

# Programsko inženjerstvo

Ak. god. 2023./2024.

## CookBooked

Dokumentacija, Rev. 2

Grupa: *Culinary Chaos Crew*

Voditelj: *Dominik Dejanović*

Datum predaje: *19.01.2024.*

Nastavnik: *Nikolina Frid*

# Sadržaj

<b>1 Dnevnik promjena dokumentacije</b>	<b>3</b>
<b>2 Opis projektnog zadatka</b>	<b>6</b>
<b>3 Specifikacija programske potpore</b>	<b>9</b>
3.1 Funkcionalni zahtjevi . . . . .	9
3.1.1 Obrasci uporabe . . . . .	11
3.1.2 Sekvencijski dijagrami . . . . .	23
3.2 Ostali zahtjevi . . . . .	30
<b>4 Arhitektura i dizajn sustava</b>	<b>31</b>
4.1 Baza podataka . . . . .	33
4.1.1 Opis tablica . . . . .	33
4.1.2 Dijagram baze podataka . . . . .	36
4.2 Dijagram razreda . . . . .	37
4.3 Dijagram stanja . . . . .	41
4.4 Dijagram aktivnosti . . . . .	43
4.5 Dijagram komponenti . . . . .	45
<b>5 Implementacija i korisničko sučelje</b>	<b>46</b>
5.1 Korištene tehnologije i alati . . . . .	46
5.2 Ispitivanje programskog rješenja . . . . .	49
5.2.1 Ispitivanje komponenti . . . . .	49
5.3 Dijagram razmještaja . . . . .	51
5.4 Upute za puštanje u pogon . . . . .	52
<b>6 Zaključak i budući rad</b>	<b>56</b>
<b>Popis literature</b>	<b>57</b>
<b>Indeks slika i dijagrama</b>	<b>59</b>

**Dodatak: Prikaz aktivnosti grupe**

**60**

# 1. Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Dodana shema baze podataka.	Čavar	19.10.2023.
0.2	Napravljen predložak.	Nikolić	22.10.2023.
0.3	Ispravak baze podataka.	Čavar	24.10.2023.
0.4	Dodan opis projektnog zadatka.	Vuković	25.10.2023.
0.5	Dodani obrasci uporabe za korisnika.	Dejanović	25.10.2023.
0.5.1	Ispravak obrazaca uporabe za korisnika. Dodani obrasci uporabe za recepte.	Dejanović	26.10.2023.
0.6	Prva verzija cjelokupnog UML dijagrama.	Volarević	27.10.2023.
0.7	Ispravak obrazaca uporabe za korisnika.	Nikolić	27.10.2023.
0.8	Završen opis projektnog zadatka.	Vuković	27.10.2023.
0.9	Ispravak baze podataka.	Čavar	27.10.2023.
0.10	Dodani sastanci u dnevnik sastanaka.	Dejanović	27.10.2023.
0.11	Dodani UML dijagrami za UC1-4.	Dejanović	30.10.2023.
0.12	Ispravak obrazaca uporabe.	Nikolić	6.11.2023.
0.13	Dodani UC6-7 i UC13-14 u sekvencijske dijagrame.	Nikolić	6.11.2023.
0.14	Ispravak baze podataka.	Vuković	12.11.2023.
0.15	Dodana dokumentacija za bazu podataka.	Trebus	13.11.2023.
0.16	Ispravak dijagrama za obrasce uporabe i sekvencijskih dijagrama.	Nikolić	15.11.2023.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodatka	Autori	Datum
0.17	Ispravak baze podataka.	Vuković	15.11.2023.
0.18	Dodani sastanci u dnevnik sastanaka.	Dejanović	15.11.2023.
0.19	Ispravak baze podataka. Dodan dijagram razreda.	Trebus	16.11.2023.
0.20	Popravljen format slika. Popunjen dnevnik promjena dokumentacije.	Vuković	16.11.2023.
0.21	Dodan opis arhitekture i dizajna sustava.	Trebus	17.11.2023.
0.22	Popravak UML dijagrama.	Nikolić	17.11.2023.
0.23	Zamjena slika UML dijagrama s onima veće kvalitete.	Dejanović	17.11.2023.
0.24	Dodani nefunkcionalni zahtjevi.	Nikolić	17.11.2023.
0.25	Dodani svi dijagrami razreda.	Trebus	17.11.2023.
0.26	Updateana lista sastanaka	Dejanović	17.11.2023.
1.0	Verzija samo s bitnim dijelovima za 1. ciklus.		17.11.2023.
1.1	Popravljen UML1 dijagram.	Nikolić	10.1.2024.
1.2	Dodan dijagram stanja.	Vuković	10.1.2024.
1.3	Dodani dijagrami aktivnosti i razmještaja, ažuriran dijagram stanja.	Vuković	13.1.2024.
1.4	Dodan dijagram komponenti, ažuriran dijagram razmještaja.	Vuković	14.1.2024.
1.5	Popravljena dokumentacija arhitekture sustava.	Trebus	14.1.2024.
1.6	Dodan opis dijagrama komponenti.	Vuković	15.1.2024.
1.7	Popravljeni sekvencijski dijagrami	Nikolić	16.1.2024.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodatka	Autori	Datum
1.8	Dodan dijagram klasa za trenutnu implementaciju.	Nikolić	17.1.2024.
1.9	Dodano poglavlje 5.1.	Nikolić	18.1.2024.
1.10	Dodani opisi dijagrama i dodano poglavlje 5.4	Nikolić	19.1.2024.
2.0	Konačna verzija dokumentacije		19.1.2024.

## 2. Opis projektnog zadatka

### Uvod

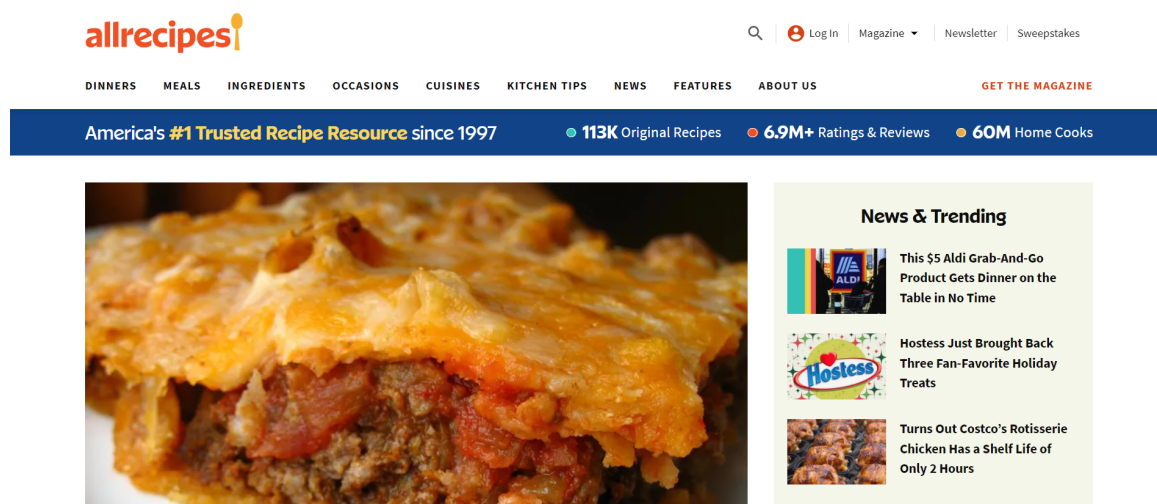
Cilj ovog projekta je razviti *CookBooked*, platformu za zajednicu ljubitelja kuhanja i pečenja kolača, koja omogućava korisnicima razmjenu i otkrivanje recepata iz cijeloga svijeta. Osim toga, platforma treba omogućiti bolje razumijevanje i komunikaciju između korisnika koji dijele istu strast prema kuhanju. Autori recepata će dobiti priliku izgraditi svoju reputaciju, povećati svoju publiku i razmijeniti iskustva s drugima. To će stvoriti motivaciju za kontinuirano stvaranje visokokvalitetnih recepata i sadržaja. S druge strane, korisnici će dobiti priliku unaprijediti vlastite vještine i pričati s autorima recepata te dobiti inspiraciju za vlastite eksperimente u kuhinji. Programska potpora za aplikaciju implementirat će se u obliku web aplikacije.



Slika 2.1: *CookBooked* logo

## Postojeća slična rješenja

Iako postoje mnoge web stranice i aplikacije za dijeljenje recepata, *CookBooked* se izdvaja po svojoj fokusiranosti na interakciju između korisnika. Primjeri postojećih sličnih rješenja uključuju *AllRecipes*, *Tasty* i *Food Network*, no *CookBooked* se razlikuje po svojim značajkama za komunikaciju među korisnicima i praćenje autora.



Slika 2.2: Izgled naslovne stranice *AllRecipes*

## Skup korisnika

Platformu *CookBooked* može koristiti bilo koji zaljubljenik u hranu i njenu pripremu. Smatramo da će platforma biti najpogodnija ovim skupinama:

- kuhari i ljubitelji kuhanja
- osobe s posebnim prehrambenim potrebama (vegetarijanci, vegani, osobe koje ne jedu gluten...)
- ljudi koji traže inspiraciju za obroke
- autori recepata koji žele podijeliti svoje vještine



## Opseg projektnog zadatka

Za ostvarenje našeg ambicioznog projekta potrebno je slijedeće:

- razvoj platforme sa svim navedenim funkcionalnostima
- stvaranje korisničkog sučelja koje je jednostavno za korištenje
- implementaciju sustava za komunikaciju među korisnicima
- kreiranje profila za sve tipove korisnika
- upravljanje korisnicima i receptima od strane administratora

## Moguće nadogradnje projektnog zadatka

Iako je ovaj projekt unio neke nove ideje i funkcionalnosti u svijet kulinarstva, još uvijek postoje neke nadogradnje i funkcionalnosti koje bi mogle biti implementirane u budućnosti. Neki primjeri nadogradnji su:

- dodavanje dodatnih značajki za napredno pretraživanje recepata
- razvijanje mobilne aplikacije za pristup platformi putem mobilnih uređaja
- uvođenje funkcionalnosti za online prodaju kuhinjskih proizvoda
- integraciju s društvenim mrežama radi širenja korisničke baze
- unapređenje sigurnosti i privatnosti podataka korisnika

## 3. Specifikacija programske potpore

### 3.1 Funkcionalni zahtjevi

#### Dionici:

1. Korisnik
2. Administrator
3. Razvojni tim
4. Vanjski sustav za komunikaciju

#### Aktori i njihovi funkcionalni zahtjevi:

1. Neregistrirani korisnik (inicijator) može:
  - (a) vidjeti listu recepata svih korisnika
  - (b) pretražiti recepte po naslovu i korisniku
    - i. filtrirati listu prema kategoriji, vrsti kuhinje i sastojcima
  - (c) pregledati detalje zasebnih recepata
  - (d) stvoriti novi korisnički račun
2. Registrirani korisnik (inicijator) može:
  - (a) ulogirati se pomoću vlastitog korisničkog računa
  - (b) objaviti nove recepte
    - i. odrediti kategoriju recepta
    - ii. dodati posebne značajke i tagove na recept
    - iii. priložiti slike ili videozapise receptu
  - (c) pregledati listu vlastitih recepata
  - (d) obrisati vlastiti recept
  - (e) komunicirati s drugim registriranim korisnicima
    - i. razmjenjivati poruke i pozive
    - ii. postaviti termine otvorene za komunikaciju
  - (f) označiti, komentirati i spremiti recepte
  - (g) pratiti omiljene autore i primati obavijesti o njihovim novim objavama

(h) promijeniti detalje korisničkog računa:

- i. korisničko ime
- ii. lozinku

(i) deaktivirati vlastiti korisnički račun

3. Administrator (inicijator) može:

- (a) pregledati objavljene recepte
- (b) promijeniti korisničke podatke registriranih korisnika
- (c) promijeniti podatke o receptu
- (d) izbrisati korisnički račun
- (e) izbrisati recept

4. Online-chat platforma (sudionik) može:

- (a) slati i primiti poruke između registriranih korisnika
- (b) pohranjivati poruke
- (c) pratiti i prikazivati dostupnost registriranih korisnika
- (d) uspostaviti videopoziv među registriranim korisnicima

### 3.1.1 Obrasci uporabe

#### UC1 - stvaranje korisničkog računa

- **Glavni sudionik:** neregistrirani korisnik
- **Cilj:** stvaranje novog korisničkog računa
- **Sudionici:** -
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Korisnik klikne na gumb za prijavu
  2. Korisnik upiše željeno korisničko ime i lozinku
  3. Korisnik klikne na gumb za registraciju
  4. Šalje se zahtjev sustavu
  5. Sustav zapisuje novog korisnika u bazu podataka
  6. Ako je registracija uspješna web stranica se preusmjeri na homepage, ali sa korisnikom prijavljenim u sustav
- **Opis mogućih odstupanja:**
  - 5.a Korisničko ime je zauzeto (već postoji)
    1. Prikazuje se poruka "Korisničko ime zauzeto!"
  - 5.b Lozinka je slaba (sadrži manje od 6 znakova i/ili sadrži samo slova)
    1. Prikazuje se poruka "Lozinka mora imati najmanje 6 znakova te barem jedan broj ili specijalan znak!"
  - 5.c Unesena adresa e-pošte nije važeća
    1. Prikazuje se poruka "Adresa e-pošte ne postoji ili nije važeća!"

#### UC2 - prijava u korisnički račun

- **Glavni sudionik:** registrirani korisnik
- **Cilj:** prijava korisnika u vlastiti račun
- **Sudionici:** -
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Korisnik klikne na gumb za prijavu
  2. Korisnik unosi korisničko ime i lozinku
  3. Korisnik klikne na gumb za prijavu
  4. Šalje se zahtjev sa korisničkim podacima sustavu
  5. Sustav provjerava točnost podataka u bazi podataka

6. U slučaju da je korisnik unio točne podatke, web stranica se preusmjeri na homepage sa korisnikom koji je sada prijavljen
- **Opis mogućih odstupanja:**
  - 5.a Korisničko ime ne postoji u bazi podataka
    1. Prikazuje se poruka "Korisničko ime ne postoji!"
  - 5.b Pogrešna lozinka
    1. Prikazuje se poruka "Kriva lozinka!"

### UC3 - promjena korisničkih podataka

- **Glavni sudionik: registrirani korisnik, administrator**
- **Cilj: promjena korisničkog imena i/ili lozinke**
- **Sudionici: -**
- **Preduvjet: korisnik je prijavljen u sustav**
- **Opis osnovnog tijeka:**
  1. Korisnik klikom na gumb pristupi postavkama
  2. Korisnik promijeni korisničko ime ili lozinku upisivanjem novog imena ili lozinke
  3. Korisnik klikne na gumb za spremanje promjena
  4. Sustavu se šalje zahtjev za promjenom podataka
  5. U bazu podataka se upisuju promjene
  6. Korisnik se automatski odjavi za stranice
- **Opis mogućih odstupanja:**
  - 5.a Novo korisničko ime je već zauzeto
    1. Prikazuje se poruka "Korisničko ime zauzeto!"
  - 5.b Lozinka je slaba (sadrži manje od 6 znakova i/ili sadrži samo slova)
    1. Prikazuje se poruka "Lozinka mora imati najmanje 6 znakova te barem jedan broj ili specijalan znak!"

