bilješki prikazan je programskim kodom 5.18.

```
const NotesPresenter: React.FunctionComponent<Props> = ({ notes }) => {
  const classes = styles();
  return (
     <Grid container>
      <Grid item xs={12}>
       {notes.map((note) => {
          const isDoctor = note.sender.role === Role.DOCTOR;
          return (
           <Grid >
            <Box className={classNames( classes.wrap, {
             [classes.doctorWrap]: isDoctor, })}>
             <Grid item xs={12} className={classNames( classes.userInfo, {
              [classes.doctorInfo]: isDoctor, })}>
              {note.sender.firstName} {note.sender.lastName } {" - "}
              { note.sender.email}
             </Grid>
             <Grid item xs={12}></Grid> <Grid item xs={12}> {note.note}</Grid>
            </Box>
           </Grid>); })}
      </Grid>
      <Grid item xs={12} className={classes.notes}>
       <CustomInputComponent name="Note" label="Add Note" fullWidth/>
       <Box width={150} ml={10} display="flex" alignSelf="center">
        <CustomButton label="Add note" style={{ backgroundColor: Colors.light_gray }}/>
       </Box>
      </Grid>
     </Grid>
  ); };
```

Programski kod 5.17. *Prikaz bilješki*

Zahtjev za dodavanje bilješki ostvaren je pomoću HTTP POST metode, a korisnička strana šalje podatke na poslužitelja o sadržaju bilješke, pošiljatelju i primatelju. Zahtjev za prikaz bilješki je ostvaren pomoću HTTP GET metode, gdje poslužitelj šalje niz objekata bilješki izmijenjenih između pacijenta i liječnika. Autentifikacija pacijenta i liječnika odrađuje se na poslužiteljskoj strani.

```
function addNotes(body: NotesValues) {
  return function(dispatch) {
    axios.post("/add_notes", body)
      .then((data) => {
      });
  };
}

function listNotes(body: NotesValues) {
  return function(dispatch) {
    axios.get("/list_notes", NotesValues)
      .then((data) => {
      });
  };
}
```

Programski kod 5.17. *Zahtjevi za slanje i dohvaćanje bilješki*

6. KORIŠTENJE I TESTIRANJE

Web aplikacija za personalizirano praćenje pacijenata oboljelih od zloćudnih bolesti omogućuje korisnicima, koji uključuju liječnike i pacijente, jednostavno i efikasno korištenje uz pregledno i intuitivno korisničko sučelje.

6.1. Pokretanje web aplikacije

Kako bi pokrenuli web aplikaciju, potrebno je pokrenuti programski kod pomoću naredbe *yarn start*. Naredba za pokretanja programskog koda prikazana je na slici 6.1.

Sl. 6.1. *Naredba za pokretanje programskog koda*

Pokretanje programskog koda prikazano je na slici 6.2.

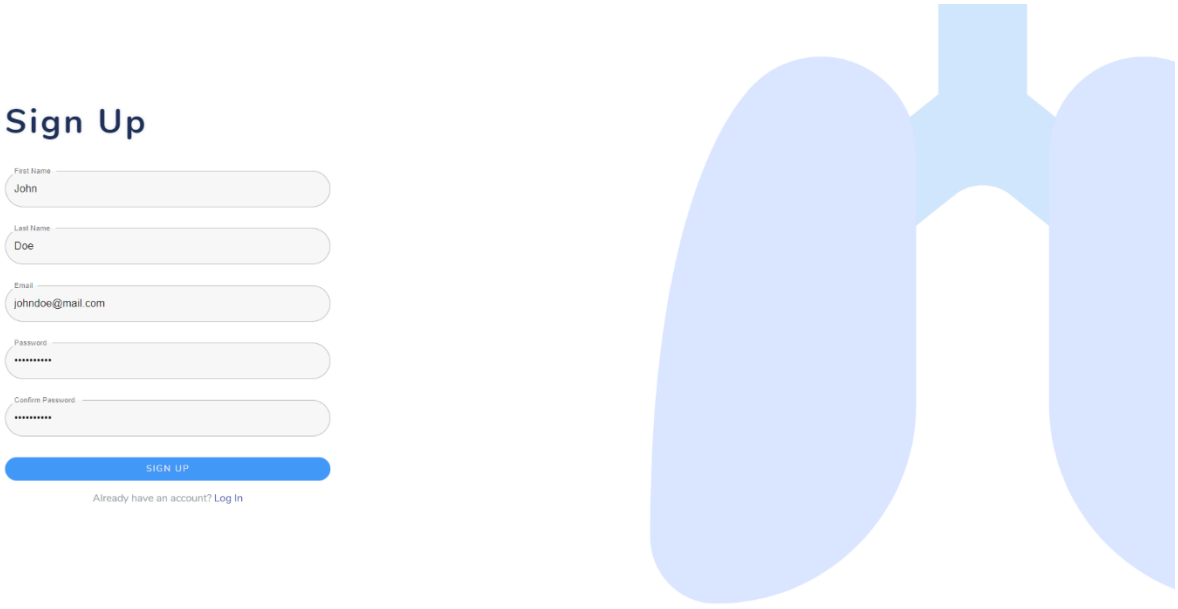
Sl. 6.2. *Pokretanje programskog koda*

6.2. Izgled web aplikacije

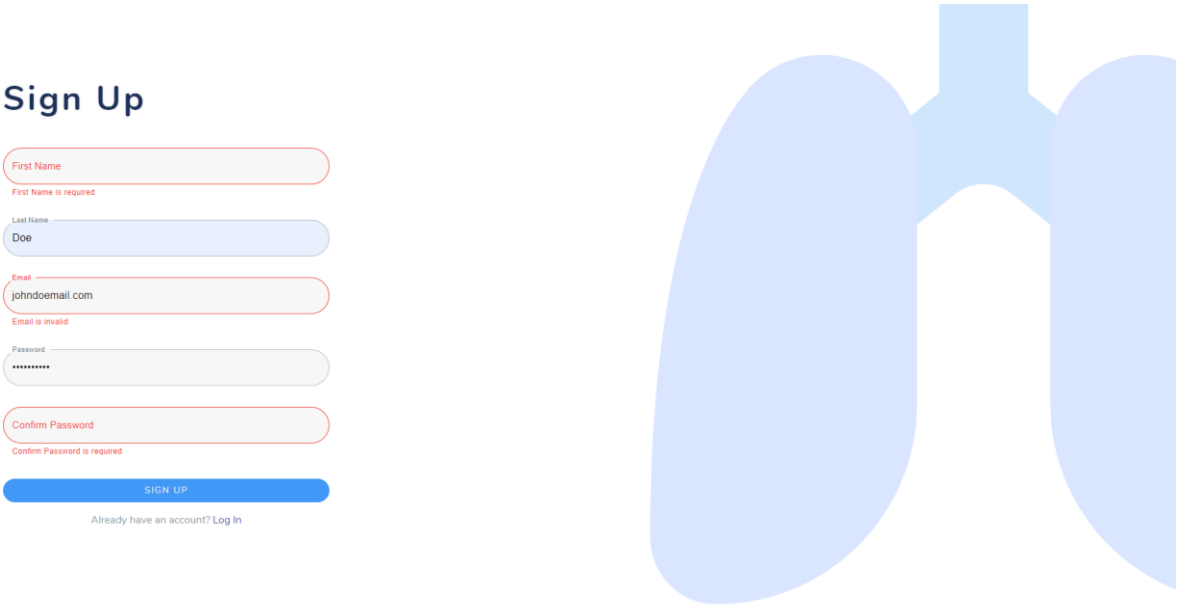
Glavna podjela pogleda aplikacije je pogled liječnika i pogled pacijenta. Uzimajući u obzir oba pogleda, aplikacija se sastoji od zaslona za prijavu, registraciju, pregled liste pacijenta, dodavanje novog pacijenta, pregled zakazanih liječničkih termina, bilješke i edukacijskih zaslon o raku i raku pluća.

6.2.1. Registracija korisnika

Zaslon za registraciju korisnika, prikazan na slici 6.3, namijenjen je liječnicima. Kao što je ranije spomenuto u poglavlju 5.2.1, prilikom registracije na poslužiteljskoj strani se dodaje korisnik u ulozi liječnika. Na sučelju za registraciju korisnik mora unijeti podatke: ime, prezime, email, lozinka i potvrđenu lozinku. Ukoliko su podaci neispravni ili nepotpuni, korisnik je o tome obaviješten, a primjer je prikazan na slici 6.4.



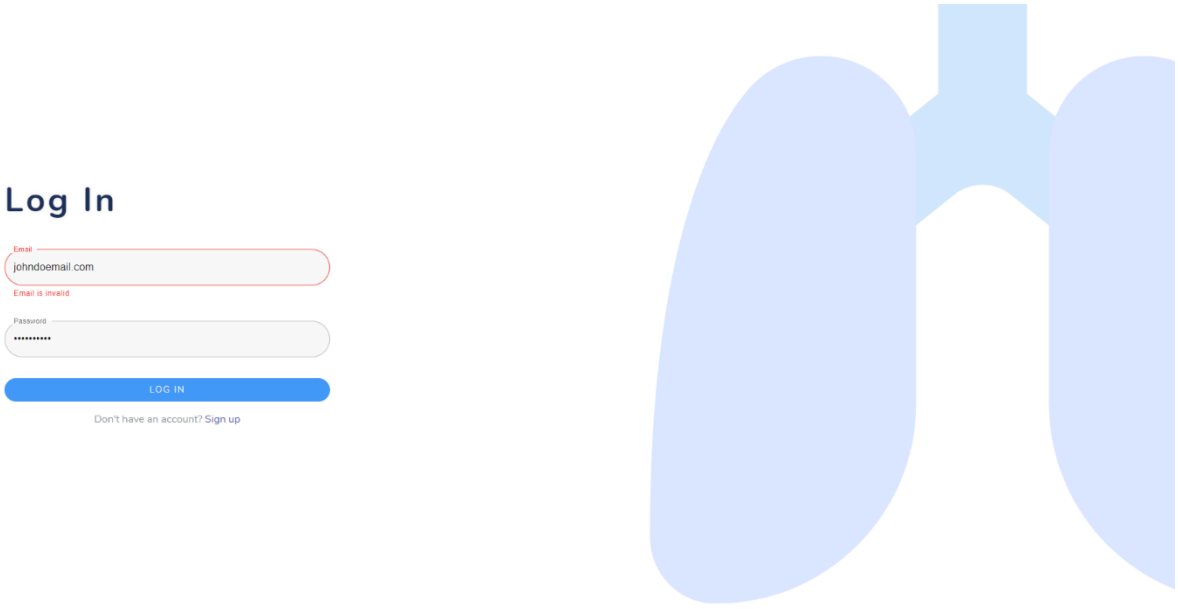
Sl. 6.3. *Zaslon za registraciju korisnika*



Sl. 6.4. *Zaslon za registraciju s obavještenjem o neispravnim podacima*

6.2.2. Prijava korisnika

Nakon uspješne registracije korisnik je preusmjeren na zaslon za prijavu, prikazan na slici 6.5. koji također ima implementirano obavještavanje korisnika o neispravnim ili nepotpunim podacima, na temelju pravila prikazani u programskom kodu 5.4. Prilikom prijave korisnika, potrebno je unijeti email i lozinku, a ako korisnik nije registriran, do zaslona za registraciju može doći pomoću kika na *Sign up* link na dnu forme.

