

Programsko inženjerstvo

Ak. god. 2023./2024.

CookBooked

Dokumentacija, Rev. 2

Grupa: Undercooked

Voditelj: Eloise Habek

Datum predaje: 19. 1. 2024.

Nastavnik: Nikolina Frid

Sadržaj

1	Dnevnik promjena dokumentacije	3
2	Opis projektnog zadatka	5
2.1	Uvod	5
2.2	Postojeća rješenja i njihovi problemi	5
2.3	Osnovne mogućnosti stranice	6
2.4	Napredne mogućnosti stranice	7
2.5	Komunikacija među korisnicima	7
2.6	Administracija	8
3	Specifikacija programske potpore	9
3.1	Funkcionalni zahtjevi	9
3.1.1	Obrasci uporabe	11
3.1.2	Sekvencijski dijagrami	24
3.2	Ostali zahtjevi	28
4	Arhitektura i dizajn sustava	29
4.1	Organizacija sustava	29
4.1.1	Uvod	29
4.1.2	Sklopovski zahtjevi	30
4.1.3	Organizacija aplikacije	30
4.2	Baza podataka	30
4.2.1	Opis tablica	31
4.2.2	Dijagram baze podataka	35
4.3	Dijagram razreda	36
4.4	Dijagram stanja	38
4.5	Dijagram aktivnosti	40
4.6	Dijagram komponenti	42
5	Implementacija i korisničko sučelje	43
5.1	Korištene tehnologije i alati	43

5.2	Ispitivanje programskog rješenja	45
5.2.1	Ispitivanje komponenti	45
5.2.2	Ispitivanje sustava	50
5.3	Dijagram razmještaja	53
5.4	Upute za puštanje u pogon	54
5.4.1	Instalacija potrebne programske potpore	54
5.4.2	Pokretanje front-enda	55
5.4.3	Pokretanje back-enda i baze podataka	56
6	Zaključak i budući rad	57
	Popis literature	59
	Indeks slika i dijagrama	60
	Dodatak: Prikaz aktivnosti grupe	61

1. Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Preuzet i popunjen predložak	Mateo Plavec	19.10.2023.
0.2	Prva verzija <i>Opisa projektnog zadatka</i>	Luka Grubišin	27.10.2023.
0.3	Skicirana prva verzija obrasca uporabe	Matej Magat	29.10.2023.
0.4	Dodani <i>Use Case</i> dijagrami i funkcionalni zahtjevi	Eloise Habek	1.11.2023.
0.5	Dodani Sekvencijski dijagrami	Andrej Stanković	3.11.2023.
0.6	Opis arhitekture sustava	Luka Grubišin	6.11.2023.
0.7	Dodana struktura podataka i ER dijagram baze podataka i dijagrami razreda	Anabel Dautović	08.11.2023.
0.8	Dorađeni dijagrami obrasca uporabe	Eloise Habek	13.11.2023.
0.9	Dorađen opis arhitekture sustava	Luka Grubišin	15.11.2023.
0.95	Dodani dorađeni dijagrami razreda	Anabel Dautović	15.11.2023.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodatka	Autori	Datum
1.0	Verzija samo s bitnim dijelovima za 1. ciklus	Luka Grubišin, Eloise Habek, Anabel Dautović	17.11.2023.
1.1	Dodan dijagram aktivnosti	Luka Grubišin	19.12.2023.
1.2	Dodan dijagram stanja	Eloise Habek	19.12.2023.
1.3	Dodan dijagram razmještaja i popravljen dijagram baze podataka	Luka Grubišin	9.1.2024.
1.4	Dodan dijagram komponenti i opis dijagrama komponenti	Anabel Dautović	13.01.2024.
1.5	Dodan zaključak	Anabel Dautović	13.01.2024.
1.6	Dodane Iskorištene tehnologije	Eloise Habek	13.01.2024.
1.7	Dodane upute za puštanje u pogon	Luka Grubišin	16.01.2024
1.8	Dodan opis ispitivanja sustava	Luka Grubišin	17.01.2024
1.9	Dodan opis ispitivanja komponenti	Luka Grubišin	18.01.2024
2.0	Posljednja verzija dokumentacije	Eloise Habek	19.01.2024

2. Opis projektnog zadatka

2.1 Uvod

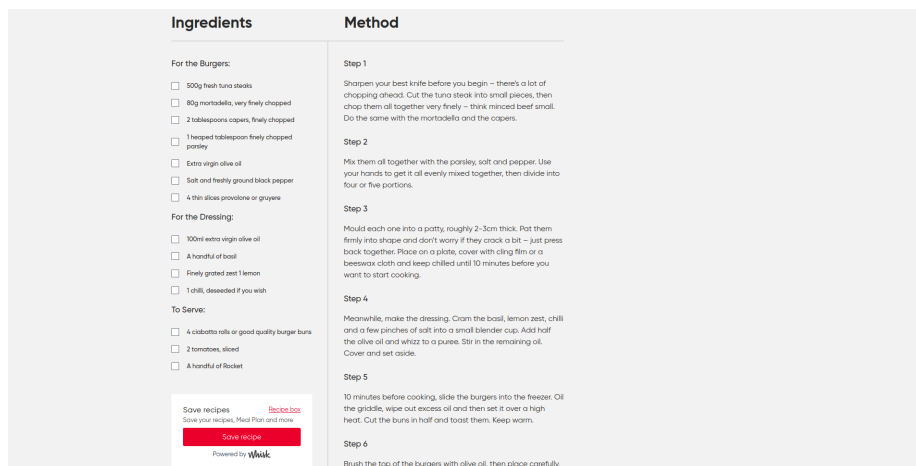
Cilj ovog projekta je razvoj web aplikacije za razmjenu recepata i diskusiju između korisnika i autora recepata koja bi ljudima omogućila jednostavno isprobavanje novih jela i poboljšanje svojih kulinarskih sposobnosti.

2.2 Postojeća rješenja i njihovi problemi

Postoje brojna popularna slična rješenja, no uglavnom pate od loše preglednosti zbog previše prevelikih slika, puno praznog prostora i velike količine popratnog teksta koji nema veze sa sastojcima ili pripremom kao što je prikazano na slikama 2.1 i 2.2 u nastavku.



Slika 2.1: Coolinarika - Prevelika slova, puno neiskorištenog prostora



Slika 2.2: Foodnetwork - Preglednije, ali i dalje puno praznog prostora

Recepti bi trebali sadržavati samo informacije važne za pripremu i možda jednu ili dvije uvodne rečenice. Većina recepata trebala bi stati na jednu stranicu kako bi bili pregledniji, ali i kako bi ih se lako moglo ispisati na papir. Ovakav izgled stranice nije uvijek jednostavno ostvariti, ali bi trebao uvelike poboljšati iskustvo korištenja.

2.3 Osnovne mogućnosti stranice

Glavna mogućnost stranice je naravno, prikazivanje sastojaka i uputa za pripremu, ali osim njih, recepti sadrže i brojne druge informacije koje olakšavaju pripremu i traženje sličnih recepata kao što su:

- kategorija jela
- vrsta kuhinje
- neuobičajeni sastojci
- alergeni
- vrijeme pripreme
- broj porcija
- ostale oznake

Stranica razlikuje dvije vrste korisnika: gosti, odnosno neregistrirani korisnici, i registrirani korisnici. Neregistrirani korisnici mogu pristupiti samo osnovnim funkcionalnostima stranice. Imaju potpuni pristup svim receptima te mogu pretraživati recepte po prije navedenim oznakama ili po njihovom nazivu. Isto tako mogu uz neka ograničenja pregledavati profile registriranih korisnika. Ovo će

većini korisnika koji samo povremeno traže nove recepte ili inspiraciju vjerojatno biti dovoljno, a oni koji žele pristupiti naprednijim pogodnostima koje stranica nudi moraju se registrirati. Postupak registracije vrlo je jednostavan. Potrebno je samo navesti sljedeće podatke kako bi korisnik stvorio novi profil:

- korisničko ime
- adresu elektroničke pošte
- lozinku

2.4 Napredne mogućnosti stranice

Nakon što se korisnik registrira i prijavi, otvaraju mu se brojne dodatne mogućnosti stranice. Najvažnija i najzanimljivija je naravno mogućnost objave vlastitih recepata. Kako bi korisnik objavio recept, mora navesti naziv recepta, sastojke, postupak pripreme i ukupno trajanje pripreme. Ostali podatci nisu neophodni, ali ih je korisno navesti kako bi zainteresirani korisnici mogli brže i jednostavnije naći recepte koji odgovaraju njihovim željama i potrebama.

Svaki registrirani korisnik također može označavati i spremati recepte na svoj osobni profil kako bi mu u budućnosti bili lako dostupni, a isto tako mogu zapratiti druge autore kako bi bili automatski obaviješteni kada netko od njihovih najdražih autora objavi novi recept. Objavljeni recepti, pratitelji i osobe koje prate korisnika javno su vidljivi na profilu svakog korisnika.

2.5 Komunikacija među korisnicima

Osim što mogu ostavljati komentare na receptima i tako komunicirati s autorom i drugim korisnicima, registrirani korisnici također imaju mogućnost izravnog kontaktiranja autora recepata ako trebaju dodatna pojašnjenja ili žele uputiti svoje pohvale (ili kritike) autoru. Aplikacija omogućava jednostavnu razmjenu poruka između korisnika, ali isto tako, autor može navesti i druge načine na koje ga se može kontaktirati, kao što su elektronička pošta ili broj telefona. Također može navesti i vremenske periode u kojima je dostupan.

2.6 Administracija

Osim gostiju i registriranih korisnika, važan čimbenik u radu stranice su i administratori. Administratori imaju vlastite korisničke profile, kao i obični registrirani korisnici, ali imaju dodatne ovlasti koje im omogućavaju upravljanje stranicom. Ovlasti administratora uključuju:

- upravljanje korisnicima
 - brisanje korisnika
 - uređivanje korisničkih profila
 - dodavanje korisnika
- upravljanje receptima
 - brisanje recepata
 - uređivanje recepata
 - mijenjanje oznaka recepata

3. Specifikacija programske potpore

3.1 Funkcionalni zahtjevi

Dionici:

1. Nastavnik
2. Demonstrator
3. Razvojni tim
4. Korisnici
5. Administrator

Aktori i njihovi funkcionalni zahtjevi:

1. korisnik (inicijator) može:
 - (a) pretražiti recept po značajkama
 - (b) pregledati recept
 - (c) osvježiti broj porcija, čime će se ažurirati mjerice sastojaka u skladu s navedenim brojem porcija
 - (d) otvoriti profile registriranih korisnika koji su postavili recept
2. Neregistrirani korisnik (inicijator) može:
 - (a) sve što i korisnik (specijalizacija korisnika)
 - (b) otvoriti novi korisnički račun za koji je potrebno unjeti korisničko ime koje nije već zauzeto, adresu elektroničke pošte i lozinku računa
3. Registrirani korisnik (inicijator) može:
 - (a) sve što i korisnik (specijalizacija korisnika)
 - (b) ažurirati profil, podaci koje je moguće ažurirati su email, broj telefona i vremenski period u kojima je korisnik dostupan
 - (c) objaviti recept s obaveznim navođenjem naziva recepta, sastojaka, postupka pripreme i ukupnog trajanja pripreme, mogućnost navođenja dodatnih karakteristika poput oznaka tipa jela i dodavanja slika i videozapisa

- (d) ažuriranje već objavljenih recepata
- (e) ocijeniti objavu
- (f) spremi recept kako bi im mogli lakše pristupiti kasnije
- (g) zapratiti drugog autora (registriranog korisnika), korisnik će dobiti obavijest kada autor objavi novi recept
- (h) pregledati vlastiti profil
- (i) dodavanje nove oznake poput kategorije ili vrste kuhinje
- (j) brisati recepte sa svojeg profila
- (k) pisanje komentara na recepte drugih korisnika
- (l) slanje poruka u privatnom chatu
- (m) pregled poruka privatnih chatova

4. Administrator (inicijator) može:

- (a) sve što i korisnik (specijalizacija korisnika)
- (b) brisanje korisničkog profila
- (c) brisanje recepata
- (d) dodati novog administratora
- (e) pregled mogućnosti administratora
- (f) pristupiti statistikama (npr. najpopularniji recept po broju profila koji su ga spremili, najaktivniji profil s najviše recepata, najpopularniji profil s najčešće spremljenim receptima)
- (g) brisanje komentara koji nisu u skladu s pravilima ophođenja na platformi

5. Baza podataka (sudionik) može:

- (a) pohranjuje sve podatke o korisnicima i njihovim ovlastima
- (b) pohranjuje sve podatke o receptima
- (c) pohranjuje informacije o odnosima među korisnika i recepata

3.1.1 Obrasci uporabe

UC1 - Pregled recepta

- **Glavni sudionik:** Korisnik
- **Cilj:** Pregledati objavljene recepte
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Korisnik odabire recept
 2. Dohvaća se recept iz baze podataka
 3. Prikaz recepta na odvojenoj stranici

UC2 - Pretraživanje recepata

- **Glavni sudionik:** Korisnik
- **Cilj:** Prikazati recepte u skladu s podacima unesenim u tražilicu
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Korisnik odabire pretraživanje recepata
 2. Upisuje ključnu riječ (kategorija, vrsta kuhinje, specifični sastojci)
 3. Baza podataka vraća recepte koji odgovaraju parametrima pretrage
 4. Dobiveni recepti prikazuju se u aplikaciji
- **Opis mogućih odstupanja:**
 - 3.a U bazi podataka ne nalazi se niti jedan recept koji odgovara unesenim parametrima
 1. Sustav korisniku prikazuje poruku da trenutno nema recepata koji odgovaraju njegovom unosu

UC3 - Registriranje novog korisnika

- **Glavni sudionik:** Neregistrirani korisnik
- **Cilj:** Stvaranje novog korisničkog računa
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Neregistrirani korisnik odabire opciju za registraciju

2. Unosi potrebne podatke (korisničko ime, adresu elektroničke pošte i lozinku)
 3. Baza podataka provjerava jesu li korisničko ime ili e-mail adresa već zauzeti
 4. Ako su korisničko ime i e-mail adresa slobodni, novog registriranog korisnika se upisuje u bazu
 5. Učitava se naslovna stranica
- **Opis mogućih odstupanja:**
 - 3.a Korisničko ime ili adresa elektroničke pošte već su zauzeti
 1. Korisnik dobiva poruku da su korisničko ime ili e-mail adresa već korišteni
 2. Korisnika se traži ponovni upis podataka

UC4 - Prikazivanje tuđeg korisničkog profila

- **Glavni sudionik:** Korisnik
- **Cilj:** Pregled profila drugog registriranog korisnika
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Korisnik odabire ikonu autora neke objave
 2. Korisniku se prikazuje profil autora
- **Opis mogućih odstupanja:**
 - 1.a Korisnik koristi tražilicu kako bi našao profil koji želi pregledati
 1. Korisnik odabire jedan od profila dobivenih pretragom
 2. Korisniku se prikazuje odabrani profil
 - 2.a Korisnik pregledava profil koji prati
 1. Korisnik otvara vlastiti profil
 2. Odabire opciju za prikaz profila koje prati
 3. Korisnik odabire jedan od navedenih profila
 4. Prikazuje se odabrani profil

UC5 - Ažuriranje broj porcija

- **Glavni sudionik:** Korisnik
- **Cilj:** Prikazati potrebne sastojke recepta u skladu s potrebnim porcijama jela
- **Sudionici:** Baza podataka
- **Preduvjet:** UC1 (Pregled recepta)

- **Opis osnovnog tijeka:**

1. Korisnik na prikazanom receptu stiskanjem tipki (+/-) ažurira broj porcija
2. Prikaz recepta se ažurira s obzirom na broj porcija

UC6 - Prijava

- **Glavni sudionik:** Registrirani korisnik, Administrator
- **Cilj:** Dobivanje pristupa mogućnostima registriranog korisnika
- **Sudionici:** Baza podataka
- **Preduvjet:** UC3 (Registracija)
- **Opis osnovnog tijeka:**
 1. Registrirani korisnik na naslovnoj stranici odabire opciju za prijavu
 2. Korisnik unosi potrebne podatke (e-mail adresu i lozinku)
 3. Baza podataka provjerava je li korisnik registriran
 4. Baza podataka provjerava je li lozinka ispravna
 5. Učitava se naslovna stranica
- **Opis mogućih odstupanja:**
 - 3.a Korisnik nije registriran
 1. Korisnik dobiva poruku da nije registriran
 2. Korisnika se preusmjerava na obrazac za registraciju
 - 4.a Lozinka nije ispravna
 1. Korisnika se traži da upiše ispravnu lozinku