### UC4 - brisanje korisničkog računa

- **Glavni sudionik: registrirani korisnik, administrator**
- **Cilj: brisanje korisničkog računa i svih podataka vezanih uz račun**
- **Sudionici: -**
- **Preduvjet: korisnik je prijavljen u sustav**
- **Opis osnovnog tijeka:**
  1. Korisnik klikne na gumb za pristup postavkama

2. Korisnik klikne na gumb za brisanje računa
  3. Pojavljuje se upozorenje i prozor sa zahtjevom za potvrdom
  4. Klikom na gumb "Da" šalje se zahtjev za brisanje računa sustavu
  5. Sustav briše račun iz baze podataka
  6. Korisnik se preusmjerava na homepage, te više nije prijavljen u sustav
- **Opis mogućih odstupanja:**
    - 4.a Korisnik klikne na gumb "Ne"
      1. Prozor sa zahtjevom se zatvara te korisnik ostaje u postavkama
    - 6.a Brisanje korisničkog računa je obavio administrator
      1. Nakon brisanja korisničkog računa administrator ostaje u postavkama

### **UC5 - pregled liste recepata**

- **Glavni sudionik:** neregistrirani korisnik, registrirani korisnik administrator
- **Cilj:** prikazati recepte na web stranici
- **Sudionici:** -
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Korisnik odabere filtere i način sortiranja recepata
  2. Korisnik u polje za pretraživanje upiše tekst
  3. Šalje se zahtjev za dohvaćanje recepata sustavu
  4. Sustav dohvaća recepte iz baze podataka
  5. Korisniku se prikaže lista recepata
- **Opis mogućih odstupanja:**
  - 1.a Odabran je pregled vlastitih recepata
    1. Korak 2 se preskače i zahtjev se šalje za receptima koje je korisnik objavio
  - 1.b Odabran je pregled omiljenih recepata
    1. Korak 2 se preskače i zahtjev se šalje za receptima koje je korisnik označio omiljenima
  - 2.a Nije upisan tekst po kojem se pretražuje
    1. Prikazuju se svi recepti

### **UC6 - stvaranje novog recepta**

- **Glavni sudionik:** registrirani korisnik
- **Cilj:** kreiranje novog recepta
- **Sudionici:** -
- **Preduvjet:** korisnik prijavljen u sustav
- **Opis osnovnog tijeka:**
  1. Korisnik klikne na gumb za početak stvaranja novog recepta
  2. Korisnik upisuje podatke o receptu kao što su postupak, slike, videi, kategorije itd.
  3. Klikom na gumb za objavu recepta šalje se zahtjev sustavu
  4. Sustav sprema recept u bazu podataka
  5. Recept postaje vidljiv drugim korisnicima
- **Opis mogućih odstupanja:**
  - 3.a Recept nema naslov
    1. Prikazuje se poruka "Potrebno je upisati ime recepta"
  - 3.b Nije upisana lista sastojaka
    1. Prikazuje se poruka "Potrebno je upisati listu sastojaka"
  - 3.c Ne postoji ni jedan korak pripreme
    1. Prikazuje se poruka "Potrebno je upisati barem jedan korak pripreme"
  - 3.d Nije uneseno vrijeme pripreme
    1. Prikazuje se poruka "Potrebno je upisati vrijeme pripreme"

#### UC7 - promjena podataka o receptu

- **Glavni sudionik:** registrirani korisnik, administrator
- **Cilj:** promjena bilo kojeg podatka o receptu kojeg je korisnik napravio
- **Sudionici:** -
- **Preduvjet:** korisnik je prijavljen u sustav te je on autor recepta kojeg želi promijeniti ili je administrator
- **Opis osnovnog tijeka:**
  1. Korisnik klikne na recept
  2. Korisnik klikne na gumb za uređivanje recepta
  3. Preusmjerava se na poveznicu za promjenu recepta
  4. Korisnik promijeni podatke o receptu i klikne na gumb za spremanje
  5. Sustavu se šalje zahtjev sa novim podacima o receptu
  6. U bazu podataka se spremaju promjene

7. Korisnika se preusmjerava na poveznicu na kojoj se može pregledati recept
- **Opis mogućih odstupanja:**
  - 4.a Korisnik klikne na gumb za brisanje recepta
    1. Pojavljuje se upozorenje i prozor sa zahtjevom za potvrdom
    2. Korak 5 se preskače
  - 5.a Problem u novim podacima (referencirati se na odstupanja u UC6)

#### UC8 - prijava na recepte korisnika

- **Glavni sudionik: registrirani korisnik**
- **Cilj: prijava korisnika na recepte autora kako bi primio obavijesti kada taj korisnik napravi novi recept**
- **Sudionici: -**
- **Preduvjet: korisnik je prijavljen u sustav**
- **Opis osnovnog tijeka:**
  1. Prijavljeni korisnik otvara recept nekog autora
  2. Klikom na gumb za praćenje se sustavu šalje zahtjev
  3. U bazu podataka se zapisuje da je korisnik prijavljen na recepte odabranog autora
  4. Recepti odabranog autora prikazuju se u listi obavijesti
- **Opis mogućih odstupanja:**
  - 3.a Korisnik je već prijavljen na recepte autora
    1. Klikom na gumb za praćenje autora se korisnik odjavljuje sa recepta tog autora

#### UC9 - pregled obavijesti

- **Glavni sudionik: registrirani korisnik**
- **Cilj: pregled recepta autora koje korisnik prati**
- **Sudionici: -**
- **Preduvjet: korisnik je prijavljen u sustav**
- **Opis osnovnog tijeka:**
  1. Korisnik klikne na gumb za pregled obavijesti
  2. Sustavu se šalje zahtjev
  3. Iz baze podataka čitaju se obavijesti korisnika
  4. Prikazuju se recepti svih autora koje korisnik prati
- **Opis mogućih odstupanja: -**



**UC10 - spremanje/bookmarkanje recepta**

- **Glavni sudionik:** registrirani korisnik
- **Cilj:** spremanje recepata kako bi se kasnije mogli pregledati omiljeni recepti
- **Sudionici:** -
- **Preduvjet:** korisnik prijavljen u sustav
- **Opis osnovnog tijeka:**
  1. korisnik klikne na gumb za spremanje recepta
  2. Šalje se zahtjev sustavu
  3. U bazu podataka se upisuje recept kao omiljen korisniku
- **Opis mogućih odstupanja:**
  - 3.a Recept je već spremljen kao omiljen
    1. Recept se miče is liste omiljenih recepata

**UC11 - pregled specifičnog recepta**

- **Glavni sudionik:** neregistrirani korisnik, registrirani korisnik
- **Cilj:** prikaz svih podataka vezanih uz recept
- **Sudionici:** -
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Korisnik klikne na recept
  2. Sustavu se šalje zahtjev za dohvaćanje podataka o receptu
  3. Iz baze podataka se čitaju podaci o receptu
  4. Korisniku se prikazuju dohvaćeni podaci
- **Opis mogućih odstupanja:** -

**UC12 - komentiranje na recept**

- **Glavni sudionik:** registrirani korisnik
- **Cilj:** ostavljanje komentara na nekom receptu
- **Sudionici:** -
- **Preduvjet:** korisnik je prijavljen u sustav te je otvorio stranicu recepta
- **Opis osnovnog tijeka:**
  1. Scrollanjem do kraja recepta se dolazi do dijela sa komentarima

2. Upisuje se tekst u polje za komentar
  3. Klikom na gumb za objavu komentara šalje se zahtjev sustavu
  4. U bazu podataka se upisuje komentar
  5. Web stranica se osvježava i komentar je sad vidljiv
- **Opis mogućih odstupanja:**
    - 3.a Tekst komentara je prazan
      1. Gumb za objavljivanje komentara je blokiran te nije moguće kliknuti na njega

### UC13 - razmjena poruka među registriranim korisnicima

- **Glavni sudionik: registrirani korisnici**
- **Cilj: razmjena poruka među korisnicima**
- **Sudionici: online-chat platforma**
- **Preduvjet: korisnik je prijavljen u sustav i omogućio je platformi pristup za komunikaciju**
- **Opis osnovnog tijeka:**
  1. Korisnik na javnom profilu drugog korisnika klikne da gumb za uspostavu kontakta
  2. Sustav preusmjerava zahtjev online-chat platformi koja obrađuje daljnje zahtjeve
  3. Online-chat platforma prikazuje njihove dosadašnje poruke (ako ih ima)
  4. Korisnik upisuje novu poruku i stisne gumb za slanje
  5. Online-chat platforma poruku sprema i šalje drugom korisniku
  6. Korisniku se poruka prikazuje među dosad izmijenjenim porukama kao poslana
  7. Korisnik izlazi iz prozora za razmjenu poruka i poziva
  8. Online-chat platforma šalje sustavu obavijest o završetku radnje
  9. Sustav preuzima daljnju obradu zahtjeva
- **Opis mogućih odstupanja:**
  - 1.a Korisnik kojeg se želi kontaktirati nije omogućio opciju komunikacije online-chat platformi
    1. Gumb za uspostavu kontakta je blokiran te nije moguće kliknuti na njega
    2. Svi daljnji koraci se ne izvode
  - 7.a Korisnik želi poslati još poruka

1. Ponavljaju se koraci 4, 5, i 6

#### UC14 - razmjena poziva među registriranim korisnicima

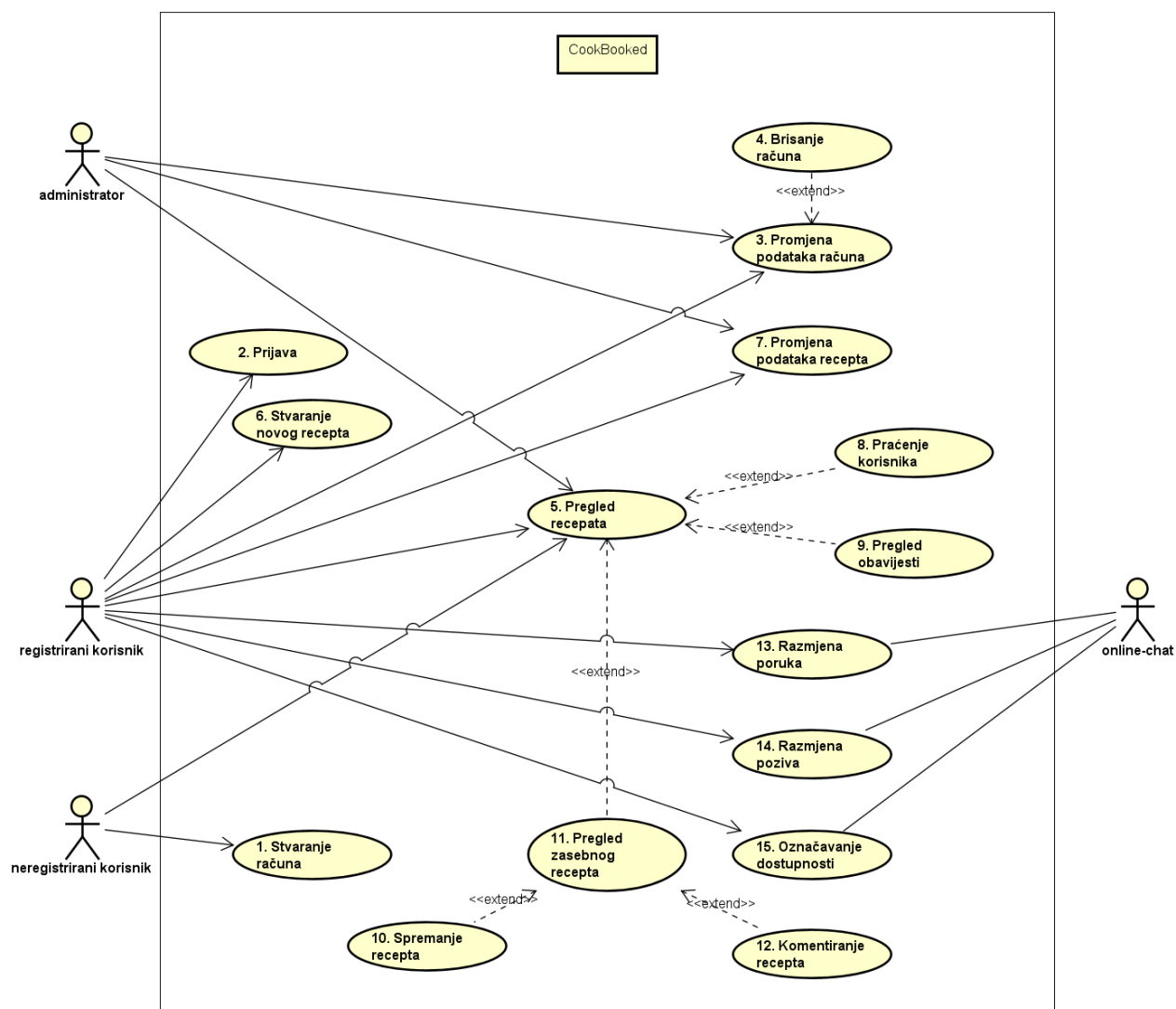
- **Glavni sudionik: registrirani korisnici**
- **Cilj: razmjena poziva među korisnicima**
- **Sudionici: online-chat platforma**
- **Preduvjet: korisnik je prijavljen u sustav i odobrio je platformi pristup mikrofonu i kameri**
- **Opis osnovnog tijeka:**
  1. Korisnik na javnom profilu drugog korisnika klikne da gumb za uspostavu kontakta
  2. Sustav preusmjerava zahtjev online-chat platformi koja obrađuje daljnje zahtjeve
  3. Online-chat platforma prikazuje stare poruke i gumb za opciju pozivanja
  4. Korisnik klikne na gumb za opciju pozivanja
  5. Online-chat platforma dohvaća podatke o dostupnosti korisnika i prikazuje ih korisniku koji inicira komunikaciju
  6. Korisnik klikne gumb za uspostavljanje poziva
  7. Online-chat platforma upućuje poziv korisniku
  8. Korisnik kojeg se zove prihvća poziv
  9. Klikom na gumb za prekid poziva šalje se zahtjev online-chat platformi koja gasi poziv
  10. Korisnik izlazi iz prozora za razmjenu poruka i poziva
  11. Online-chat platforma šalje sustavu obavijest o završetku radnje
  12. Sustav preuzima daljnju obradu zahtjeva
- **Opis mogućih odstupanja:**
  - 1.a Korisnik kojeg se želi kontaktirati nije omogućio opciju komunikacije online-chat platformi
    1. Gumb za uspostavu kontakta je blokiran te nije moguće kliknuti na njega
    2. Svi daljnji koraci se ne izvode
  - 6.a Korisnik kojeg se pokušava nazvati nije dostupan u tom vremenu
    1. Gumb za uspostavu poziva je blokiran te nije moguće kliknuti na njega

2. Prikazuje se kalendar u kojem su označeni termini u kojima je traženi korisnik dostupan
  3. Korisnik inicijator može označiti termin u kojem želi razgovarati
  4. Online-chat platforma zapisuje taj podatak i prenosi ga traženom korisniku
  5. Koraci 7, 8 i 9 se ne izvode
- 8.a Korisnik kojeg se zove odbija poziv
1. Poziv se prekida

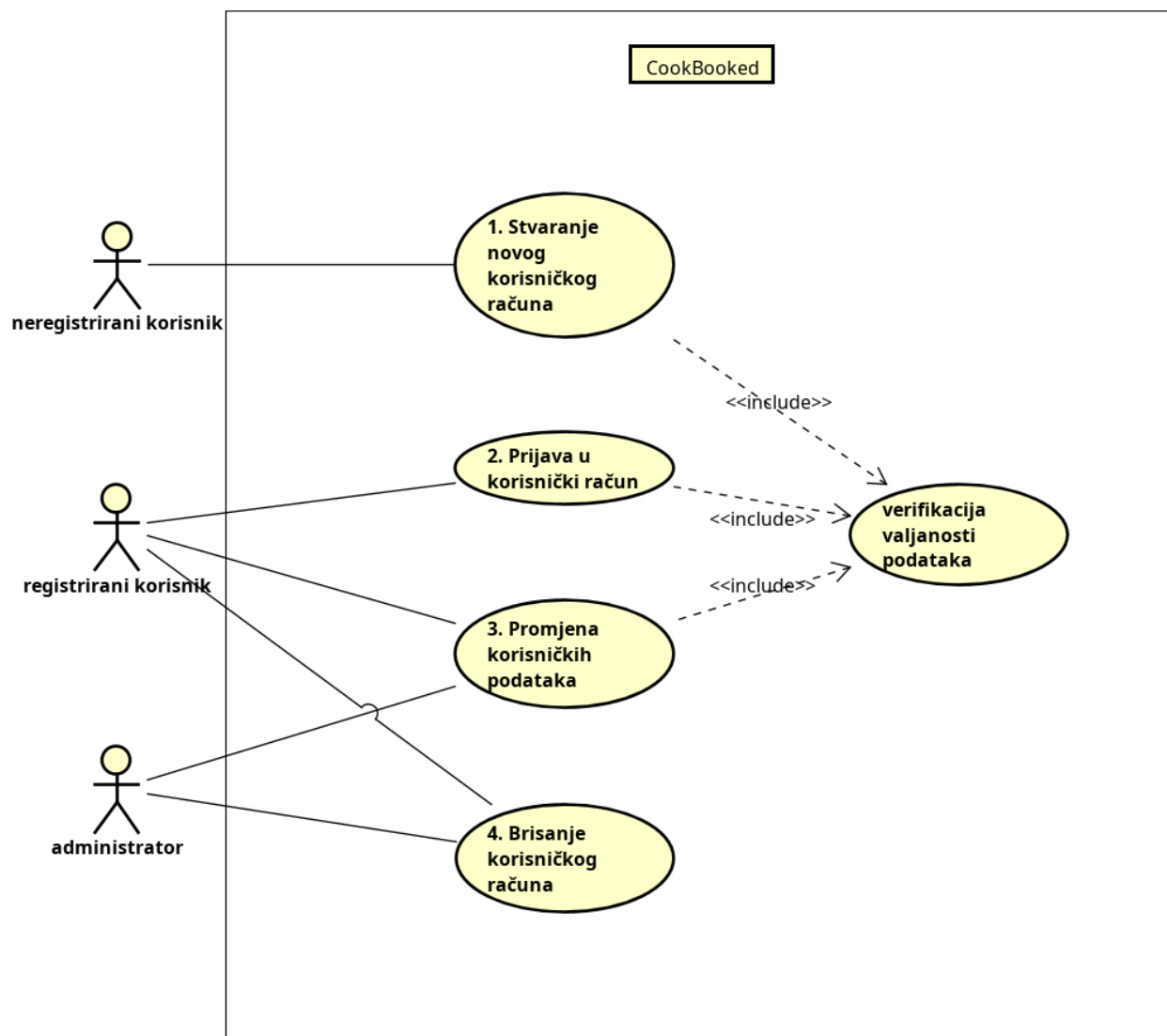
#### **UC15 - označavanje dostupnosti za pozive**

- **Glavni sudionik: registrirani korisnik**
- **Cilj: spremanje termina slobodnih za pozive i njihov prikaz drugim korisnicima**
- **Sudionici: online-chat platforma**
- **Preduvjet: korisnik je prijavljen u sustav i omogućio je platformi pristup za komunikaciju**
- **Opis osnovnog tijeka:**
  1. Korisnik klikne na gumb za postavke
  2. Korisnik klikne na gumb za uređivanje kalendara
  3. Šalje se zahtjev sustavu za dohvaćanje dosad unesenih podataka u kalendaru
  4. Šalje se zahtjev online-chat platformi za podatke o zatraženim pozivima
  5. Korisniku se prikazuje kalendar s njegovim označenim terminima i zahtjevima za poziv
  6. Korisnik klikom na određeni dan upisuje vrijeme kad je slobodan i stisne gumb za spremanje
  7. Sustav pohranjuje promjene kalendara u bazu podataka
  8. Sustav šalje platformi novouneseni podatak
  9. Korisnik izlazi iz postavki
- **Opis mogućih odstupanja:**
  - 6.a Korisnik klikne na termin zatraženog poziva
    1. Pojavljuje se prozor za prihvaćanje odnosno odbijanje zahtjeva
    2. Korisnik bira klikom na gumb za prihvaćanje odnosno odbijanje