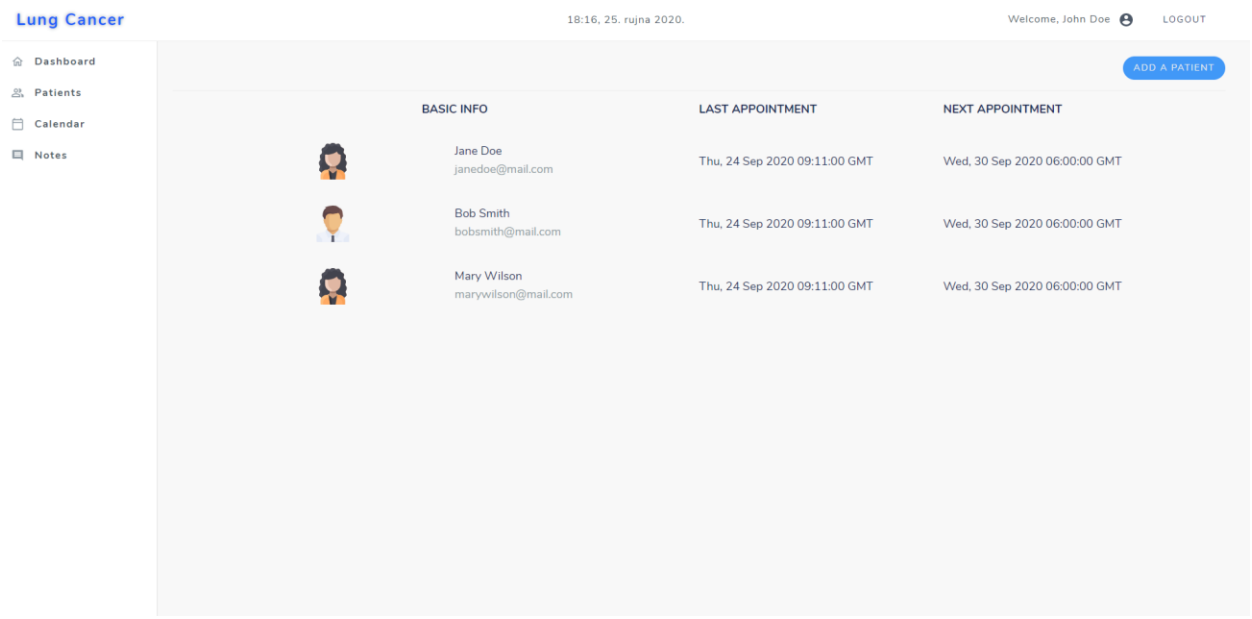


Sl. 6.5. *Zaslon za prijavu korisnika*

6.2.3. Prikaz i uređivanje pacijenata

Zaslon za prikaz popisa pacijenata , prikazan na slici 6.6, dostupan je korisnicima u ulozi liječnika, a provjera uloge odvija se pri spremanju korisnika u Redux store aplikacije nakon uspješne prijave, što je detaljnije objašnjeno u poglavlju 5.2.1 i prikazano programskim kodom 5.6. Do zaslona za pregled popisa i profila pacijenta se dolazi klikom na poveznicu *Patients* u navigaciji aplikacije.

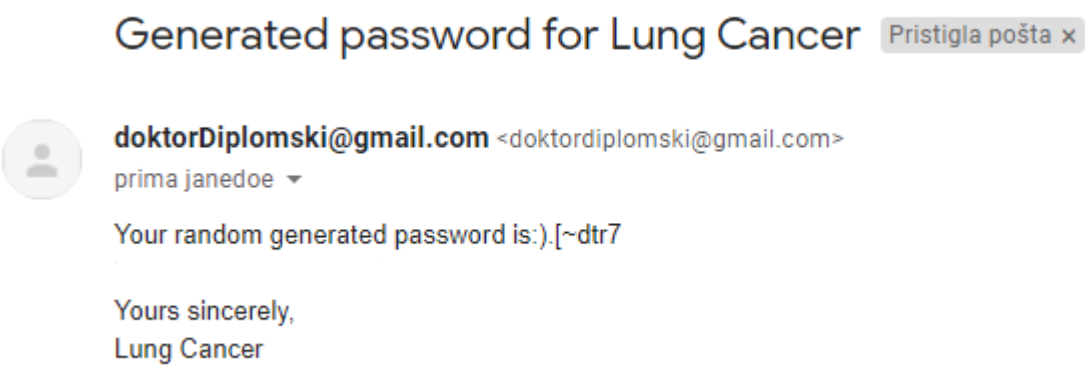
Pacijenti se prikazuju tablično s osnovnim podacima: ime, prezime i email, također uz podatke o posljednjem i idućem liječničkom terminu.



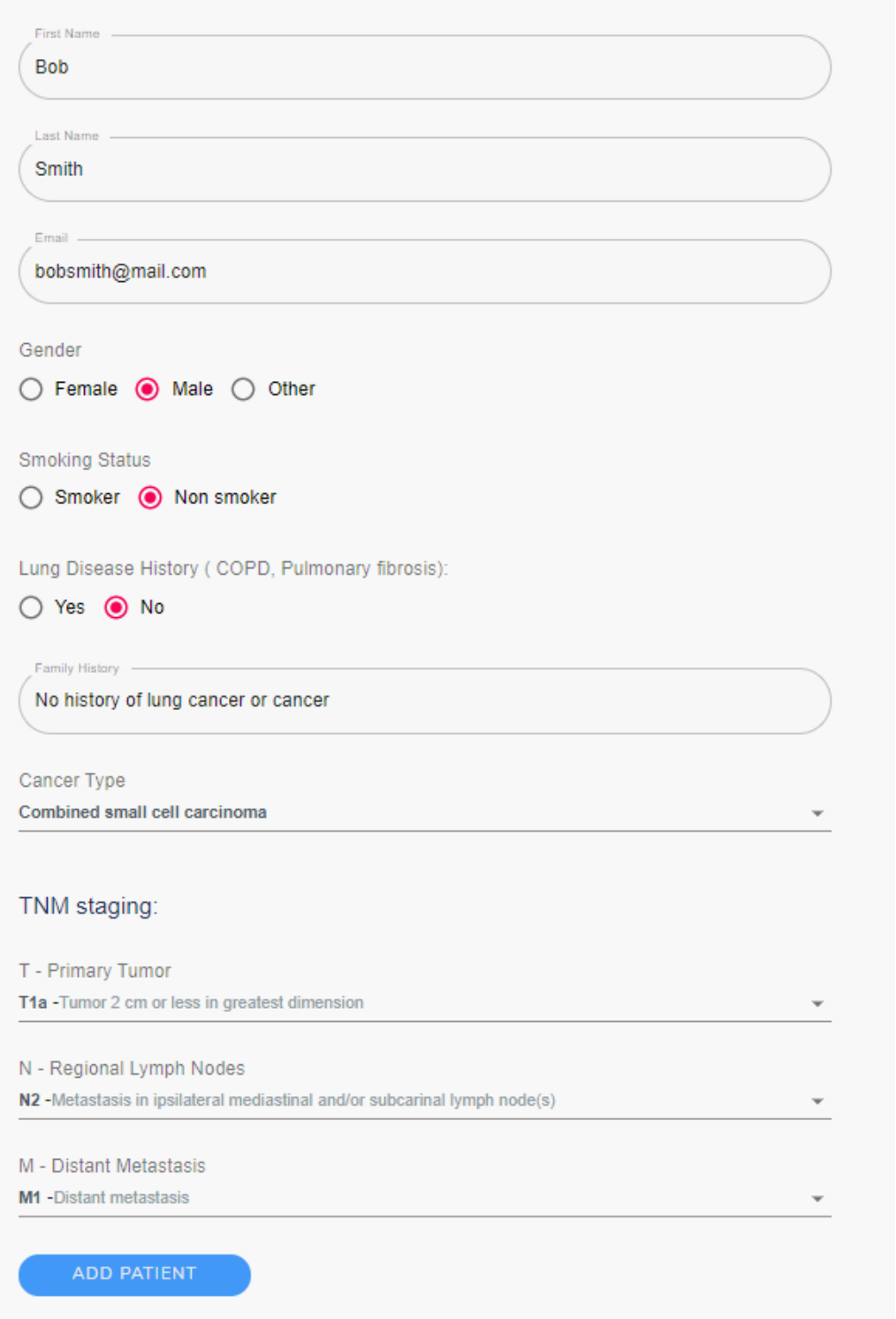
Sl. 6.6. *Zaslon za prikaz popisa pacijenata*

S istog zaslona se otvara zaslon za dodavanje novog pacijenta pritiskom tipke *Add a Patient*. Zaslon za dodavanje novog pacijenta u sustav prikazan je na slici 6.8. te je to jedini način dodavanja pacijenta u sustav, s obzirom da se registrirati mogu samo liječnici.