UC7 - Prikaz najnovijih recepta praćenih korisnika

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Pregled najnovijih recepta praćenih korisnika
- **Sudionici:** Baza podataka
- **Preduvjet:** UC6 (Prijava)
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju prikaza noviteta recepata
 2. Za korisnike koje prati sudionik se dohvaćaju najnoviji recepti iz baze podataka
 3. Dohvaćeni recepti se prikazuju na novoj stranici

UC8 - Objava recepta

- **Glavni sudionik:** Registrirani korisnik

- **Cilj:** Objava novog recepta
- **Sudionici:** Baza podataka
- **Preduvjet:** UC6 (Prijava)
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju za novu objavu
 2. Korisnik unosi naslov recepta, sastojke, upute za pripremu i vrijeme pripreme te po izboru dodaje i oznake kategorija, slike i videozapise
 3. Baza podataka zapisuje novi recept i moguće nove oznake unesene
 4. Svi pratitelji autora dobivaju obavijest o novom receptu

UC9 - Uređivanje objave

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Urediti prethodno objavljeni recept
- **Sudionici:** Baza podataka
- **Preduvjet:** UC7 (Objava recepta)
- **Opis osnovnog tijeka:**
 1. Korisnik odabire pregled vlastitog profila
 2. Korisnik odabire objavu koju želi urediti
 3. Odabire opciju za uređivanje
 4. Korisnik uređuje sadržaj recepta
 5. Po završetku odabire opciju za kraj uređivanja
 6. Baza podataka ažurira recept i dodaje moguće nove oznake

UC10 - Brisanje objave

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Brisanje prethodno objavljenog recepta
- **Sudionici:** Baza podataka
- **Preduvjet:** UC7 (Objava recepta)
- **Opis osnovnog tijeka:**
 1. Korisnik odabire pregled vlastitog profila
 2. Korisnik odabire objavu koju želi obrisati
 3. Korisnik odabire opciju za brisanje
 4. Baza podataka briše odabrani recept

UC11 - Slanje poruke drugom registriranom korisniku

- **Glavni sudionik:** Registrirani korisnik

- **Cilj:** Komunikacija među korisnicima putem poruka
- **Sudionici:** Baza podataka
- **Preduvjet:** UC4 (Prikazivanje tuđeg profila)
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju za slanje poruke
 2. Otvara se polje za pisanje poruke
 3. Kada napiše poruku, korisnik odabire opciju za slanje
 4. Poruka se upisuje u bazu podataka u inbox primatelja

UC12 - Zaprati korisnika

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Praćenje drugog korisnika kako bi dobio obavijesti o novim objavama
- **Sudionici:** Baza podataka
- **Preduvjet:** UC4 (Prikazivanje tuđeg profila)
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju za praćenje
 2. Baza podataka ažurira popis praćenih korisnika

UC13 - Pozovi korisnika na video poziv

- **Glavni sudionik:** Registrirani korisnik (Pozivatelj)
- **Cilj:** Uspostava komunikacije između dva korisnika putem videopoziva
- **Sudionici:** Registrirani korisnik (Korisnik kojeg se zove)
- **Preduvjet:** UC4 (Prikazivanje tuđeg profila)
- **Opis osnovnog tijeka:**
 1. Pozivatelj odabire opciju za pokretanje videopoziva
 2. Korisnik kojeg se zove dobiva obavijest o pozivu
 3. Korisnik kojeg se zove prihvća poziv
 4. Korisnici razgovaraju
 5. Jedan od korisnika prekida poziv po završetku razgovora
- **Opis mogućih odstupanja:**
 - 3.a Korisnik kojeg se zove odbija pozivi ili se ne javlja
 1. Pozivatelj dobiva poruku da je korisnik kojeg zove nedostupan

UC14 - Pregled inboxa

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Pregled svih poruka i obavijesti

- **Sudionici:** Baza podataka
- **Preduvjet:** UC6 (Prijava)
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju za pregled inboxa
 2. Otvara se stranica s popisom svih poruka i obavijesti

UC15 - Odgovaranje na poruku

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Poslati odgovor na dobivenu poruku
- **Sudionici:** Baza podataka
- **Preduvjet:** UC14 (Otvaranje inboxa)
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju za otvaranje inboxa
 2. Korisnik odabire poruku na koju želi odgovoriti
 3. Odabire opciju za odgovor
 4. Otvara se polje za sastavljanje poruke
 5. Korisnik piše odgovor
 6. Kada je gotov, odabire opciju za slanje
 7. Poruka se upisuje u bazu podataka u inbox primatelja

UC16 - Ocijeni objavu

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Ocijeniti recept s ocjenom od 1 do 5
- **Sudionici:** Baza podataka
- **Preduvjet:** UC6 (Prijava)
- **Opis osnovnog tijeka:**
 1. Korisnik otvara recept koji želi ocijeniti
 2. Korisnik odabire ocjenu koju želi dati
 3. U bazi podataka se ažurira ocjena

UC17 - Spremi recept

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Spremiti recept kako bi ga se kasnije moglo lakše opet pronaći
- **Sudionici:** Baza podataka
- **Preduvjet:** UC6 (Prijava)
- **Opis osnovnog tijeka:**

1. Korisnik otvara recept koji želi spremiti
2. Odabire opciju za spremanje recepta
3. U bazi podataka se ažurira relacija spremanja u odnosu na korisnika i recepta

UC18 - Komentiraj objavljeni recept

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Ostaviti komentar na nekoj objavi
- **Sudionici:** Baza podataka
- **Preduvjet:** UC6 (Prijava)
- **Opis osnovnog tijeka:**
 1. Korisnik otvara objavljeni recept
 2. Korisnik otvara komentare na objavi
 3. Korisnik odabire opciju za pisanje komentara
 4. Otvara se polje za pisanje komentara
 5. Korisnik piše komentar i zatim odabire opciju za objavu
 6. Komentar se zapisuje u bazu podataka

UC19 - Pregled vlastitog profila

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Pregledati stanje vlastitog korisničkog profila
- **Sudionici:** Baza podataka
- **Preduvjet:** UC6 (Prijava)
- **Opis osnovnog tijeka:**
 1. Korisnik na početnoj stranici odabire svoju ikonu
 2. Otvara se stranica na kojoj je prikazan korisnikov profil i njegove objave

UC20 - Pregled spremljenih recepata

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Prikazati korisnikove najdraže recepte
- **Sudionici:** Baza podataka
- **Preduvjet:** UC19 (Pregledavanje vlastitog profila)
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju za pregled spremljenih recepata
 2. Odgovarajući recepti dohvaćaju se iz baze i prikazuju korisniku

UC21 - Pregled profila koje korisnik prati

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Pregled popisa profila koje korisnik prati
- **Sudionici:** Baza podataka
- **Preduvjet:** UC19 (Pregledavanje vlastitog profila)
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju za pregled profila koje prati
 2. Odgovarajući profili dohvaćaju se iz baze i prikazuju korisniku

UC22 - Pregled osobnih postavki

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Pregledavanje postavki korisničkog profila
- **Sudionici:** Baza podataka
- **Preduvjet:** UC19 (Pregledavanje vlastitog profila)
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju za pregled postavki profila
 2. Prikazuju se postavke profila

UC23 - Brisanje vlastitog profila

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Brisanje vlastitog korisničkog profila
- **Sudionici:** Baza podataka
- **Preduvjet:** UC22 (Pregled osobnih postavki)
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju za brisanje računa
 2. Profil se briše iz baze podataka, recepti i komentari ostaju, ali piše da ih je objavio obrisani korisnik

UC24 - Uređivanje osobnih postavki

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Promjena postavki profila
- **Sudionici:** Baza podataka
- **Preduvjet:** UC22 (Pregled osobnih postavki)
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju za uređivanje postavki
 2. Korisnik mijenja željene postavke
 3. Postavke se ažuriraju u bazi podataka

UC25 - Dodavanje novog administratora

- **Glavni sudionik:** Administrator
- **Cilj:** Dodati novog administratora
- **Sudionici:** Baza podataka
- **Preduvjet:** UC6 (Prijava)
- **Opis osnovnog tijeka:**
 1. Administrator odabire opciju stvaranja novog administratora
 2. Upisuje potrebne informacija za kreiranje admina, korisničko ime i lozinka
 3. Baza podataka se ažurira i dodaje novog administratora
- **Opis mogućih odstupanja:**

UC26 - Brisanje recepta

- **Glavni sudionik:** Administrator
- **Cilj:** Brisanje recepta koji nije prikladan ili slično
- **Sudionici:** Baza podataka
- **Preduvjet:** UC6 (Prijava)
- **Opis osnovnog tijeka:**
 1. Administrator odabire recept koji želi obrisati
 2. Odabire opciju za brisanje
 3. Recept, njegovi komentari i ostali povezani podatci brišu se iz baze podataka

UC27 - Brisanje korisnika

- **Glavni sudionik:** Administrator
- **Cilj:** Brisanje korisničkog profila
- **Sudionici:** Baza podataka
- **Preduvjet:** UC6 (Prijava)
- **Opis osnovnog tijeka:**
 1. Administrator odabire profil koji želi obrisati
 2. Odabire opciju za brisanje
 3. Korisniku se elektroničkom poštom šalje obavijest o brisanju
 4. Profil se briše iz baze podataka

UC28 - Pregled mogućnosti administratora

- **Glavni sudionik:** Administrator

- **Cilj:** Pristup svim administratorskim mogućnostima
- **Sudionici:** Baza podataka
- **Preduvjet:** UC6 (Prijava)
- **Opis osnovnog tijeka:**
 1. Administrator odabire opciju za prikaz administratorskih mogućnosti
 2. Prikazuje se pregled i mogućnosti za upravljanje bazom podataka

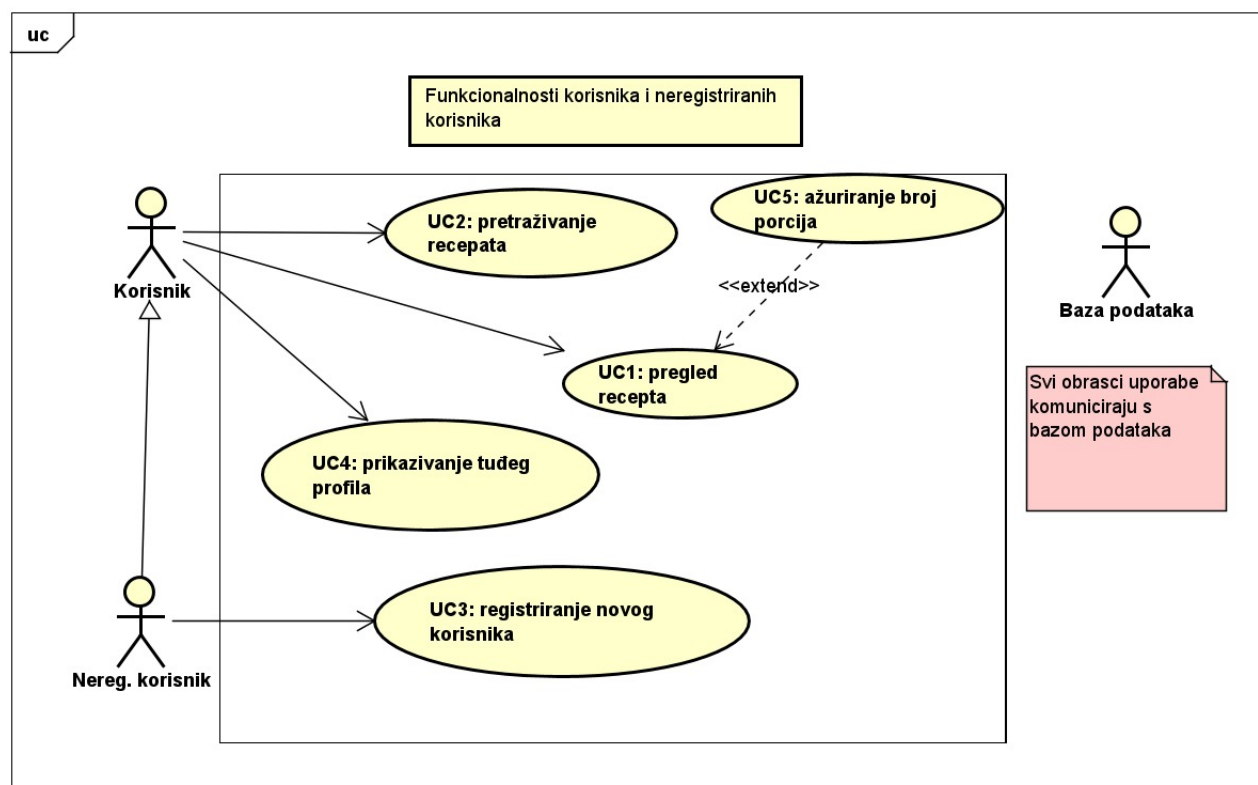
UC29 - Brisanje komentara s recepata

- **Glavni sudionik:** Administrator
- **Cilj:** Obrisati neprimjerene komentare s objava
- **Sudionici:** Baza podataka
- **Preduvjet:** UC6 (Prijava)
- **Opis osnovnog tijeka:**
 1. Administrator odabire komentar koji želi obrisati
 2. Odabire opciju za brisanje
 3. Korisniku se elektroničkom poštom šalje obavijest o brisanju njegovog napisanog komentara
 4. Komentar se briše iz baze podataka

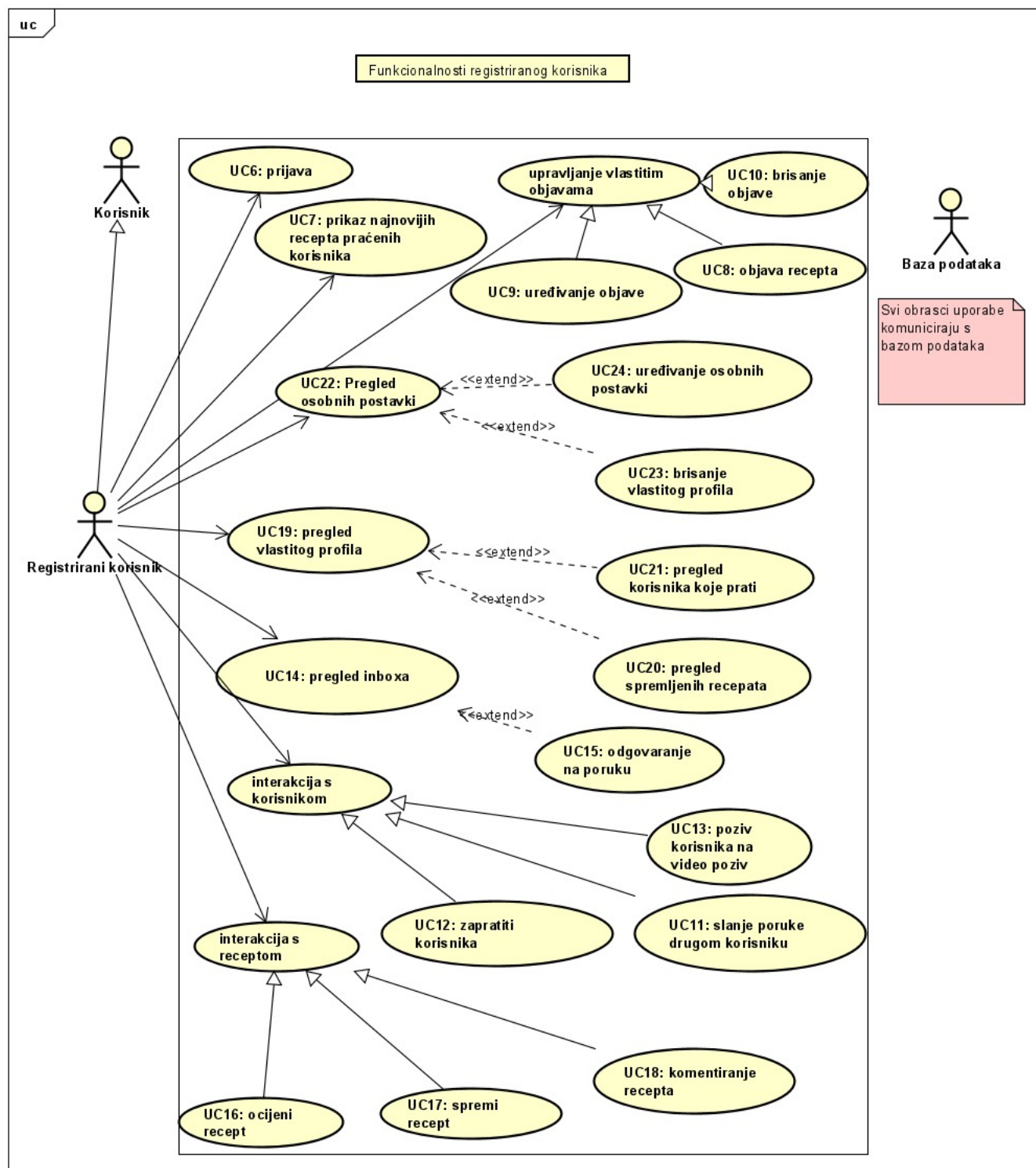
UC30 - Pregled statistike platforme

- **Glavni sudionik:** Administrator
- **Cilj:** Dobiti uvid u najbolji recept (po ocjeni), najaktivnije korisnike (po broju objava) i najpopularniji recept (po broju spremanja)
- **Sudionici:** Baza podataka
- **Preduvjet:** UC28 (Pregled mogućnosti administratora)
- **Opis osnovnog tijeka:**
 1. Administrator odabire opciju za pregled statistika
 2. Prikazuje se pregled