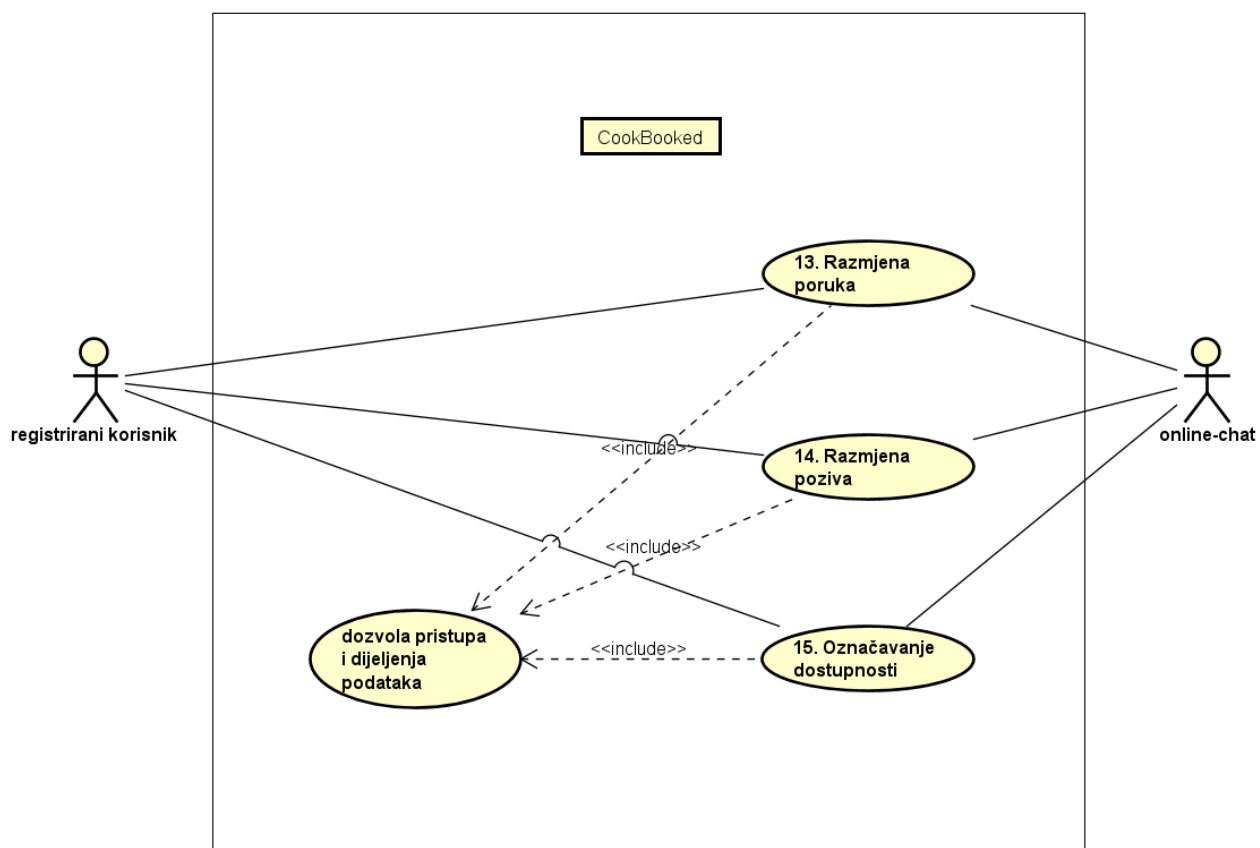
## Dijagrami obrazaca uporabe



Slika 3.1: Cijeli UML dijagram



Slika 3.2: UML dijagram za UC1-UC4



Slika 3.3: UML dijagram za UC13-UC15

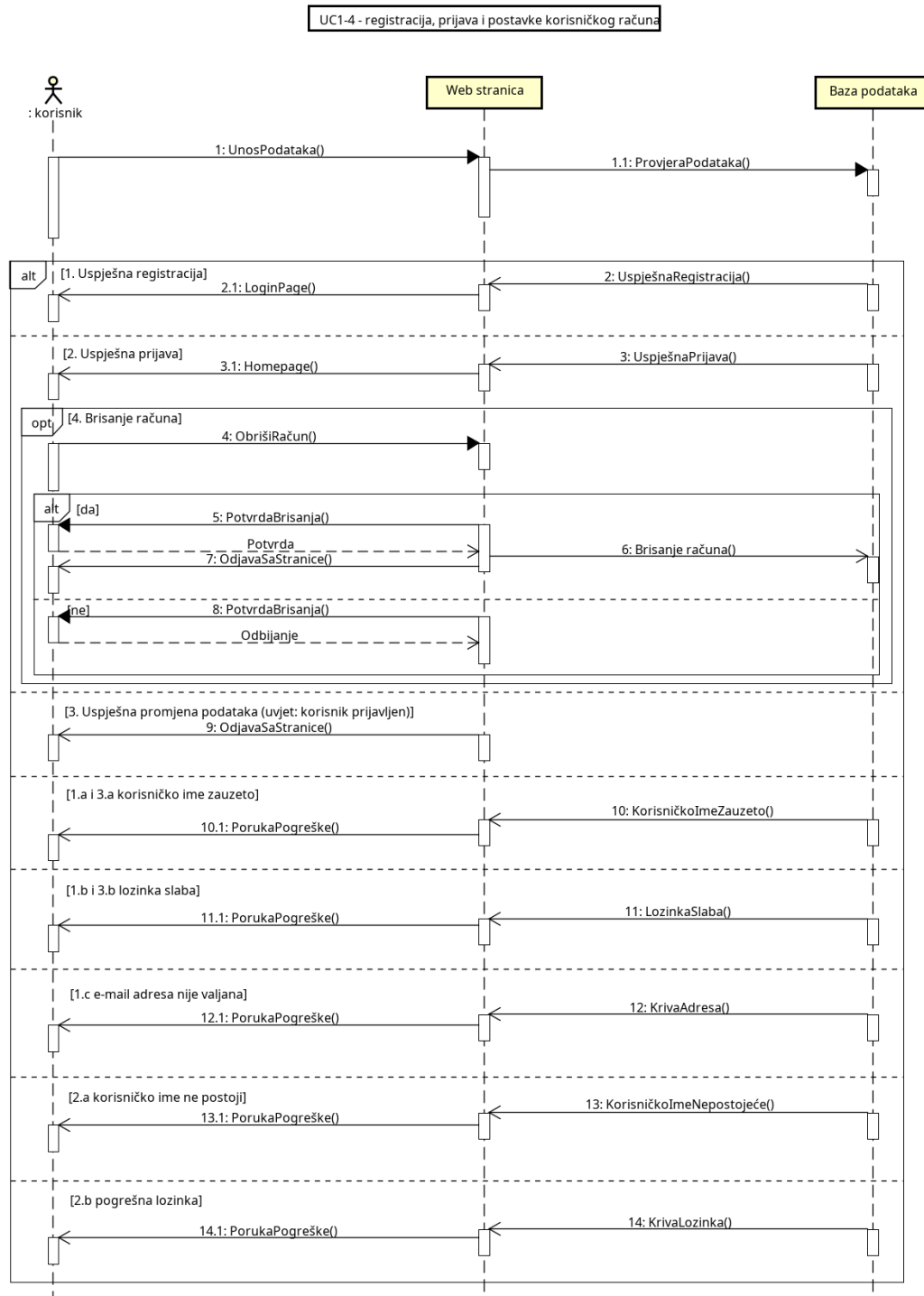
### 3.1.2 Sekvencijski dijagrami

#### Registracija, prijava i postavke korisničkog računa

Korisnik unosi podatke u svrhu registracije, prijave u sustav ili promjene vlastitih podataka profila (za što treba već biti prijavljen u sustav). Ako je uspješno unio podatke, korisnik se vraća na početnu stranicu odjavljen (osim u slučaju da se htio prijaviti). U suprotnome, korisniku se prikazuje jedna od grešaka (korisničko ime zauzeto, lozinka slaba, e-mail adresa nije valjana, korisničko ime ne postoji ili pogrešna lozinka).

Nakon što se korisnik uspješno prijavio u sustav, može odabrati opciju brisanja računa. Prilikom toga sustav mu šalje zahtjev za potvrdom gdje može birati DA ili NE. U slučaju da korisnik izabere opciju DA, vraća se na početnu stranicu odjavljen, a njegov se profil briše iz sustava.

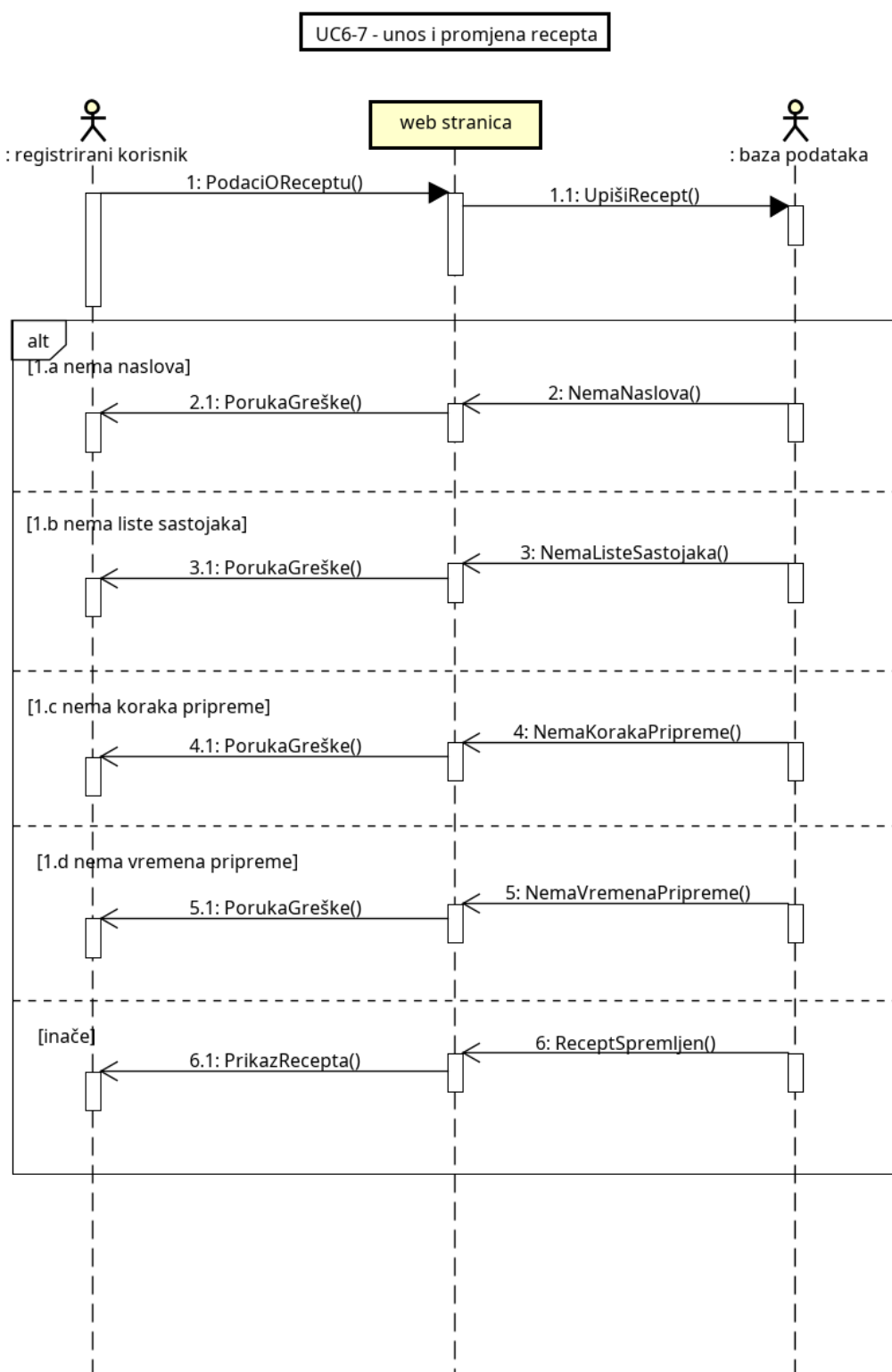




Slika 3.4: Sekvencijski dijagram za UC1-4

**Unos recepta i promjena podataka recepta**

Registrirani korisnik može unijeti novi recept ili urediti podatke svog već objavljenog recepta. Prilikom spremanja novih podataka sustav može odgovoriti porukama greške (nema naslova, nema liste sastojaka, nema koraka pripreme, nema vremena pripreme) ukoliko neki od uvijeta nije ispunjen. U suprotnom, recept s novim podacima se sprema u sustav i prikazuje korisniku.

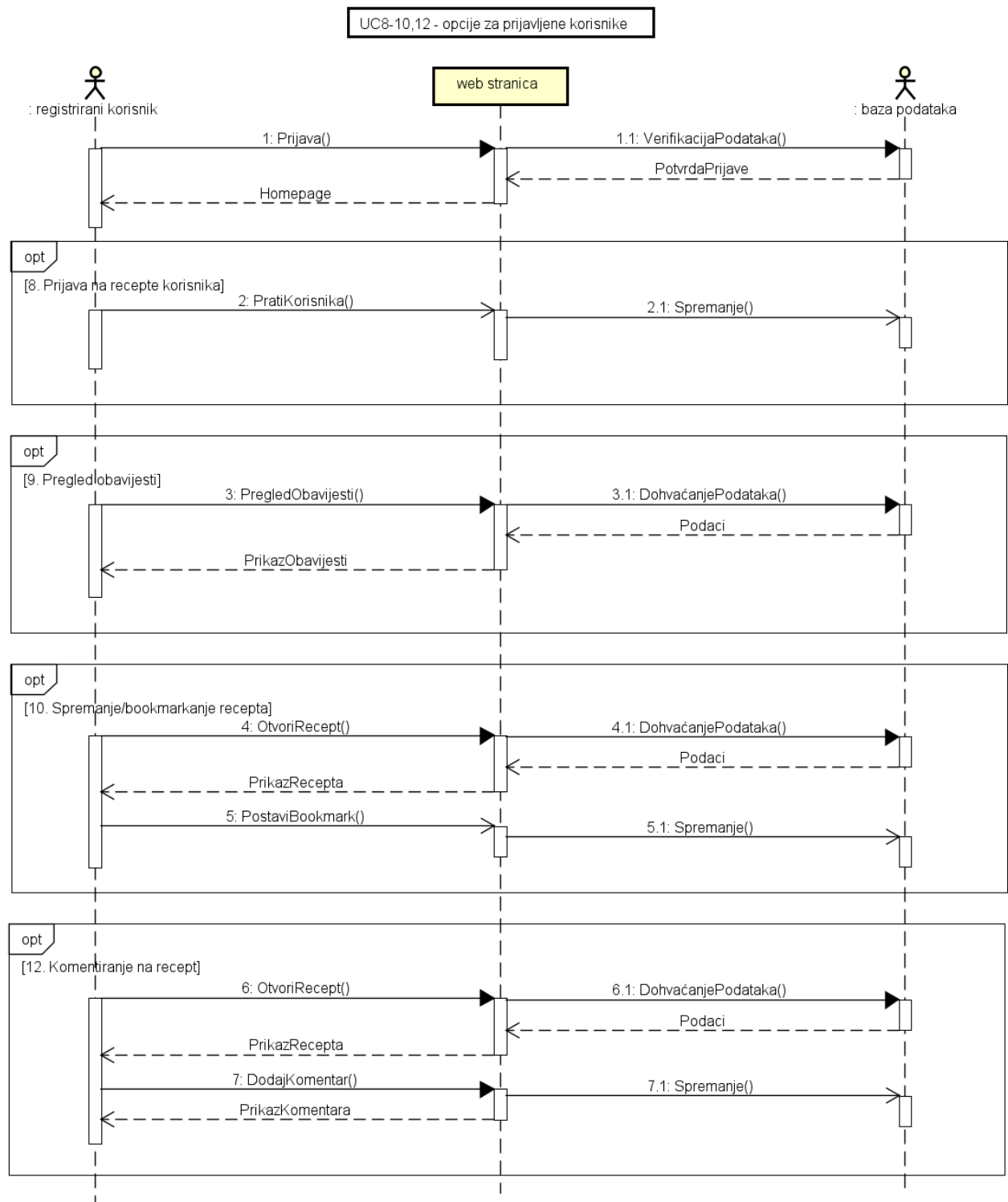


Slika 3.5: Sekvencijski dijagram za UC6-7

### **Opcije prilikom pregleda recepata**

Korisnik, kada je prijavljen u sustav i pregledava listu recepata, može zapratiti nekog od korisnika čiji recept mu je na listi prikazan, može pregledati obavijesti o aktivnostima autora koje već prati ili može pregledati odabrani recept s liste.