Nakon uspješnog dodavanja pacijenta, pacijent će primiti mail sa generiranom lozinkom pomoću koje se može prijaviti u sustav, što je već detaljnije objašnjeno u poglavlju 5.2.2. Izgled maila prikazan je na slici 6.7.



Sl. 6.7. *Mail s podacima za prijavu u sustav*

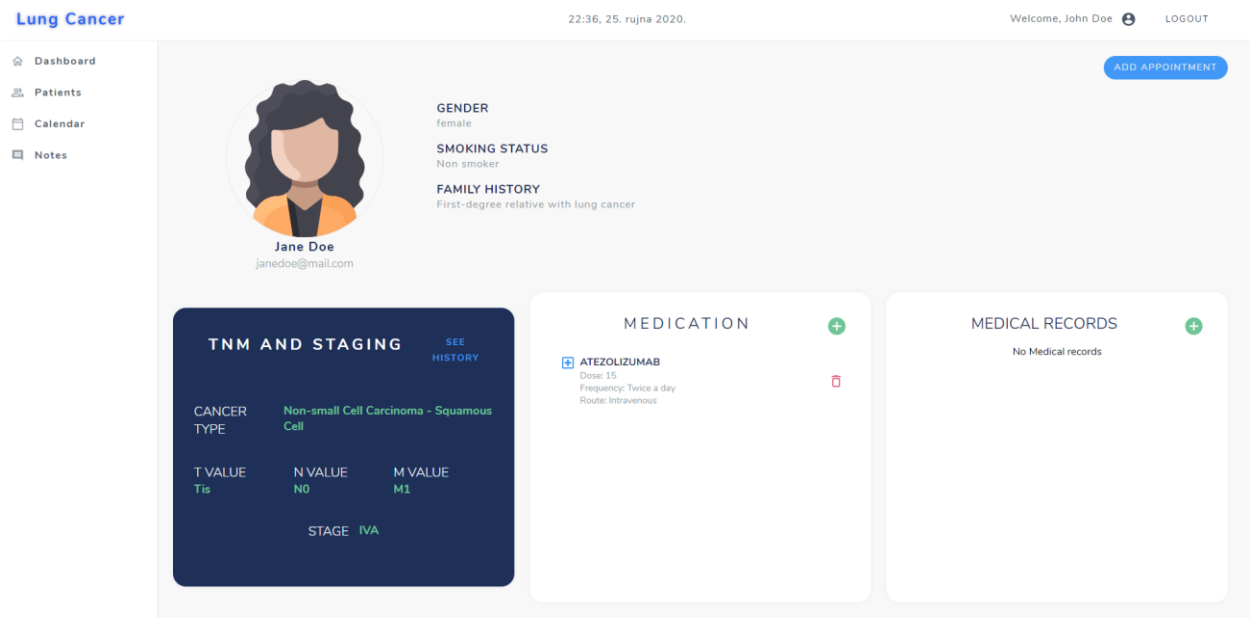


Sl. 6.8. *Forma za dodavanje novog pacijenta*

Kako bi liječnik dodao pacijenta, potrebno je ispuniti formu sa opći podacima i podacima vezanim uz rak pluća. Opći podaci su ime, prezime, email, spol i pušački status. Podaci vezani uz rak pluća su povijest bolesti vezanih uz rak pluća, obiteljska anamneza, tip raka pluća, t, n i m vrijednosti

prema kojima se određuje stadij raka pluća, čiji je postupak detaljnije opisan u poglavlju 5.2.2, a t, n i m vrijednosti tablicom 5.1.

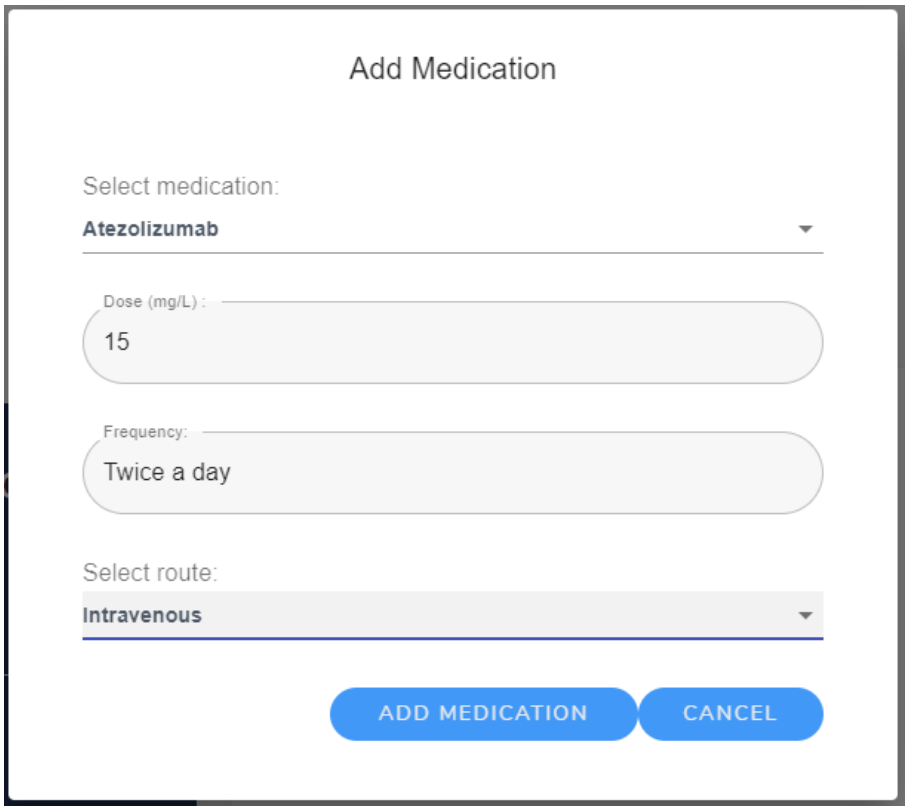
Također, s istog zaslona pritiskom na redak u tablici na pacijenta otvara se profil s detaljnijim prikazom podataka o odabranom pacijentu. Zaslon sa profilom pacijenta u pogledu liječnika prikazan je slikom 6.9. Detaljan prikaz pacijenta sadrži opće podatke: ime, prezime, email, spol, i povijest bolesti. Zatim, stadij i tip raka, te propisane lijekove i liječničke nalaze.



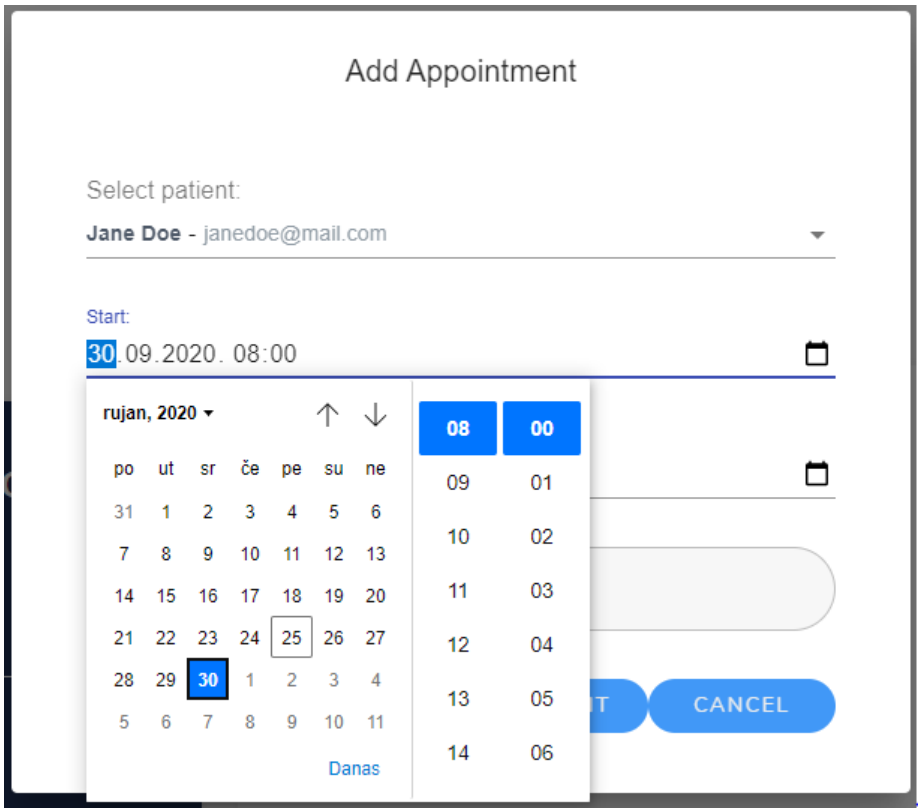
Sl 6.9. *Zaslon profila pacijenta u pogledu liječnika*

Sa pacijentovog profila moguće je zakazati liječnički termin, dodati lijekove ili nalaze te pogledati povijest stadija raka. Dodavanje termina omogućeno je pritiskom na tipku *Add Appointment*, koja otvara dijalog sa formom prikazanom na slici 6.10. U formu je potrebno unijeti podatke o pacijentu za kojeg se zakazuje termin, koje je inicijalno postavljeno na pacijenta s čijeg profila je otvoren dijalog, zatim početak i kraj termina, za što se koristi komponenta s zajedničkim biranjem datuma i vremena prikazana na slici 6.11, te opis liječničkog termina.