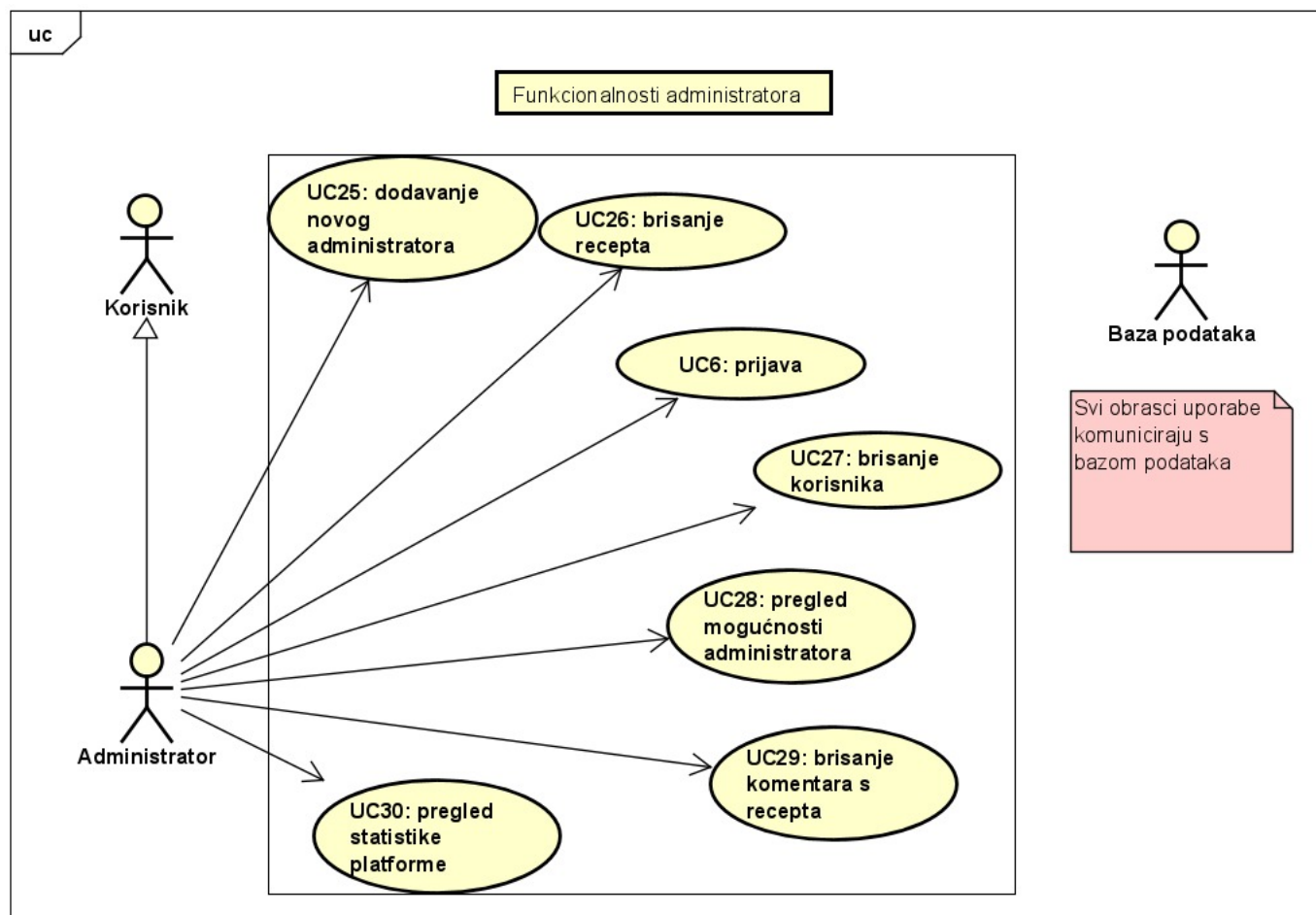
Dijagrami obrazaca uporabe



Slika 3.1: Dijagram obrasca uporabe, funkcionalnost korisnika i neregistriranog korisnika



Slika 3.2: Dijagram obrasca uporabe, funkcionalnost registriranog korisnika

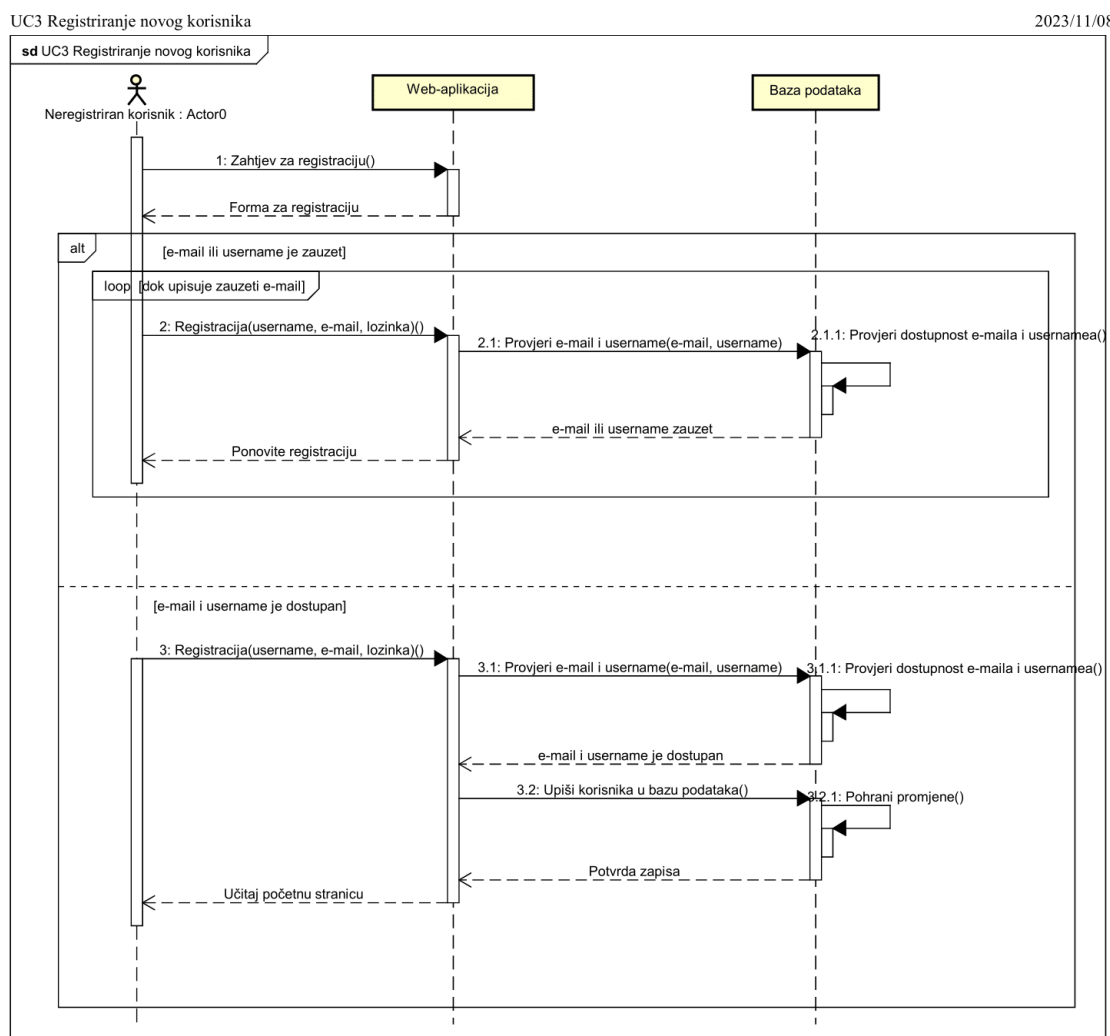


Slika 3.3: Dijagram obrasca uporabe, funkcionalnost administratora

3.1.2 Sekvencijski dijagrami

Obrazac uporabe UC3 - Registriranje novog korisnika

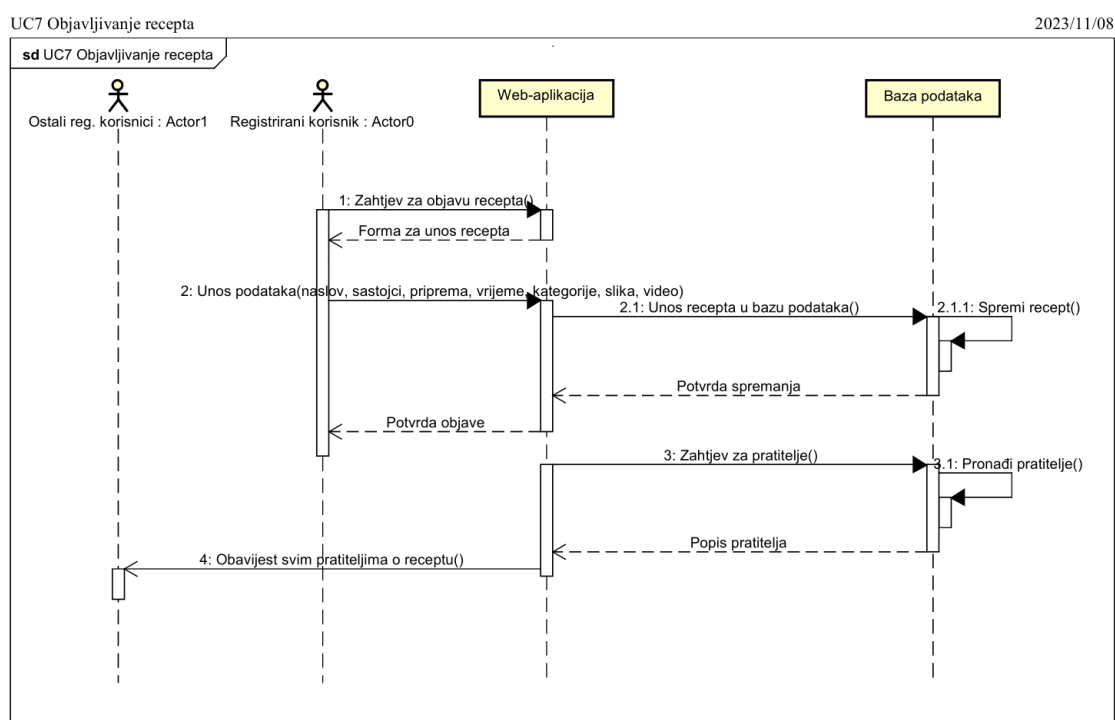
Neregistrirani korisnik šalje zahtjev za registraciju te mu web-aplikacija šalje odgovor formom za registraciju. Dokle god web-aplikacija u bazi podataka provjerom dobije odgovor da su email ili lozinka zauzeti korisnik dobiva poruku o zauzetosti emaila ili lozinke te ponovnu formu za upis podataka. Nakon unosa kod kojeg su email i username dostupni korisnikove informacije se zapisuju u bazu podataka. Korisnik dobiva poruku o uspješnoj registraciji.



Slika 3.4: Sekvencijski dijagram za UC3

Obrazac uporabe UC7 - Objava recepta

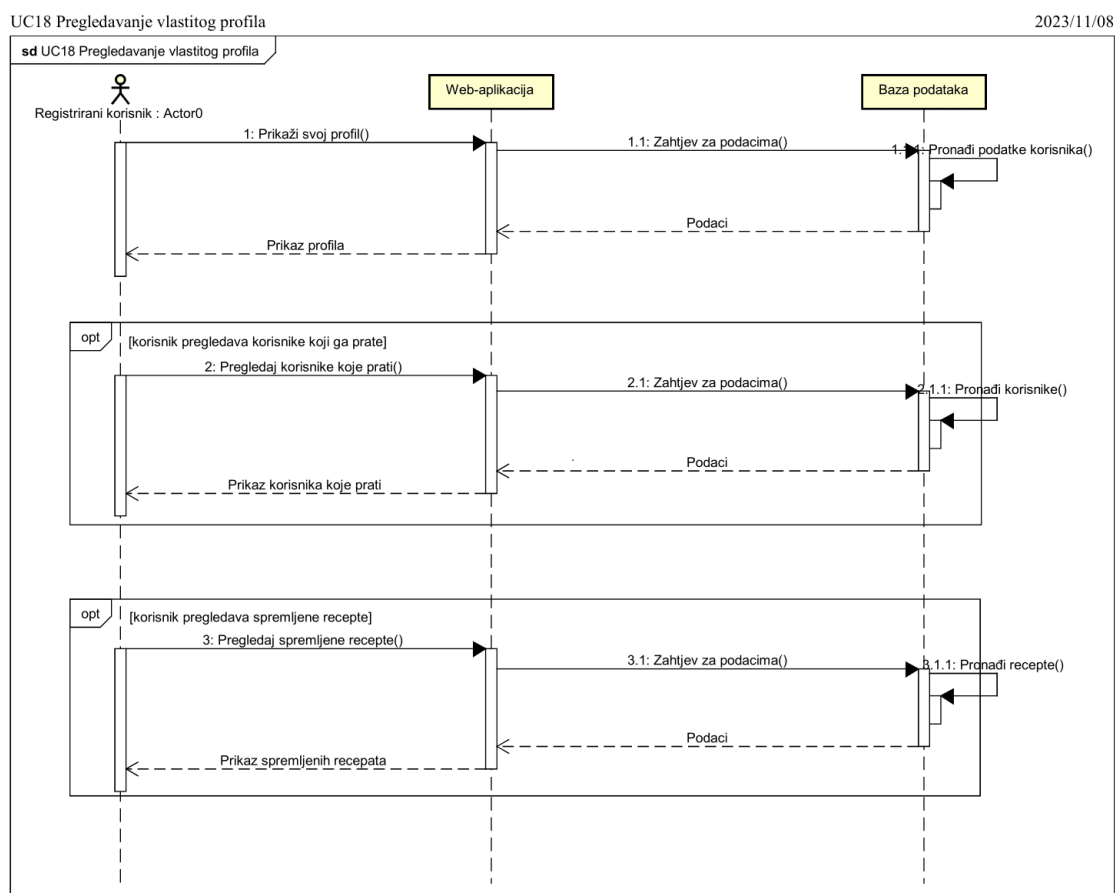
Registrirani korisnik šalje zahtjev za objavu recepta, a web-aplikacija mu odgovara formom za unos recepta. Registrirani korisnik unosi podatke o receptu: naslov, sastojci, priprema, vrijeme, kategorije, slike, video. Novonastali recept se zatim sprema u bazu podataka i web-aplikacija dobiva potvrdu o spremljenim podacima. Nakon potvrde web-aplikacija bazi podataka šalje zahtjev sa svim korisnicima koji su pratitelji autora objavljenog recepta te im svima šalje obavijest o novoj objavi.