U slučaju da korisnik odabere pregled zasebnog recepta, prikazuje mu se odabrani recept. Prilikom pregleda nude mu se opcije da recept označi kako bi ga spremio za kasnije ili da ostavi komentar na recept. Ukoliko ostavi komentar, on se sprema u sustav i odmah prikazuje svim korisnicima uz odabrani recept.



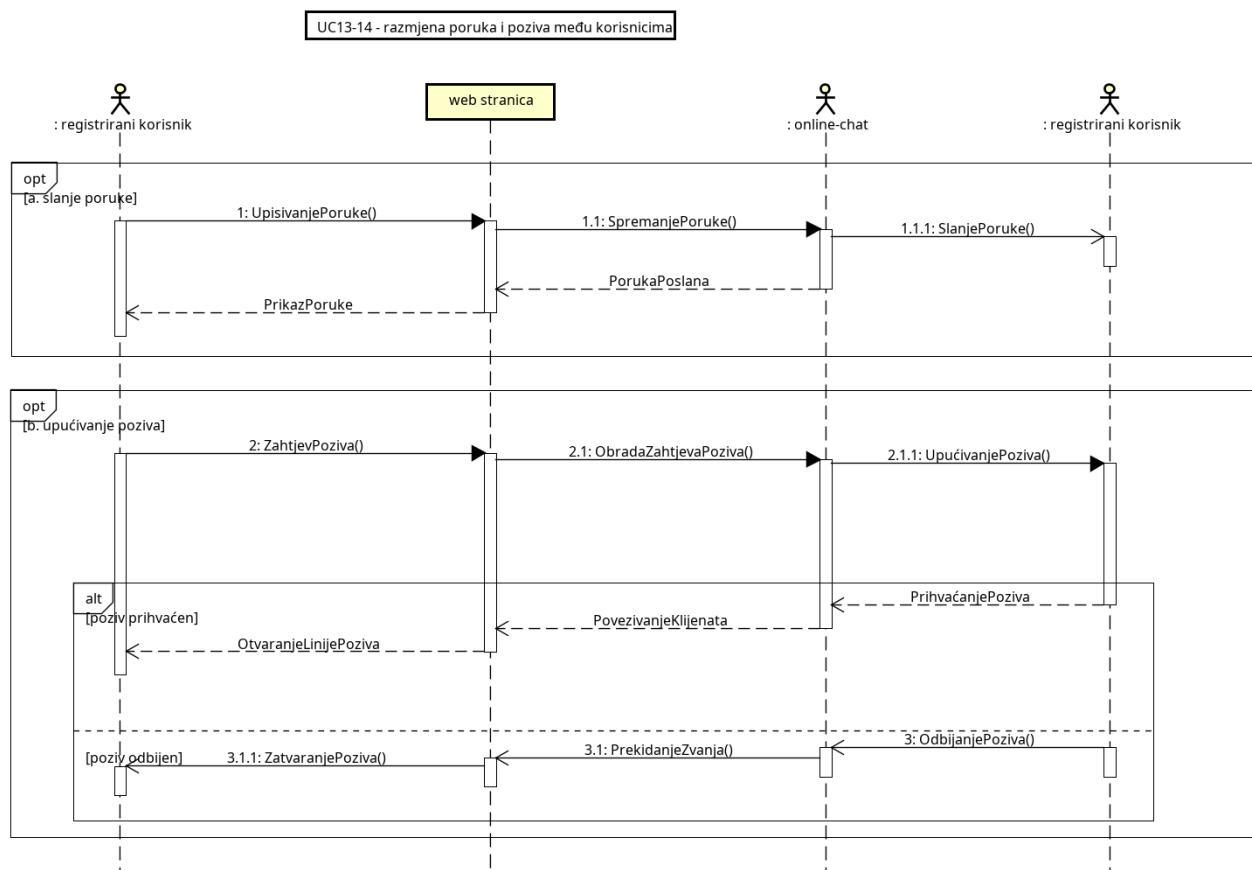
Slika 3.6: Sekvencijski dijagram za UC8-10;12

## Razmjena poruka i poziva među korisnicima

Registrirani korisnik ima opciju slanja poruka i upućivanja poziva drugim registriranim korisnicima. Kontakt se izvodi preko vanjske online-chat platforme koja djeluje kao posrednik.

Prilikom slanja poruke vanjska platforma poruku sprema i prosljeđuje drugom korisniku. Ona se zatim prikazuje kod oba korisnika kao dio razgovora.

Prilikom upućivanja poziva vanjska platforma šalje zahtjev za pozivom drugom korisniku koji ima opciju prihvatiti ili odbiti poziv. S obzirom na odabir pozvanog korisnika, platforma otvara liniju između korisnika ili pozivatelju javlja da je poziv odbijen.



Slika 3.7: Sekvencijski dijagram za UC13-14

## 3.2 Ostali zahtjevi

- CookBooked će podržavati višejezičnost s odabirom između najmanje 5 različitih jezika. Korisnici će moći prilikom prijave na aplikaciju birati jezik, a bit će moguće zamijeniti ga u bilo kojem željenom trenutku.
- Maksimalno vrijeme odziva sustava na zahtjeve korisnika neće prelaziti 5 sekundi. To će korisnicima pružiti ugodno i brzo kretanje kroz funkcionalnosti aplikacije i interakciju sa sustavom s kratkim vremenom učitavanja.
- Aplikacija će biti prilagođena za pristup putem web preglednika kao i mobilnih uređaja. Uz to ostvarit ćemo podršku za najmanje 3 glavne mobilne platforme (npr. iOS, Android).
- Platforma će biti zaštićena SSL i TLS protokolima osiguravajući kriptiranu komunikaciju između korisnika i poslužitelja. Lozinke će se čuvati u kriptiranom obliku, a sustav će implementirati mehanizme zaštite od neovlaštenog pristupa.
- Sustav će podržavati formate visoke kvalitete za slike (npr. JPEG, PNG) i videozapise (npr. MP4) kako bi se osigurala visoka razina vizualnog doživljaja korisnika. Korisničko sučelje biti će intuitivno i jednostavno za razumjeti korisnicima s različitim razinama educiranosti.
- Svi objavljeni recepti bit će moderirani kako bi se osigurala točnost, autentičnost i odgovarajući sadržaj prema pravilima platforme prije nego što postanu javno dostupni. Sadržaj će se arhivirati u bazu kako ne bi došlo do gubitaka podataka.

## 4. Arhitektura i dizajn sustava

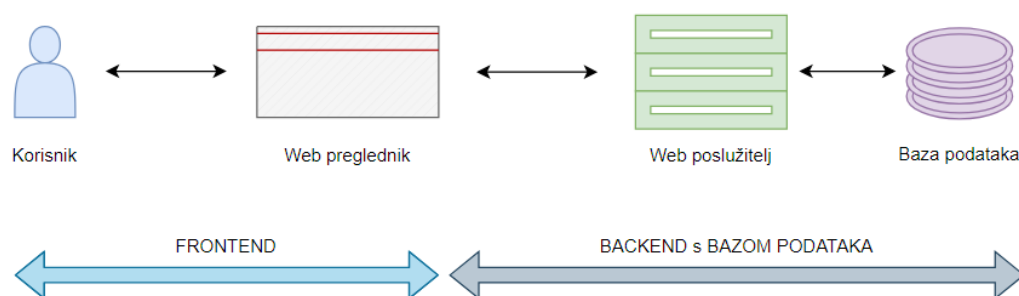
Arhitekturu sustava dijelimo na tri različita dijela: **frontend**, **backend** i **bazu podataka**. Jedino koordinirana implementacija ovih podsustava zajedno dovodi do kvalitetnog finalnog proizvoda u obliku web platforme koju ostvarujemo.

**Frontend** predstavlja sučelje između korisnika i same platforme. Konkretno, on pruža korisniku pregled sadržaja web stranice i omogućava mu komunikaciju s aplikacijom putem *HTTP (HyperText Transfer Protocol)* zahtjeva. Frontend je implementiran uz pomoć *JavaScripta* (radnog okvira *React.js*), *HTML-a* i *CSS-a*.

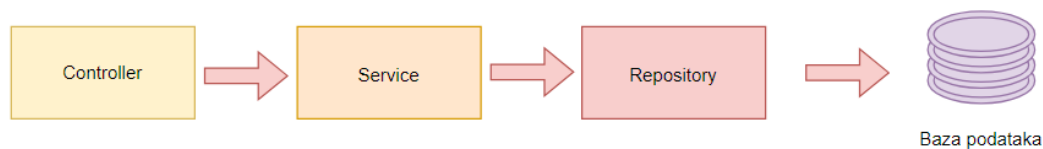
**Backend** je taj koji te *HTTP* zahtjeve obrađuje i upravlja podacima u bazi podataka. Neke od funkcionalnosti koje omogućuje backend su primjerice registracija novog korisnika ili brisanje korisnika od strane sistemskog administratora. **Controller - Service - Repository** troslojna arhitektura temelj je backend implementacije aplikacije. *Controller* sloj povezuje frontend i backend, pozivajući *Service* sloj da obradi zahtjeve korisnika. Ako je za obradu tih zahtjeva potreban pristup bazi podataka, *Service* sloj poziva funkcije *Repository* sloja koji to omogućuje. Za ostvaranje backenda korišten je programski jezik *Java* (radni okvir *Spring Boot*), a odabrana je razvojna okolina *Intellij IDEA*. Ispitivanje backend funkcionalnosti omogućeno je korištenjem *Postman* platforme.

**Baza podataka** spremište je entiteta koji su prikazani u obliku tablica s atributima. Iz nje se često čitaju, u nju dodaju i po potrebi u njoj ažuriraju podatci. Kritičan je dio arhitekture sustava. Baza podataka izgrađena je korištenjem *PostgreSQL* sustava za upravljanje bazom podataka.





Slika 4.1: Skica arhitekture sustava

Slika 4.2: Skica **Controller - Service - Repository** troslojne arhitekture backenda

## 4.1 Baza podataka

Za našu platformu koristit ćemo relacijsku bazu podataka. Objekt takve baze je relacija, odnosno tablica koja je definirana svojim imenom i skupom atributa. Atributi unutar jednog entiteta mogu poprimiti funkciju primarnog ili stranog ključa. Baza podataka u našem se slučaju sastoji se od 5 entiteta - **profile**, **recipe**, **follow**, **likes** i **comment**.

### 4.1.1 Opis tablica

#### *profile*

Ovaj entitet sadrži osnovne informacije o korisničkom profilu i sastoji se od sljedećih atributa: *userID* (autogenerirani identifikator korisnika), *email* (e-mail adresa), *username* (korisničko ime), *password* (lozinka), *name* (ime), *surname* (prezime) i *age* (dob). Atribut *userID* primarni je ključ entiteta **profile**.

profile		
userID	VARCHAR	jedinstveni identifikator korisnika
email	VARCHAR	jedinstvena e-mail adresa korisnika
name	VARCHAR	ime korisnika (opcionalno)
surname	VARCHAR	prezime korisnika (opcionalno)
age	INT	dob korisnika (opcionalno)
password	VARCHAR	lozinka korisnika
username	VARCHAR	jedinstveno korisničko ime

#### *recipe*

Ovaj entitet opisuje recept postavljen na platformu od strane registriranog korisnika. Primarni je ključ entiteta *recipeID* (autogenerirani identifikator recepta), a strani ključ *userID* koji povezuje entitete **recipe** i **profile**.

Ostali atributi su: *ingredients* (sastojci jela koje recept opisuje), *name* (ime recepta), *instructions* (niz uputa pripreme jela opisanog receptom), *origin* (mjesto podrijetla jela), *category* (kategorija jela), *specialTag* (dodatne oznake), *imageUrl* (URL slike koju korisnik prilaže uz postavljeni recept), *videoURL* (URL videa koji korisnik prilaže uz postavljeni recept) te *preparationTime* (vrijeme potrebno za spremanje

receptnog jela).

recipe		
recipeID	VARCHAR	jedinstveni identifikator recepta
ingredients	VARCHAR	popis sastojaka receptnog jela
name	VARCHAR	ime recepta
instructions	VARCHAR	niz uputa za pripremu jela prema receptu
origin	VARCHAR	mjesto podrijetla receptnog jela (opcionalno)
category	VARCHAR	vrsta (kategorija) receptnog jela
specialTag	VARCHAR	specijalne dodatne oznake receptnog jela (opcionalno)
videoURL	VARCHAR	URL videa priloženog uz recept (opcionalno)
imageURL	VARCHAR	URL slike priložene uz recept
preparationTime	INT	vrijeme kuhanja jela u minutama
userID	VARCHAR	jedinstveni identifikator korisnika

### *follow*

Ovaj entitet predstavlja vezu koja se stvara između 2 korisnika kada dođe do praćenja i sastoji se od sljedećih atributa: *userID* i *followerId* koji u paru čine primarni ključ relacije, a oba su također i strani ključevi uzeti iz relacije **profile**.

follow		
userID	VARCHAR	jedinstveni identifikator korisnika
followerId	VARCHAR	jedinstveni identifikator korisnika

### *likes*

Ovaj entitet predstavlja vezu koja se stvara prilikom označavanja recepata i sastoji se od sljedećih atributa: *userID* i *recipeId* koji u paru čine primarni ključ relacije, a oba su također i strani ključevi uzeti iz relacija **profile** i **recipe**

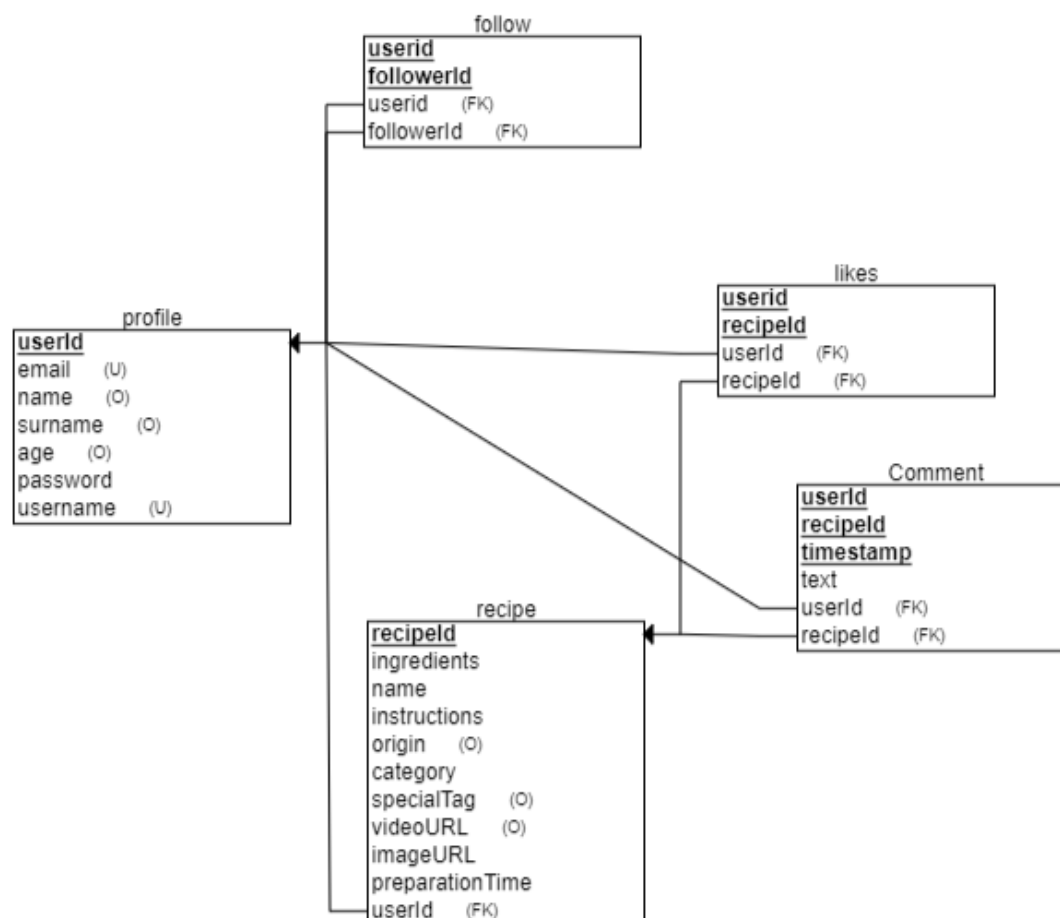
likes		
userID	VARCHAR	jedinstveni identifikator korisnika
recipeId	VARCHAR	jedinstveni identifikator recepta

**comment**

Ovaj entitet sadrži osnovne informacije o komentarima i sastoji se od sljedećih atributa: *userID*, *recipeId*, *timestamp* (vremenska oznaka komentara) i *text* (tekst komentara). Atributi *userId* i *recipeId* su strani ključevi relacija **profile** i **recipe**, a zajedno s atributom *timestamp* čine primarni ključ ove relacije.

comment		
userID	VARCHAR	jedinstveni identifikator korisnika
recipeId	VARCHAR	jedinstveni identifikator recepta
timestamp	DATE	vremenska oznaka komentara
text	CHAR(2048)	tekst ostavljenog komentara