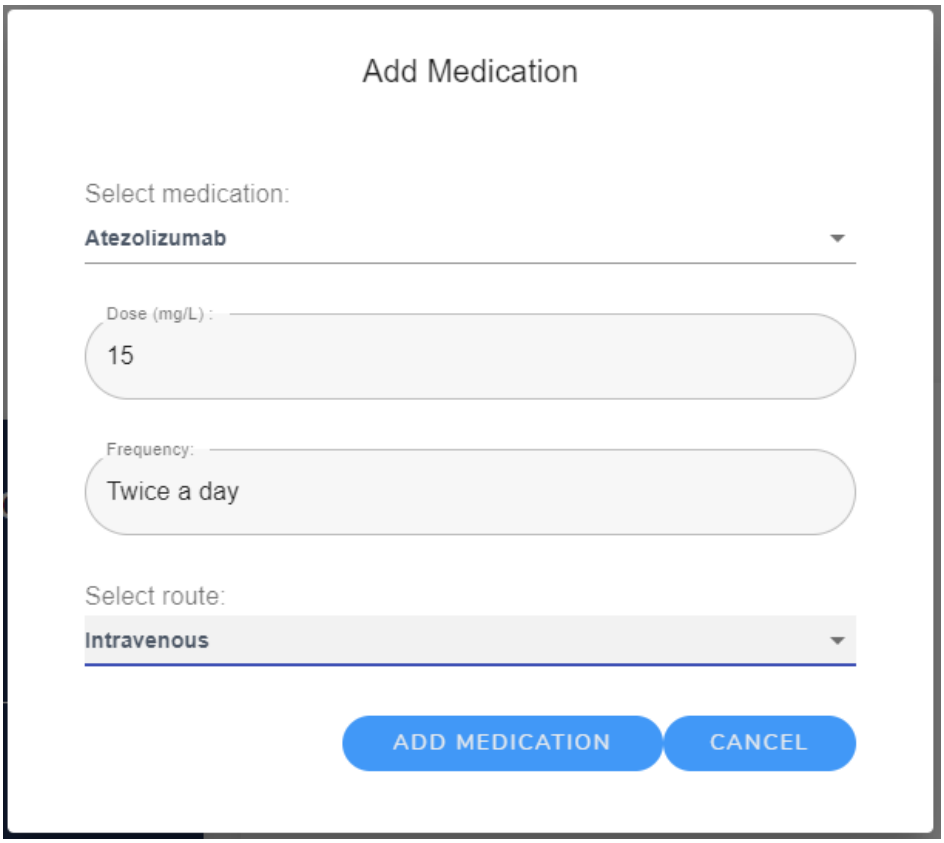
Jednako tako, za dodavanje lijekova se otvara dijalog prikazan na slici 6.12., gdje se treba popuniti forma s podacima o nazivu lijeka, dozi, učestalosti korištenja i načinu primjene. Svaki dodani lijek, samo korisnik u ulozi liječnika može ukloniti.



Sl. 6.10. *Forma za dodavanje liječničkog termina*

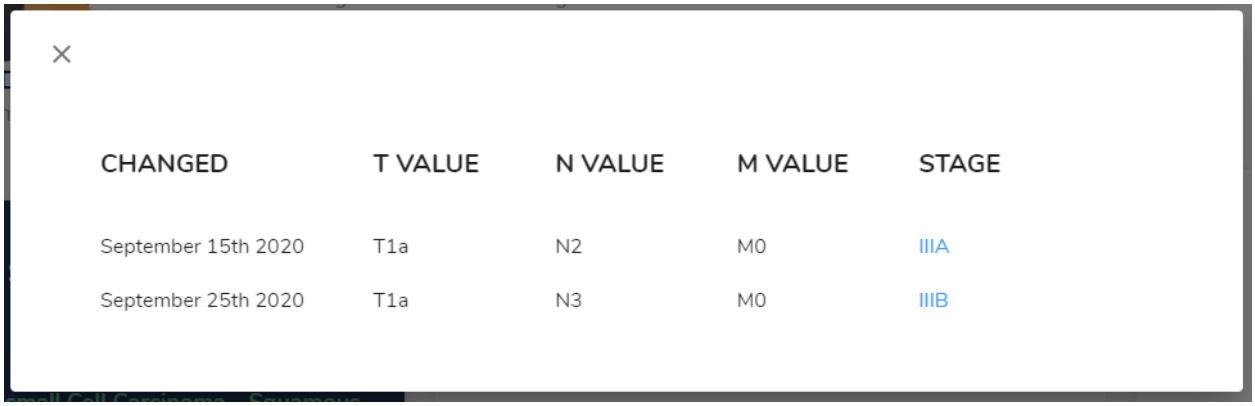


Sl. 6.11. *Komponenta za odabir datuma i vremena*



Sl. 6.12. *Forma za dodavanje lijeka*

Dijalogu koji prikazuje povijest stadija raka pacijenta moguće je pristupiti sa profila pacijenta pritiskom na tipku *See History* koja otvara dijalog prikazan na slici 6.13. u dijalogu se nalaze podaci o t, n i m vrijednostima, stadiju raka i datum izmjene. Dijalogu mogu pristupiti korisnik u ulozi pacijenta i liječnika.

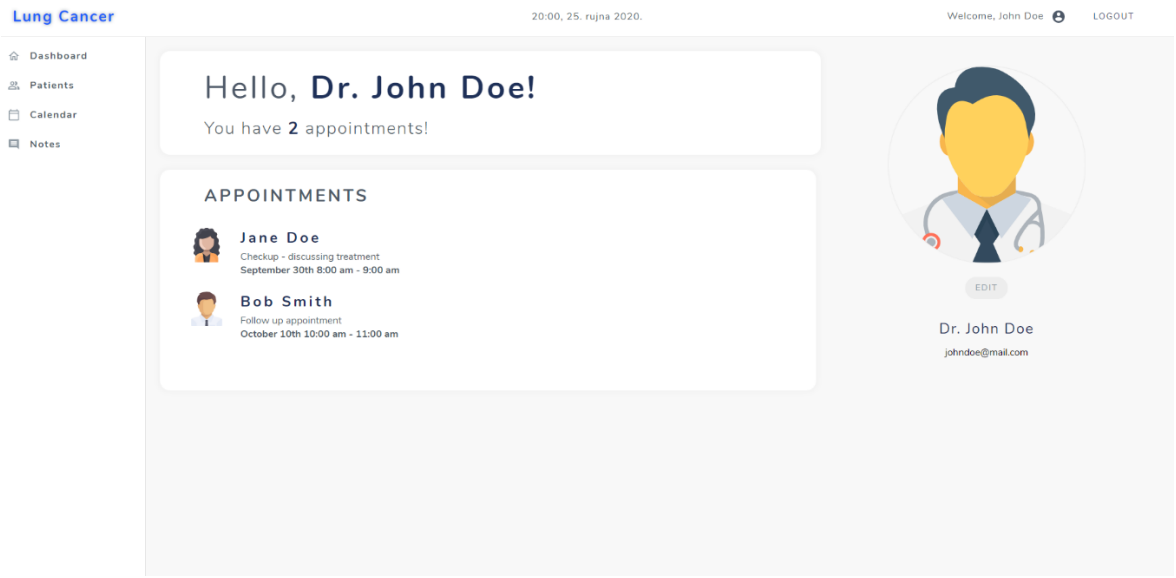


Sl. 6.13. *Dijalog povijesti stadija raka*

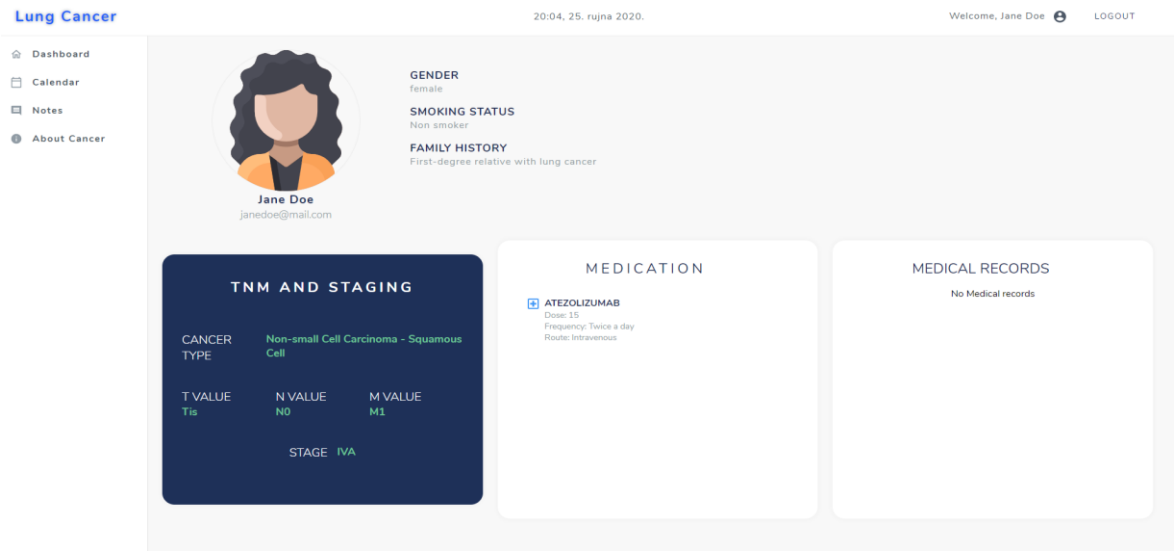
6.2.4. Zaslon profila u pogledu pacijenta i liječnika

Korisnik u ulozi liječnika, nakon prijave je preusmjeren na naslovni zaslon prikazan na slici 6.14 gdje se nalaze podaci o profilu, koji može urediti i podaci o liječničkim terminima koji sadrže vrijeme, pacijenta i opis termina. Korisnik u ulozi liječnika može u bilo kojem trenutku pristupiti profilu klikom na *Dashboard* link na navigaciji aplikacije.

Korisnik u ulozi pacijenta također je nakon prijave preusmjeren na naslovni zaslon prikazan na slici 6.15 gdje se nalaze detaljni podaci o korisniku, te je zaslon jednak zaslonu prikazanom na slici 6.9 kojem ima pristup samo korisnik u ulozi liječnika, osim što korisnik u ulozi pacijenta nema mogućnost dodavanja termina, lijekova i nalaza.