Slika 3.5: Sekvencijski dijagram za UC7

Obrazac uporabe UC18 - Pregled vlastitog profila

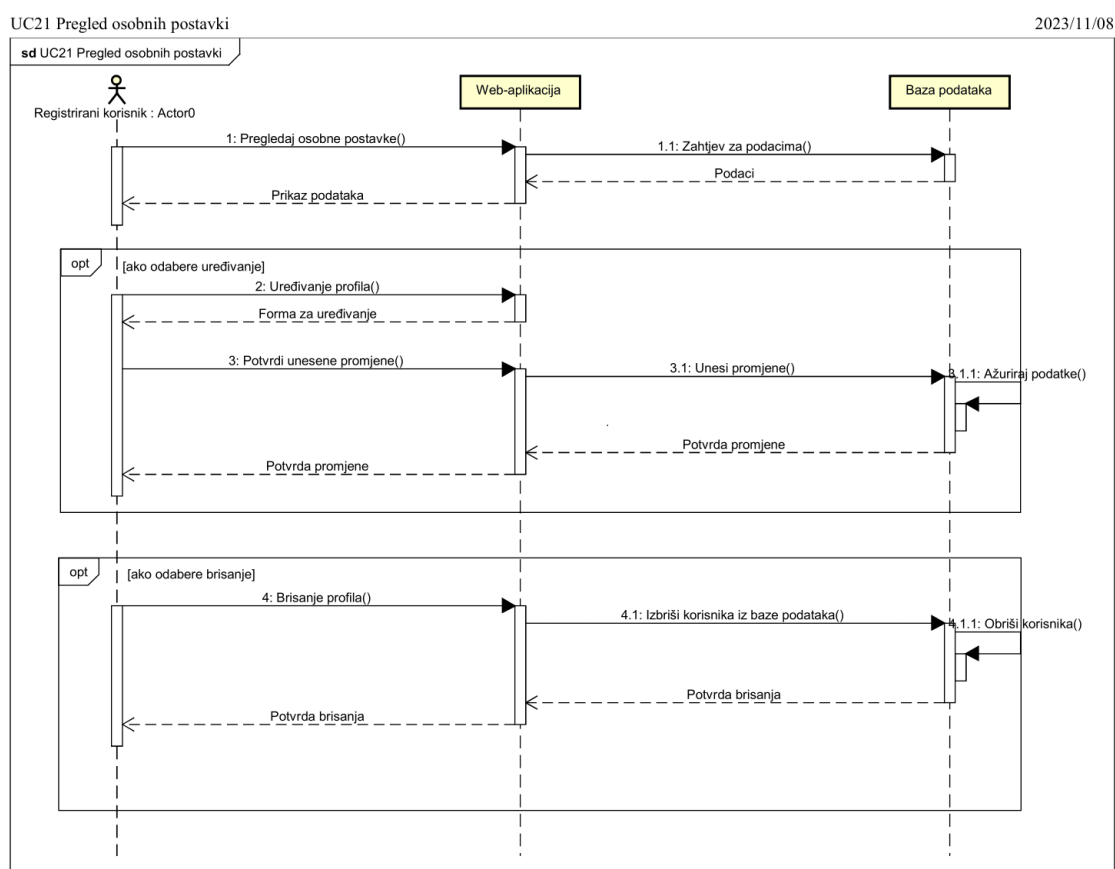
Registrirani korisnik šalje zahtjev za prikaz svog profila i web aplikaciju mu prikazuje sve informacije dobivene iz baze podataka. Nakon toga korisnik može tražiti popis svih korisnika koji ga prate (UC20 Pregled profila koje korisnik prati), a web-aplikaciju bi mu odgovorila popisom korisnika koju bi dobila slanjem zahtjeva bazi podataka. Korisnik može odabrati i opciju pregledavanja spremljenih recepata (UC19 Pregledavanje spremljenih recepata) koji se prikazuju tako što web-aplikacija dobiva popis spremljenih recepata iz baze podataka.



Slika 3.6: Sekvencijski dijagram za UC18

Obrazac uporabe UC21 - Pregled osobnih postavki

Registrirani korisnik šalje zahtjev za pregled osobnih postavki. Web-aplikacija prihvata zahtjev i pošalje upit bazi podataka koja šalje sve podatke korisnika koje može ažurirati/brisati. Korisnik može odabrati opciju uređivanja profila (UC23 Uređivanje osobnih postavki), a web-aplikacija mu šalje formu za uređivanje postavki. Nakon što korisnik upiše promjene, potvrda promjena se šalje web-aplikaciji koja prosljeđuje podatke bazi podataka koja za tog korisnika ažurira promjene. Druga opcija koju korisnik može odabrati je brisanje profila (UC22 Brisanje vlastitog profila). Korisnik šalje zahtjev za brisanje profila web-aplikaciji koja prosljeđuje podatke korisnika bazi podataka koja ga na kraju briše.



Slika 3.7: Sekvencijski dijagram za UC21

3.2 Ostali zahtjevi

- Aplikaciji se pristupa s pomoću internetskog preglednika korištenjem HTTPS protokola
- Aplikaciji treba moći pristupiti više korisnika istovremeno
- Korisničko sučelje mora biti intuitivno i jednostavno za korištenje
- Dizajn korisničkog sučelja mora biti responzivan kako bi se aplikaciji moglo pristupiti i s mobilnih uređaja
- Sustav mora biti otporan na greške uzrokovane neispravnim korištenjem korisničkog sučelja
- Postupak prijave korisnika mora biti siguran, a lozinke ne smiju biti pohranjene kao otvoreni tekst
- Baza podataka mora sadržavati odgovarajuće indekse kako bi pristup podacima bio brz
- Baza podataka mora biti odvojena od ostatka sustava, ne smije joj se moći pristupiti izravno, već samo preko back-enda aplikacije
- Korisničko sučelje i baza podataka moraju podržavati diakritičke znakove
- Pri implementaciji moraju biti korištena načela objektno-orijentiranog programiranja
- Razvojna verzija aplikacije mora biti u potpunosti odvojena od produkcijske kako rad na nadogradnjama ne bi utjecao na postojeće funkcionalnosti

4. Arhitektura i dizajn sustava

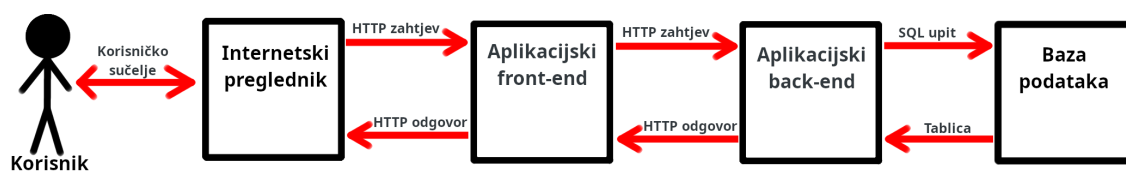
4.1 Organizacija sustava

4.1.1 Uvod

Arhitekturu sustava čine tri glavna dijela:

- Baza podataka
- Aplikacijski front-end
- Aplikacijski back-end

Front-end aplikacije ključan je dio u uspostavljanju komunikacije između korisnika i aplikacije. Korisnik aplikaciji pristupa uz pomoć internetskog preglednika na svom računalu, mobitelu ili nekom drugom uređaju. Preglednik komunicira s poslužiteljem preko HTTP protokola slanjem odgovarajućih zahtjeva. Front-end korisnikove zahtjeve zatim prosljeđuje back-endu koji predstavlja samu aplikaciju. Aplikacija preuzima zahtjev te ga obrađuje sukladno njegovoj vrsti i parametrima. Obrada zahtjeva uključuje i pristupanje bazi podataka kako bi se dohvatili podatci potrebni za rad. Po završetku obrade zahtjeva, aplikacija front-endu vraća odgovor, a on ga dalje šalje korisniku. Baza podataka može se nalaziti na istom računalu kao i ostali dijelovi sustava ili različitom, no komunikacija se uvijek odvija na isti način, preko dobro poznatih vrata na transportnom sloju i korištenjem odgovarajućeg protokola.



Slika 4.1: Arhitektura sustava

4.1.2 Sklopovski zahtjevi

Ako se aplikacijski poslužitelj i baza podataka nalaze na različitim računalima, za optimalan rad ona bi trebala imati odgovarajuće karakteristike. Računalo na kojem će raditi poslužitelj treba imati dovoljnu veliku procesorsku moć kako bi moglo što brže odgovarati na zahtjeve i kako bi više korisnika moglo koristiti aplikaciju bez značajnog usporavanja sustava. Računalo na kojem će se nalaziti baza podataka treba imati dovoljno velike i brze diskove za pohranu podataka, idealno uz neku vrstu zaštite od gubitka podataka u slučaju kvara (na primjer, korištenje RAID sustava ili automatskog redovitog stvaranja sigurnosne kopije). Ako se pak poslužitelj i baza podataka pokreću na istom računalu, ono treba kombinirati maloprije navedene karakteristike.

4.1.3 Organizacija aplikacije

Za izradu aplikacije odabrani su programski jezik Java uz razvojni okvir Spring te Javascript i razvojni okvir React. Dva glavna sloja aplikacije su frontend, koji komunicira s korisnikom (Javascript + React) i backend, koji obrađuje HTTP zahtjeve i komunicira s bazom podataka (Java + Spring).

4.2 Baza podataka

Naš će sustav koristiti relacijsku bazu podataka čija je osnovna jedinica baze relacija, odnosno tablica, koja je definirana svojim imenom i skupom atributa. Baza podataka ima zadatak brzo i jednostavno pohranjivati, mijenjati i dohvaćati podatke za daljnju obradu. Baza podataka stranice sastoji se od sljedećih entiteta:

- Korisnik
- Recept
- Oznaka
- Komentar
- Ocjena
- Poruka
- Sastojak

4.2.1 Opis tablica

Korisnik Ovaj entitet sadržava sve važne informacije o korisniku aplikacije. Sadrži attribute: Korisničko ime, lozinku, ime, prezime, razinu ovlasti korisnika i e-mail korisnika. Ovaj entitet u vezi je *One-to-Many* s entitetima Recept, Komentar i Ocjena te je s entitetom poruka u ulozi primatelj i pošiljatelj isto u *One-to-Many* vezi, sve preko ID-a korisnika.

Korisnik		
ID Korisnika	INT	jedinstveni identifikator korisnika
Lozinka	VARCHAR	lozinka korisnika
Korisničko ime	VARCHAR	korisničko ime
Ime	VARCHAR	ime korisnika
Prezime	VARCHAR	prezime korisnika
Razina Ovlasti	VARCHAR	razina ovlasti korisnika
Email	VARCHAR	e-mail adresa korisnika

Pretplata Ovaj entitet sadrži informacije vezane za praćenje korisnika. Njegovi atributi su: ID korisnika kojeg se prati (praćenik), ID pratitelja i ID pretplate. Entitet je u *Many-to-Many* vezi s korisnikom preko ID-a pratitelja i praćenika.

Pretplata		
IDpretplata	INT	jedinstveni identifikator pretplate
ID pratitelj	INT	ID korisnika koji prati
ID praćenik	INT	ID korisnika kojeg se prati

Recept Ovaj entitet sadržava sve važne informacije o receptu. Sadrži attribute: ID recepta, naziv recepta, vrijeme pripreme, postupak pripreme, opis recepta, sliku recepta, datum recepta i vrijeme objave recepta i prosječnu ocjenu recepta. Recept je u vezi *Many-to-One* s korisnikom koji ga je objavio, u vezi *One-to-Many* s ocjenom, sastojkom u receptu i komentarom preko ID-a recepta te u *Many-to-Many* s oznakom recepta preko ID-a oznake.

Recept		
ID Recepta	INT	jedinstveni identifikator recepta
Naziv recepta	VARCHAR	naziv recepta
Vrijeme pripreme	INTERVAL	vrijeme pripreme
Postupak pripreme	VARCHAR	postupak pripreme
Opis recepta	VARCHAR	opis recepta
Slika recepta	LONGBOB	slika recepta
Datum i vrijeme recepta	DATETIME	datum recepta i vrijeme objave recepta
Prosječna ocjena recepta	INT	prosječna ocjena recepta
ID Korisnika	INT	ID korisnika koji je objavio recept
ID oznake	INT	ID oznake recepta

Oznaka Ovaj entitet sadržava sve važne informacije o oznaci recepta. Sadrži attribute: ID oznake i naziv oznake. Oznaka je u *Many-to-Many* vezi s receptom preko ID-a oznake.

Oznaka		
ID oznake	INT	jedinstveni identifikator oznake
Naziv oznake	VARCHAR	naziv oznake

Komentar Ovaj entitet sadržava sve važne informacije o komentaru recepta. Sadrži attribute: ID komentara, ID komentatora, ID recepta, tekst komentara i datum i vrijeme komentara. Komentar je u *Many-to-One* vezi s korisnikom preko ID-a korisnika koji ga objavi i receptom preko ID-a recepta.

Komentar		
ID komentar	INT	ID komentara

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Komentar		
Tekst komentara	VARCHAR	tekst komentara
ID komentatora	INT	ID korisnika komentatora
ID recepta	INT	ID recepta na kojem je komentar postavljen
Datum i vrijeme komentara	DATETIME	datum i vrijeme komentara

Ocjena Ovaj entitet sadržava sve važne informacije o pojedinoj ocjeni recepta. Sadrži attribute: ID ocjenitelja, ID recepta i datum i vrijeme ocjene. Entitet je u vezi *Many-to-One* s ocjeniteljem preko ID-a korisnika koji je ocijenio recept i *Many-to-One* s receptom preko ID-a recepta.

Ocjena		
Ocjena	VARCHAR	ocjena
ID ocjenitelja	INT	ID korisnika koji je ostavio poruku
ID recepta	INT	ID recepta koji je ocjenjen
Datum i vrijeme ocjene	DATETIME	datum i vrijeme ocjene

Poruka Ovaj entitet sadržava sve važne informacije o poruci između dva korisnika. Sadrži attribute: ID poruke, ID pošiljatelja, ID primatelja, tekst poruke i datum i vrijeme poruke. Poruka je u vezi *Many-to-One* s primateljem i pošiljateljem preko ID-a korisnika koji šalje i prima poruku.

Poruka		
ID Poruka	INT	ID poruka
Tekst poruka	VARCHAR	tekst poruka
ID pošiljatelja	INT	ID korisnika koji šalje poruku
ID primatelja	INT	ID korisnika koji prima poruku

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Poruka		
Datum i vrijeme poruke	TIMESTAMP	datum i vrijeme poruke

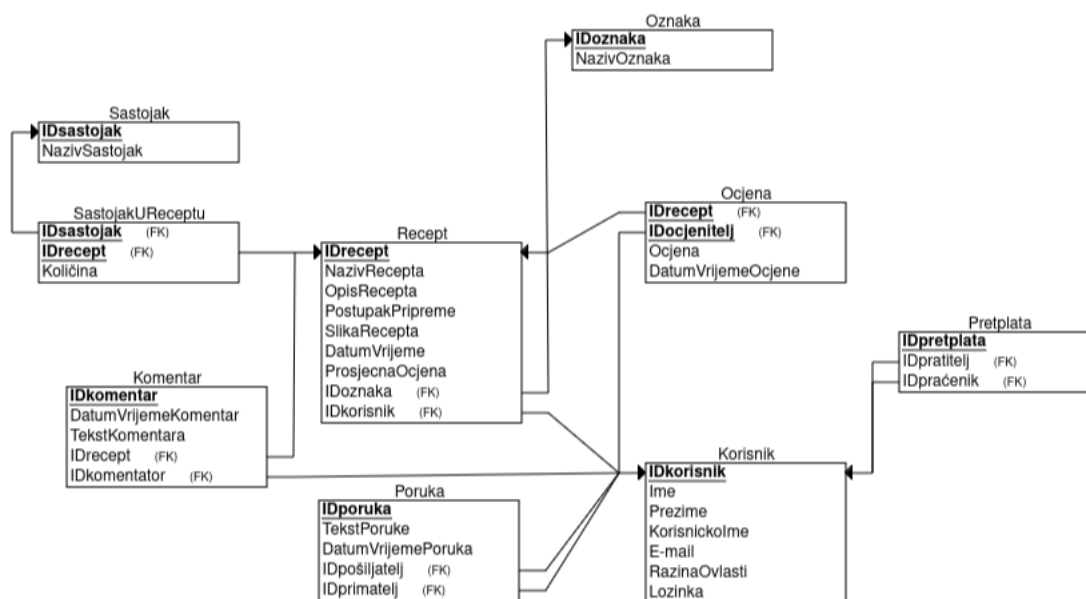
Sastojak Ovaj entitet sadržava sve važne informacije o sastojku navedenom u nekom receptu. Sadrži attribute: ID sastojka, naziv sastojka. Entitet je u vezi *Many-to-Many* sa sastojkom u receptu preko ID-a recepta.