### 4.1.2 Dijagram baze podataka

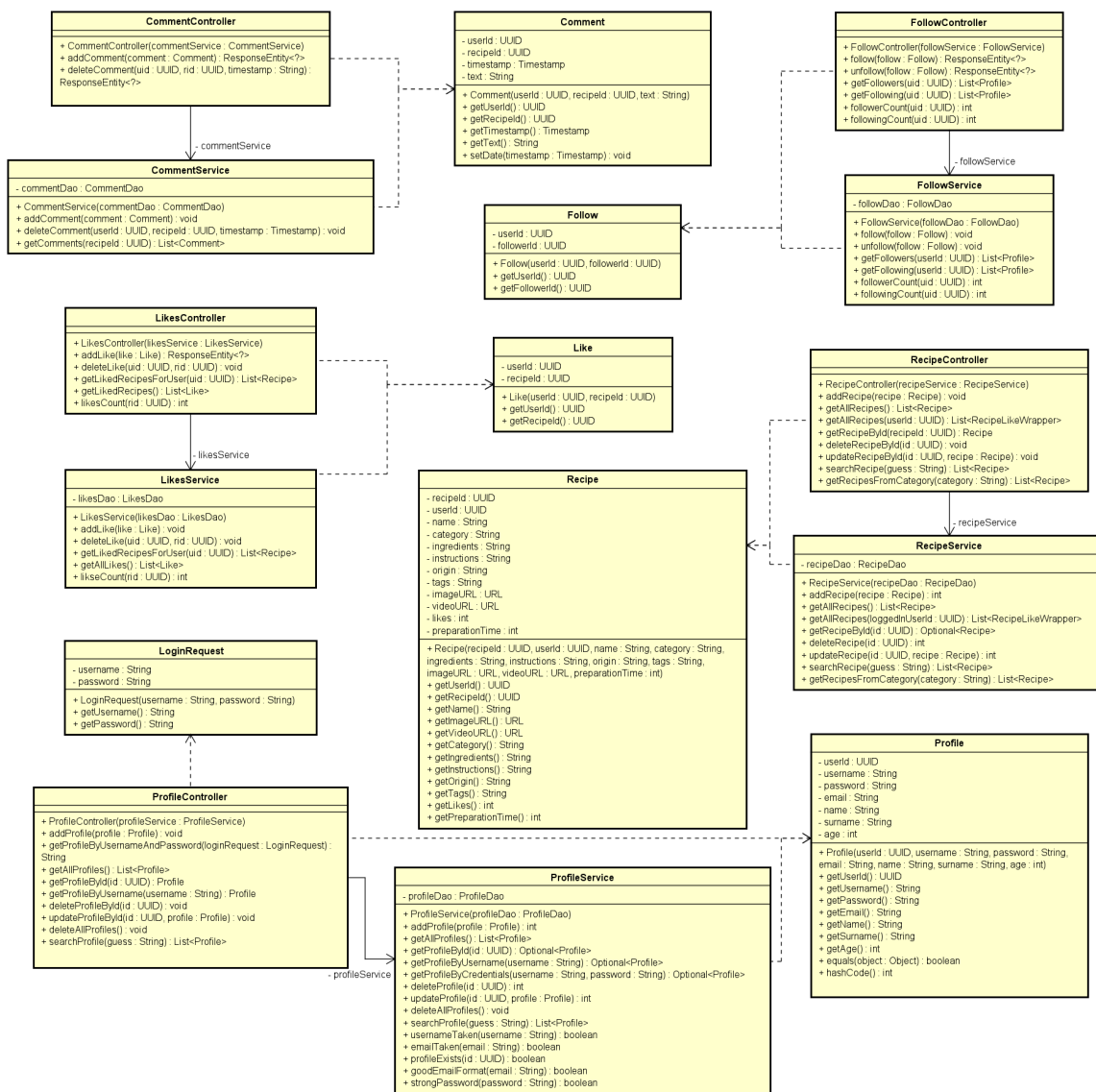


Slika 4.3: Dijagram baze podataka

## 4.2 Dijagram razreda

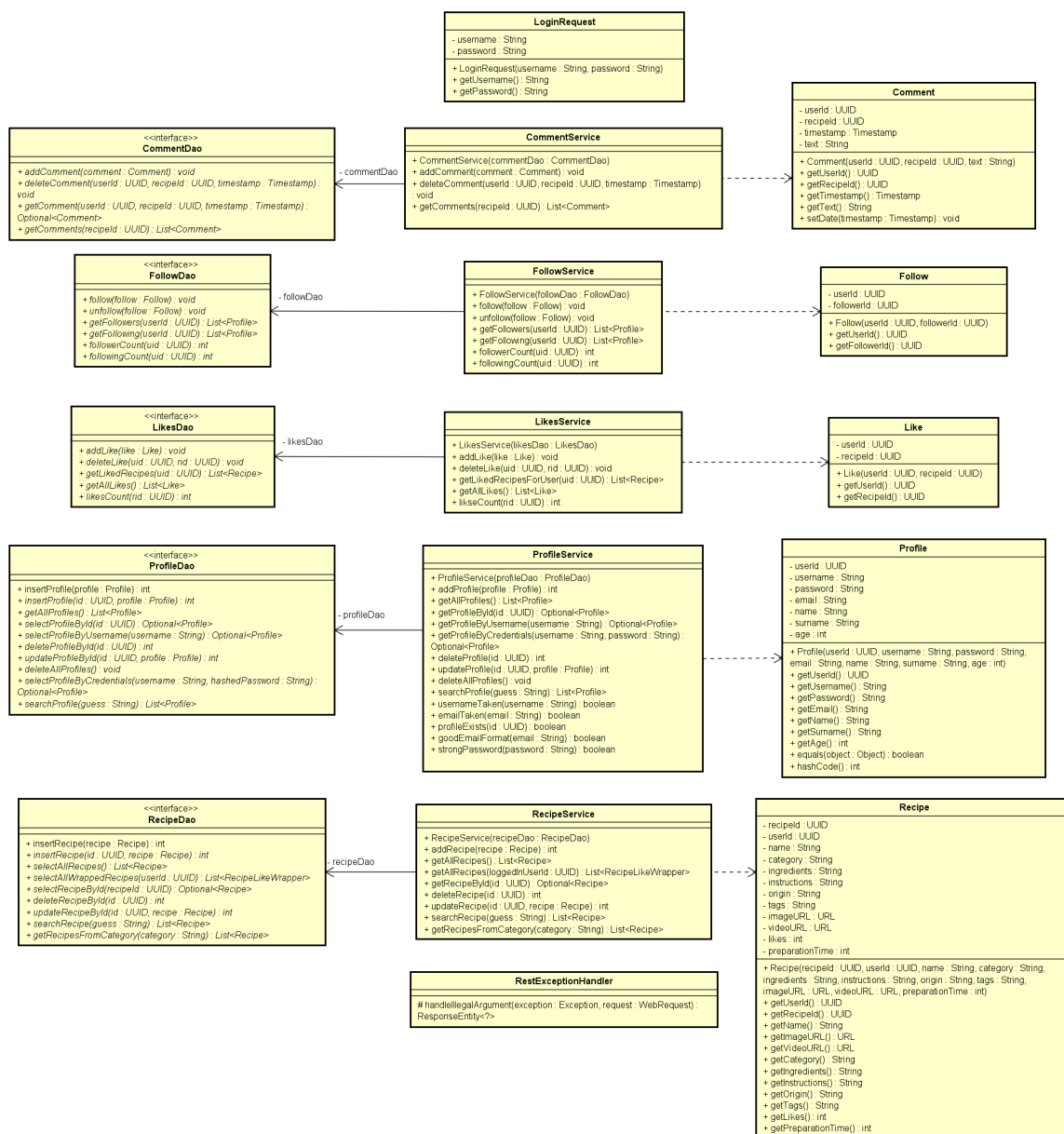
U nastavku su prikazani i opisani razredni dijagrami trenutne implementacije prikazanim 2. predaje dokumentacije. Podijeljeni su u 3 dijagrama radi preglednosti, a svaki opisuje posebni sloj u arhitekturi software-a.

**Controller** je sloj softvera koji upravlja dolaznim zahtjevima, obrađuje ih i koordinira tijekom izvođenja programa. Njegove klase su kontroleri (*CommentController*, *FollowController*, *LikeController*, *ProfileController* i *RecipeController*) koji koordiniraju različite zahtjeve i usmjeravaju ih prema odabranim service klasama (*CommentService*, *FollowService*, *LikeService*, *ProfileService* i *RecipeService*). *Controller* klase su sa service klasama vezane asocijacijom, a obje ovise o osnovnim model klasama (*Comment*, *Follow*, *Like*, *Profile* i *Recipe*).



Slika 4.4: Dijagram razreda - Controller

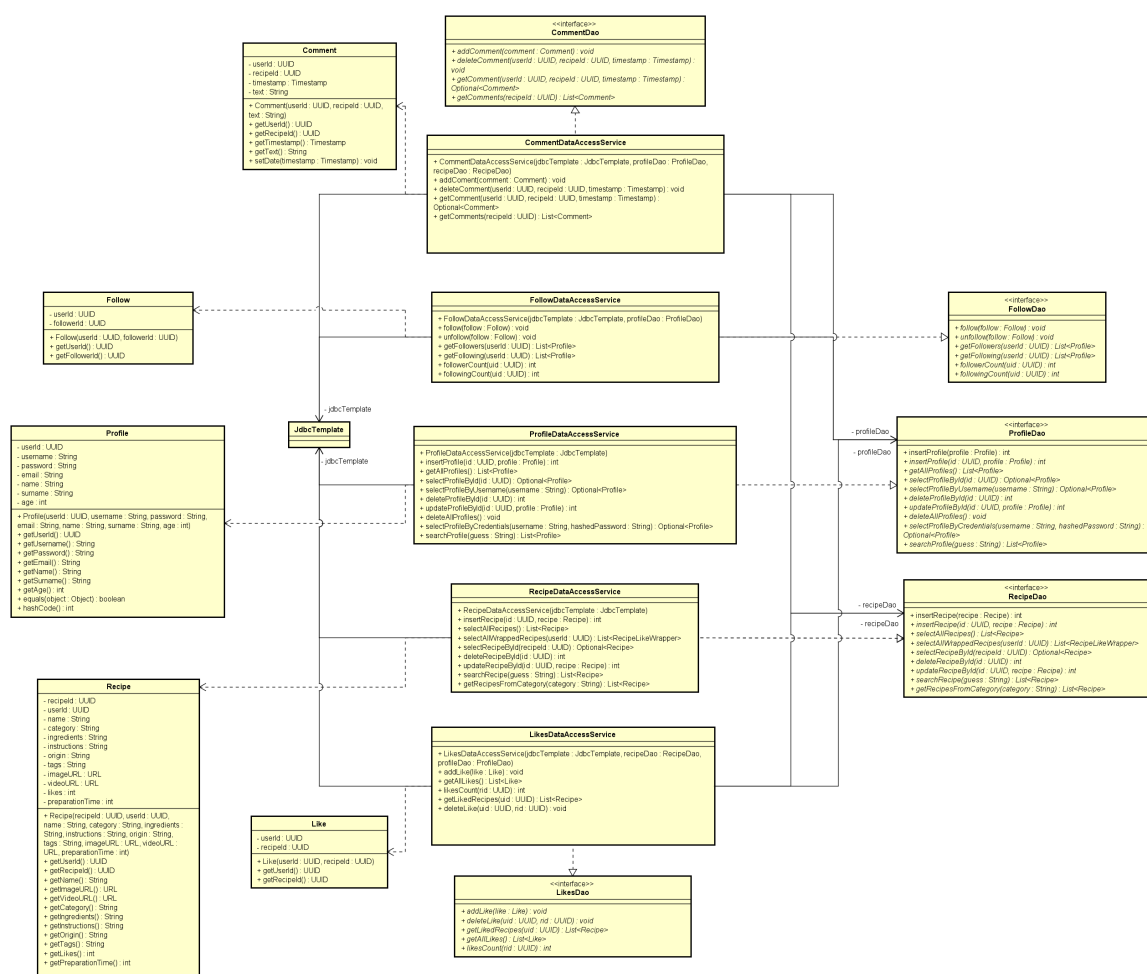
**Service** je sloj softvera koji pruža sučelje između kontrolera i sloja pristupa podacima. Service klase (*CommentService*, *FollowService*, *LikeService*, *ProfileService* i *RecipeService*) asociraju se sa sučeljem (*CommentDao*, *FollowDao*, *LikesDao*, *ProfileDao* i *RecipeDao*) preko kojeg pristupaju traženim podacima i upravljaju njima pa stoga ovise o model razredima (*Comment*, *Follow*, *Like*, *Profile* i *Recipe*). U *service* sloju nalaze se još klase *LoginRequest* i *RestExceptionHandler* koje ne pristupaju podacima direktno nego su zadužene za druge funkcionalnosti Service sloja.



Slika 4.5: Dijagram razreda - Service



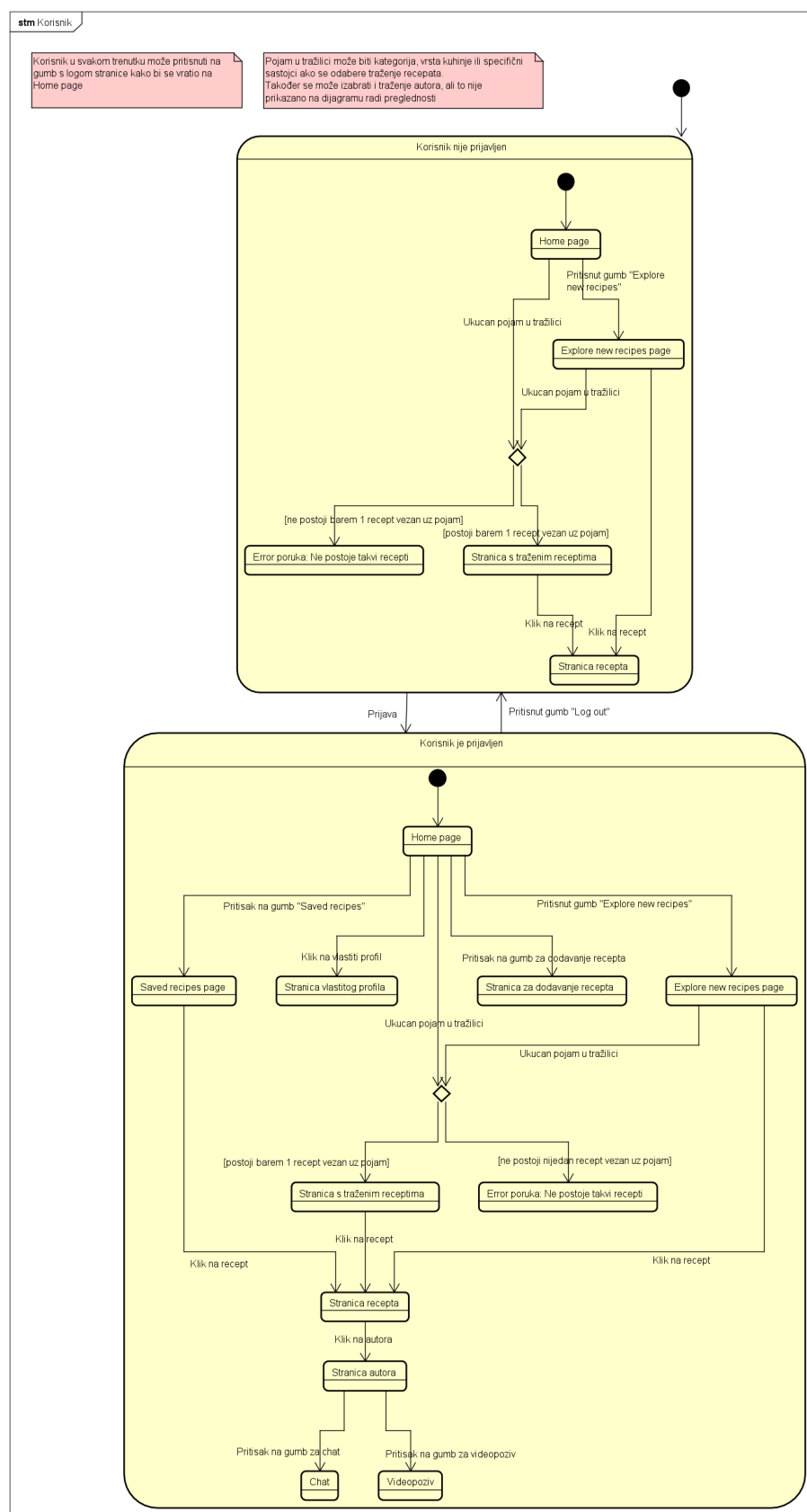
DAO (Data Access Object) predstavlja objekt koji pruža sučelje za izvođenje osnovnih operacija pristupa podacima, dodavanja, čitanja, ažuriranja i brisanja. Unutar projekta koristili smo 5 DAO sučelja (*CommentDao*, *FollowDao*, *LikesDao*, *ProfileDao* i *RecipeDao*) te njihove implementacije u obliku service klasa (*CommentDataAccessService*, *FollowDataAccessService*, *LikesDataAccessService*, *ProfileDataAccessService* i *RecipeDataAccessService*) s omogućenim pristupom podacima. Uz to, implementirane klase asocijacijom su vezane na klasu *JdbcTemplate* radi formatiranja u svrhu jednostavnije implementacije.