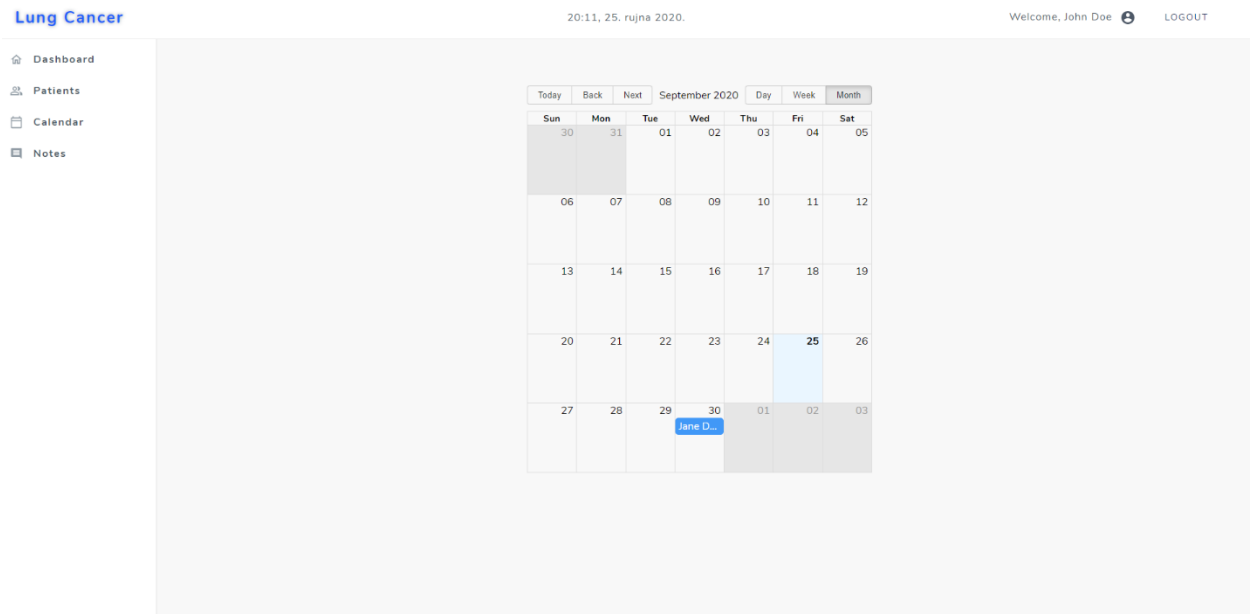
Sl. 6.14. *Prikaz profila korisnika u ulozi liječnika*



Sl. 6.15. *Prikaz profila korisnika u ulozi pacijenta*

6.2.5. Zaslon kalendara liječničkih termina

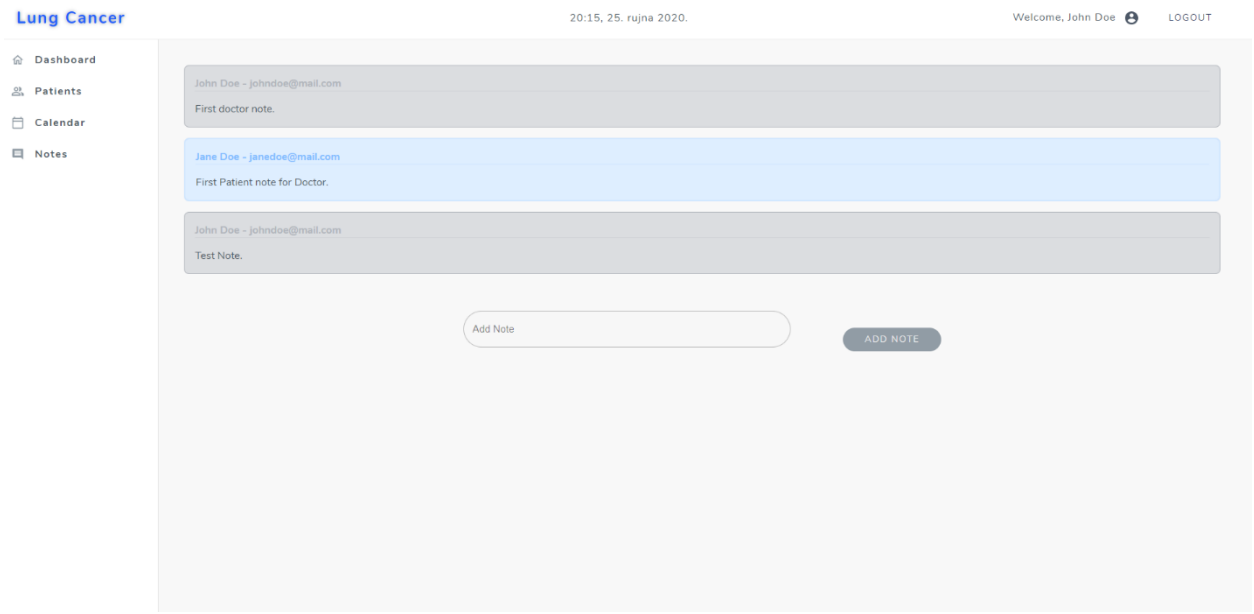
Zaslon kalendara, prikazan slikom 6.16, s označenim liječničkim terminima jednak je za oba korisnika. Razlika je što korisnik u ulozi pacijenta ima prikazano termine vezane uz sebe, a korisnik u ulozi liječnika ima prikazane termine za sve pacijente. Kalendar omogućava biranje dnevnog, tjednog i mjesečnog pregleda. Kalendaru korisnik može pristupiti odabirom *Calendar* linka u navigaciji aplikacije.



Sl. 6.16. *Prikaz kalendara*

6.2.6. Zaslon bilješki

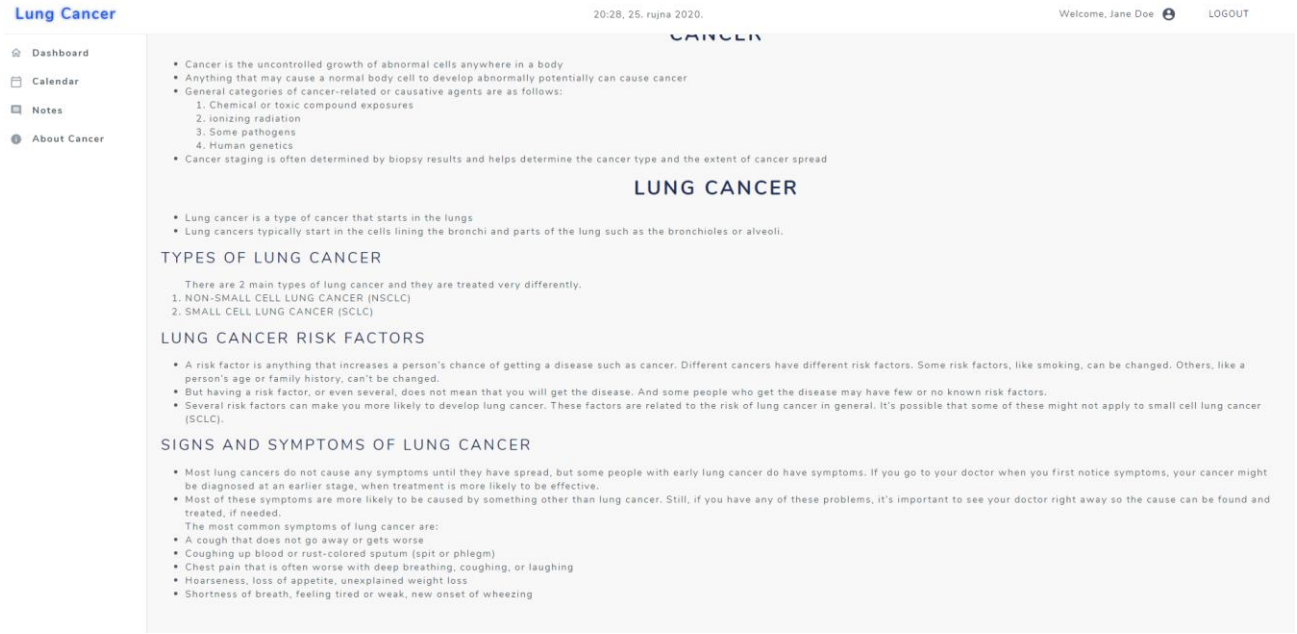
Zaslon na kojem su prikazane bilješke između pacijenta i liječnika također je jednak za oba korisnika. Izgled zaslona prikazan je na slici 6.17. Svaka bilješka ima navedeno ime, prezime i mail autora te sadržaj bilješke. Također, na dnu je predviđeno polje za upis nove bilješke.