Sastojak		
ID Sastojka	INT	ID sastojka
Naziv sastojka	VARCHAR	naziv sastojka

Sastojak u receptu Ovaj entitet sadržava sve važne informacije o sastojku navedenom u nekom receptu. Sadrži attribute: ID sastojka, ID recepta, količinu. Entitet je u vezi *Many-to-One* s receptom preko ID-a recepta i u vezi *Many-to-Many* sa sastojkom preko ID sastojak.

Sastojak u receptu		
ID Sastojka	INT	ID sastojka
ID recepta	INT	ID recepta u kojem je sastojak
Količina	VARCHAR	količina

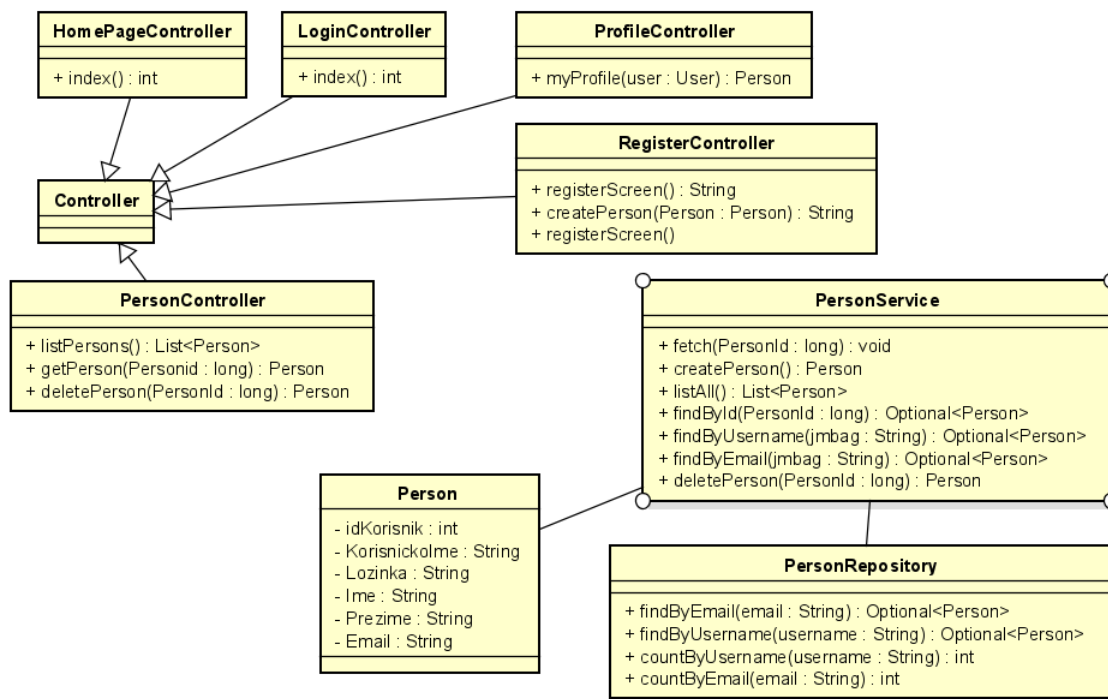
4.2.2 Dijagram baze podataka



Slika 4.2: Relacijski dijagram baze podataka

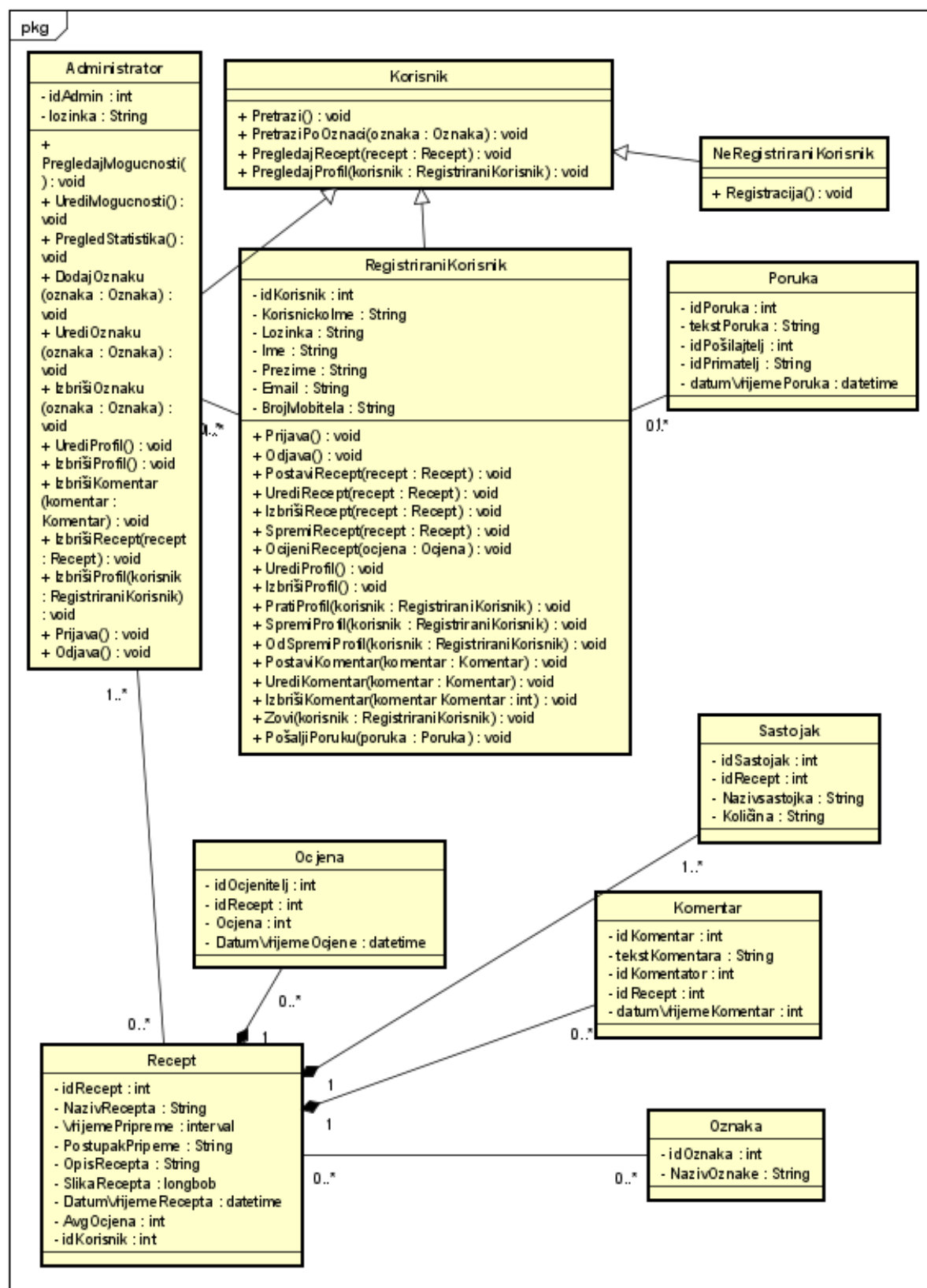
4.3 Dijagram razreda

Prvi dijagram prikazuje trenutno stanje stranice. Dijagram ima jedan repozitorij, entitet i servis te 5 kontrolera.



Slika 4.3: Dijagram razreda - dio Controllers

Drugi dijagram prikazuje glavne entitete i veze između njih, tj. preslikava strukturu baze podataka. Razred Korisnik predstavlja neregistriranog korisnika koji se može registrirati u Razred RegistriraniKorisnik predstavlja korisnika koji je registriran u sustav i koji može koristiti više od osnovnih funkcija korisnika. NeRegistriraniKorisnik predstavlja vrstu korisnika i može se registrirati. Razred Administrator predstavlja administratora sustava koji ima najveće ovlasti.

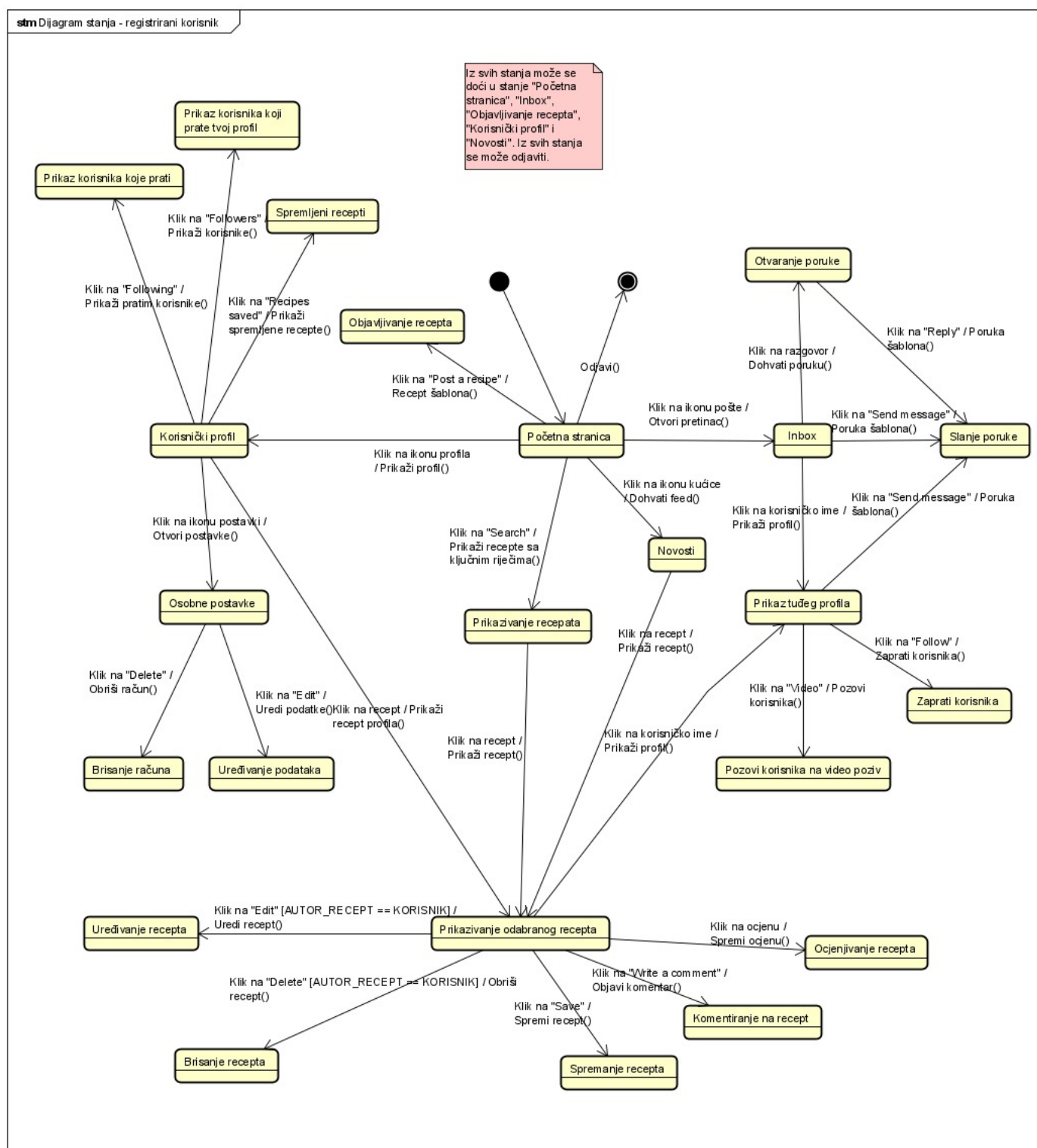


Slika 4.4: Dijagram razreda - dio Data transfer objects

4.4 Dijagram stanja

Dijagram stanja na slici 4.5 opisuje stanja u kojima se registrirani korisnik može naći i njihove okidače te prijelaze. Nakon uspješne prijave, korisniku se prikazuje početna stranica na kojoj može pretražiti recepte pomoću tražilice, pregledati svoj pretinac poruka, pregledati svoj profil, objaviti recept ili pogledati nove recepte koje su objavili korisnici koje prati. Na vlastitom profilu se prikazuju osobni podaci, objavljeni recepti i gumbi za dodatne mogućnosti. Dodatne funkcionalnosti na stranici profila su opcije za daljnji pregled osobnih postavki, prikaz spremjenih recepata, prikaz korisnika koje prati te prikaz korisnika koji prate tvoj profil. Klikom na "Postavke" korisnik može ažurirati svoje podatke ili obrisati svoj račun. Klikom na "Pretinac poruka" prikazuju se izmijenjene poruke. U pretincu se može otvoriti poruka njezinim odabirom te poslati nova poruka, čime se pritiskom gumba otvara šablona za poruku.

Odabirom opcije "Objavi recept" otvara se nova stranica na kojoj se nalazi šablona za stvaranje recepta. Upisivanjem ključnih riječi u tražilicu prikaže se stranica s receptima koji se poklapaju s danim opisom. Odabirom recepta otvara se pojedinačna stranica sa samo odabranim receptom. Na stranici pregleda recepta korisnik može ocijeniti, spremi ili komentirati recept, u slučaju da je korisnik autor prikazanog recepta pojavljuje se opcija uređivanja ili brisanja recepta. Korisnik također može pritisnuti na korisničko ime korisnika koji je objavio recept, time se otvara stranica tuđeg profila. Prikazivanjem tuđeg profila moguće je zapratiti korisnika, poslati privatnu poruku korisniku, čime se ponovno otvara šablona za poruku ili pozvati korisnika na video poziv. Korisnik uvijek ima mogućnost povratka na početnu stranicu, otvaranja pretinca poruka, pregleda korisničkog profila, pregleda novosti i objavljivanja recepta.