Slika 4.6: Dijagram razreda - Dao

## 4.3 Dijagram stanja

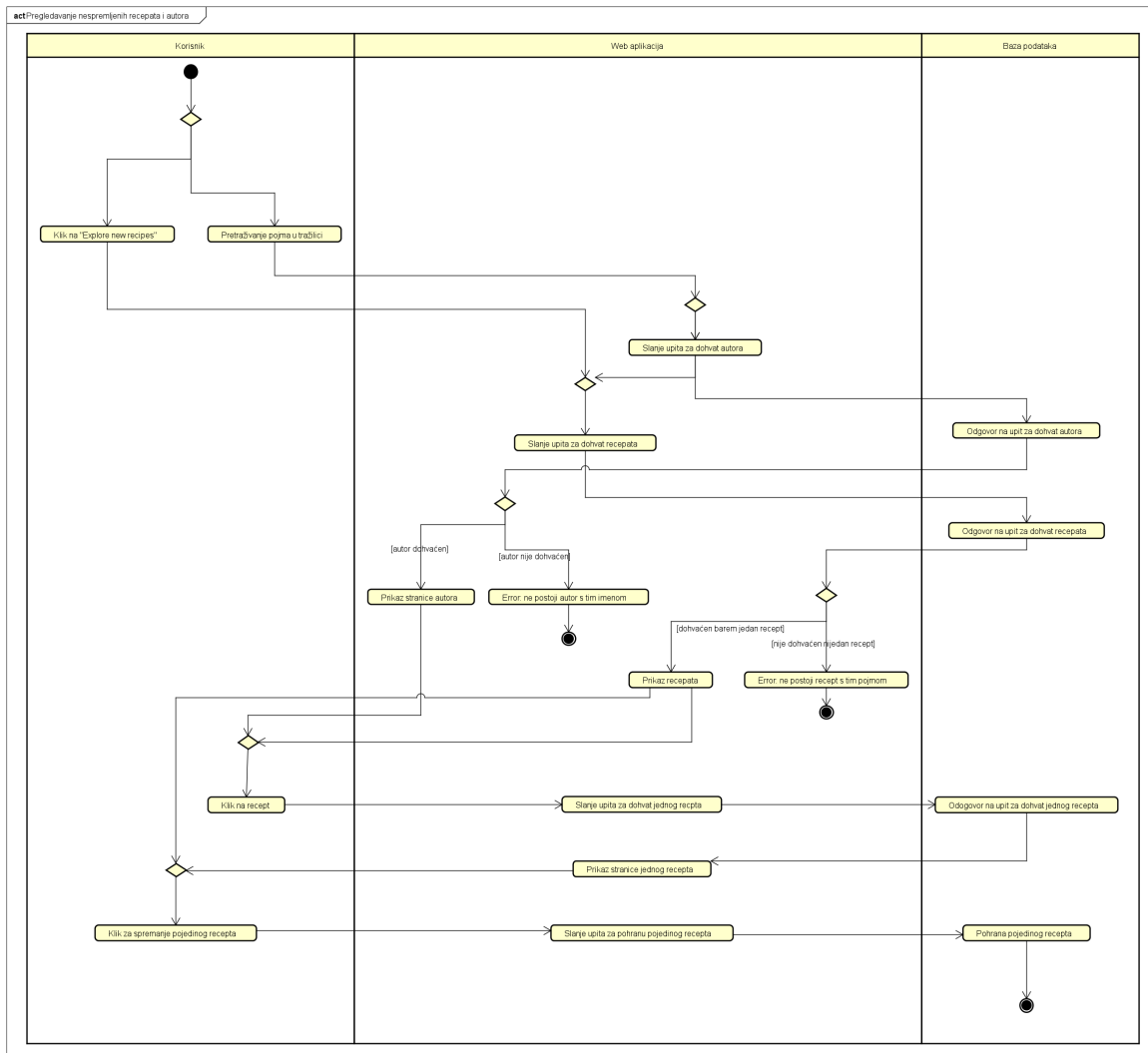
Dijagram stanja prikazuje stanja prijavljenog i neprijavljenog korisnika. Korisnik inicijalno nije prijavljen. Neprijavljeni korisnik iz home pagea ima opciju stisnuti gumb za pretraživanje novih recepata ili pregledavati recepte (ako ih ima) temeljem kategorija (npr. predjela, deserti), vrsta kuhinje (talijanska, kineska) ili specifičnih sastojaka uz pomoć tražilice. Neprijavljeni korisnik također može potražiti autora recepta temeljem korisničkog imena putem tražilice. Za pristup svim ostalim mogućnostima platforme, korisnik se mora registrirati ili prijaviti. Prijavljeni korisnik može sve što i neprijavljeni korisnik, a može koristiti i još neke stvari. Može stisnuti za gumb za dodavanje vlastitog recepta ili na gumb za pregled svih recepata koje je spremio. Također je moguće i pregledati te uređivati vlastiti profil na stranici profila. Još jedna pogodnost koju prijavljeni korisnici imaju je ta da mogu komunicirati s autorima putem chata ili videopoziva. Naravno, korisnik u bilo kojem trenutku može stisnuti na gumb s logoim stranice kako bi se vratio na Home page.



Slika 4.7: Dijagram stanja za korisnika

## 4.4 Dijagram aktivnosti

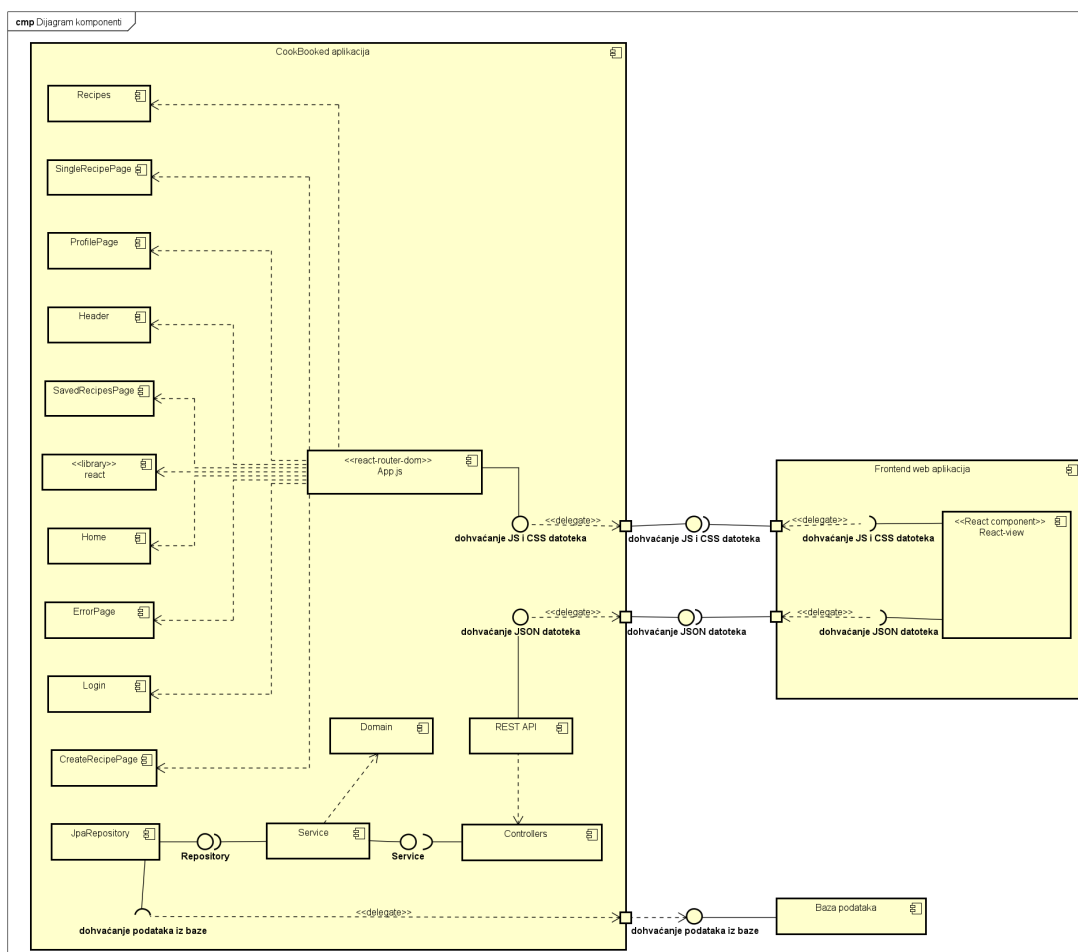
Dijagram aktivnosti prikazuje aktivnosti pregledavanja nespremljenih recepata i autora. Korisnik može naći nove recepte uz pomoć gumba "Explore new recipes" ili uz pomoć pretraživanja pojmova u tražilici. Kada korisnik odabere jednu od ovih dviju opcija, web aplikacija šalje upit za dohvat tih recepata bazi podataka koja daje odgovor o tome postoje li podaci i ako da, onda ih vraća. Nakon toga aplikacija prikazuje recepte na koje korisnik može kliknuti kako bi ih pohranio ili kako bi ih detaljnije pregledao. Naravno, u oba slučaja aplikacija mora slati upit bazi podataka kako bi dohvatila / pohranila podatke. Osim toga, korisnik u tražilici može pretraživati i autore pa tada baza dohvaća podatke iz tablice profiles. Nakon dohvaćanja autora, aplikacija prikazuje njegovu stranicu koja sadrži njegove podatke i recepte koje korisnik opet može pregledati i pohraniti.



Slika 4.8: Dijagram aktivnosti za pregledavanje nespremljenih recepata i autora

## 4.5 Dijagram komponenti

Dijagram komponenti prikazuje organizaciju i odnose komponenti koje čine programsku potporu za našu aplikaciju. Naša aplikacija koristi dva sučelja, jedno za dohvat JS i CSS datoteka, a druga za dohvat JSON datoteka. Preko sučelja za dohvat JS i CSS datoteka poslužuju se datoteke za frontend. Komponenta App.js poslužuje jednu od JS datoteka na sučelje. Te datoteke su logičke cjeline nazvane po svojoj funkciji (npr. home page, login, header). Sve te datoteke ovisne su o biblioteci *react*. Preko sučelja za dohvat JSON datoteka dobivamo pristup REST API komponenti. REST API koristi HTTP zahtjeve kako bi pristupio podacima na backendu. JPA repozitorij dohvaća podatke iz baze podataka koristeći SQL upite. Komponenta React view ovisno o korisnikovim unosima prikazuje stvari s frontenda i dohvaća podatke s backenda.



Slika 4.9: Dijagram komponenti

## 5. Implementacija i korisničko sučelje

### 5.1 Korištene tehnologije i alati

Projekt je izveden primjenom različitih tehnologija i alata u svrhu implementacije kao i za dokumentaciju. U nastavku slijedi lista alata i tehnologija koje smo koristili uz kratke opise njihove uporabe.

**VisualStudioCode**<sup>1</sup> je besplatni visoko prilagodljiv tekstualni editor namijenjen širokom spektru programskih jezika i platformi. U sklopu ovoga projekta koristio se za pri pomoći u izradi dokumentacije. **LaTeX**<sup>2</sup> je visokokvalitetan sustav za pisanje teksta koji se koristi za izradu tehničkih i znanstvenih dokumentacija zato što pruža konzistentan izgled dokumenta. Dokumentacija ovog projekta pisana je u LaTeX-u uporabom različitih okruženja kao što su **TeXStudio**<sup>3</sup> (integrirano okruženje koje pruža sučelje za jednostavno pisanje, uređivanje i kompiliranje LaTeX koda), **Overleaf**<sup>4</sup> (online suradnički editor koji omogućuje zajednički rad više korisnika na istom dokumentu u stvarnom vremenu) i **LaTeX Workshop**<sup>5</sup> (extension unutar VSC-a koji pruža potpore za pisanje, uređivanje i kompiliranje LaTeX koda). Za izradu dijagrama korišten je **Astah**<sup>6</sup> alat za modeliranje UML dijagrama. To je alat koji omogućuje vizualno prikazivanje struktura, interakcija i dinamike sustava što pomaže pri razumijevanju i planiranju arhitekture aplikacija. Također, dio dijagrama iscrtan je pomoću web-aplikacije za crtanje dijagrama **draw.io**<sup>7</sup> koja pruža mogućnosti kao i Astah uz intuitivno korisničko sučelje koje olakšava vizualizaciju i komunikaciju ideja.

**PostgreSQL**<sup>8</sup> je snažan, open-source sustav za upravljanje bazama podataka koji

---

<sup>1</sup><https://code.visualstudio.com/>

<sup>2</sup><https://www.latex-project.org/>

<sup>3</sup><https://www.texstudio.org/>

<sup>4</sup><https://www.overleaf.com/>

<sup>5</sup><https://marketplace.visualstudio.com/items?itemName=James-Yu.latex-workshop>

<sup>6</sup><https://astah.net/>

<sup>7</sup><https://app.diagrams.net/>

<sup>8</sup><https://www.postgresql.org/>

podržava relacijski model podataka. Ova baza pruža napredne značajke poput transakcija, podrške za vanjske ključeve, indeksiranja i proširivosti. U ovome projektu PostgreSQL je korišten unutar integriranog razvojnog okruženja **DataGrip**<sup>9</sup> koji nudi napredne alate uz brzu navigaciju, pisanje i testiranje SQL upita, upravljanje shemama i izvođenje drugih zadataka nad bazom podataka. Od ponuđenih administratorskih alata koristio se **pgAdmin4**<sup>10</sup> koji omogućuje korisnicima pregled i manipulaciju bazom, shemama, tablicama i drugim objektima. Za izradu relacijskog modela baze podataka korišten je **ERDPlus**<sup>11</sup> besplatni online programski alat koji omogućuje izradu i uređivanje dijagrama baze podataka te njihov izvoz u .png formatu.

**IntelliJ IDEA**<sup>12</sup> je integrirano razvojno okruženje namijenjeno potpori razvoju softvera u različitim programskim jezicima s naglaskom na Javi. Ono pruža bogat skup alata za pisanje koda, rafaktoriranje i debugiranje, a podržava širok spektar tehnologija. Backend razvoj ovog projekta izveden je unutar IntelliJ IDEA razvojne okoline upotrebom **Java**<sup>13</sup> objektno orijentiranog programskog jezika u radnom okviru **Spring Boot**<sup>14</sup> (otvoreni okvir za razvoj Java aplikacija koji pruža konvencije nad konfiguracijama, ugrađene funkcionalnosti poput ugrađenog servera i podršku za mikrosereve). Za ispitivanje funkcionalnosti backenda koristila se **Postman**<sup>15</sup> platforma za suradnju u razvoju API-ja. Ovaj alat pruža korisničko sučelje za stvaranje, testiranje i upravljanje HTTP API zahtjevima i odgovorima.

**WebStorm**<sup>16</sup> je integrirano razvojno okruženje dizajnirano za web razvoj koje pruža napredne značajke kao što su automatsko dovršavanje koda, analiza u stvarnom vremenu, duboko integriranje s alatima za sustave za upravljanje verzijama i drugim alatima za web razvoj. Frontend se razvijao unutar WebStorm-a upotrebom označnog jezika **HTML**<sup>17</sup> (za prikaz sadržaja na web-u), stilske jezika **CSS**<sup>18</sup> (koji se koristi za opis izgleda i drugih stilskih karakteristika na web-u) i skriptnog pro-

---

<sup>9</sup><https://www.jetbrains.com/datagrip/>

<sup>10</sup><https://www.pgadmin.org/download/>

<sup>11</sup><https://erdplus.com/>

<sup>12</sup><https://www.jetbrains.com/idea/>

<sup>13</sup><https://www.java.com/en/>

<sup>14</sup><https://spring.io/projects/spring-boot/>

<sup>15</sup><https://www.postman.com/>

<sup>16</sup><https://www.jetbrains.com/webstorm/?var=1>

<sup>17</sup><https://www.w3schools.com/html/>

<sup>18</sup><https://www.w3schools.com/css/>



gramskog jezika **JavaScript**<sup>19</sup> (koji se u web pregledniku izvršava na strani korisnika) te njegove biblioteke otvorenog koda **ReactJS**<sup>20</sup> (koja se koristi za komponentni pristup razvoju i učinkovito ažuriranje sučelja bez potrebe za punim ponovnim iscrtavanjem).

**GitHub**<sup>21</sup> je online platforma korištena za upravljanje projektom i pohranu programskog koda i dokumentacije, a oslanja se na Git verzioniranje koda. Ona omogućuje suradnju među članovima grupe, olakšava praćenje zadataka, zaduženja i promjena na projektu te razvoj na posebnim granama i njihovo spajanje. **GitHub Desktop**<sup>22</sup> je besplatna aplikacija koja olakšava upotrebu Git-a s intuitivnim sučeljem koje olakšava dodavanje vlastitih promjena u repozitorij projekta. Korištena je kao alat za dodavanje i dohvaćanje podataka projekta. Za komunikaciju o detaljima projekta i raspodjelu posla i zadataka te međusobnu pomoć u izvođenju istih, tim je koristio **Discord**<sup>23</sup> server. Discord je platforma za komunikaciju putem glasovnih, video i tekstualnih razgovora, a omogućuje izradu različitih kanala za organizirano komuniciranje.