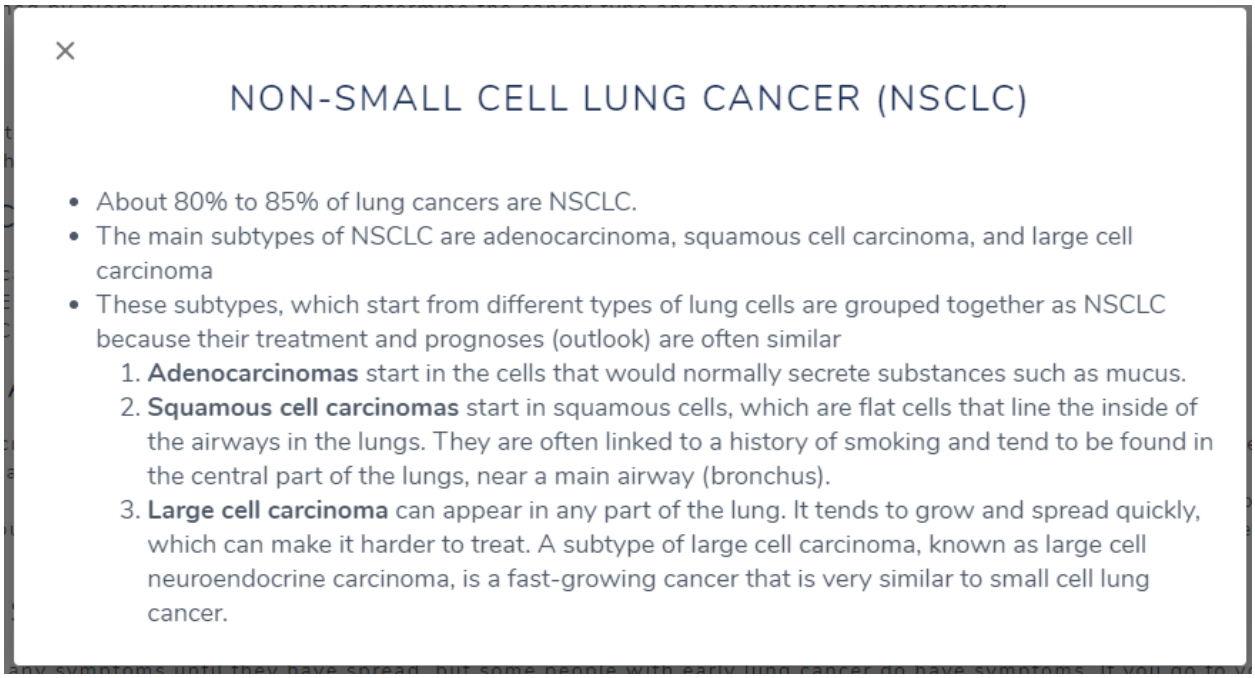
Sl. 6.17. *Prikaz zaslona s bilješkama*

6.2.7. Zaslon edukacijskog sadržaja o raku

Posljednji zaslon je edukacijski sadržaj o raku i raku pluća prikazan na slici 6.16. Edukacijskom sadržaju mogu pristupiti samo korisnici u ulozi pacijenta, pritiskom na *About* link u navigaciji. Navedene su informacije o raku općenito, raku pluća, tipovima raka pluća, faktorima rizika i simptomima raka pluća koji su detaljno objašnjeni u 3. poglavlju. Pritiskom na tipove raka pluća otvara se dijalog sa detaljnijim informacijama prikazan na slici 6.18.



Sl. 6.17. *Prikaz zaslona s edukacijskim sadržajem*



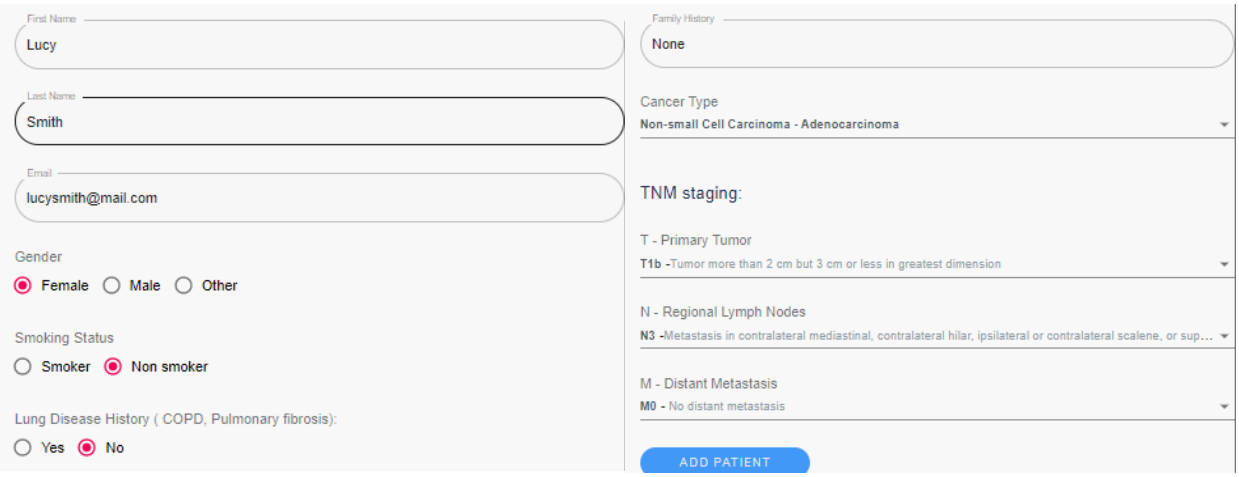
Sl. 6.18. *Prikaz dijaloga s detaljnijim opisom tipova raka pluća*

6.3.Testiranje i analiza rada sustava

Cjelokupno programsko rješenje sustava za personalizirano praćenje stanja oboljelih od zloćudnih bolesti testirano je uspoređivanjem podudaraju li se ulazni podaci s očekivanim rezultatom. Poglavljem 6.2 prikazana je vjerodostojnost i ispravnost prikazivanja ulaznih podataka na postojećim zaslonima aplikacije. U ovom poglavlju, osvrnut ćemo se na ispravnost rezultata pri dodavanju pacijenta i izračunu stadija raka. Izračun stadija raka testiran je na 3 skupa ulaznih podataka te se rezultat uspoređivao sa očekivanim vrijednostima iz tablice 5.1.

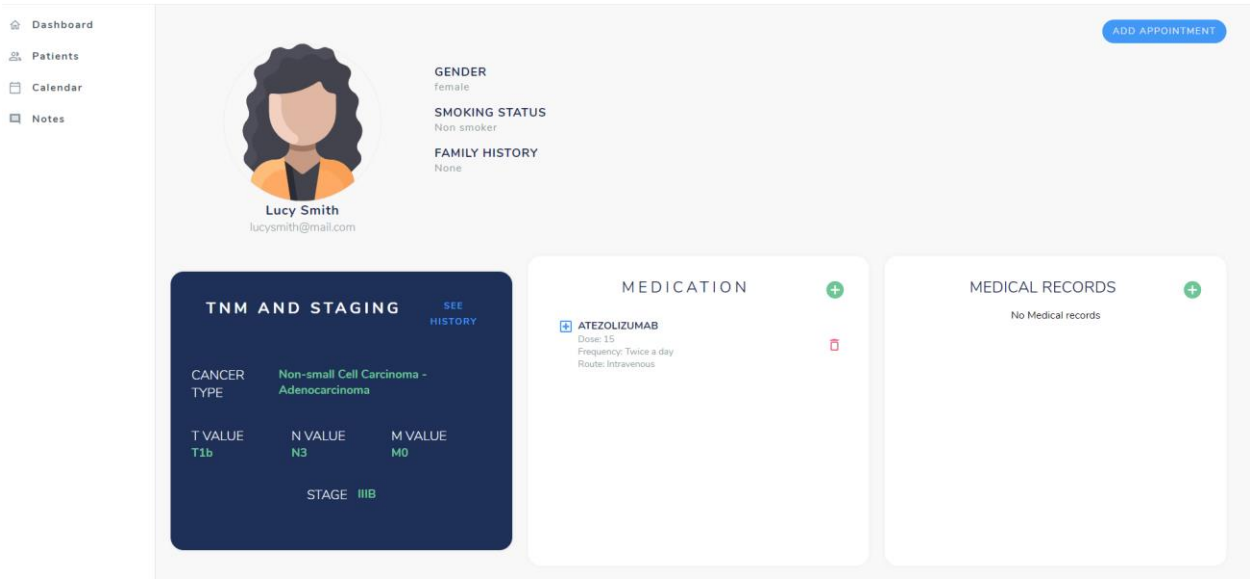
6.3.1. Prvi testni skup

Prvi skup ulaznih podataka prikazan je slikom 6.19.



Sl. 6.19. *Prvi skup ulaznih podataka za izračun stadija raka*

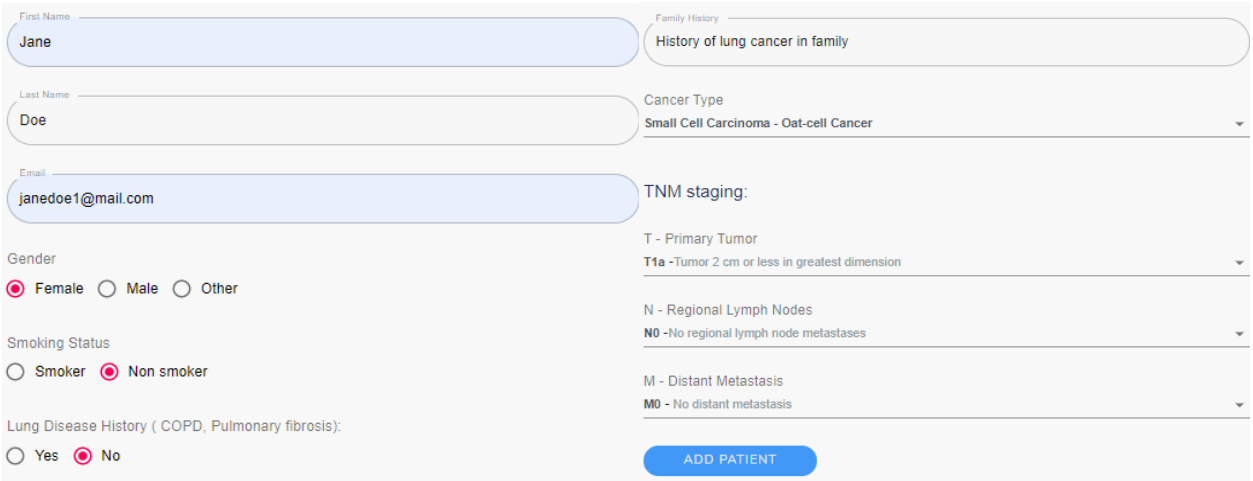
Na temelju unesenih *t*, *n* i *m* vrijednosti, očekivani rezultat prema tablici 5.1 za parametar stadija raka je IIIB, što na slici 6.20 možemo vidjeti da je ispravno izračunato na poslužiteljskoj stani i prikazano na korisničkoj strani. Također možemo vidjeti da su vrijednosti ime, prezime, email, pušački status, spol, povijest bolesti i tip raka ispravno prikazane na zaslonu profila pacijenta kojem mogu pristupiti liječnik i pacijent.