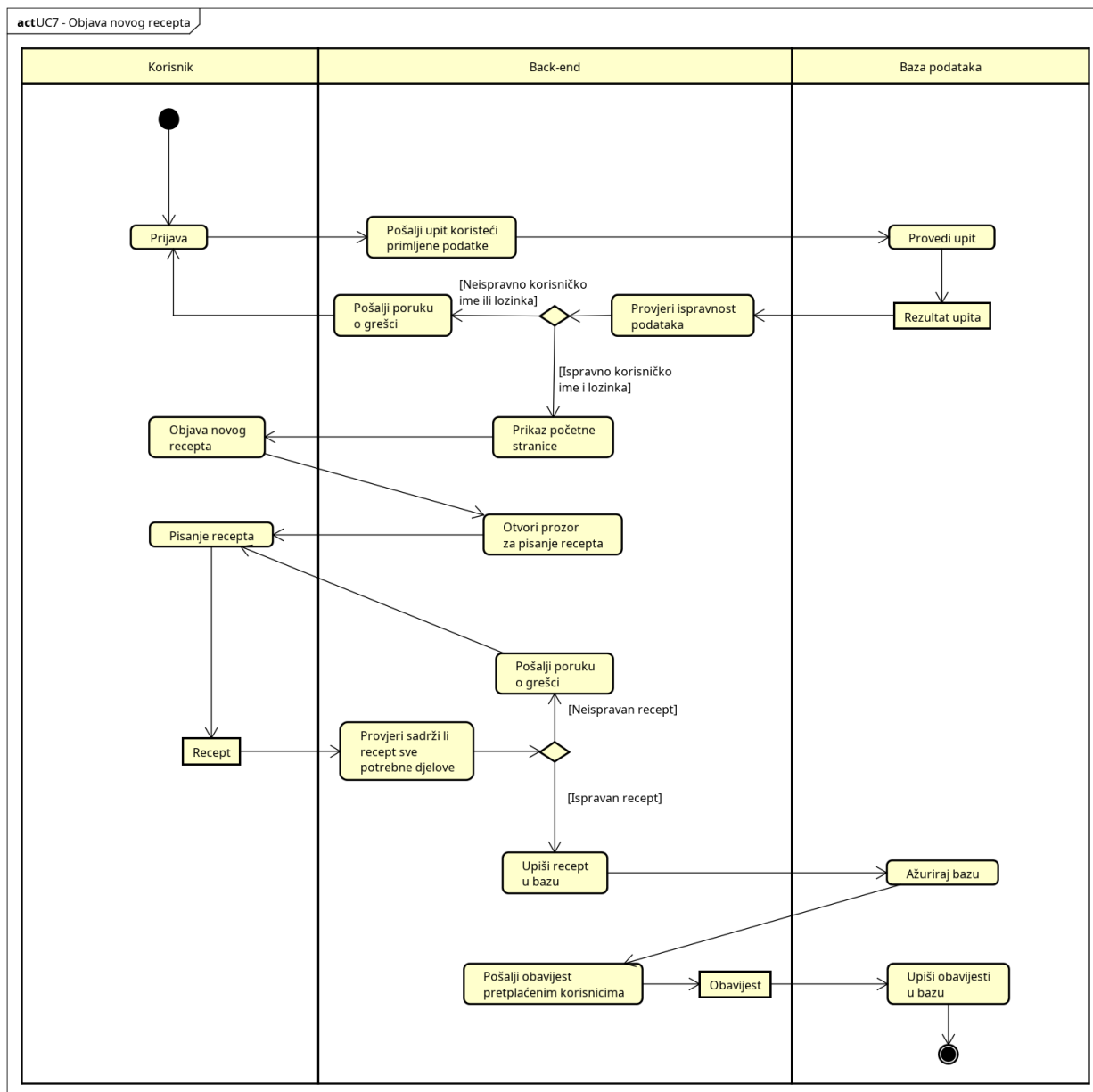


Slika 4.5: Dijagram stanja - registrirani korisnik

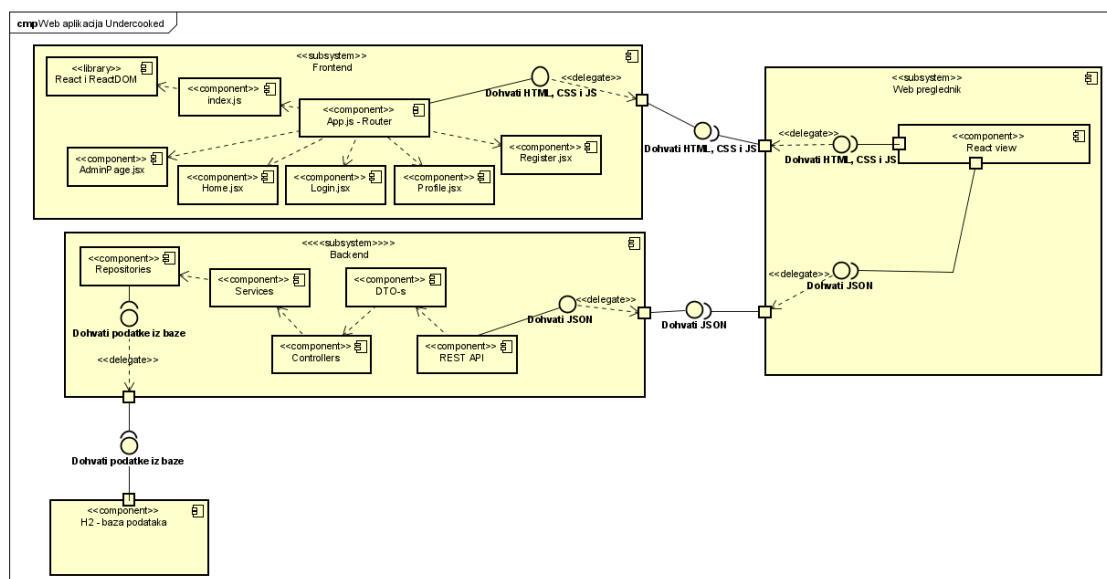
4.5 Dijagram aktivnosti

Dijagram aktivnosti u nastavku prikazuje tok upravljanja jedne od najvažnijih funkcionalnosti stranice, a to je objava novog recepta. Dva glavna sudionika su korisnik koji objavljuje recept te baza podataka u koju se zapisuje, a cijelim postupkom upravlja sama aplikacija, odnosno back-end. Dijagram je sukladno tome podijeljen na tri particije.

Korisnik se najprije mora prijaviti u aplikaciju. Nakon uspješne prijave prikazuje mu se početna stranica s koje korisnik može odabrati opciju za objavu novog recepta. Odabirom te opcije otvara se novi prozor te korisnik može započeti s pisanjem recepta. Nakon što je korisnik gotov s pisanjem, back-end provjerava sadrži li recept sve potrebne podatke. Ako je recept valjan, upisuje ga se u bazu podataka te se na kraju korisnicima koji prate autora u obliku poruke šalje obavijest o novoj objavi.



Slika 4.6: Dijagram aktivnosti objave novog recepta



Slika 4.7: Dijagram komponenti

4.6 Dijagram komponenti

Slika 4.7 pruža prikaz organizacije komponenata, internih struktura, njihovih međusobnih odnosa te njihovih veza s okolinom. Spring Boot korišten pri izradi aplikacije, koristi se za brzu i lagano pokretljivu izgradnju servera koji obrađuje HTTP zahtjeve putem definiranih REST API endpointa. Preko sučelja za dohvat HTML, CSS i JS datoteka poslužuju se datoteke koje pripadaju frontend dijelu aplikacije. App.js, tj. Router omogućava navigaciju između različitih stranica, Home, Login, Register, Profile i AdminPage. To je komponenta koja na upit s url određuje koja datoteka će se poslužiti na sučelje.

Na korisnički zahtjev Korisnik interaktira s frontend-om, pokrećući HTTP zahtjeve prema backend API-ju. API endpointi u kontrolerima backend-a prihvataju i vraćaju DTO-ove. DTO-ovi se koriste kako bi podatke omotali na način koji ima smisla za frontend, ne izlažući pritom unutarnji podatkovni model backend-a. Repozitoriji omogućavaju interakciju s H2 in-memory bazom podataka. Ona se koristi za pohranu podataka tijekom trajanja aplikacije. Automatski se stvara prilikom pokretanja backend-a i nestaje nakon gašenja. Dakle, backend prima zahtjev, obrađuje ga putem odgovarajućih kontrolera i servisa i ako je potrebno komunicira s H2 bazom podataka putem Repozitorija. Backend onda generira odgovor u JSON formatu i šalje ga frontend-u koji ažurira svoje stanje i reagira na promjene korisničkog sučelja.

5. Implementacija i korisničko sučelje

5.1 Korištene tehnologije i alati

Komunikacija grupe se odvijala putem aplikacije WhatsApp¹, dok je za održavanje sastanaka na daljinu korištena aplikacija MicrosoftTeams², koja pruža opciju video poziva. Sustav Git³ je omogućio upravljanje različitim verzijama programskog koda i dokumentacije uz udaljeni repozitorij na web platformi GitHub⁴. Dokumentacija je pisana u programskom jeziku LaTeX⁵, a za izradu UML dijagrama korišteni su alati ERDPlus⁶ i Astah UML⁷ sa studentskom licencom.

Korištena je kombinacija razvojnih okruženja ovisno o developeru. Aplikacija je napisana koristeći Eclipse IDE⁸ i IntelliJ IDEA⁹, dok je za pisanje dokumentacije korišteno razvojno okruženje Visual Studio Code¹⁰. Eclipse je *open-source* integrirano razvojno okruženje koje se dominantno koristi za razvoj Java aplikacija uz Java razvojne alate. Moguće je prilagoditi okruženje za razvoj web-stranica, web-aplikacija i mobilnih aplikacija uz alate za razvoj programskih aplikacija. IntelliJ IDEA je integrirano razvojno okruženje tvrtke JetBrains za razvoj računalnih programa u Javi, Kotlinu te drugim programskim jezicima koji se oslanjaju na Java virtualni stroj. Sadrži brojne značajke koje znatno olakšavaju pisanje aplikacija kao što su posebni alati za radne okvire Spring i Spring Boot, pametno popunjavanje koda i podrška pri radu s HTTP porukama. Visual Studio Code je integrirano razvojno okruženje tvrtke Microsoft koje se koristi za razvoj računalnih programa u brojnim računalnim jezicima poput Python, C/C++ i Java. Sva korištena razvojna okruženja su podržana na operacijskim sustavima Microsoft, Linux i macOS.

¹<https://www.whatsapp.com/>

²<https://www.microsoft.com/en-us/microsoft-teams/group-chat-software/>

³<https://git-scm.com/>

⁴<https://github.com/>

⁵<https://www.latex-project.org/>

⁶<https://erdplus.com/>

⁷<https://astah.net/products/astah-uml/>

⁸<https://eclipseide.org/>

⁹<https://www.jetbrains.com/idea/>

¹⁰<https://code.visualstudio.com/>

Aplikacija je napravljena u radnom okviru Spring Boot¹¹ kao Maven¹² projekt. Za izradu *backenda* aplikacije korišten je programski jezik Java¹³, a za izradu *frontenda* je korišten React¹⁴ i programski jezik JavaScript¹⁵. Spring Boot je specijalizacija radnog okvira Spring koji se temelji na višeslojnoj arhitekturi čiji je cilj ubrzati i pojednostaviti razvoj web-aplikacija. Spring boot pruža podršku za automatsko definiranje programskih zahtjeva prilikom stvaranja projekata, za ugrađeni server Tomcat i za bazu podataka. Maven je programska podrška za organizaciju projekata i automatizaciju pokretanja aplikacija. Koristimo *in-memory* H2¹⁶ bazu podataka koju pruža Spring Boot. React je biblioteka u JavaScriptu za izgradnju korisničkih sučelja temeljenih na komponentama koju održava Facebook. Glavne značajke React-a su ponovno korištenje web komponenti i ponovno prikazivanje samo komponente koje su promijenjene, a ne cijelu stranicu. Uspostava sobe za video poziv je omogućena preko Daily¹⁷ WebRTC usluga.

¹¹<https://spring.io/projects/spring-boot/>

¹²<https://maven.apache.org/>

¹³<https://www.java.com/en/>

¹⁴<https://react.dev/>

¹⁵<https://www.javascript.com/>

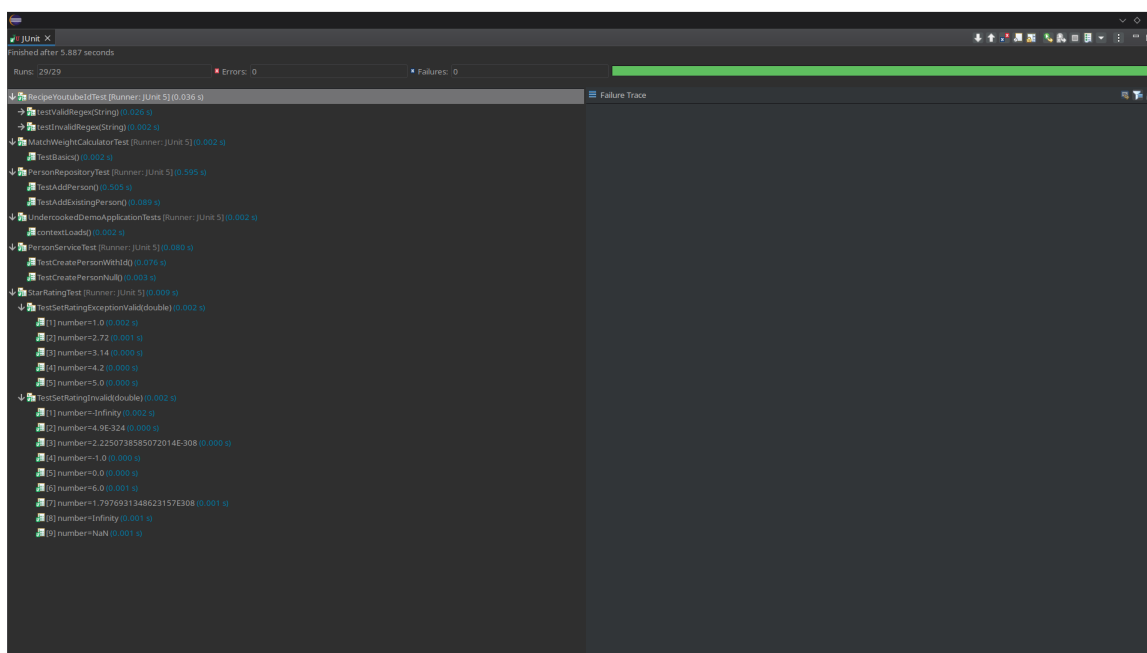
¹⁶<https://www.h2database.com/html/main.html>

¹⁷<https://www.daily.co/>

5.2 Ispitivanje programskog rješenja

5.2.1 Ispitivanje komponenti

Ispitivanje jedinica provedeno je uz pomoć okvira za ispitivanje JUnit 5¹⁸. Provedeno je ispitivanje za važne elemente klase SearchService, PersonService, Recipe i StarRating te sučelja PersonRepository. Rezultat pokretanja testova prikazan je na slici, a sami testovi opisani su u nastavku.



Slika 5.1: Rezultati unit testova

Ispitivanje klase SearchService

Klasa SearchService omogućava pretraživanje recepata prema ključnim riječima naslova i opisa. Za ispravan rad pretraživanja važna je ugniježđena klasa MatchWeightCalculator koja računa sličnost između argumenata pretrage i rezultata. Metoda pow kao parametar uzima decimalni broj i vraća decimalni broj koji asimptotski raste prema dva za veću vrijednost parametra.

```
public class MatchWeightCalculatorTest {
    @Test
```

¹⁸<https://junit.org/junit5/>

```
public void TestBasics() {
    Assert.isTrue(SearchService.MatchWeightCalculator.pow(0) == 0, "");
    // Nula u nuli,
    Assert.isTrue(SearchService.MatchWeightCalculator.pow(1) == 1, "");
    // jedan u jedan,
    Assert.isTrue(SearchService.MatchWeightCalculator.pow(2) == 1.5,
        ""); // raste
    Assert.isTrue(SearchService.MatchWeightCalculator.pow(3) == 1.75,
        ""); // i raste
    Assert.isTrue(SearchService.MatchWeightCalculator.pow(Double.POSITIVE_INFINITY)
        == 2, ""); // asimptotski prema dva.
}
}
```

Ispitivanje klase PersonService

Klasa PersonService služi za upravljanje podacima vezanim uz korisnike stranice. Ispitano je kako se sustav ponaša pri neispravnom dodavanju osobe u bazu podataka.

```
@SpringBootTest
```

```
@AutoConfigureTestDatabase(connection = EmbeddedDatabaseConnection.H2)
```

```
public class PersonServiceTest {
```

```
    @Autowired
```

```
    PersonService personService;
```

```
    @Test
```

```
    public void TestCreatePersonNull() {
```

```
        Person person = null;
```

```
        String expected = "Person object must be given";
```

```
        Exception e = assertThrows(IllegalArgumentException.class,
```

```
            () -> personService.createPerson(person));
```

```
        assertEquals(expected, e.getMessage());
```

```
    }
```

```
    @Test
```

```
    public void TestCreatePersonWithId() {
```

```
Person person = new Person();
person.setUsername("RMS");
person.setEmail("rms@gnu.org");
person.setName("Richard");
person.setSurname("Stallman");
person.setPassword(new
    BCryptPasswordEncoder().encode("superSecretPassword123"));
person.setAdmin(false);
person.setId((long) 1337);
String expected = "Person ID must be null, not: 1337";

Exception e = assertThrows(IllegalArgumentException.class,
    () -> personService.createPerson(person));
assertEquals(expected, e.getMessage());
}
}
```

Ispitivanje klase Recipe

Klasa Recipe služi za modeliranje sadržaja recepta. Metoda koja se ovdje testira služi za provjeru je li ID YouTube videa povezanog s receptom valjan.

```
public class RecipeYoutubeIdTest {
    private Recipe recipe;

    @BeforeEach
    public void beforeEach() {
        this.recipe = new Recipe();
    }

    @ParameterizedTest
    @NullSource
    @ValueSource(strings = {"gocwRvLhDf8", "dQw4w9WgXcQ", "_POWKV-6G9M"})
    public void testValidRegex(String youtubeId) {
        recipe.setYoutubeEmbedId(youtubeId);
        Assertions.assertEquals(youtubeId, recipe.getYoutubeEmbedId());
    }

    @ParameterizedTest
```

```
@ValueSource(strings = {
    "https://www.youtube.com/watch?v=dQw4w9WgXcQ",
    "https://youtu.be/dQw4w9WgXcQ",
    "sm3504435",
    ""
})
public void testInvalidRegex(String invalidId) {
    Assertions.assertThrows(IllegalArgumentException.class, () ->
        recipe.setYoutubeEmbedId(invalidId));
}
```

Ispitivanje klase StarRating

Klasa StarRating služi za modeliranje ocjene recepta. Metoda koja se ovdje testira provjerava je li dana ocjena između 1 i 5.

```
public class StarRatingTest {

    private StarRating rating;

    @BeforeEach
    public void setup() {
        this.rating = new StarRating();
    }

    @ParameterizedTest
    @ValueSource(doubles = {Double.NEGATIVE_INFINITY, Double.MIN_VALUE,
        Double.MIN_NORMAL, -1, 0, 6, Double.MAX_VALUE,
        Double.POSITIVE_INFINITY, Double.NaN})
    public void TestSetRatingInvalid(double number) {
        String expected = "Invalid rating: " + number + ". Must be in range 1-5.";

        Throwable exception = assertThrows(IllegalArgumentException.class,
            () -> rating.setRating(number)
        );
        assertEquals(expected, exception.getMessage());
    }
}
```

```
@ParameterizedTest
@ValueSource(doubles = {1, 2.72, 3.14, 4.20, 5})
public void TestSetRatingExceptionValid(double number) {
    assertDoesNotThrow(() -> rating.setRating(number));
}
}
```

Ispitivanje sučelja PersonRepository

Sučelje PersonRepository modelira sadržaj baze podataka vezan uz korisnike stranice. Testirano je dodavanje nove osobe, a nakon toga dodavanje osobe koja je već upisana u bazu podataka.

```
@SpringBootTest
@AutoConfigureTestDatabase(connection = EmbeddedDatabaseConnection.H2)
public class PersonRepositoryTest {

    @Autowired
    private PersonRepository personRepository;

    @Test
    public void TestAddPerson() {

        Person person = new Person();
        person.setUsername("RMS");
        person.setEmail("rms@gnu.org");
        person.setName("Richard");
        person.setSurname("Stallman");
        person.setPassword(new
            BCryptPasswordEncoder().encode("superSecretPassword123"));
        person.setAdmin(false);

        Person savedPerson = personRepository.save(person);

        assertEquals(person.getUsername(), savedPerson.getUsername());
        assertNotNull(savedPerson.getId());
    }
}
```

```
@Test
public void TestAddExistingPerson() {
    //Vazno je da se ovaj test pokrene nakon prethodnog kako bi se ista
    osoba dvaput pokusala dodati u bazu
    Person person = new Person();
    person.setUsername("RMS");
    person.setEmail("rms@gnu.org");
    person.setName("Richard");
    person.setSurname("Stallman");
    person.setPassword(new
        BCryptPasswordEncoder().encode("superSecretPassword123"));
    person.setAdmin(false);

    assertThrows(DataIntegrityViolationException.class,
        () -> personRepository.save(person)
    );
}
```

5.2.2 Ispitivanje sustava

Svi testovi provedeni su automatski uz pomoć radnog okvira *Selenium*¹⁹. Ispitane su neke od ključnih funkcionalnosti sustava:

- UC6 - Prijava
- UC8 - Objava recepta
- UC16 - Ocjenjivanje objave

Ispitni slučaj 1: Prijava s ispravnim korisničkim imenom i lozinkom

Ulaz:

1. Otvaranje stranice za prijavu u web pregledniku
2. Unos ispravnog korisničkog imena i lozinke
3. Kliktanje na gumb za prijavu

Očekivani rezultat: Korisnika se prebacuje na stranicu njegovog profila (*/profile/[korisničko ime]*).

¹⁹<https://www.seleniumhq.org/>

Rezultat: Očekivani rezultat je zadovoljen čime je aplikacija prošla ovaj test.

Ispitni slučaj 2: Prijava s ispravnim korisničkim imenom i neispravnom lozinkom

Ulaz:

1. Otvaranje stranice za prijavu u web pregledniku
2. Unos ispravnog korisničkog imena i neispravne lozinke
3. Kliktanje na gumb za prijavu

Očekivani rezultat: Korisnika se ostavlja na stranici za prijavu (*/login*) i šalje se poruka o neispravnom korisničkom imenu ili lozinki.

Rezultat: Očekivani rezultat je zadovoljen čime je aplikacija prošla ovaj test.

Ispitni slučaj 3: Ocjenjivanje recepta uz ispravnu prijavu

Ulaz:

1. Otvaranje stranice za prijavu u web pregledniku
2. Unos ispravnog korisničkog imena i neispravne lozinke
3. Kliktanje na gumb za prijavu
4. Navigiranje do stranice nekog recepta, u ovom slučaju */recipe/1*
5. Kliktanje na gumb ocjenu recepta

Očekivani rezultat: Na stranici se mijenja broj žuto obojenih zvjezdica kako bi se označilo uspješno ocjenjivanje recepta.

Rezultat: Očekivani rezultat je zadovoljen čime je aplikacija prošla ovaj test.

Ispitni slučaj 4: Ocjenjivanje recepta bez prijave

Ulaz:

1. Navigiranje do stranice nekog recepta, u ovom slučaju */recipe/1*
2. Kliktanje na gumb ocjenu recepta

Očekivani rezultat: Na stranici se ne mijenja broj žuto obojenih zvjezdica koje označavaju ocjenu recepta.

Rezultat: Očekivani rezultat je zadovoljen čime je aplikacija prošla ovaj test. Valja napomenuti da pri ovome testu, poslužitelj također vraća odgovor sa šifrom 401, međutim Selenium ne omogućava čitanje zaprimljenih odgovora pa se pri

testiranju moramo oslanjati na CSS vrijednosti odgovarajućih elemenata stranice.

Ispitni slučaj 5: Objava recepta

Ulaz:

1. Otvaranje stranice za prijavu u web pregledniku
2. Unos ispravnog korisničkog imena i neispravne lozinke
3. Kliktanje na gumb za prijavu
4. Navigiranje do stranice za objavu recepta - */recipe/post*
5. Popunjavanje polja s informacijama o receptu - naslov, opis, sastojci...
6. Kliktanje na gumb za objavu recepta

Očekivani rezultat: Korisnika se prebacuje na novonastalu stranicu na kojoj se vidi objavljeni recept (*/recipe/[id recepta]*).

Rezultat: Očekivani rezultat je zadovoljen čime je aplikacija prošla ovaj test.

Ispitni slučaj 6: Objava recepta bez navođenja uputa za pripremu

Ulaz:

1. Otvaranje stranice za prijavu u web pregledniku
2. Unos ispravnog korisničkog imena i neispravne lozinke
3. Kliktanje na gumb za prijavu
4. Navigiranje do stranice za objavu recepta - */recipe/post*
5. Popunjavanje polja s informacijama o receptu osim onog s uputama
6. Kliktanje na gumb za objavu recepta

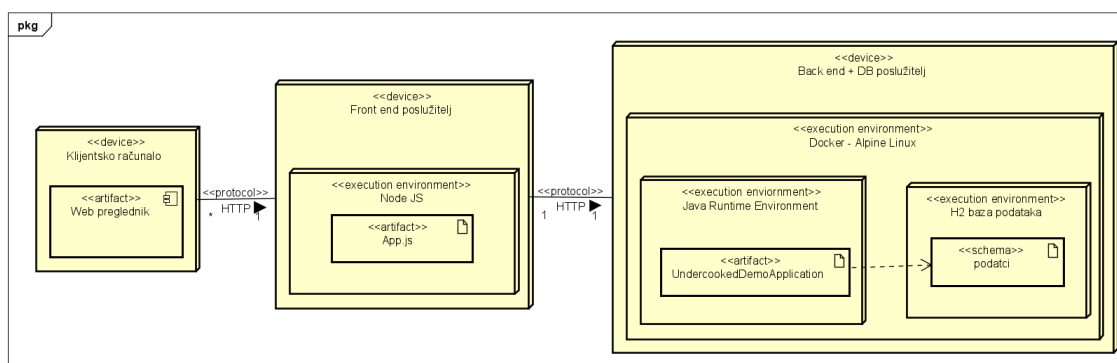
Očekivani rezultat: Korisnik ostaje na stranici za objavu recepta i dobiva obavijest da je potrebno popuniti opis pripreme.

Rezultat: Recept se objavljuje bez uputa za pripremu i korisnika se prebacuje na stranicu s novim receptom. Ova funkcionalnost nije implementirana pa aplikacija pada ovaj test, ali dalje nastavlja raditi normalno unatoč tome.

5.3 Dijagram razmještaja

Dijagramom razmještaja opisuje se topologija sustava i odnos različitih sklopovskih i programskih komponenti sustava. Sustav je temeljen na arhitekturi klijent-poslužitelj, a sastoji se od dva poslužiteljska računala. Na prvom se poslužitelju u okruženju NodeJS pokreće front-end aplikacije. Klijenti uz pomoć internetskog preglednika pristupaju aplikacije preko front-enda korištenjem protokola HTTP.

Front-end će po potrebi proslijediti HTTP zahtjev drugom poslužitelju na kojem se nalaze back-end aplikacije i baza podataka koji se radi jednostavnosti i prenosivosti pokreću u Docker kontejneru. Kada dobije HTTP zahtjev od front-enda, back-end bazi podataka šalje odgovarajući upit, a nakon što dobije odgovor, šalje ga front-endu koji ga onda koristi kako bi odgovorio klijentu.



Slika 5.2: Dijagram razmještaja

5.4 Upute za puštanje u pogon

5.4.1 Instalacija potrebne programske potpore

Kako bi se aplikacija ispravno pokrenula potrebno je preuzeti sljedeće programske pakete:

- *NodeJS*²⁰ i *npm*²¹ za pokretanje front-enda
- *Java development kit* (Verzija 17)²² i *Maven*²³ za back-end i bazu podataka.

```
[20:20] [root@ ~] # pacman -S nodejs npm jre17-openjdk maven
warning: nodejs-21.5.0-1 is up to date -- reinstalling
warning: npm-10.2.5-1 is up to date -- reinstalling
warning: jre17-openjdk-17.0.9.u8-2 is up to date -- reinstalling
warning: maven-3.8.7-1 is up to date -- reinstalling
resolving dependencies...
looking for conflicting packages...

Packages (4) jre17-openjdk-17.0.9.u8-2  maven-3.8.7-1  nodejs-21.5.0-1  npm-10.2.5-1

Total Installed Size: 82.86 MiB
Net Upgrade Size:    0.00 MiB

:: Proceed with installation? [Y/n] y
(4/4) checking keys in keyring [#####] 100%
(4/4) checking package integrity [#####] 100%
(4/4) loading package files [#####] 100%
(4/4) checking for file conflicts [#####] 100%
(4/4) checking available disk space [#####] 100%
:: Processing package changes...
(1/4) reinstalling nodejs [#####] 100%
(2/4) reinstalling npm [#####] 100%
(3/4) reinstalling jre17-openjdk [#####] 100%
(4/4) reinstalling maven [#####] 100%
```

Slika 5.3: Instalacija potrebne programske potpore na Linuxu

Alternativno, back-end i baza podataka mogu se jednostavnije pokrenuti u *Docker*²⁴ kontejneru.

Nakon instalacije navedenih programa, potrebno je klonirati repozitorij aplikacije na proizvoljnu lokaciju ili ga preuzeti kao arhivu i raspakirati.

²⁰<https://nodejs.org>

²¹<https://www.npmjs.com>

²²<https://www.oracle.com/java/technologies/downloads/#java17>

²³<https://maven.apache.org>

²⁴<https://www.docker.com>

```
[20:26] [redacted@redacted] ~ $ git clone git@github.com:Eloise-Habek/undercooked.git
Cloning into 'undercooked'...
remote: Enumerating objects: 3524, done.
remote: Counting objects: 100% (1080/1080), done.
remote: Compressing objects: 100% (346/346), done.
remote: Total 3524 (delta 635), reused 1007 (delta 600), pack-reused 2444
Receiving objects: 100% (3524/3524), 20.96 MiB | 1.82 MiB/s, done.
Resolving deltas: 100% (1896/1896), done.
```

Slika 5.4: Kloniranje repozitorija

5.4.2 Pokretanje front-enda

Kako bi se pokrenuo front-end aplikacije, potrebno je pozicionirati se u direktorij *IzvorniKod/undercooked-frontend* i pokrenuti *npm install* kako bi se instalirale potrebne biblioteke.

```
[20:30] [redacted@redacted] ~/undercooked/IzvorniKod/undercooked-frontend $ npm install
npm WARN deprecated stable@0.1.8: Modern JS already guarantees Array#sort() is a stable sort, so this library is deprecated. See the compatibility table on MDN: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/sort#browser_compatibility
npm WARN deprecated rollup-plugin-terser@7.0.2: This package has been deprecated and is no longer maintained. Please use @rollup/plugin-terser
npm WARN deprecated @babel/plugin-proposal-nullish-coalescing-operator@7.18.6: This proposal has been merged to the ECMAScript standard and thus this plugin is no longer maintained. Please use @babel/plugin-transform-nullish-coalescing-operator instead.
npm WARN deprecated @babel/plugin-proposal-numeric-separator@7.18.6: This proposal has been merged to the ECMAScript standard and thus this plugin is no longer maintained. Please use @babel/plugin-transform-numeric-separator instead.
npm WARN deprecated @babel/plugin-proposal-private-methods@7.18.6: This proposal has been merged to the ECMAScript standard and thus this plugin is no longer maintained. Please use @babel/plugin-transform-private-methods instead.
npm WARN deprecated @babel/plugin-proposal-class-properties@7.18.6: This proposal has been merged to the ECMAScript standard and thus this plugin is no longer maintained. Please use @babel/plugin-transform-class-properties instead.
npm WARN deprecated @babel/plugin-proposal-optional-chaining@7.21.0: This proposal has been merged to the ECMAScript standard and thus this plugin is no longer maintained. Please use @babel/plugin-transform-optional-chaining instead.
npm WARN deprecated w3c-hr-time@1.0.2: Use your platform's native performance.now() and performance.timeOrigin.
npm WARN deprecated sourcemap-codec@1.4.8: Please use @jridgewell/sourcemap-codec instead
npm WARN deprecated workbox-cacheable-response@6.6.0: workbox-background-sync@6.6.0
npm WARN deprecated svgo@1.3.2: This SVGO version is no longer supported. Upgrade to v2.x.x.

added 1540 packages, and audited 1541 packages in 7s

247 packages are looking for funding
  run `npm fund` for details

10 vulnerabilities (4 moderate, 6 high)

To address issues that do not require attention, run:
  npm audit fix

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.
```

Slika 5.5: Instalacija NodeJS biblioteka

Nakon toga, front-end aplikacije može se pokrenuti s *npm start*. *React* front-end pokreće se na vratima 3000, osim ako nije postavljena varijabla okruženja *PORT*.

```
[20:32] [redacted@redacted] ~/undercooked/IzvorniKod/undercooked-frontend $ npm start
```

Slika 5.6: Pokretanje front-enda

5.4.3 Pokretanje back-enda i baze podataka

Kako bi se pokrenuli back-end aplikacije i baza podataka, potrebno je pozicionirati se u direktorij *IzvorniKod/Undercooked-Demo* i pokrenuti *mvn install* čime se instalira sve što je potrebno za pokretanje back-enda i baze podataka te se gradi sama aplikacija.

```
[20:32] [redacted@redacted] ~/undercooked/IzvorniKod/Undercooked-Demo $ mvn install
[INFO] Scanning for projects...
[INFO]
[INFO] -----< hr.fer.progi:Undercooked-Demo >-----
[INFO] Building Undercooked-Demo 0.0.1-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO]
```

Slika 5.7: Instalacija uz pomoć Mavena

Ova naredba će također pokrenuti i testove, no svi ispitni slučajevi koji koriste *Selenium* će pasti jer se oslanjaju na to da back-end radi, a pokreću se prije nego što se on pokrene. Testovi se mogu izvršiti naknadno, na primjer iz IDE-a. Nakon što je izgrađena, aplikacija se može pokrenuti navigiranjem u novonastali direktorij *target* i izvršavanjem naredbe *java -jar Undercooked-Demo-0.0.1-SNAPSHOT.jar*. Back-end aplikacije sluša na vratima 8080, a pokretanjem back-enda pokreće se i baza podataka.

```
[20:36] [e] ~/undercooked/IzvorniKod/Undercooked-Demo/target $ java -jar Undercooked-Demo-0.0.1-SNAPSHOT.jar
```

Slika 5.8: Pokretanje back-enda i baze podataka

6. Zaključak i budući rad

Na ovom je projektu uspješno izrađena web stranica za dijeljenje recepata i diskusiju između korisnika. Projekt se odvio u dvije glavne faze, prva faza se odvijala u prvom ciklusu semestra koju obilježava upoznavanje s projektom, te druga faza koja se odvijala tokom drugog ciklusa gdje smo se usredotočili na implementaciju web stranice.

U prvoj fazi projekta, okupljen je tim od 7 članova. Najprije su odabrane tehnologije koje će se koristiti tijekom projekta te je napravljena podjela projektnih zadataka po interesima i sposobnostima članova, gdje je ujedno bio cilj da svi dođu u dodir sa svim fazama razvoja i da količina posla bude ravnomjerno podijeljena. Rana podjela odgovornosti u timu omogućila je članovima da rano krenu učiti o tehnologijama u kojima će raditi te istraživati odabrane alate i programske jezike. Uslijedio je rad na dokumentaciji i pokretanje razvoja aplikacije. U prvoj fazi razvoja web stranice najteži dio je bio definirati obrasce uporabe te funkcionalnosti kako se ne bi trebali vraćati popravljati greške zbog manjka diskusije i razmišljanja na početku. Sprječili smo ovakve probleme uz puno otvorenih timskih diskusija gdje smo uskladili ideje i očekivanja koje smo imali o projektu. U drugoj fazi projekta uslijedila je realizacija rješenja i izrada preostale dokumentacije. Kako bi se rad članova što više uskladio i komunikacija olakšala, članovi su se sastajali jednom tjedno uživo i preko teamsa tijekom praznika. Osim toga, napravljena je i Whatsapp grupa u kojoj je voditeljica nakon sastanaka slala zapisnik o sastanku i koja je služila kao glavni medij komunikacije među članovima.

Kroz zajednički rad na ovom projektu u složnom i dobro organiziranom timu, svi su članovi stekli iskustvo koje im u budućnosti može služiti kao primjer tima u kakvom žele biti. Zajedno smo osvijestili važnost dobre vremenske organiziranosti i koordinacije među članovima tima, ali i osjećaj odgovornosti prema drugim članovima koji nemamo u samostalnom radu i učenju. Zadovoljni smo postignutim rezultatima ali svjesni smo da ima puno prostora za napredak naših vještina, kako tehničkih tako i organizacijskih, timskih i komunikacijskih.

Logičan korak u razvoju projekta koji bi dalje mogao slijediti je ostvarenje dodatnih funkcionalnosti na web stranici, dodatan rad na UI stranice ili izrada mo-

bilne aplikacije.

Popis literature

1. Programsko inženjerstvo, FER ZEMRIS, <http://www.fer.hr/predmet/proinz>
2. SpringBoot framework, <https://spring.io/projects/spring-framework/>
3. React documentation, <https://devdocs.io/react/>
4. Selenium with Python, <https://selenium-python.readthedocs.io/>

Indeks slika i dijagrama

2.1	Coolinarika - Prevelika slova, puno neiskorištenog prostora	5
2.2	Foodnetwork - Preglednije, ali i dalje puno praznog prostora	6
3.1	Dijagram obrasca uporabe, funkcionalnost korisnika i neregistriranog korisnika	21
3.2	Dijagram obrasca uporabe, funkcionalnost registriranog korisnika	22
3.3	Dijagram obrasca uporabe, funkcionalnost administratora	23
3.4	Sekvencijski dijagram za UC3	24
3.5	Sekvencijski dijagram za UC7	25
3.6	Sekvencijski dijagram za UC18	26
3.7	Sekvencijski dijagram za UC21	27
4.1	Arhitektura sustava	29
4.2	Relacijski dijagram baze podataka	35
4.3	Dijagram razreda - dio Controllers	36
4.4	Dijagram razreda - dio Data transfer objects	37
4.5	Dijagram stanja - registrirani korisnik	39
4.6	Dijagram aktivnosti objave novog recepta	41
4.7	Dijagram komponenti	42
5.1	Rezultati unit testova	45
5.2	Dijagram razmještaja	53
5.3	Instalacija potrebne programske potpore na Linuxu	54
5.4	Kloniranje repozitorija	55
5.5	Instalacija NodeJS biblioteka	55
5.6	Pokretanje front-enda	55
5.7	Instalacija uz pomoć Mavena	56
5.8	Pokretanje back-enda i baze podataka	56
6.1	Dijagram komponenti - prvi dio	66
6.2	Dijagram komponenti - drugi dio	67

Dodatak: Prikaz aktivnosti grupe

Dnevnik sastajanja

1. sastanak

- Datum: 19. listopada 2023.
- Prisustvovali: E.Habek, M.Plavec, A.Dautović, L.Grubišin, A.Đurđević, M.Magat, A.Stanković
- Teme sastanka:
 - procjena vještina i motivacija članova tima
 - izmijena ideja o načinu provedbe projekta
 - upoznavanje
 - odabir tehnologija

2. sastanak

- Datum: 24. listopada 2023.
- Prisustvovali: E.Habek, M.Plavec, A.Dautović, L.Grubišin, A.Đurđević, M.Magat, A.Stanković
- Teme sastanka:
 - razrada projekta
 - zajedničko iznošenje ideja za obrasce uporabe
 - podjela posla za pisanje dokumentacije za naredni tjedan
 - međusobno pomaganje i dijeljenje znanja o Gitu

3. sastanak

- Datum: 31. listopada 2023.
- Prisustvovali: E.Habek, M.Plavec, A.Dautović, L.Grubišin, A.Đurđević, M.Magat, A.Stanković
- Teme sastanka:
 - skica baze podataka
 - skica grafičkog sučelja
 - skica ruta
 - razrješiti nedoumice

4. sastanak

- Datum: 7. studenog 2023.
- Prisustvovali: E.Habek, M.Plavec, A.Dautović, L.Grubišin, A.Đurđević, M.Magat, A.Stanković
- Teme sastanka:
 - iznošenje svojih problema te zajedničko pronalaženje rješenja
 - diskutiranje o specifičnim svojstvima baze podataka

5. sastanak

- Datum: 14. studenog 2023.
- Prisustvovali: E.Habek, M.Plavec, A.Dautović, L.Grubišin, A.Đurđević, M.Magat, A.Stanković
- Teme sastanka:
 - programeri pokazuju i objašnjavaju ono što su učinili
 - diskutiranje kako ćemo deployati

6. sastanak

- Datum: 19. prosinca 2023.
- Prisustvovali: E.Habek, M.Plavec, A.Dautović, L.Grubišin, A.Đurđević, M.Magat, A.Stanković
- Teme sastanka:
 - dijeljenje trenutnog napretka
 - dogovor o daljnjem ophođenju preko zimskih praznika

7. sastanak

- Datum: 4. siječnja 2024.
- Prisustvovali: E.Habek, M.Plavec, A.Dautović, L.Grubišin, A.Đurđević, M.Magat, A.Stanković
- Teme sastanka:
 - dijeljenje trenutnog napretka
 - iznošenje problema i razmišljanje o rješenjima

8. sastanak

- Datum: 9. siječnja 2024.
- Prisustvovali: E.Habek, M.Plavec, A.Dautović, L.Grubišin, A.Đurđević, M.Magat, A.Stanković
- Teme sastanka:
 - pokazivanje programske potpore

9. sastanak

- Datum: 16. siječnja 2024.
- Prisustvovali: E.Habek, M.Plavec, A.Dautović, L.Grubišin, A.Đurđević, M.Magat, A.Stanković
- Teme sastanka:
 - dogovoravanje puštanje u pogon

Tablica aktivnosti

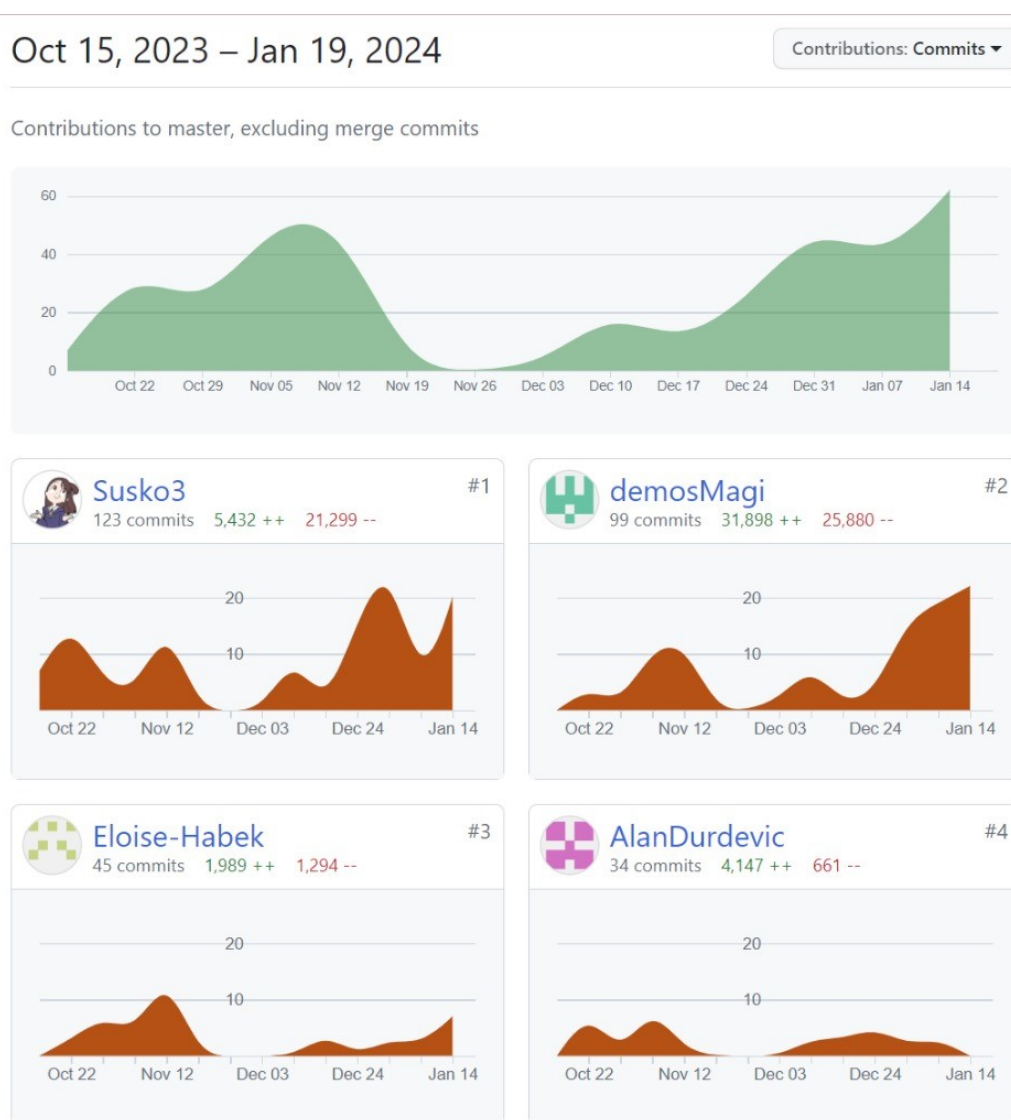
	Eloise Habek	Anabel Dautović	Mateo Plavec	Luka Grubišin	Alan Đurđević	Matej Magat	Andrej Stanković
Upravljanje projektom	7						
Opis projektnog zadatka				5			
Funkcionalni zahtjevi	3			5			
Opis pojedinih obrazaca	5		1			2	
Dijagram obrazaca	5						
Sekvencijski dijagrami							4
Opis ostalih zahtjeva				1			
Arhitektura i dizajn sustava				2			
Baza podataka		8					
Dijagram razreda		10					
Dijagram stanja	6						
Dijagram aktivnosti				2			
Dijagram komponenti		10	1				
Korištene tehnologije i alati	2						
Ispitivanje programskog rješenja				11			
Dijagram razmještaja				1			
Upute za puštanje u pogon				2		1	
Dnevnik sastajanja	1						
Zaključak i budući rad	1	3					
Popis literature	1						

Nastavljeno na idućoj stranici

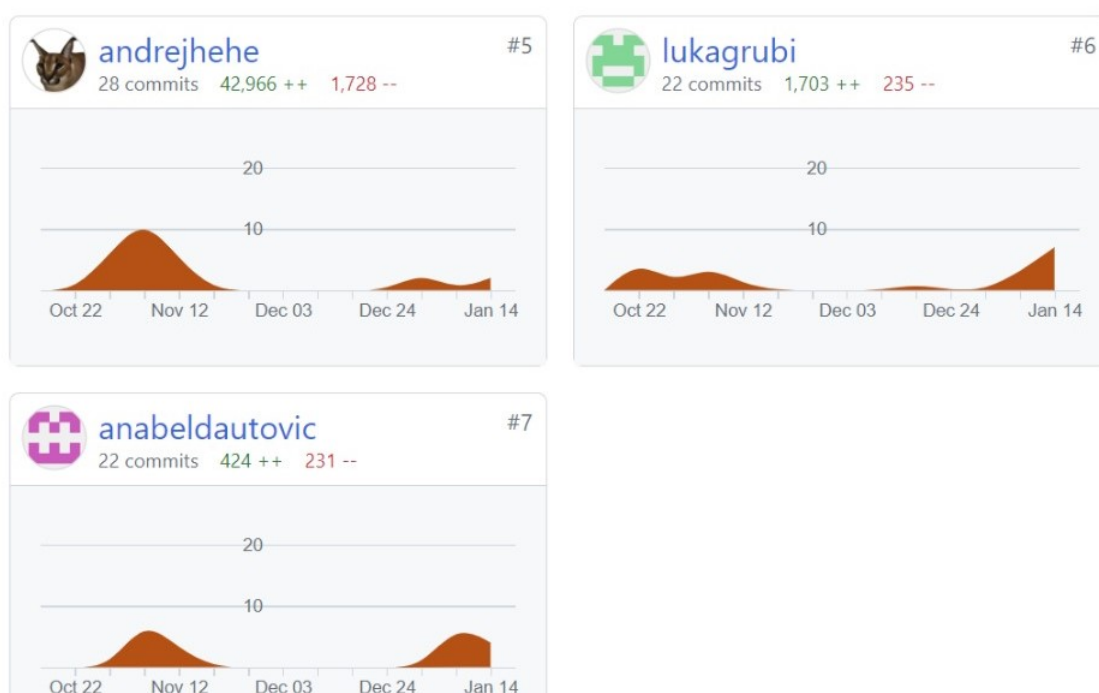
Nastavljeno od prethodne stranice

	Eloise Habek	Anabel Dautović	Mateo Plavec	Luka Grubišin	Alan Đurđević	Matej Magat	Andrej Stanković
<i>Postavljanje programske potpore</i>	1		3	2	2		
<i>GitHub admin</i>			4				
<i>Proučavanje backend i frontend tehnologija</i>				1	8	6	2
<i>Dizajn aplikacije</i>							20
<i>Izrada frontenda</i>						29	2
<i>izrada baze podataka</i>			1				
<i>spajanje s bazom podataka</i>							
<i>Izrada backenda</i>			29		20	3	
<i>deploy</i>	1		4			1	
<i>provjera dokumentacije</i>	6		1	1			
<i>izrada sustava poruka</i>					5		
<i>traženje video API</i>	2						
<i>izrada sustava pratitelja</i>	2				5		
<i>testiranje</i>			3				
<i>izrada prezentacije</i>		3					

Dijagrami pregleda promjena



Slika 6.1: Dijagram komponenti - prvi dio



Slika 6.2: Dijagram komponenti - drugi dio