---

<sup>19</sup><https://www.javascript.com/>

<sup>20</sup><https://react.dev/>

<sup>21</sup><https://github.com/>

<sup>22</sup><https://desktop.github.com/>

<sup>23</sup><https://discord.com/>

## 5.2 Ispitivanje programskog rješenja

### 5.2.1 Ispitivanje komponenti

Napravili smo testove komponenti koje se nalaze u direktoriju `src/test/java/com/PROGI/backend`. Izvorni kod izgleda ovako:

```
@SpringBootTest
class BackendApplicationTests {

    private final validEmail underTest = new validEmail();
    private final validPassword underTest1 = new validPassword();

    @Test
    public void correctMail() {
        assertThat(underTest.test("hello@gmail.com")).isTrue();
    }

    @Test
    public void mailWithNoAt() {
        assertThat(underTest.test("hellogmail.com")).isFalse();
    }

    @Test
    public void mailWithNoDot() {
        assertThat(underTest.test("hello@gmail")).isFalse();
    }

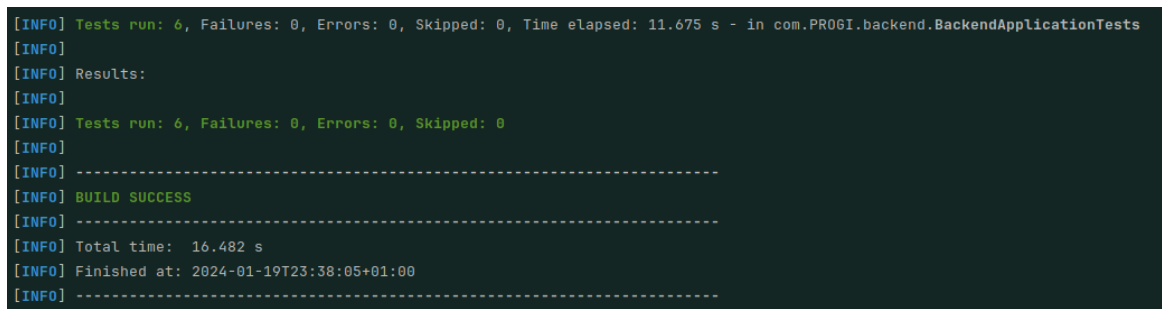
    @Test
    public void correctPass() {
        assertThat(underTest1.test("baba1223!")).isTrue();
    }

    @Test
    public void passUnder8Char() {
        assertThat(underTest1.test("adsa12")).isFalse();
    }
}
```

```
}

@Test
public void passWithInvalidChars() {
    assertThat(underTest1.test("pass@{}123;")).isFalse();
}
}
```

Nakon testova, u terminalu dobijemo:

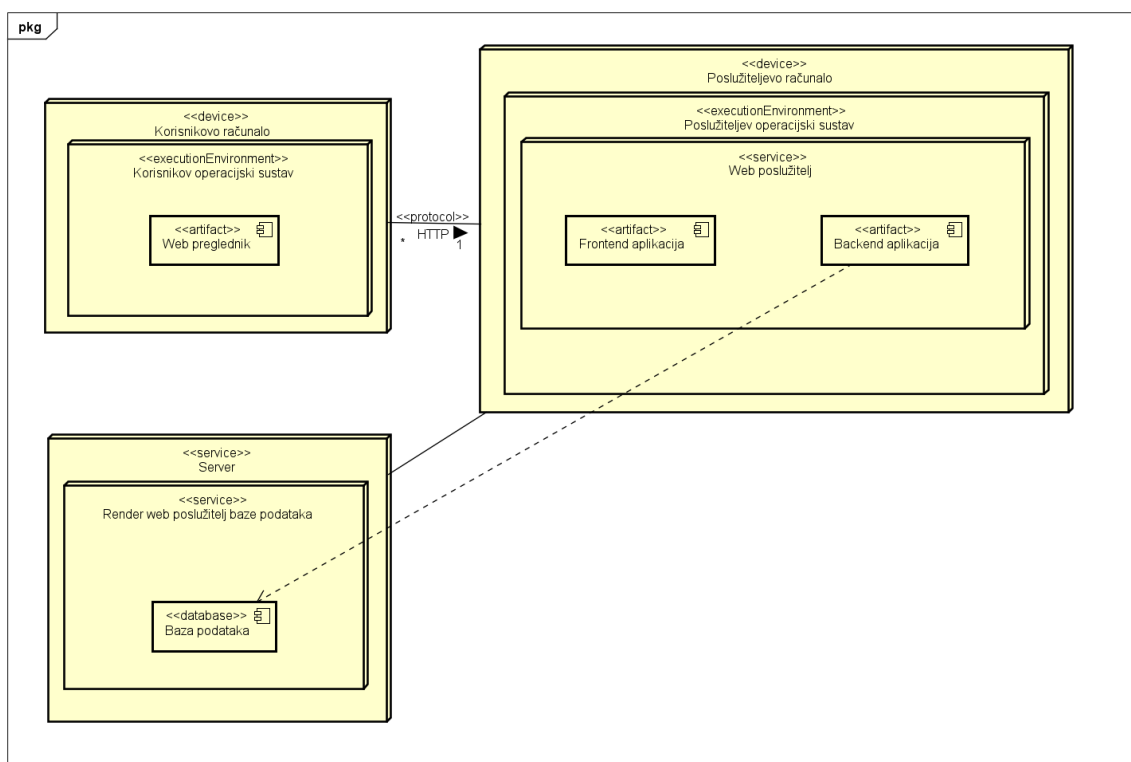


```
[INFO] Tests run: 6, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 11.675 s - in com.PRO6I.backend.BackendApplicationTests
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 6, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 16.482 s
[INFO] Finished at: 2024-01-19T23:38:05+01:00
[INFO] -----
```

Slika 5.1: Rezulati API testova

## 5.3 Dijagram razmještaja

Dijagram razmještaja prikazuje fizičku arhitekturu i konfiguraciju razmještaja našeg programskog sustava. Dijagrami razmještaja koji ne sadrže imena instanci nazivaju se specifikacijski dijagrami razmještaja. Korisnici koriste web preglednik kako bi mogli pristupiti web aplikaciji. Na poslužiteljskom računalu nalazi se web poslužitelj, a komunikacija između korisnika i poslužitelja se održava preko HTTP protokola. Baza podataka nalazi se na serveru, a o njoj, naravno, ovisi backend aplikacija s poslužiteljskog računala.



Slika 5.2: Dijagram razmještaja

## 5.4 Upute za puštanje u pogon

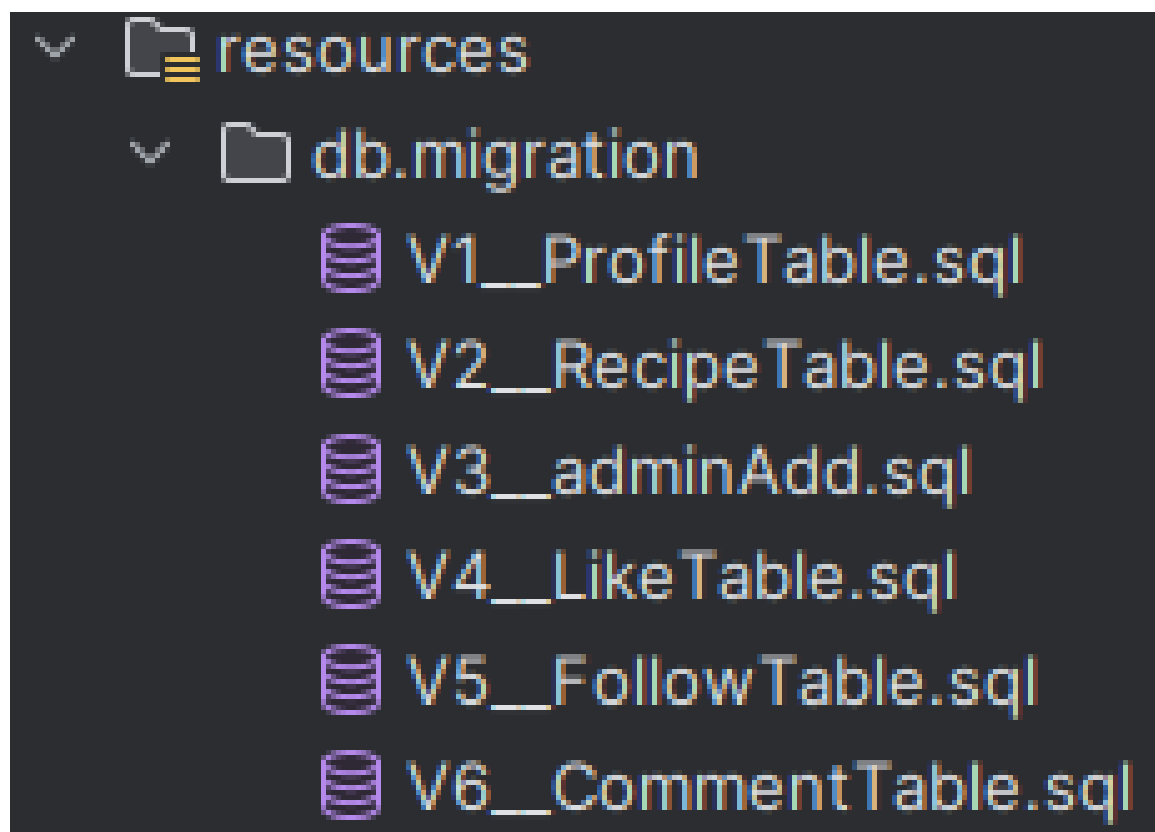
Aplikacija je dostupna online na linku:

```
\url{https://culinary-chaos-website.onrender.com/}.
```

Za postavljanje aplikacije na mrežu koristili smo Render. Na master grani dostupna je konfiguracija potrebna za puštanje u pogon preko Rendera. U ovom dijelu ćemo opisati kako se aplikacija pokreće na dev grani.

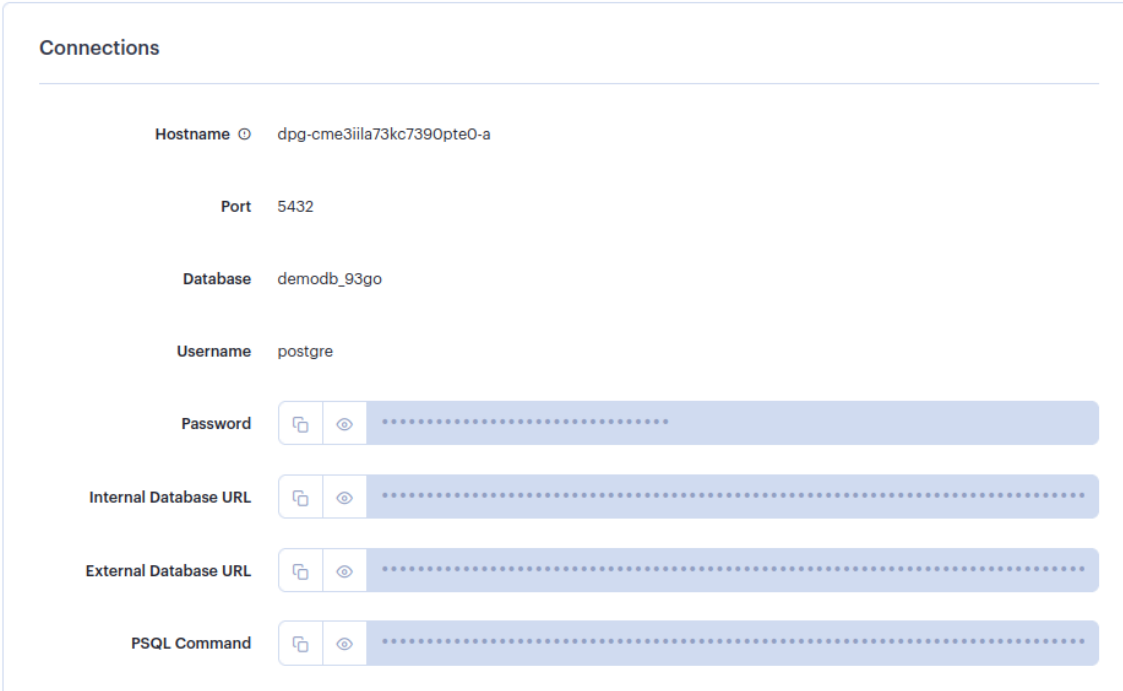
### Baza podataka

Prilikom pokretanja backenda, baza podataka se (ako već nije stvorena ranijim pokretanjem) stvara kroz migracije.



Slika 5.3: Migracije baze podataka

Baza se nalazi online i koristi PostgreSQL verziju 15. Sljedeći parametri su potrebni za spajanje na bazu (specifično external database URL i password):



The screenshot shows a 'Connections' window with the following fields:

- Hostname: dpg-cme3iila73kc7390pte0-a
- Port: 5432
- Database: demodb\_93go
- Username: postgres
- Password: [Redacted]
- Internal Database URL: [Redacted]
- External Database URL: [Redacted]
- PSQL Command: [Redacted]

Slika 5.4: Parametri potrebni za spajanje na bazu

## Backend

Backend web aplikacija je Java Spring Boot projekt. Za pokretanje servera potrebno je na računalu imati instalirano Java Development Kit 19 koji sadrži virtualni stroj koji pokreće aplikaciju i prevoditelja koji prevodi kod za taj virtualni stroj. Projekt u izgradnji i pokretanju koristi alat Maven. Za spajanje na bazu potrebno je u terminalu upisati:

```
PGPASSWORD=I7J0U3PrbY0jr9nQWoWCiLsK1c7JtSqA psql -h dpg-cme3iila73kc7390pte0-a.oregon-postgres.render.com -U postgres demodb_93go
```

Za pokretanje aplikacije, potrebno je pokrenuti razred:

```
src/main/java/com/PROGI/backend/BackendApplication.java.
```

Za puštanje aplikacije na Render potrebno je napraviti Docker file iz kojeg se može složiti sustav kontejner koji će pokretati aplikaciju. Nakon što se neka promjena stavi na master granu, backend se automatski restarta i pokreće se novi kod na toj grani.

### Build & Deploy

**Repository**  
The repository used for your Web Service.

https://github.com/ReflexLevel0/CulinaryChaosCrew

Edit

**Branch**  
The repository branch used for your Web Service.

dev

Edit

**Root Directory** Optional  
Defaults to repository root. When you specify a **root directory** that is different from your repository root, Render runs all your commands in the specified directory and ignores changes outside the directory.

Izvornikod/backend

Edit

**Build Filters**  
Filter for files and paths to monitor for automatic deploys. Learn about [build filters](#).  
**Filter paths are absolute.** If you've defined a root directory, you can still define paths outside of the root directory.

**Included Paths**  
Changes that match these paths will trigger a new build.

+ Add Included Path

**Ignored Paths**  
Changes that match these paths will **not** trigger a new build.

+ Add Ignored Path

**Registry Credential**  
Use a credential to access private images.  
[Manage credentials in Settings](#)

No credential

Edit

**Dockerfile Path**  
Path to your Dockerfile relative to the repository root. This is *not* relative to your Docker build context. For example, `./subdir/Dockerfile`.

Izvornikod/backend/ ./Dockerfile

Edit

Slika 5.5: Deployment backenda

## Frontend

Front-end web aplikacija je React projekt. Za pokretanje servera potrebno je na računalu imati instaliran Node.js i npm (Node Package Manager). Kada su instalirani moduli, možemo pokrenuti server tako da upišemo u terminal:

Culinary Chaos Crew

stranica 54/67

20. siječnja 2024.

```
npm run start.
```

Svi HTTP zahtjevi za back-end šalju se na:

```
https://culinary-chaos-backend.onrender.com/api
```

### Build & Deploy

**Repository**  
The repository used for your Web Service.

https://github.com/ReflexLevel0/CulinaryChaosCrew

Edit

**Branch**  
The repository branch used for your Web Service.

dev

Edit

**Root Directory** Optional  
Defaults to repository root. When you specify a [root directory](#) that is different from your repository root, Render runs all your commands in the [specified directory](#) and ignores changes outside the directory.

IzvorniKod/website

Edit

**Build Filters**  
Filter for files and paths to monitor for automatic deploys. Learn about [build filters](#).  
**Filter paths are absolute.** If you've defined a root directory, you can still define paths outside of the root directory.

**Included Paths**  
Changes that match these paths will trigger a new build.

+ Add Included Path

**Ignored Paths**  
Changes that match these paths will **not** trigger a new build.

+ Add Ignored Path

**Build Command**  
This command runs in the root directory of your repository when a new version of your code is pushed, or when you deploy manually. It is typically a script that installs libraries, runs migrations, or compiles resources needed by your app.