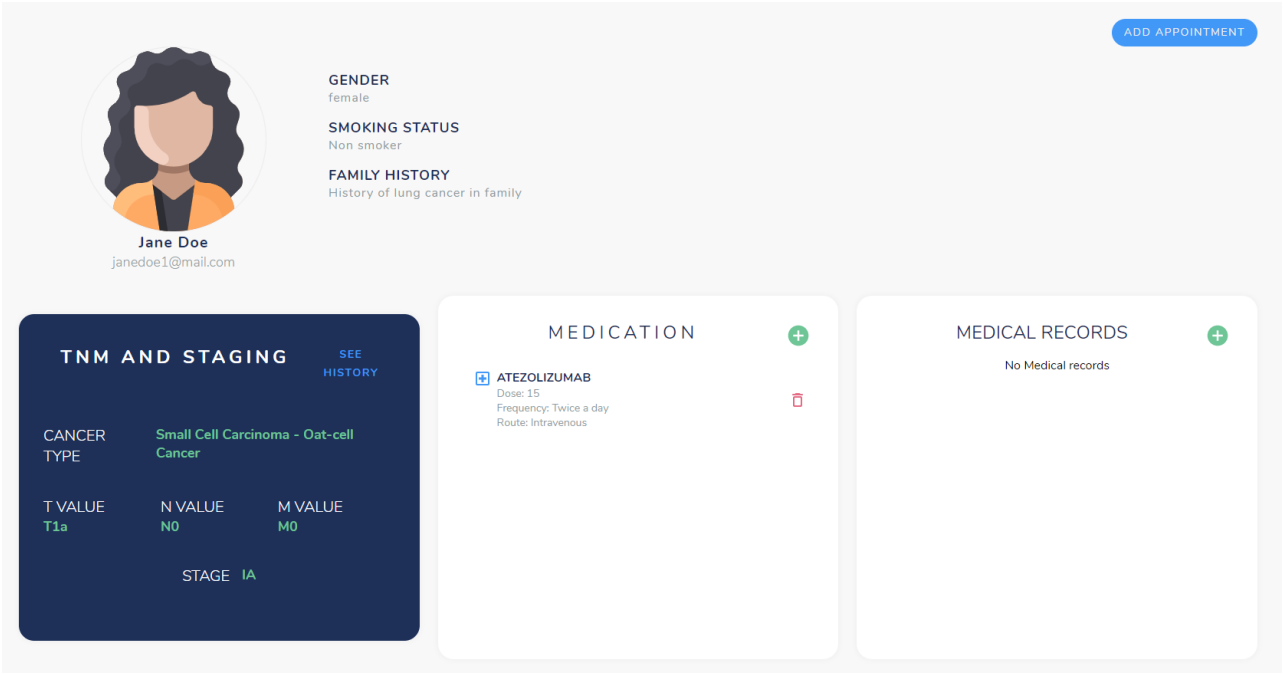
Sl. 6.20. *Rezultat prvog testnog skupa*

6.3.2. Drugi testni skup

Drugi skup ulaznih podataka prikazan je slikom 6.21. Na temelju unesenih vrijednosti, očekivani stadij raka prema tablici 5.1. je IA. Na slici 6.22 možemo vidjeti da je parametar stadija raka ispravno izračunat na poslužiteljskoj stani i prikazan na korisničkoj strani. Također možemo vidjeti da su vrijednosti ime, prezime, email, pušački status, spol, povijest bolesti i tip raka ispravno prikazane na zaslonu profila pacijenta kojem mogu pristupiti liječnik i pacijent.



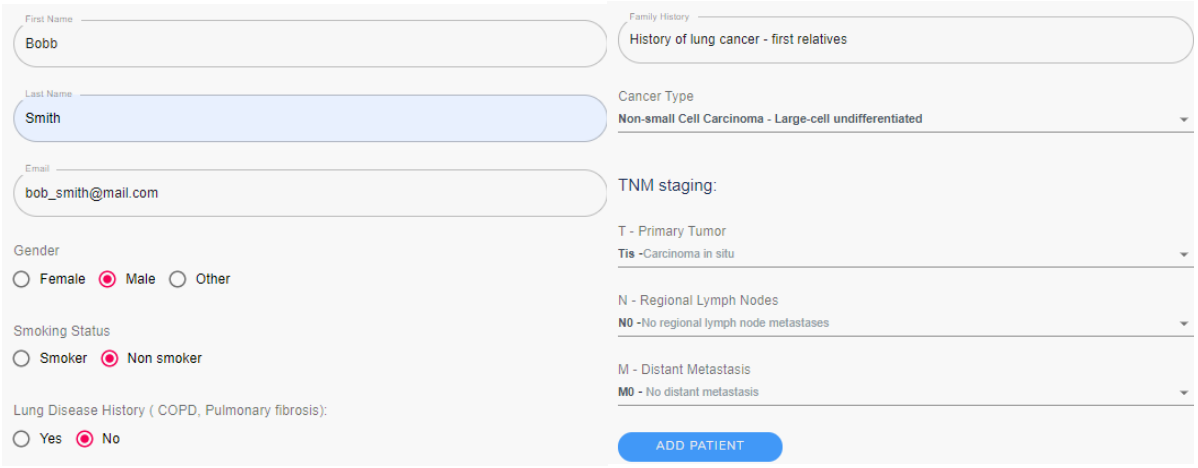
Sl. 6.21. *Drugi skup ulaznih podataka za izračun stadija raka*



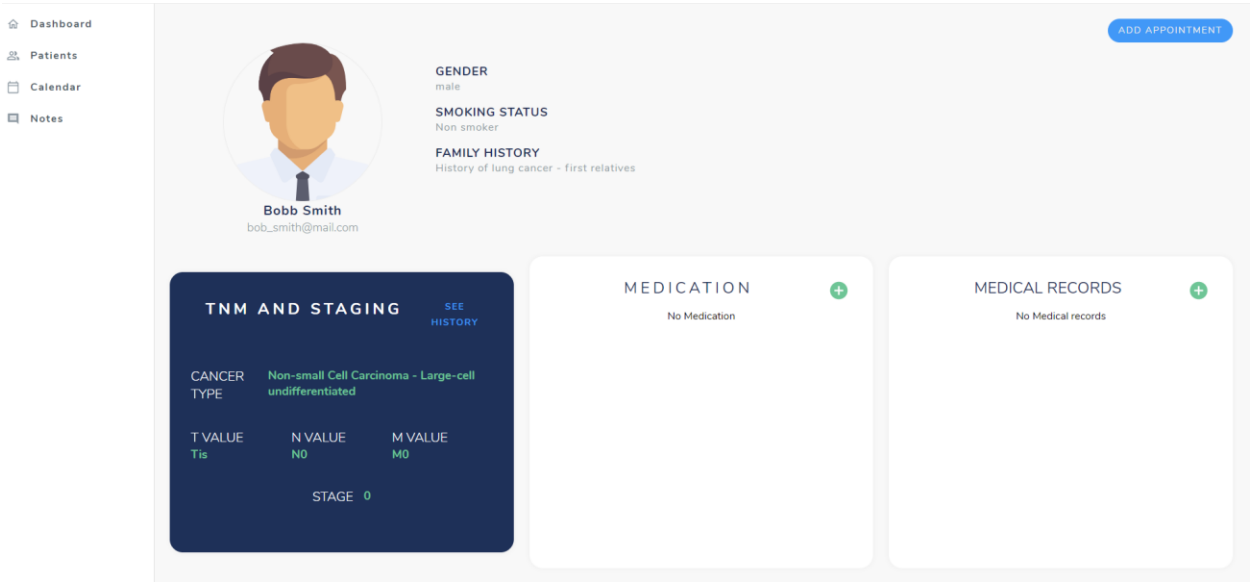
Sl. 6.22. *Rezultat drugog testnog skupa*

6.3.3. Treći testni skup

Treći skup ulaznih podataka prikazan je slikom 6.23. Na temelju unesenih vrijednosti, očekivani stadij raka prema tablici 5.1. je 0. Na slici 6.24 možemo vidjeti da je parametar stadija raka ispravno izračunat na poslužiteljskoj stani i prikazan na korisničkoj strani. Također možemo vidjeti da su vrijednosti ime, prezime, email, pušački status, spol, povijest bolesti i tip raka ispravno prikazane na zaslonu profila pacijenta kojem mogu pristupiti liječnik i pacijent.



Sl. 6.22. *Drugi skup ulaznih podataka za izračun stadija raka*



Sl. 6.23. *Rezultat trećeg testnog skupa*

6.3.4.Usporedba rezultata testnih skupova

Ručno testiranje izračuna stadija raka na poslužiteljskoj strani i prikaz stadija raka na korisničkoj strani provelo se usporedbom ulaznih i očekivanih podataka prikazanih tablicom 6.1.

Tab. 6.1. *Testni skupovi*

<table>
<tr><th colspan="2">Ulazni podaci</th><th colspan="2">Očekivani rezultat</th></tr>
<tr><td colspan="4">Prvi testni skup podataka</td></tr>
<tr><td>T</td><td>T1b</td><td rowspan="3">Stadij raka</td><td rowspan="3">IIIB</td></tr>
<tr><td>N</td><td>N3</td></tr>
<tr><td>M</td><td>M0</td></tr>
<tr><td colspan="4">Drugi testni skup podataka</td></tr>
<tr><td>T</td><td>T1a</td><td rowspan="3">Stadij raka</td><td rowspan="3">IA</td></tr>
<tr><td>N</td><td>N0</td></tr>
<tr><td>M</td><td>M0</td></tr>
<tr><td colspan="4">Treći testni skup podataka</td></tr>
</table>

T	Tis	Stadij raka	0
N	N0		
M	M0		

Ulazni podaci uspoređeni su s očekivanim podacima koji su određeni prema tablici 5.1.u kojoj je opisan prikaz ovisnosti *t, n* i *m* parametara o stadiju raka. Vrijednost t parametra predstavlja rast primarnog tumora, vrijednost n predstavlja stadij širenja raka na limfne čvorove, a m parametar predstavlja stadij metastaze raka. Navedeni parametri su detaljnije objašnjeni u poglavlju 3.5.1. Očekivani podaci u tablici za svaki testni skup ulazni podataka određeni su prema pravilima priručnika Američkog odbora za rak [27] te su trenutno aktualni s obzirom da se podaci periodički nadopunjavaju na temelju novih saznanja u području medicine i onkoloških istraživanja.

7. ZAKLJUČAK

Razvijeni sustava na strani korisnika za personalizirano praćenje pacijenata oboljelih od zloćudnih bolesti prikazuje prijeko potrebnu primjenu računalne tehnologije i programskih rješenje u medicini. Ciljana skupina korisnika su liječnici, pacijenti oboljeli od raka pluća i pacijenti u remisiji. Razvijeni sustav poboljšava praćenje napretka liječničke dijagnoze i tretmana, kao i zdravstvenog stanja oboljelih, te unaprjeđuje proces komunikacije liječnika i pacijenata koji aktivno sudjeluju u vlastitom oporavku.

Za izradu programskog rješenja definirani su zahtjevi i funkcionalnosti koje bi aplikacija trebala sadržavati. Prema definiranim zahtjevima, razvijeno je idejno rješenje i model sustava i izabrane odgovarajuće tehnologije. Razvijeni sustav razlikuje dva pogleda, pogled liječnika i pacijenta. Liječnik ima pristup sučelju za registraciju, prijavu, prikaz profila, prikaz popisa pacijenata kao i njihovih profila te dodavanja pacijenta, prikaz kalendara sa zakazanim terminima i pristup sučelju za bilješke između pacijenta i liječnika. Pacijent ima pristup sučelju prijave, prikaza vlastitog profila, kalendara, sučelja za bilješke s doktorom i sučelja sa informativnim sadržajem o raku i raku pluća. Testiranjem i analizom potvrđena je ispravnost prikaza ulaznih podataka na korisničkoj strani sučelja.

Moguća unaprjeđenja web sustava mogla bi se baviti daljnjim razvojem sustava za komunikaciju pacijenta i liječnika, te obavještavanja o mogućim ili postojećim nepravilnostima razvojem algoritma za određivanje stadija raka pluća.

LITERATURA

[1] NCCN GUIDELINES FOR PATIENTS, Lung Cancer: Early and Locally Advanced, https://www.nccn.org/patients/guidelines/content/PDF/lung-early-stage-patient.pdf, pristupljeno: 12.7.2020.

[2] NCCN GUIDELINES FOR PATIENTS, Lung Cancer Screening, https://www.nccn.org/patients/guidelines/content/PDF/lung_screening-patient.pdf, pristupljeno: 12.7.2020.

[3] NCCN GUIDELINES FOR PATIENTS, Lung Cancer Metastatic, https://www.nccn.org/patients/guidelines/content/PDF/lung-metastatic-patient.pdf, pristupljeno: 12.7.2020.

[4] World Health Organization, WHO report on cancer: setting priorities, investing wisely and providing care for all, https://www.who.int/publications/i/item/who-report-on-cancer-setting-priorities-investing-wisely-and-providing-care-for-all, pristupljeno: 10.7.2020.

[5] Cancer Treatment: Centers of America, Lung Cancer, https://www.cancercenter.com/cancer-types/lung-cancer, pristupljeno: 9.7.2020.

[6] ESMO Patient Guide Series, Što je rak pluća nemalih stanica?, https://www.esmo.org/content/download/124824/2364857/file/HR-Rak-Pluca-Nemalih-Stanica-NSCLC-Vodic-za-Bolesnike.pdf, pristupljeno: 9.7.2020.

[7] MSD Medicinski priručnik za pacijente, Rak pluća, http://www.msd-prirucnici.placebo.hr/msd-za-pacijente/bolesti-pluca-i-disnih-putova/rak-pluca, pristupljeno: 10.7.2020.

[8] HRVATSKA LIGA PROTIV RAKA, Rak pluća, http://hlpr.hr/rak/vijest/rak-pluca, pristupljeno: 11.7.2020.

[9] Radio Chirurgia Zagreb, Što je MSCT angiografija?, https://radiochirurgiazagreb.com/hr/2019/04/10/sto-je-msct-angiografija, pristupljeno: 10.7.2020.

[10] WHO, Guide to Cancer: Early Diagnosis, https://apps.who.int/iris/bitstream/handle/10665/254500/9789241511940-eng.pdf?sequence=1, pristupljeno: 12.7.2020.

[11] C.M. Ionescu, The Human Respiratoy System, Springer-Verlag, London, 2013.

[12] J.A. Roth, W.K. Hong, J. Cox, Lung Cancer, Blackwell Publishing, Third edition 2008.

[13] A. Torres, Lung Cancer: Strategies for diagnosis and treatment, British Library, London, 2018

[14] I.S. Graham, The HTML Sourcebook, Wiley, New York, 1995.

[15] R.W. Sebesta, Programming the World Wide Web, Pearson, 2014.

[16] A. Fedosejev, React.js Essentials, Packt Publishing, 2015.

[17] D. Bugl, Learning Redux, Packt Publishing, 2017.

[18] A. Boduch, React Material-UI Cookbook, Packt Publishing, 2019.

[19] A. Jemal, F. Bray, J. Frelay, E. Ward, D. Forman, CA: A Cancer Journal for Clinicians, Global cancer statistics, br. 2, sv. 60, str. 69-90, Ožujak – Travanj 2011.

[20] F. Denis, E. Basch, A. Septans, Two-Year Survival Comparing Web-Based Symptom Monitoring vs Routine Surveillance Following Treatment for Lung Cancer, siječanj 2019.

[21] F. Denis, l. Viger, A. Charron, Detection of lung cancer relapse using self-reported symptoms transmitted via an Internet Web-application: pilot study of the sentinel follow-up, siječanj 2014

[22] LUNGevity Foundation introduces a new mobile app to help patients understand and manage life with lung cancer. LUNGevity website. http://www.multivu.com/players/English/7865551-lungevity-lung-cancer-navigator-app/, pristupljeno: 17.9.2020.

[23] Cancer.Net Mobile, https://www.cancer.net/navigating-cancer-care/managing-your-care/cancernet-mobile, pristupljeno: 17.9.2020.

[24] Y. Omar, A. Tasleem, M. Pasquierm, A. Sagahyroon, Proceedings of the 11th International Joint Conference on Biomedical Engineering Systems and Technologies, Lung Cancer Prognosis System using Data Mining Techniques, sv. 5, str. 361 – 368

[25] C. Nance, TypeScript Essentials, Packt Publishing, 2014.

[26] M. Antunović, Programsko rješenje web sustava na strani poslužitelja za personalizirano praćenje pacijenata oboljelih od zloćudnih bolesti zasnovano na funkcijskom programiranju, Diplomski rad, Fakultet elektrotehnike, računarstva i informacijskih tehnologija, 2020.

[27] American Joint Committee On Cancer, AJCC CANCER STAGING MANUAL 7th edition, Springer, 2015.

[28] K. Arslan, S. Yasar, C. Colak, An Intelligent System for the Classification of Lung Cancer Based on Deep Learning Strategy, International Artificial Intelligence and Data Processing Symposium, 2019

SAŽETAK

Sustav na strani korisnika za personalizirano praćenje pacijenata oboljelih od zloćudnih bolesti razvijen u ovom radu temelji se na potrebama poboljšanja prevencije oboljenja od raka pluća, sprječavanja izlaska oboljelih iz remisije detaljnim praćenjem oporavka, te poboljšanjem kvalitete i uspješnosti liječničkih dijagnoza. Na primjeru razvijenog sustava možemo uočiti prijeko potrebna primjena računalnih tehnologija i programskih rješenja u medicinskom području. Razvijeni web sustav se sastoji od web aplikacije koja nakon unesenih podataka, prikazuje obrađene podatke te šalje povratnu informaciju liječniku i pacijentu o zdravstvenom stanju oboljelog. Kroz rad je na temelju praktičnog dijela opisan razvoj web sustava i njegove funkcionalnosti. Rezultati analize sustava pokazuju učinkovitost i potencijal u poboljšanju praćenja zdravstvenog stanja oboljelih od raka pluća te komunikacije pacijenta i liječnika.

Ključne riječi: personalizirana medicina, rak pluća, web aplikacija, zdravstveno stanje, zloćudne bolesti

ABSTRACT

Client-side based system for personalized monitoring of cancer patients developed for this final paper is based on the need to minimize the impact of living with cancer and developing better care for cancer patients in the remission, as well as improving quality of life with cancer. By developing and using web application it is easy to notice the great need and advantage of computer science in the medical field. The main concept of the application is to estimate the cancer stage of the patient based on data input provided by the doctor, and to help the patient as well as the doctor to monitor the health status more efficiently and to help the patient better understand his situation. In this paper are elaborated all the functionalities of the web application, along with the application performance analysis.

Key words: personalized medicine, lung cancer, web application, health, malignant diseases

ŽIVOTOPIS

Maja Ivezić rođena 8.veljače 1997. godine u Osijeku. U Osijeku stječe osnovnoškolsko obrazovanje u Osnovnoj školi Antuna Mihanovića od 2003. do 2011. godine. Prirodoslovno matematičku gimnaziju Osijek upisuje 2011.godine. Nakon završetka srednjoškolskog obrazovanja, 2015. godine upisuje Fakultet Elektrotehnike, Računarstva i Informacijskih tehnologija u Osijeku, preddiplomski studij računarstva. Na trećoj godini studija zapošljava se u informatičkoj tvrtki Kod Savjetovanje gdje radi do dan danas.

_____

Maja Ivezić

PRILOZI

Prilog 1: Diplomski rad u .pdf formatu.

Prilog 2: Diplomski rad u .docx formatu

Prilog 3: Programski kod aplikacije i testova u .zip datoteci.