IzvorniKod/website/ \$ yarn build

Edit

Slika 5.6: Deployment frontenda



## 6. Zaključak i budući rad

Izuzetno je mnogo vremena bilo potrošeno kako bi nadoknadili manjak jednog člana. Sve tehnologije koje smo koristili su nam bile nepoznate te smo zato čvsto odredili uloge kako bi smanjili obujam učenja pojedinog člana. Backend je morao savladati Spring Boot i Maven sa svim svojim značajkama. Frontend je morao ovladati React-om za uređivanje komponenti stranice i npm-om za instalaciju paketa. Ljudi zaduženi za dokumentaciju su morali ovladati LaTeX-om. No kako to obično biva u malim timovima na kraju je svatko radio sve. Kada bi isti tim ljudi duže programirao zajedno te se dogovorio oko normi oblikovanja mogli bi biti brži i efektivniji, no to je bilo izuzetno teško sa našim popunjenim rasporedima. Perspektiva koju smo svi usvojili je značajnost rada svakog od nas. Koliko god se pomoć činila mala na kraju bi njena značajnost imala veliku težinu. **Ukratko, najbitnija je kontinuiranost u radu jer projekt iako kratak nije gotov u jednom danu.**

Unatoč volji, perspektivi te požrtvovnosti svih članova tima sljedeće značajke nismo uspjeli implementirati:

- video-chat između korisnika
- admin page (sa svojim značajkama)
- praćenje korisnika
- mijenjanje recepata

# Popis literature

## *Kontinuirano osvježavanje*

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

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

# Indeks slika i dijagrama

2.1	<i>CookBooked</i> logo . . . . .	6
2.2	Izgled naslovne stranice <i>AllRecipes</i> . . . . .	7
3.1	Cijeli UML dijagram . . . . .	20
3.2	UML dijagram za UC1-UC4 . . . . .	21
3.3	UML dijagram za UC13-UC15 . . . . .	22
3.4	Sekvencijski dijagram za UC1-4 . . . . .	24
3.5	Sekvencijski dijagram za UC6-7 . . . . .	26
3.6	Sekvencijski dijagram za UC8-10;12 . . . . .	28
3.7	Sekvencijski dijagram za UC13-14 . . . . .	29
4.1	Skica arhitekture sustava . . . . .	32
4.2	Skica <b>Controller - Service - Repository</b> troslojne arhitekture backenda	32
4.3	Dijagram baze podataka . . . . .	36
4.4	Dijagram razreda - Controller . . . . .	38
4.5	Dijagram razreda - Service . . . . .	39
4.6	Dijagram razreda - Dao . . . . .	40
4.7	Dijagram stanja za korisnika . . . . .	42
4.8	Dijagram aktivnosti za pregledavanje nespremljenih recepata i autora	44
4.9	Dijagram komponenti . . . . .	45
5.1	Rezultati API testova . . . . .	50
5.2	Dijagram razmještaja . . . . .	51
5.3	Migracije baze podataka . . . . .	52
5.4	Parametri potrebni za spajanje na bazu . . . . .	53
5.5	Deployment backenda . . . . .	54
5.6	Deployment frontenda . . . . .	55
6.1	. . . . .	65
6.2	. . . . .	65
6.3	. . . . .	66
6.4	. . . . .	66

6.5	.....	66
6.6	.....	67
6.7	.....	67
6.8	.....	67

# Dodatak: Prikaz aktivnosti grupe

## Dnevnik sastajanja

### *Kontinuirano osvježavanje*

*U ovom dijelu potrebno je redovito osvježavati dnevnik sastajanja prema predlošku.*

#### 1. sastanak

- Datum: 19. listopada 2023.
- Prisustvovali: Dominik Dejanović, Ante Čavar, Borna Dramalija, Adam Vuković, Nikola Trebus, Ljubica Nikolić
- Teme sastanka:
  - upoznavanje tima
  - raspodjela zadataka
  - rasprava o korištenim tehnologijama

#### 2. sastanak

- Datum: 27. listopada 2023.
- Prisustvovali: Dominik Dejanović, Ante Čavar, Adam Vuković, Nikola Trebus, Ante Volarević, Ljubica Nikolić
- Teme sastanka:
  - dogovor oko dizajna baze podataka
  - dogovor oko rada na dokumentaciji
  - rasprava o budućim ciljevima

#### 3. sastanak

- Datum: 2. studeni 2023.
- Prisustvovali: Ljubica Nikolić, Ante Čavar
- Teme sastanka:
  - izrada dokumentacije
  - izrada osnovne verzije API-a

#### 4. sastanak

- Datum: 9. listopada 2023.

- Prisustvovali: Dominik Dejanović, Ljubica Nikolić
- Teme sastanka:
  - izrada početne web stranice
  - razdijela stranica za prijavu i registraciju
  - dokumentacija - UML i sekvencijski dijagrami

#### 5. sastanak

- Datum: 14. listopada 2023.
- Prisustvovali: Dominik Dejanović, Ante Čavar, Adam Vuković, Nikola Trebus, Ljubica Nikolić, Ante Volarević
- Teme sastanka:
  - raspodijela zadataka
  - razgovor o deployanju stranice

#### 6. sastanak

- Datum: 16. listopada 2023.
- Prisustvovali: Dominik Dejanović, Ante Čavar, Adam Vuković, Nikola Trebus, Ante Volarević
- Teme sastanka:
  - raspodijela završnih zadataka

#### 7. sastanak

- Datum: 7. prosinca 2023.
- Prisustvovali: Dominik Dejanović, Ante Čavar
- Teme sastanka:
  - osvrt na prvi ciklus

#### 8. sastanak

- Datum: 21. prosinca 2024.
- Prisustvovali: Dominik Dejanović, Ante Čavar, Adam Vuković, Nikola Trebus, Ante Volarević, Ljubica Nikolić
- Teme sastanka:
  - budući rad kao šesteročlana grupa

#### 9. sastanak

- Datum: 11. siječnja 2024.
- Prisustvovali: Dominik Dejanović, Ante Čavar, Adam Vuković, Nikola Trebus, Ante Volarević, Ljubica Nikolić
- Teme sastanka:
  - raspodjela preostalog posla

## 10. sastanak

- Datum: 19. siječnja 2024.
- Prisustvovali: Dominik Dejanović, Ante Čavar, Adam Vuković, Nikola Trebus, Ljubica Nikolić, Ante Volarević
- Teme sastanka:
  - troubleshooting
  - završne promjene

## Tablica aktivnosti

Aktivnost izražena u satima.

	Dominik Dejanović	Adam Vuković	Ljubica Nikolić	Ante Volarević	Nikola Trebus	Ante Čavar	
Upravljanje projektom	2						
Opis projektnog zadatka		3					
Funkcionalni zahtjevi			2				
Opis pojedinih obrazaca	4		2				
Dijagram obrazaca	1		3	1			
Sekvencijski dijagrami	1		7				
Opis ostalih zahtjeva			5				
Arhitektura i dizajn sustava	3				2	1	
Baza podataka		1	2	1	1	6	
Dijagram razreda			30		15		
Dijagram stanja		4					
Dijagram aktivnosti		3					
Dijagram komponenti		3					
Korištene tehnologije i alati			2				
Ispitivanje programskog rješenja	10					15	
Dijagram razmještaja		2					
Puštanje u pogon	10			2		10	
Dnevnik sastajanja	1				2		
Zaključak i budući rad						1	

Nastavljeno na idućoj stranici



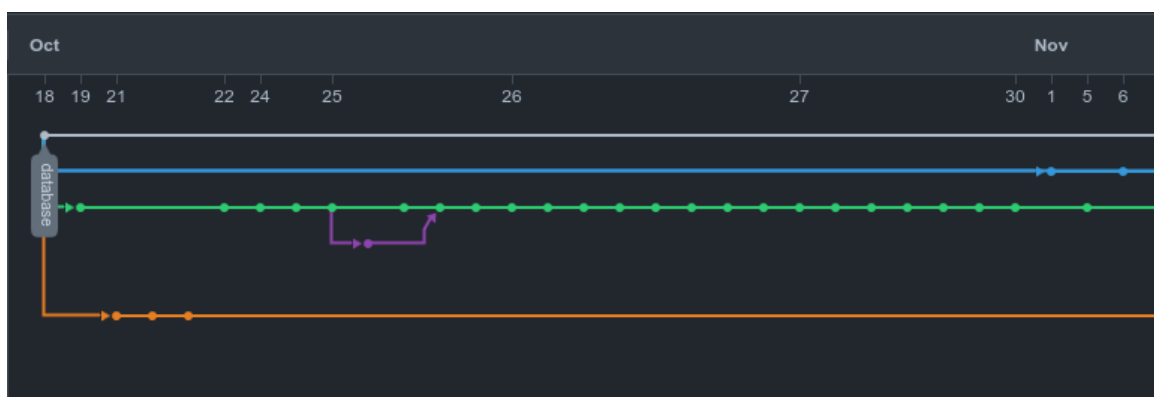
Nastavljeno od prethodne stranice

	Dominik Dejanović	Adam Vuković	Ljubica Nikolić	Ante Volarević	Nikola Trebus	Ante Čavar	
Popis literature							
Web stranica	12	35					
<i>Dodatne stavke kako ste podijelili izradu aplikacije</i>							
<i>npr. izrada početne stranice</i>							
<i>izrada baze podataka</i>		2		1			
<i>spajanje s bazom podataka</i>		2		1			
<i>back end</i>	5	10		6	3	30	

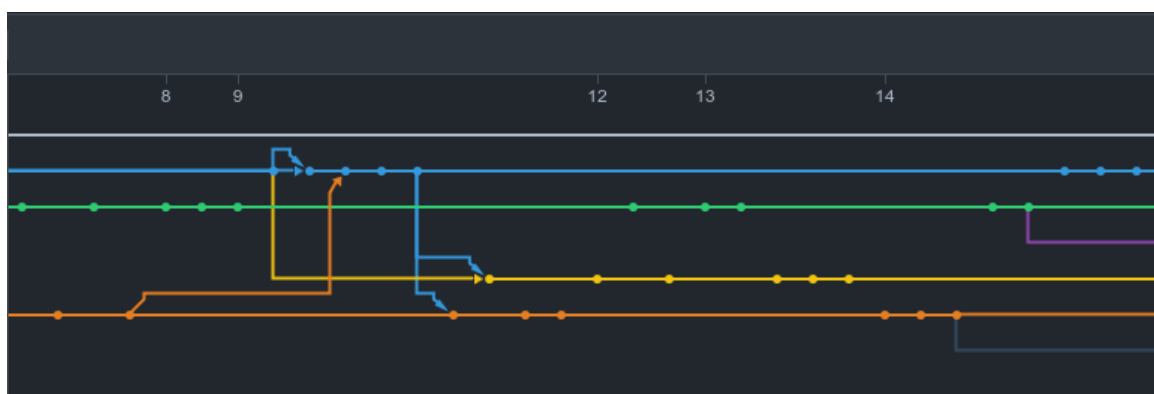
## Dijagrami pregleda promjena

### dio 2. revizije

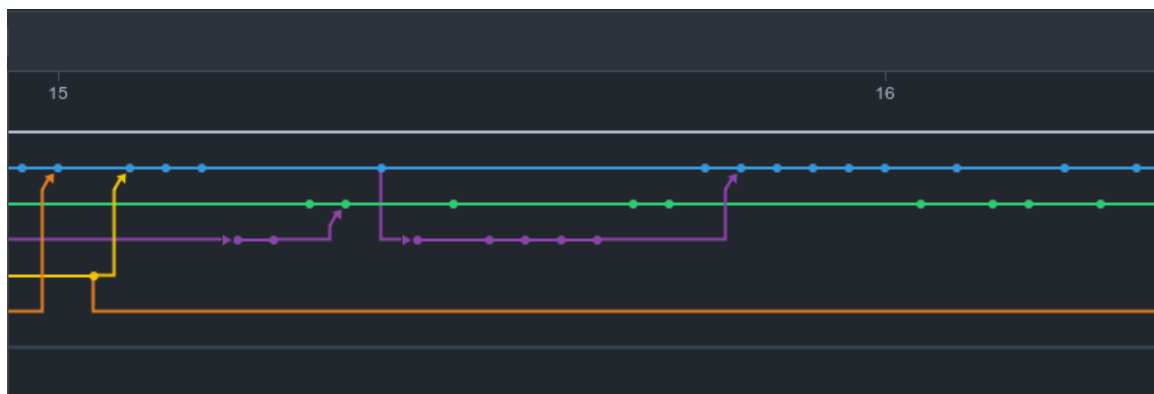
Prenijeti dijagram pregleda promjena nad datotekama projekta. Potrebno je na kraju projekta generirane grafove s gitlaba prenijeti u ovo poglavlje dokumentacije. Dijagrami za vlastiti projekt se mogu preuzeti s [gitlab.com](https://gitlab.com) stranice, u izborniku Repository, pritiskom na stavku Contributors.



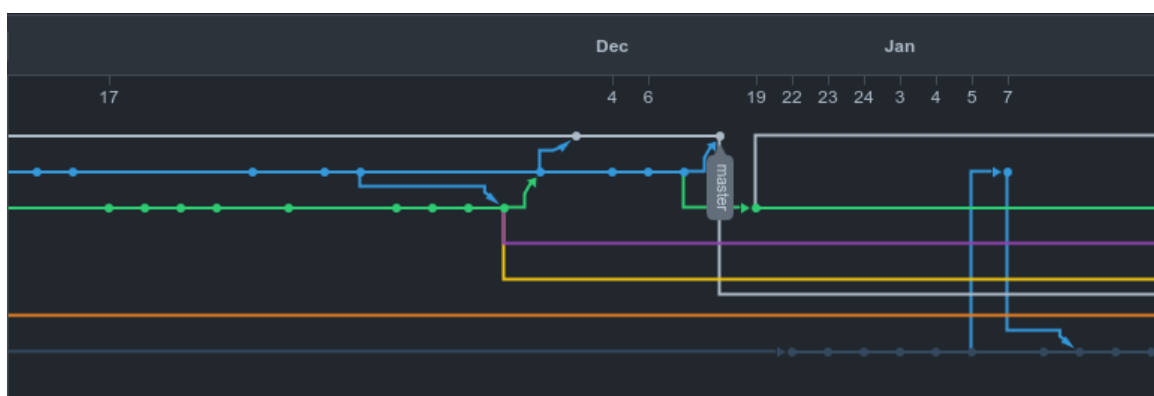
Slika 6.1:



Slika 6.2:



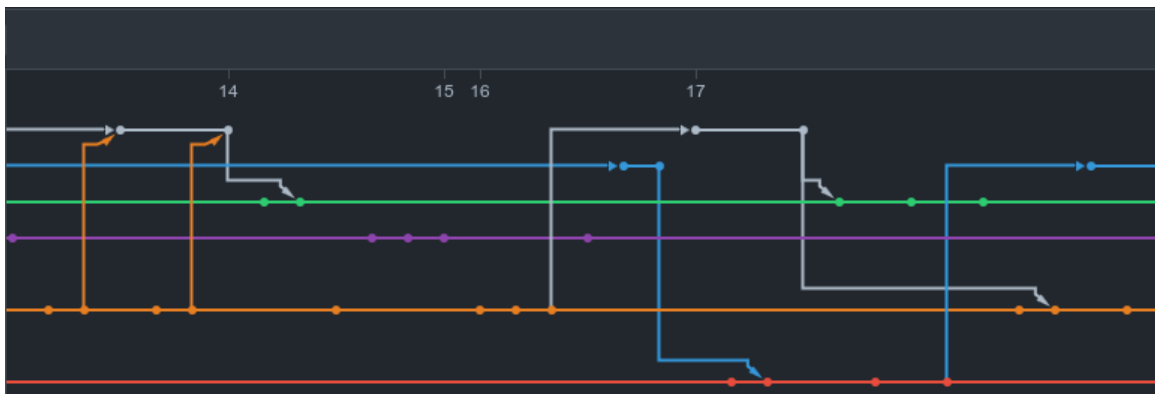
Slika 6.3:



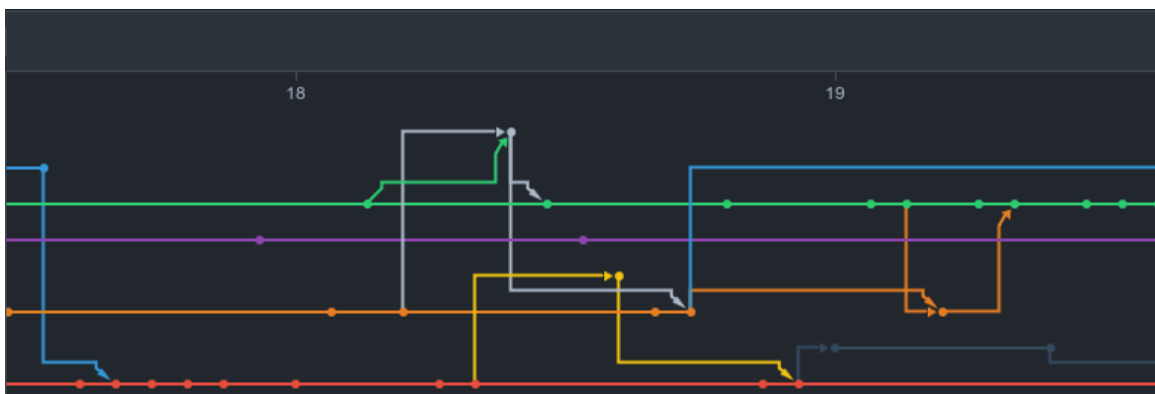
Slika 6.4:



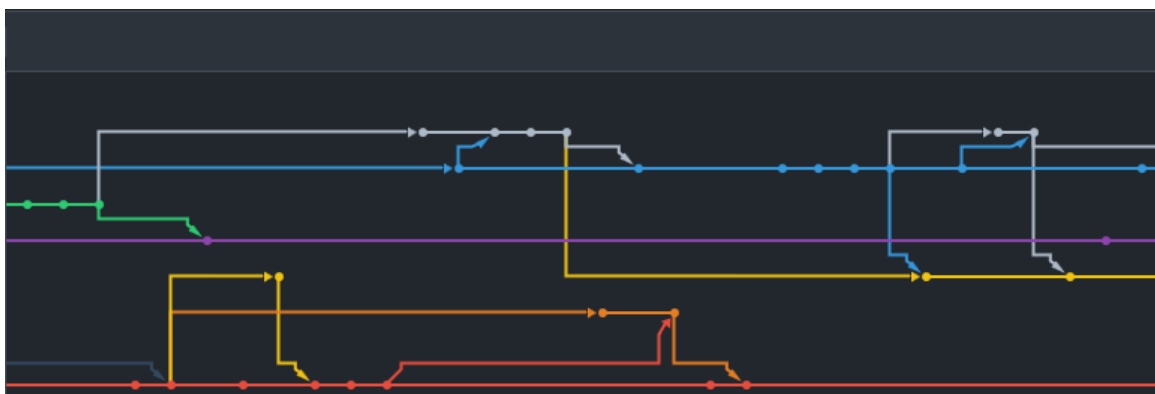
Slika 6.5:



Slika 6.6:



Slika 6.7:



Slika 6.8: