

Programsko inženjerstvo

Ak. god. 2023./2024.

InterFER

Dokumentacija, Rev. 1.0

Grupa: Burek

Voditelj: Oleksandr Ichenskyi

Datum predaje: 17. studenog 2023.

Nastavnik: Igor Stančin

Sadržaj

1 Dnevnik promjena dokumentacije	2
2 Opis projektnog zadatka	4
2.1 Dionici InterFER Platforme	6
2.2 Zaključak	7
3 Specifikacija programske potpore	9
3.1 Funkcionalni zahtjevi	9
3.1.1 Obrasci uporabe	11
3.1.2 Sekvencijski dijagrami	25
3.2 Ostali zahtjevi	31
4 Arhitektura i dizajn sustava	32
4.1 Baza podataka	35
4.1.1 Opis tablica	35
4.1.2 Dijagram baze podataka	38
4.2 Dijagram razreda	39
4.3 Dijagram stanja	47
4.4 Dijagram aktivnosti	50
4.5 Dijagram komponenti	52
5 Implementacija i korisničko sučelje	53
5.1 Korištene tehnologije i alati	53
5.1.1 Komunikacijske platforme	53
5.1.2 Razvoj aplikacije	54
5.2 Ispitivanje programskog rješenja	55
5.2.1 Ispitivanje komponenti	55
5.2.2 Ispitivanje sustava	59
5.3 Dijagram razmještaja	63
5.4 Upute za puštanje u pogon	65
5.4.1 Stvaranje baze na javnom poslužitelju	65

5.4.2 Konfiguriranje backenda	66
5.4.3 Posluživanje backenda	66
5.4.4 Konfiguriranje frontenda	67
5.4.5 Posluživanje frontenda	67
6 Zaključak i budući rad	68
Popis literature	70
Indeks slika i dijagrama	71
Dodatak: Prikaz aktivnosti grupe	72

1. Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Napravljen predložak.	Anton La-dan	3.11.2023.
0.2	Funkcionalni zahtjevi - dionici i aktori	Anton La-dan	10.11.2023.
0.3	Funkcionalni zahtjevi - UC dijagrami	Anton La-dan	13.11.2023.
0.4	Funkcionalni zahtjevi - dijagrami obrazaca uporabe	Oleksandr Ichenskyi	14.11.2023.
0.5	Arhitektura i dizajn sustava - opis arhitekture	Oleksandr Ichenskyi	15.11.2023.
0.6	Opis projektnog zadatka i ostali zahtjevi	Nikola Antolović	16.11.2023.
0.7	Sekvencijski dijagrami obrazaca uporabe	Antonia Šarčević	16.11.2023.
0.8	Dnevnik sastajanja	Anton La-dan	17.11.2023.
0.9	Dijagrami razreda	Antonia Šarčević	17.11.2023.
0.10	Opis baze podataka	Karlo Španović	17.11.2023.
1.0	Korigiranje i provjera dokumentacije	Oleksandr Ichenskyi	11.09.2033.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodataka	Autori	Datum
1.1	Dijagrami stanja i aktivnosti	Anton Ladan, Olek- sandr Ichenskyi	19.01.2024.
1.2	Dnevnik promjena dokumentacije	Anton La- dan	19.01.2024.
1.3	Tablica aktivnosti	*	19.01.2024.
1.4	Dijagrami pregleda promjena	Anton La- dan	19.01.2024.
1.5	Zaključak	Oleksandr Ichenskyi	19.01.2024.
1.6	Dijagram aktivnosti - opis	Oleksandr Ichenskyi	19.01.2024.
1.7	Upute za puštanje u pogon	Antonia Šarčević	19.01.2024.
1.8	Dijagram razmještaja	Nikola Antolović	19.01.2024.
1.9	Korištene tehnologije	Nikola Antolović, Anton Ladan	19.01.2024.
1.10	Ispitivanje programskog rješenja	Anton La- dan	19.01.2024.
1.11	Dijagram komponenti	Karlo Španović	19.01.2024.

2. Opis projektnog zadatka

InterFER, platforma za pisanje i objavu studentskih članaka, predstavlja inovativan projektni zadatak čija je svrha olakšati studentima stvaranje, dijeljenje i objavljivanje svojih akademskih radova. Nastala iz potrebe za jednostavnom razmjenom iskustava i znanja, ova platforma ima ambiciozan cilj poticanja suradnje unutar akademske zajednice. Kroz niz funkcionalnosti, InterFER će omogućiti registraciju korisnika, personalizaciju profila, kreiranje i uređivanje članaka, objavu i dijeljenje sadržaja, kategorizaciju i označavanje članaka, komentiranje, ocjenjivanje te moderaciju neprikladnog sadržaja. Pored toga, identificirani su ključni dionici, uključujući studente, razvojni tim, profesore, asistente i Sveučilište u Zagrebu.

Razvojna platforma će osigurati prostor za stvaranje i razmjenu akademskih ideja, potičući tako kreativnost i raznolikost izraza unutar virtualnog prostora posvećenog akademskom dijalogu. U ovom uvodu, istražit ćemo svaku funkcionalnost platforme, analizirati uloge dionika te naglasiti potencijalne koristi koje InterFER može pružiti studentskoj zajednici.

InterFER platforma će pružiti niz funkcionalnosti kako bi zadovoljila potrebe svojih korisnika:

1. Registracija, Prijava i Personalizacija Profila

Korisnici će se moći registrirati na platformi pružanjem osnovnih informacija i potrebnih podataka. Nakon registracije, prijava će biti omogućena putem korisničkih podataka. Dodatno, personalizacija profila omogućit će korisnicima da dodaju informacije o sebi, uključujući sliku profila, opis i druge relevantne podatke. Ova opcija personalizacije profila će im omogućiti bolje povezivanje s drugim korisnicima, stvarajući tako snažniju online zajednicu.

2. Kreiranje i Uređivanje Članaka

Korisnicima će biti omogućeno stvaranje novih članaka pomoću ugrađenog uređivača teksta. Ovaj uređivač će podržavati različite oblike formatiranja sadržaja, uključujući stilizaciju teksta, dodavanje slika i drugih multimedijalnih elemenata te omogućiti izradu skica članaka. Korisnicima će se također pružiti mogućnost suradnje na člancima, što će poticati timski rad i raznolikost perspektiva.

3. Objava i Dijeljenje Članaka

Kada korisnici budu zadovoljni sadržajem svojih članaka, moći će ih objaviti. Objavljeni članci će postati javno dostupni i bit će vidljivi svim korisnicima platforme. Osim pregleda, korisnicima će biti omogućeno dijeljenje objavljenih članaka na društvenim mrežama, što će proširiti doseg platforme i potaknuti širenje akademске informacije izvan same platforme.

4. Kategorizacija i Označavanje Članaka

Članci će biti svrstani u određene kategorije, a korisnicima će biti omogućeno dodavanje odgovarajućih oznaka radi lakšeg pretraživanja i organizacije. Također, korisnici će moći pretraživati članke prema ključnim riječima, kategorijama ili oznakama. Filtre će moći primijeniti za sortiranje članaka prema datumu, popularnosti ili kategoriji, čime će se poboljšati iskustvo pretraživanja i pronalaženja relevantnih informacija.

5. Komentiranje i Ocjena Članaka

Prijavljeni korisnici će moći ostavljati komentare na člancima kako bi pružili povratne informacije ili sudjelovali u raspravi. Svaki članak će prikazivati broj pregleda, pozitivnih i negativnih ocjena te komentara. Osim toga, korisnicima će biti omogućeno označavanje kvalitetnih komentara, čime će se poticati konstruktivna rasprava i zajedništvo unutar platforme.

6. Moderator i Administrator Uloge

Na platformi će postojati uloge moderatora i administratora. Moderatori će imati ovlasti za moderiranje i brisanje komentara te skrivanje neprikladnih članaka ili slanje istih na doradu. Administratori će imati dodatne ovlasti za dodjelu uloge moderatora i administratora. Također, administratori će moći pratiti omjer moderiranih i objavljenih članaka kako bi pravilno upravljali opterećenjem moderatora.

Osim toga, razmatrat će se mogućnost implementacije sustava nagrađivanja za moderatore kako bi se potaknula aktivnost u održavanju kvalitete sadržaja.

7. Upravljanje Neprikladnim Sadržajem

U slučaju širenja neprikladnog sadržaja od strane korisnika, moderatori će imati ovlasti za privremenu ili trajnu zabranu komentiranja i ocjenjivanja članaka, te privremeno ili trajno oduzimanje ovlasti objave članaka. Osim toga, razmotrit će se implementacija sustava automatske detekcije neprikladnog sadržaja kako bi se smanjila potreba za ručnim moderiranjem.

2.1 Dionici InterFER Platforme

Razmatramo i identificiramo ključne dionike u ovom projektu kako bismo bolje razumjeli tko će biti uključen u korištenje i upravljanje platformom:

1. Korisnici

Razvojni tim, sastavljen od programera, dizajnera i administratora, ima ključnu ulogu u ostvarivanju ciljeva InterFER platforme. Njihova odgovornost obuhvaća razvoj, održavanje i poboljšanje platforme kako bi osigurali suvremeno i intuitivno korisničko iskustvo. Planira se redovito ažuriranje platforme kako bi se uključile nove funkcionalnosti, a feedback korisnika bit će neprestano uzet u obzir. Razvojni tim također može igrati ključnu ulogu u provođenju edukacija za korisnike o novim značajkama i mogućnostima InterFER platforme, potičući ih da maksimalno iskoriste njene potencijale.

2. Razvojni Tim

Razvojni tim, sastavljen od programera, dizajnera i administratora, ima ključnu ulogu u ostvarivanju ciljeva InterFER platforme. Njihova odgovornost obuhvaća razvoj, održavanje i poboljšanje platforme kako bi osigurali suvremeno i intuitivno korisničko iskustvo. Planira se redovito ažuriranje platforme kako bi se uključile nove funkcionalnosti, a feedback korisnika bit će neprestano uzet u obzir. Razvojni tim također može igrati ključnu ulogu u provođenju edukacija za korisnike o novim značajkama i mogućnostima InterFER platforme, potičući ih da maksimalno iskoriste njene potencijale.

3. Profesori i Asistenti

Uloga profesora i asistenata na InterFER platformi proširuje se izvan njihove tradicionalne uloge u učionici. Oni će preuzeti ulogu moderatora na platformi, nadzirući sadržaj, komentare i članke kako bi osigurali da se pravila i standardi poštuju. Osim toga, razmatrat će se mogućnost organizacije radionica i edukacija direktno putem platforme, čime će se podržati razvoj vještina pisanja i komunikacije kod studenata. Njihova prisutnost na platformi kao mentora također će pridonijeti stvaranju poticajnog okruženja za učenje, gdje će se akademske ideje razvijati kroz dijalog i konstruktivnu kritiku.

4. Sveučilište u Zagrebu

Sveučilište u Zagrebu, kao sekundarni dionik, igra ključnu ulogu u podršci i promociji InterFER platforme. Njegova podrška će pridonijeti širenju znanja i dijaloga unutar sveučilišne zajednice. Razmatrat će se mogućnost suradnje s drugim sveučilištima kako bi se proširio utjecaj InterFER-a i potaknula interakcija između studenata različitih institucija. Sveučilište također može pružiti dodatne resurse i poticaje, potičući suradnju između različitih fakulteta i disciplina.

Razumijevanje uloge svakog dionika ključno je za uspješno vođenje InterFER platforme i postizanje njezinog punog potencijala. Kroz ovo prošireno razmatranje, naglasak se stavlja na međusobnu interakciju između korisnika, razvojnog tima, mentora i institucija, stvarajući tako bogato i dinamično okruženje za dijalog i razmjenu akademskih ideja.

2.2 Zaključak

InterFER predstavlja ne samo platformu za pisanje i objavu članaka već i poticaj za rast i razvoj akademske kulture dijeljenja znanja. Kroz poticanje studenata da dijele svoje ideje i iskustva, ova platforma postavlja temelje za jaču povezanost unutar akademske zajednice. Njegove funkcionalnosti omogućuju korisnicima unaprjeđivanje vlastitih vještina pisanja i komunikacije, a suradnja na člancima potiče timski rad i raznolikost perspektiva.

Identificirani dionici, poput profesora i asistenata, imat će ključnu ulogu u održavanju kvalitete sadržaja, prateći pravila i standarde. Razvojni tim će kontinuirano raditi na poboljšanjima i ažuriranjima kako bi osigurao optimalno korisničko iskustvo.

Sveučilište u Zagrebu kao sekundarni dionik pružit će podršku i promovirati platformu, doprinoseći širenju znanja unutar sveučilišne zajednice.

U zaključku, InterFER ne samo da odražava potrebu za inovacijama u akademском prostoru već i potiče rast zajedništva i razmjene ideja među studentima. Ova platforma predstavlja korak prema stvaranju dinamičnog virtualnog prostora posvećenog akademskom dijalogu i poticanju raznolikosti u izražavanju znanja.

3. Specifikacija programske potpore

3.1 Funkcionalni zahtjevi

Dionici:

1. Studenti
2. Profesori
3. Asistenti
4. Sveučilište u Zagrebu
5. Razvojni tim

Aktori i njihovi funkcionalni zahtjevi:

1. Anonimni korisnik (inicijator) može:
 - (a) prijaviti se u aplikaciju
 - (b) stvoriti profil
 - (c) pretraživati članke
 - i. po kategorijama
 - ii. po naslovu
 - (d) vidjeti određeni članak
 - (e) podijeliti članak na društvenim mrežama
 - (f) pregledati tuđe profile
2. Prijavljeni korisnik (inicijator) može:
 - (a) se odjaviti
 - (b) stvoriti
 - i. članak
 - ii. komentar na članak
 - (c) urediti
 - i. svoj članak
 - ii. svoj komentar na članak

iii. svoj profil

(d) obrisati

i. svoj članak

ii. svoj komentar na članak

(e) vidjeti obavijesti

(f) ocijeniti tuđi članak

(g) pregledati svoj profil

(h) obrisati svoj profil

(i) prijaviti članak

3. Moderator (inicijator) može:

(a) zatražiti izmjenu članka

(b) zatražiti izmjenu komentara

(c) obrisati tuđi članak

(d) orisati tuđi komentar

(e) slati obavijesti

(f) privremeno/trajno banati korisnika

4. Administrator (inicijator) može:

(a) dodati moderatora/administratora

(b) ukloniti moderatora

(c) odstupiti s mjesta administratora

3.1.1 Obrasci uporabe

Opis obrazaca uporabe

UC1 - Prijava u aplikaciju

- **Glavni sudionik:** Anonimni korisnik
- **Cilj:** Prijaviti se u aplikaciju
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je registriran
- **Opis osnovnog tijeka:**
 1. Korisnik ulazi u sučelje za prijavu
 2. Korisnik unosi email adresu i šifru
 3. Provjera valjanosti podataka
 4. Korisnik ulazi u aplikaciju kao prijavljeni korisnik
- **Opis mogućih odstupanja:**
 - 3.a Unesena je neispravna email adresa ili šifra
 1. Sustav obavještava korisnika o neispravnosti unesenih podataka, i predlaže korisniku promjenu unesenih podataka

UC2 - Stvaranje računa

- **Glavni sudionik:** Anonimni korisnik
- **Cilj:** Stvoriti račun
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Korisnik ulazi u sučelje za stvaranje računa
 2. Korisnik unosi ime, email adresu i šifru
 3. Provjera valjanosti podataka
 4. Sustav dojavljuje korisniku kako je registracija uspješna
- **Opis mogućih odstupanja:**
 - 3.a Račun s već navedenom email adresom već postoji
 1. Sustav predlaže korisniku promjenu unesene email adrese ili prijavu u postojeći profil pod navedenom adresom
 - 3.b Šifra je preslab
1. Sustav predlaže korisniku promjenu unesene šifre
 - 3.c Drugi unos šifre se ne podudara s prvim

1. Sustav obavještava korisnika da se šifre razlikuju

UC3 - Pretraga članaka po kategorijama

- **Glavni sudionik:** Anonimni/prijavljeni korisnik
- **Cilj:** Pretražiti članke po kategorijama
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Korisnik bira kategorije za pretragu
 2. Baza podataka dohvaća članke sa zadanim kategorijama
 3. Korisniku se prikazuju dohvaćeni članci
- **Opis mogućih odstupanja:**
 - 2.a U bazi podataka ne postoji članak sa navedenim kategorijama
 1. Sustav obavještava korisnika da nije dohvaćen niti jedan članak, te da proba suziti izbor kategorija

UC4 - Pretraga članaka po naslovu

- **Glavni sudionik:** Anonimni/prijavljeni korisnik
- **Cilj:** Pretražiti članke po naslovu
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Korisnik unosi tekst za pretragu
 2. Baza podataka dohvaća članke koji sadrže zadani tekst u naslovu
 3. Korisniku se prikazuju dohvaćeni članci
- **Opis mogućih odstupanja:**
 - 2.a U bazi podataka ne postoji članak sa danim tekstrom u naslovu
 1. Sustav obavještava korisnika da nije dohvaćen niti jedan članak, te da proba promijeniti tekst za pretragu

UC5 - Pregledavanje članka

- **Glavni sudionik:** Anonimni/prijavljeni korisnik
- **Cilj:** Pregledati određeni članak
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**

1. Dohvaćanje članka sa baze podataka
2. Prikaz članka na sučelju

UC6 - Dijeljenje članaka

- **Glavni sudionik:** Anonimni/prijavljeni korisnik
- **Cilj:** Podijeliti članak na društvenim mrežama
- **Sudionici:** -
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Korisnik bira opciju za dijeljenje članka
 2. Korisniku se prikazuje izbornik za društvene mreže
 3. Preusmjeravanje na izabranu društvenu mrežu ili kopiranje adrese u korisnikov međuspremnik
- **Opis mogućih odstupanja:**

UC7 - Pregled profila

- **Glavni sudionik:** Anonimni/prijavljeni korisnik
- **Cilj:** Pregledati tuđi profil
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Korisnik ulazi na stranicu profila
 2. Baza podataka dohvaća javne podatke profila
 3. Korisniku se prikazuju dohvaćeni podaci profila
- **Opis mogućih odstupanja:**
 - 2.a Profil ne postoji, ili je izbrisani
 1. Korisnik se preusmjerava na matičnu stranicu, sa dojavom da profil ne postoji

UC8 - Odjava sa aplikacije

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:** Odjaviti se sa aplikacije
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik bira opciju za odjavu iz aplikacije

2. Baza podataka briše zapis o prijavi
3. Korisnik se preusmjerava na stranicu za anonimnog korisnika

UC9 - Stvaranje članaka

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:** Stvoriti novi članak
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen, korisnik nema ograničen pristup
- **Opis osnovnog tijeka:**
 1. Korisnik otvara sučelje za pisanje članka
 2. Korisnik upisuje tekst članka
 3. Korisnik objavljuje članak
 4. Baza podataka sprema članak
 5. Sustav dojavljuje korisniku da je članak objavljen
- **Opis mogućih odstupanja:**
 - 3.a Uneseni tekst je prazan
 1. Sustav dojavljuje korisniku da tekst ne smije biti prazan

UC10 - Uređivanje članaka

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:** Urediti vlastiti članak
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen, članak pripada korisniku
- **Opis osnovnog tijeka:**
 1. Korisnik bira opciju za uređivanje članka
 2. Otvara se sučelje za pisanje članka
 3. Korisnik unosi promjene u članak
 4. Korisnik objavljuje članak
 5. Baza podataka sprema promjene članka
 6. Sustav dojavljuje korisniku da je članak uređen
- **Opis mogućih odstupanja:**
 - 3.a Članak je prazan/nepromijenjen
 1. Sustav dojavljuje korisniku da tekst ne smije biti prazan/nepromijenjen

UC11 - Brisanje vlastitog članka

- **Glavni sudionik:** Prijavljeni korisnik

- **Cilj:** Obrisati vlastiti članak
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen, članak pripada korisniku
- **Opis osnovnog tijeka:**
 1. Korisnik bira opciju za brisanje članka
 2. Otvara se sučelje za potvrdu brisanja članka
 3. Korisnik potvrđuje brisanje članka
 4. Baza podataka briše članak
- **Opis mogućih odstupanja:**
 - 3.a Korisnik nije potvrdio brisanje članka
 1. Članak se ne briše

UC12 - Pregled obavijesti

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:** Pregledati obavijesti
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik otvara sučelje za prikaz obavijesti
 2. Baza podataka dohvata obavijesti
- **Opis mogućih odstupanja:**
 - 2.a Nema zapisa o obavijestima za korisnika
 1. Sustav dojavljuje korisniku da nema obavijesti

UC13 - Ocjenjivanje članaka

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:** Ocijeniti tuđi članak
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik ocjenjuje članak jednom od predodređenih ocjena
 2. Baza podataka sprema zapis o ocjeni od korisnika na članak
 3. Sučelje osvježava prikaz o ocjenama članka
- **Opis mogućih odstupanja:**
 - 2.a Korisnik je već ostavio ocjenu na članak
 1. Baza podataka unosi promjene na već postojeći zapis

- 2.b Korisnik pokušava ostaviti ocjenu na vlastiti članak
 1. Sustav dojavljuje korisniku da ne može ocijeniti vlastiti članak

UC14 - Komentiranje članaka

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:** Komentirati članak
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen, korisnik nema ograničen pristup
- **Opis osnovnog tijeka:**
 1. Korisnik otvara sučelje za pisanje komentara
 2. Korisnik upisuje komentar
 3. Korisnik objavljuje komentar
- **Opis mogućih odstupanja:**
 - 3.a Komentar je prazan
 1. Sustav dojavljuje korisniku da komentar ne smije biti prazan

UC15 - Brisanje vlastitih komentara

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:** Obrisati vlastiti komentar na članak
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen, komentar pripada korisniku
- **Opis osnovnog tijeka:**
 1. Korisnik bira opciju za brisanje komentara
 2. Otvara se sučelje za potvrdu brisanja komentara
 3. Korisnik potvrđuje brisanje komentara
 4. Baza podataka briše komentar
- **Opis mogućih odstupanja:**
 - 3.a Korisnik nije potvrdio brisanje komentara
 1. Komentar se ne briše

UC16 - Pregled vlastitog profila

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:** Pregledati vlastiti profil
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**

1. Korisnik bira opciju za pregled profila
2. Baza podataka dohvaća podatke profila korisnika
3. Korisniku se prikazuju dohvaćeni podatci

UC17 - Uređivanje profila

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:** Urediti vlastiti profil
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik bira opciju za uređivanje profila
 2. Baza podataka dohvaća trenutne podatke profila korisnika
 3. Korisniku se dohvaćeni podatci prikazuju
 4. Korisnik uređuje podatke
 5. Korisnik spremi promjene profila
 6. Baza podataka spremi promjene profila
- **Opis mogućih odstupanja:**
 - 6.a Neki od podataka nisu valjni
 1. Sustav korisniku dojavljuje da profil nije spremlijen, te koji su podatci nevaljni

UC18 - Brisanje profila

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:** Izbrisati vlastiti profil
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik bira opciju za brisanje profila
 2. Otvara se sučelje za potvrdu brisanja profila
 3. Korisnik potvrđuje brisanje profila
 4. Baza podataka briše zapis o korisničkom profilu
- **Opis mogućih odstupanja:**
 - 3.a Korisnik nije potvrdio brisanje profila
 1. Profil se ne briše

UC19 - Prijavljivanje članaka

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:** Prijaviti neprimjeren članak
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik bira opciju za prijavljivanje članka
 2. Korisniku se prikazuje sučelje za opis neprimjerenog sadržaja
 3. Korisnik unosi podatke
 4. Korisnik prijavljuje članak
 5. Baza podataka spremna prijavu članka
- **Opis mogućih odstupanja:**
 - 5.a Podatci su neispravni ili nedostaju
 1. Sustav dojavljuje korisniku da su podaci neispravni ili nedostaju

UC20 - Pregledavanje komentara

- **Glavni sudionik:** Anonimni/prijavljeni korisnik
- **Cilj:** Pregledati komentare na nečiji članak
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Korisnik bira opciju za prikaz komentara na članku
 2. Baza podataka dohvata komentare na članak
 3. Korisniku se prikazuju dohvачeni komentari
- **Opis mogućih odstupanja:**
 - 2.a Na članku nema komentara
 1. Sustav dojavljuje korisniku da objava nema komentara

UC21 - Prijavljanje komentara

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:** Prijaviti neprimjeren komentar
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik bira opciju za prijavljivanje komentara
 2. Korisniku se prikazuje sučelje za opis neprimjerenog sadržaja
 3. Korisnik unosi podatke

4. Korisnik prijavljuje komentar
 5. Baza podataka sprema prijavu komentara
- **Opis mogućih odstupanja:**
 - 5.a Podatci su neispravni ili nedostaju
 1. Sustav dojavljuje korisniku da su podatci neispravni ili nedostaju

UC22 - Izmjena nečijeg članka

- **Glavni sudionik:** Moderator
- **Cilj:** Predložiti drugom korisniku izmjenu članka
- **Sudionici:** Baza podataka, prijavljeni korisnik
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Moderator bira opciju za izmjenu nečijeg članka
 2. Moderator unosi poruku korisniku i/ili predloženu izmjenu
 3. Moderator šalje obavijest korisniku
 4. Baza podataka stvara zapis obavijesti s porukom za korisnika

UC23 - Izmjena nečijeg komentara

- **Glavni sudionik:** Moderator
- **Cilj:** Predložiti drugom korisniku izmjenu komentara
- **Sudionici:** Baza podataka, prijavljeni korisnik
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Moderator bira opciju za izmjenu nečijeg komentara
 2. Moderator unosi poruku korisniku i/ili predloženu komentara
 3. Moderator šalje obavijest korisniku
 4. Baza podataka stvara zapis obavijesti s porukom za korisnika

UC24 - Brisanje članka

- **Glavni sudionik:** Moderator
- **Cilj:** Obrisati članak nekog korisnika
- **Sudionici:** Baza podataka, prijavljeni korisnik
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Moderator bira opciju za brisanje članka
 2. Moderatoru se prikazuje sučelje za brisanje članka

3. Moderator unosi dodatne informacije u vezi brisanja članka
 4. Moderator potvrđuje brisanje članka
 5. Baza podataka briše zapis članka i stvara zapis obavijesti o brisanju članka za korisnika
- **Opis mogućih odstupanja:**
 - 4.a Moderator nije potvrdio brisanje članka
 1. Članak se ne briše

UC25 - Brisanje komentara

- **Glavni sudionik:** Moderator
- **Cilj:** Obrisati komentar nekog korisnika
- **Sudionici:** Baza podataka, prijavljeni korisnik
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Moderator bira opciju za brisanje komentara
 2. Moderatoru se prikazuje sučelje za brisanje komentara
 3. Moderator unosi dodatne informacije u vezi brisanja komentara
 4. Moderator potvrđuje brisanje komentara
 5. Baza podataka briše zapis članka i stvara zapis obavijesti o brisanju komentara za korisnika
- **Opis mogućih odstupanja:**
 - 4.a Moderator nije potvrdio brisanje komentara
 1. Komentar se ne briše

UC26 - Slanje obavijesti

- **Glavni sudionik:** Moderator
- **Cilj:** Poslati obavijest jednom ili više korisnika
- **Sudionici:** Baza podataka, prijavljeni korisnik
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Moderator bira opciju za slanje obavijesti
 2. Moderatoru se prikazuje sučelje za slanje obavijesti
 3. Moderator unosi grupu korisnika kojoj šalje obavijesti i sadržaj obavijesti
 4. Moderator šalje obavijest korisnicima
 5. U bazi podataka se stvara zapis o obavijesti za sve navedene korisnike

- **Opis mogućih odstupanja:**

- 5.a Jedno od unesenih polja je prazno
 1. Sustav dojavljuje moderatoru da treba ispuniti prazno polje

UC27 - Bananje korisnika

- **Glavni sudionik:** Moderator

- **Cilj:** Zabraniti pristup korisniku na neke funkcionalnosti aplikacije

- **Sudionici:** Baza podataka, korisnik

- **Preduvjet:** Korisnik nije prethodno dobio ograničenje pristupa

- **Opis osnovnog tijeka:**

1. Moderator bira opciju za bananje korisnika
2. Moderatoru se prikazuje sučelje za bananje korisnika
3. Moderator unosi dodatne informacije o banu
4. Moderator potvrđuje bananje korisnika
5. Baza podataka spremi zapis o banu korisnika i obavijesti za korisnika o banu

- **Opis mogućih odstupanja:**

- 4.a Moderator nije potvrdio bananje korisnika

1. Korisnik se ne bana

UC28 - Dodavanje privilegiranog korisnika

- **Glavni sudionik:** Administrator

- **Cilj:** Dodati korisnika na mjesto moderatora ili administratora

- **Sudionici:** Baza podataka, prijavljeni korisnik ili moderator

- **Preduvjet:** Korisnik nije prethodno bio privilegiran

- **Opis osnovnog tijeka:**

1. Administrator bira opciju za promociju korisnika ili moderatora
2. Administratoru se prikazuje sučelje za promociju korisnika ili moderatora
3. Administrator bira želi li korisnika promovirati na moderatora i administratora
4. Administrator potvrđuje promociju
5. Baza podataka spremi novu poziciju korisnika i zapis o obavijesti o promociji korisnika

- **Opis mogućih odstupanja:**

- 4.a Administrator nije potvrdio promociju korisnika

1. Korisniku ostaju iste ovlasti koje je imao

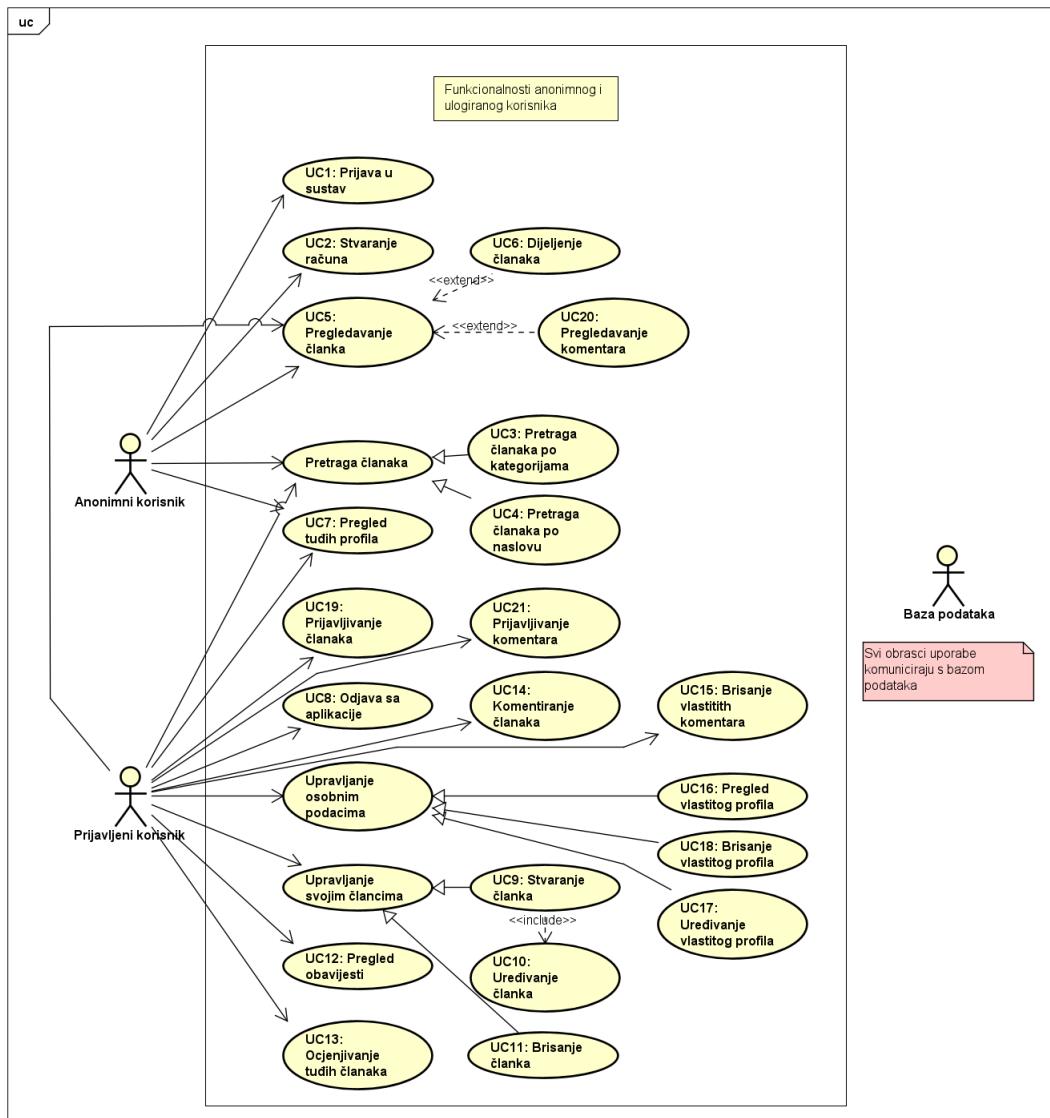
UC29 - Uklanjanje moderatora

- **Glavni sudionik:** Administrator
- **Cilj:** Ukloniti korisniku moderatorske ovlasti
- **Sudionici:** Baza podataka, moderator
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Administrator bira opciju za uklanjanje moderatora
 2. Administratoru se prikazuje sučelje za potvrdu uklanjanja moderatora
 3. Administrator potvrđuje uklanjanje moderatora
 4. Baza podataka spremi novu poziciju moderatora kao korisnika i zapis o obavijesti gubitka moderatorskih ovlasti
- **Opis mogućih odstupanja:**
 - 3.a Administrator nije potvrdio uklanjanje moderatora
 1. Moderatoru ostaju iste ovlasti koje je imao

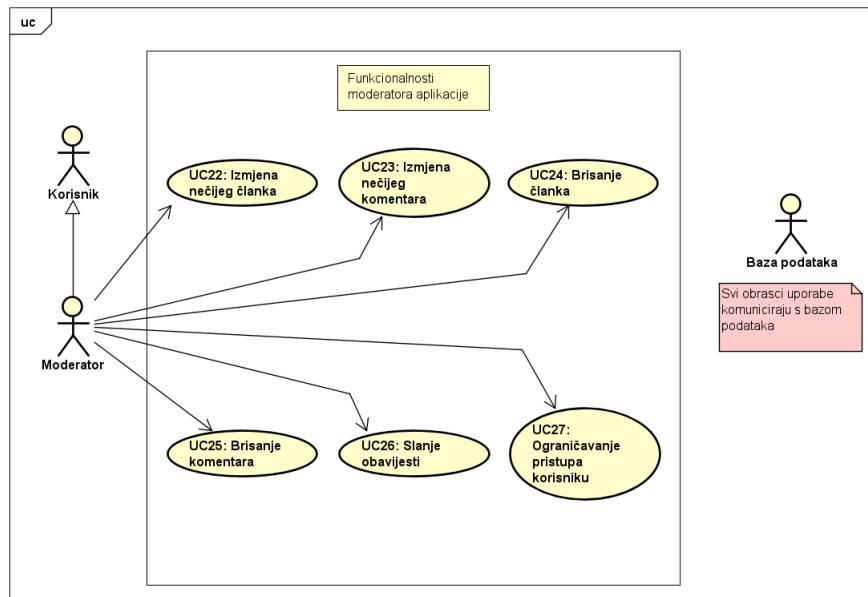
UC30 - Odstupanje s mjesta administratora

- **Glavni sudionik:** Administrator
- **Cilj:** Ukloniti sebi administratorske ovlasti
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Administrator bira opciju za odstupanje s mjesta administratora
 2. Administratoru se prikazuje sučelje za potvrdu za odstupanje
 3. Administrator potvrđuje odstupanje
 4. Baza podataka spremi novu poziciju administratora
- **Opis mogućih odstupanja:**
 - 3.a Administrator nije potvrdio odstupanje
 1. Administratoru ostaju iste ovlasti koje je imao

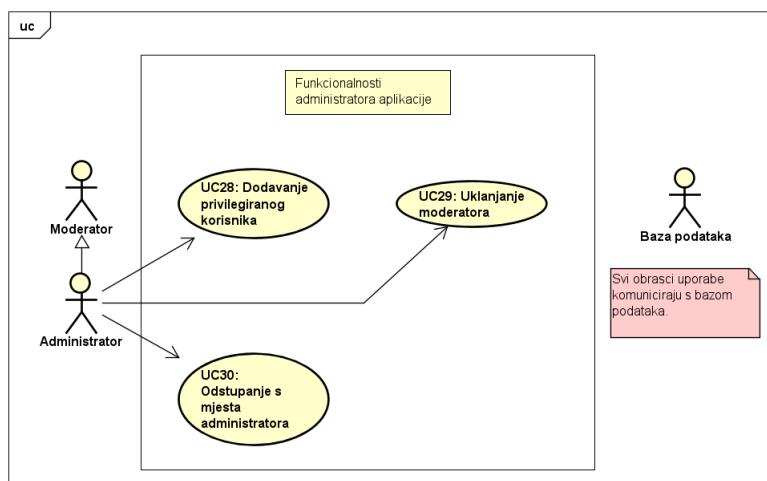
Dijagrami obrazaca uporabe



Slika 3.1: Dijagram obrasca uporabe, funkcionalnost anonimnog i prijavljenog korisnika



Slika 3.2: Dijagram obrasca uporabe, funkcionalnost moderatora sustava



Slika 3.3: Dijagram obrasca uporabe, funkcionalnost administratora sustava

3.1.2 Sekvencijski dijagrami

Obrazac uporabe UC3 - Pretraga članaka po kategorijama

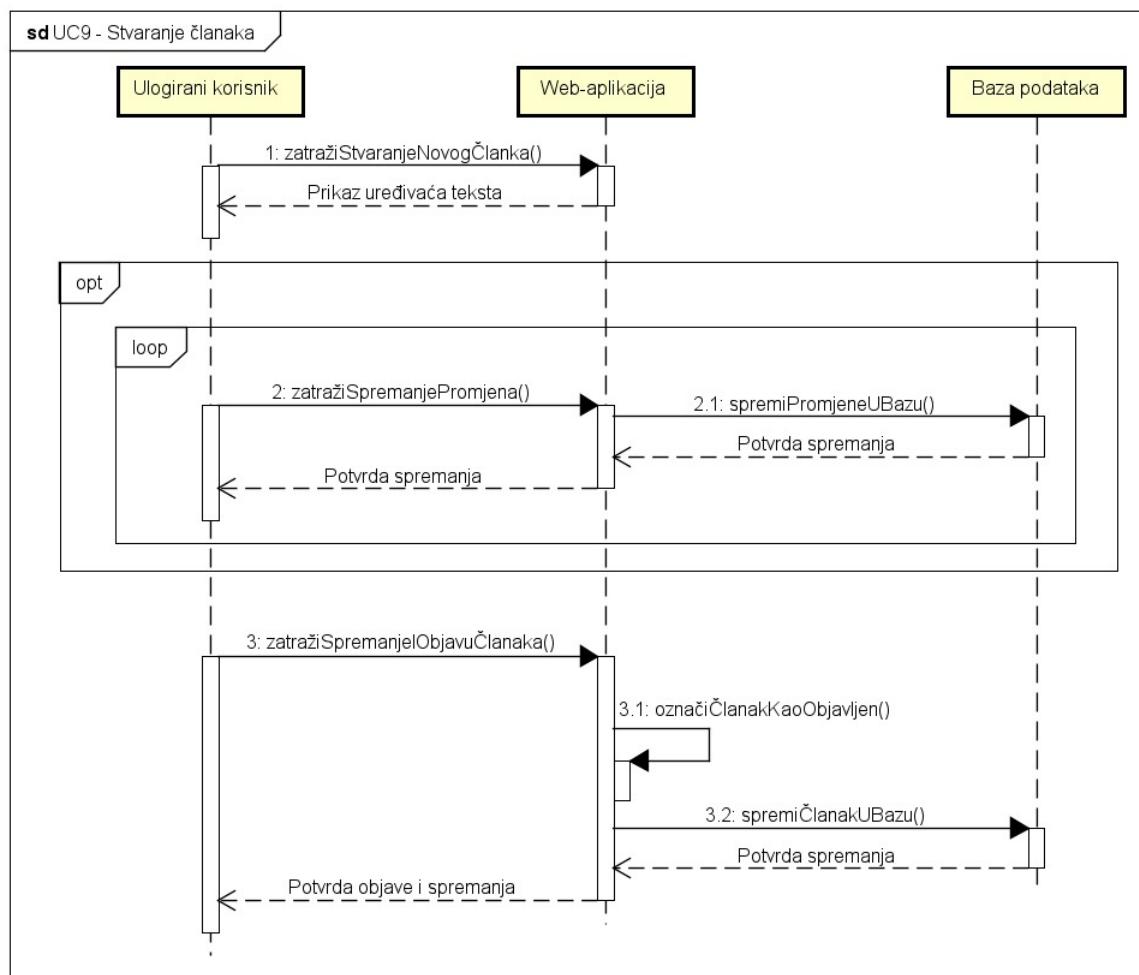
Korisnik šalje zahtjev za prikaz članaka određene kategorije. Poslužitelj dohvaća tražene članke i prikazuje ih. Ako korisnik zatraži dodavanje određenih filtera, poslužitelj dohvati tražene članke iz prethodno definirane kategorije i prikaže ih.



Slika 3.4: Sekvencijski dijagram za UC3

Obrazac uporabe UC9 - Stvaranje članaka

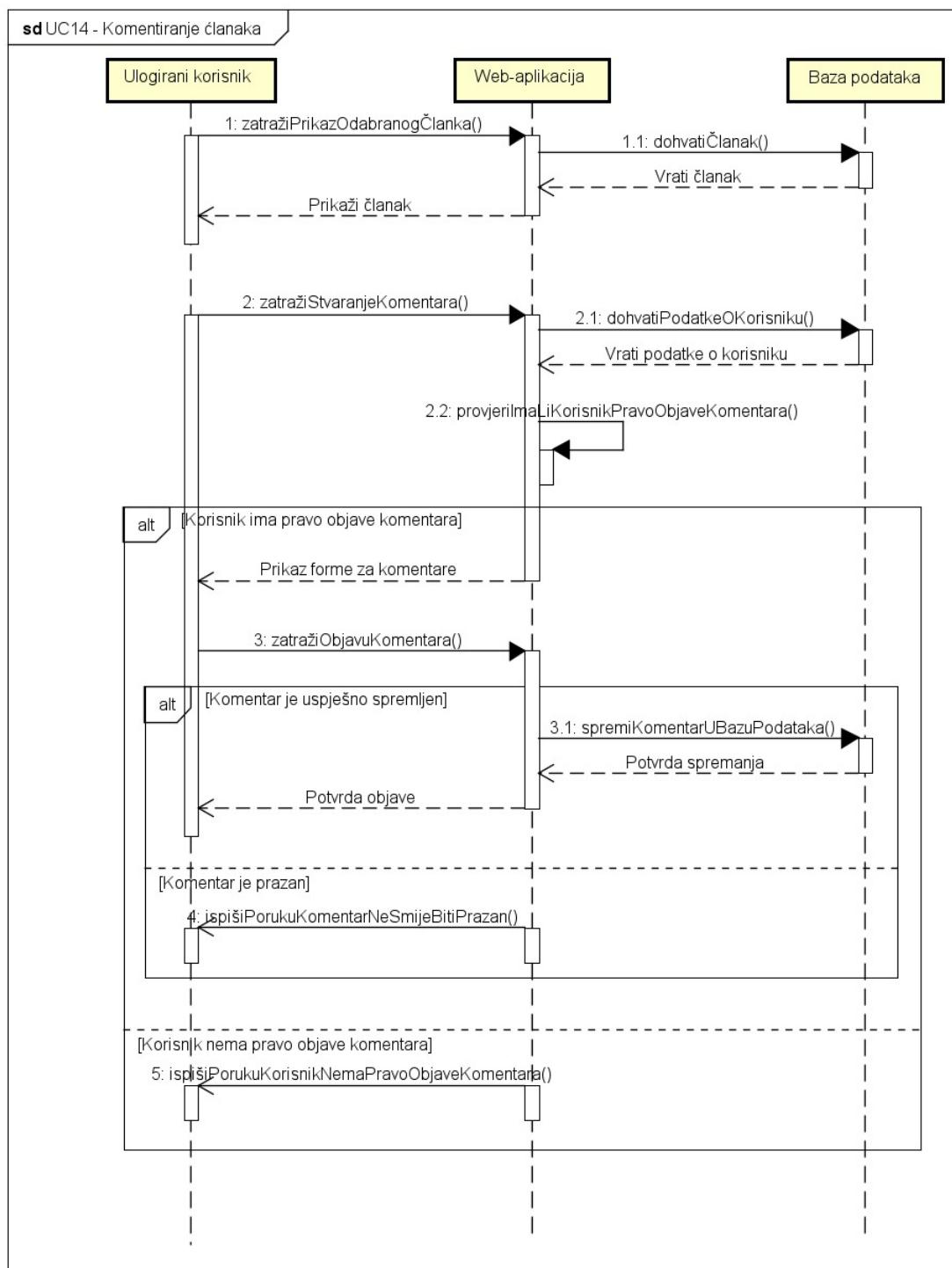
Korisnik zatraži od aplikacije stvaranje novog članka. Aplikacija prikaže uređivač teksta. Tijekom uređivanja teksta, korisnik može zatražiti pohranu promjena koje poslužitelj spremi u bazu podataka. Konačno, korisnik zatraži spremanje i objavu članka. Poslužitelj označi članak kao objavljen, spremi ga u bazu te potvrdi spremanje i objavljinje.



Slika 3.5: Sekvencijski dijagram za UC9

Obrazac uporabe UC14 - Komentiranje članaka

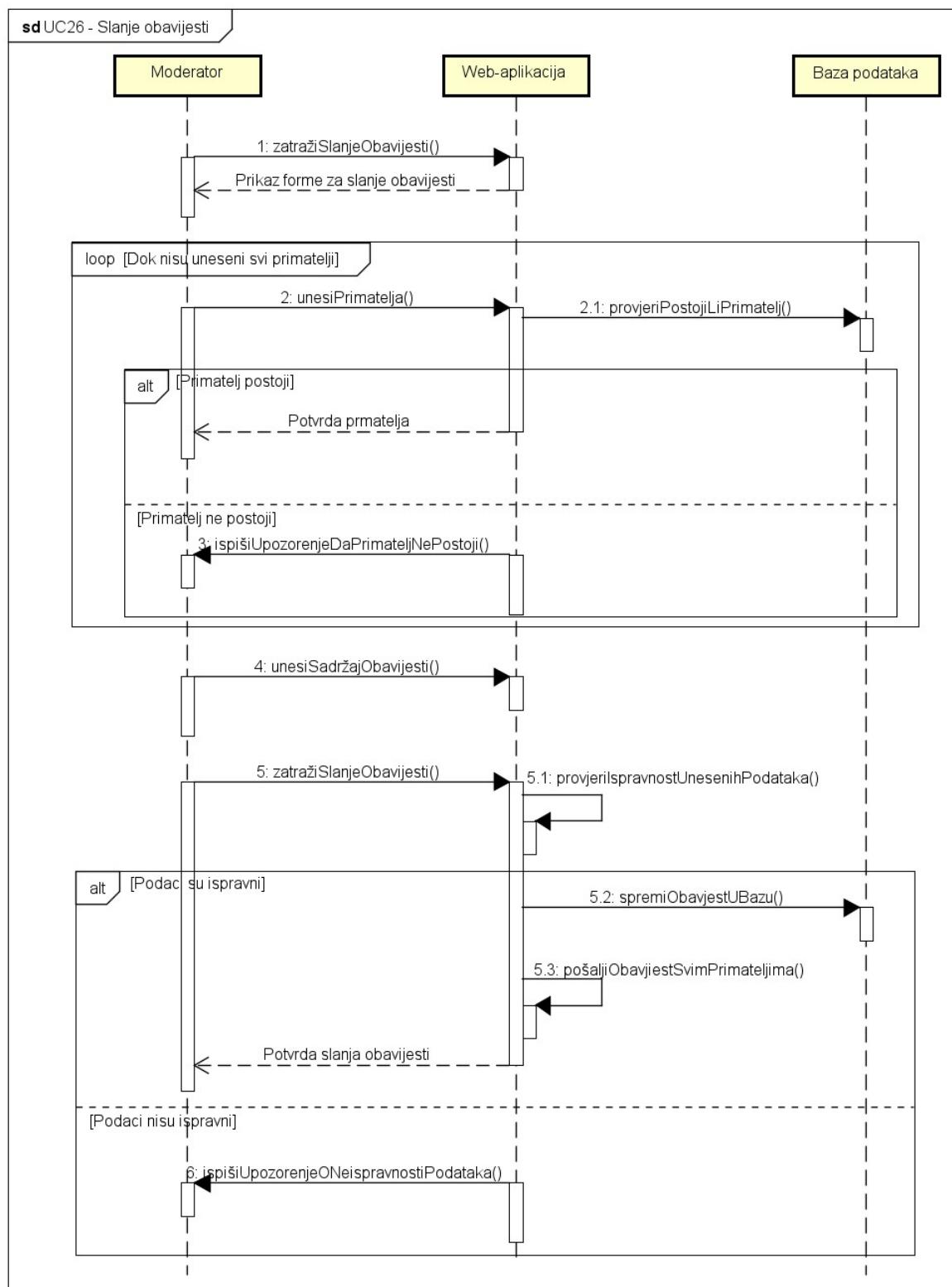
Korisnik zatraži prikaz članka koji želi komentirati. Poslužitelj dohvati članak iz baze podataka te ga prikaže korisniku zajedno s formom za ostavljanje komentara. Odabirom forme korisnik zatraži stvaranje komentara za navedeni članak. Poslužitelj prvo dohvati podatke o korisniku kako bih provjerio ima li korisnik pravo objavljivanja komentara. Ako ima, korisniku se prikaže forma za uređivanje komentara. Nakon napisanog komentara korisnik zatraži njegovu objavu. Ako je komentar uspješno spremlijen, poslužitelj ga spremi u bazu podataka te korisniku potvrdi objavu njegovog komentara. Ako je komentar prazan, korisniku se ispiše poruka da komentar ne smije biti prazan. Ako se ustanovi da korisnik nema pravo na objavu komentara, odgovarajuća forma se ne prikazuje te se korisniku ispiše odgovarajuća poruka.



Slika 3.6: Sekvencijski dijagram za UC14

Obrazac uporabe UC26 - Slanje obavijesti

Moderator zatraži slanje obavijesti (notifikacije) korisniku. Poslužitelj moderatoru prikaže formu za uređivanje obavijesti. Dok nisu uneseni svi primatelji, moderator unosi primatelje putem forme. Poslužitelj provjerava postoji li navedeni primatelj. Ako postoji, poslužitelj potvrdi moderatoru unos primatelja. U suprotnom slučaju moderatoru se ispiše upozorenje da navedeni primatelj ne postoji. Nakon upisivanja primatelja, moderator uređuje sadržaj obavijesti. Nakon što je moderator gotov s uređivanjem, zatraži slanje obavijesti. Poslužitelj provjeri ispravnost unesenih podataka. Ako su podaci ispravni, poslužitelj spremi obavijest u bazi te istu pošalje svim navedenim primateljima nakon čega potvrdi moderatoru slanje odgovarajuće obavijesti. Ako podaci nisu ispravni, moderatora se upozori da podaci nisu ispravni.



Slika 3.7: Sekvencijski dijagram za UC26

3.2 Ostali zahtjevi

1. Performanse sustava:

- Omogućiti rad više korisnika u stvarnom vremenu
- Optimizirati brzinu pristupa bazi podataka
- Osigurati učinkovitost interakcije korisnika s platformom

2. Podrška jeziku i pismu:

- Implementirati podršku hrvatskoj abecedi i dijakritičkim znakovima

3. Fleksibilnost tehnologije:

- Implementirati InterFER kao web aplikaciju
- Koristiti objektno-orientirane jezike za razvoj

4. Stabilnost i prilagodljivost:

- Osigurati da nadogradnje ne naruše postojeće funkcionalnosti
- Održavati stabilnost sustava tijekom promjena i dodataka

5. Korisničko iskustvo:

- Osigurati jednostavno i intuitivno korisničko sučelje
- Poticati aktivno sudjelovanje korisnika bez potrebe za dugim uputama

6. Zaštita baze podataka:

- Zaštititi bazu podataka od vanjskih prijetnji
- Održavati brzinu i otpornost na vanjske greške

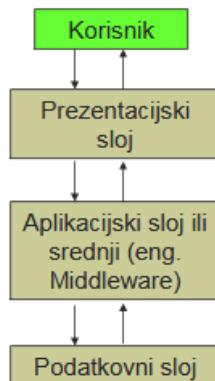
7. Pristup iz javne mreže:

- Osigurati enkripciju podataka i privatnost korisnika prilikom pristupa izvan zaštićene mreže

4. Arhitektura i dizajn sustava

Arhitektura naše aplikacije može se intuitivno podijeliti na prezentacijski, aplikacijski te podatkovni sloj. Njima redom odgovaraju idući podsustavi:

- Web aplikacija
- Web poslužitelj
- Baza podataka



Slika 4.1: Arhitektura sustava

Kako bih se korisniku omogućilo korištenje naše aplikacije, korisnik na svojem uređaju (osobno računalo, laptop, mobitel ili tablet) mora koristiti *web preglednik*, aplikaciju koja korištenjem stroja za vizualizaciju (u koji je uključen i interpreter JavaScript-a) omogućava pretvaranje koda naše web aplikacije (HTML, CSS i JavaScript) u vizualni multimedijijski sadržaj.

Za osnovnu komunikaciju s korisnikom koristi se *web aplikacija*, program koji se pomoću HTTP (engl. *Hyper Text Transfer Protocol*) protokola na zahtjev vraća korisnikovom web pregledniku. Web aplikacija služi kao sučelje između korisnika i poslovne logike aplikacije na web poslužitelju. Pomoću vizualnih komponenti korisniku se omogućava slanje HTTP zahtjeva na web poslužitelj te obrada i vizualizacija primljenih podataka u JSON (engl. *JavaScript Object Notation*) formatu u ljudski razumljivom obliku.

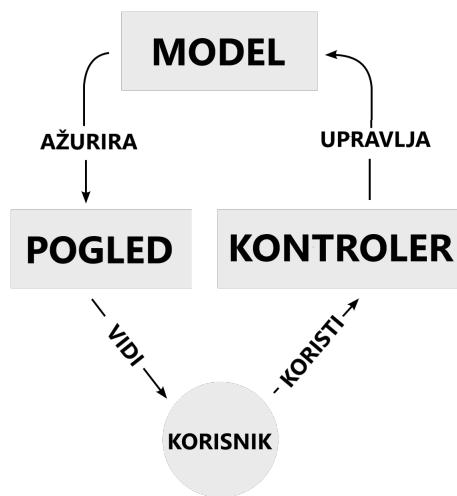
Primarni alat za komunikaciju s bazom podataka i za obradu poslovne logike je web poslužitelj kojem je omogućena komunikacija s web aplikacijom putem REST API (engl. *Representational State Transfer Application Programming Interface*) tehnologije koja uvodi standardizirani način komunikacije s klijentom razmjenom JSON poruka. Na taj se način omogućava jasno određena i istovremeno fleksibilna razmjena podataka.

Za izradu klijentske aplikacije izabrali smo jezik JavaScript zajedno s knjižnicom React koji omogućavaju jednostavnu i nativnu implementaciju arhitekture, zasnovane na događajima u klijentskom dijelu web aplikacije. Ovakav pristup omogućava dinamičko ažuriranje vizualnog prikaza na temelju korisničkih radnji i podataka, unesenih korisnikom. Kako bih se aplikacija mogla prikazati u korisnikovom web pregledniku, koristi se skupljač modula Webpack koji generira sve potrebne HTML, CSS i JavaScript datoteke koje se s poslužitelja šalju korisniku te se konačno prikazuju koristeći ugrađeni stroj za vizualizaciju web preglednika.

Za opisivanje poslovne logike naše aplikacije odabrali smo programski jezik Java zajedno s radnim okvirom Spring Boot. Ovakav izbor tehnologija obrazložen je nativnom podrškom objektno usmјerenog pristupa programskim jezicima Java te jednostavnom implementacijom MVC (engl. *Model View Controller*) arhitekturnog modela u radnom okviru Spring Boot.

MVC obrazac je stilistička varijacija arhitekture zasnovane na događajima koja se u našoj aplikaciji koristi kao alat za pojednostavljenje implementacije, testiranja i korištenja svih arhitekturnih podsustava. MVC obrazac dopušta paralelnu i neovisnu izradu tri komponente sustava na koje se intuitivno može podijeliti naša aplikacija:

- Model - Poslovna logika aplikacije koja definira i mijenja stanje koje aplikacija treba imati na temelju podataka iz kontrolera.
- View - Vizualno sučelje ili korisnička strana aplikacije koja se koristi za vizualizaciju svih podataka unutar aplikacije, primarno dobivenih s web poslužitelja.
- Controller - Upravljački dio aplikacije koji komunikacijom s korisnikom kroz vizualni sloj ažurira stanje modela i/ili vizualnog stanja aplikacije.



Slika 4.2: MVC Obrazac

4.1 Baza podataka

Za potrebe naseg sustava koristit ćemo sustav za upravljanje relacijskim bazama podataka MySQL. Odlučili smo se koristiti ovaj sustav zbog toga što je softver otvorenog koda, ima izvrsnu potporu zajednice i jer je pouzdano i robusno rješenje. Glavna značajka relacijske baze podataka jest da se podaci organiziraju u tablice definirane svojim imenom i skupom atributa, a u shemi baze podataka se definira kako se pospremaju podaci i povezuju tablice. Zadaća baze podataka je brza i jednostavna pohrana, izmjena i dohvata podataka za daljnju obradu. Baza podataka ove aplikacije sastoji se od sljedećih entiteta:

- Korisnik
- Članak
- Komentar
- Ocjena
- Notifikacija

4.1.1 Opis tablica

Korisnik Ovaj entitet sadržava sve važne informacije o korisniku aplikacije. Sadrži atribute: id, ime, prezime, email, uloga, lozinka i potvrđen. Ovaj entitet u vezi je One-to-Many s entitetom Članak preko atributa id korisnika, u vezi One-to-Many s entitetom Komentar preko atributa id korisnika, te u dvostrukoj One-to-Many vezi s entitetom Notifikacija preko atributa id korisnika.

Korisnik		
id	BIGINT	jedinstveni brojčani identifikator
Ime	VARCHAR	ime korisnika
Prezime	VARCHAR	prezime korisnika
Email	VARCHAR	e-mail adresa korisnika
Uloga	TINYINT	oznaka razine privilegije korisnika (korisnik, moderator, administrator)

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Korisnik		
Lozinka	VARCHAR	hash lozinke
Potvrđen	BOOL	oznaka da je korisnik potvrdio e-mail adresu

Članak Ovaj entitet sadržava sve važne informacije vezane za pospremljene članke. Sadrži atribute: id, autorid, naslov, sadržaj, objavljen, datumobjave, tagovi i moderiran. Ovaj entitet u vezi je Many-to-One s entitetom Korisnik preko atributa autorid, u vezi One-to-Many s entitetom Komentar preko atributa id članka, te u One-to-Many vezi s entitetom Ocjena preko atributa id članka.

Članak		
id	BIGINT	jedinstveni brojčani identifikator
autorid	BIGINT	jedinstveni brojčani identifikator autora (korisnik.id)
Naslov	VARCHAR	naslov članka
Sadržaj	TEXT	tekst članka
Objavljen	BOOL	oznaka da je članak javno dostupan
DatumObjave	DATETIME	datum i vrijeme objave članka
Tagovi	VARCHAR	popis tagova prema kojima se može pronaći članak
Moderiran	BOOL	oznaka da je članak pregledao moderator

Komentar Ovaj entitet sadržava sve važne informacije vezane za komentare na članke. Sadrži atribute: id, autorid, članakid, sadržaj, vidljivost i datumobjave. Ovaj entitet u vezi je Many-to-One s entitetom Korisnik preko atributa autorid te u vezi Many-to-One s entitetom Članak preko atributa članakid.

Komentar		
id	BIGINT	jedinstveni brojčani identifikator

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Komentar		
autorid	BIGINT	jedinstveni brojčani identifikator autora (korisnik.id)
članakid	BIGINT	jedinstveni brojčani identifikator članka (članak.id)
Sadržaj	TEXT	tekst komentara
Vidljivost	BOOL	oznaka da je komentar javno dostupan
DatumObjave	DATETIME	datum i vrijeme objave komentara

Ocjena Ovaj entitet sadržava sve važne informacije vezane za ocjene članaka. Sadrži atribute: id, članakid, ocjena. Ovaj entitet u vezi u vezi Many-to-One s entitetom Članak preko atributa članakid.

Ocjena		
id	BIGINT	jedinstveni brojčani identifikator
članakid	BIGINT	jedinstveni brojčani identifikator članka (članak.id)
Ocjena	TINYINT	ocjena članka (cijeli broj od 1 do 5)

Notifikacija Ovaj entitet sadržava sve važne informacije vezane za komentare na članke. Sadrži atribute: id, idpošiljatelj, idprimatelj, prioritet, naslov i sadržaj. Ovaj entitet je u dvostrukoj vezi Many-to-One s entitetom Korisnik preko atributa idpošiljatelj i idprimatelj.

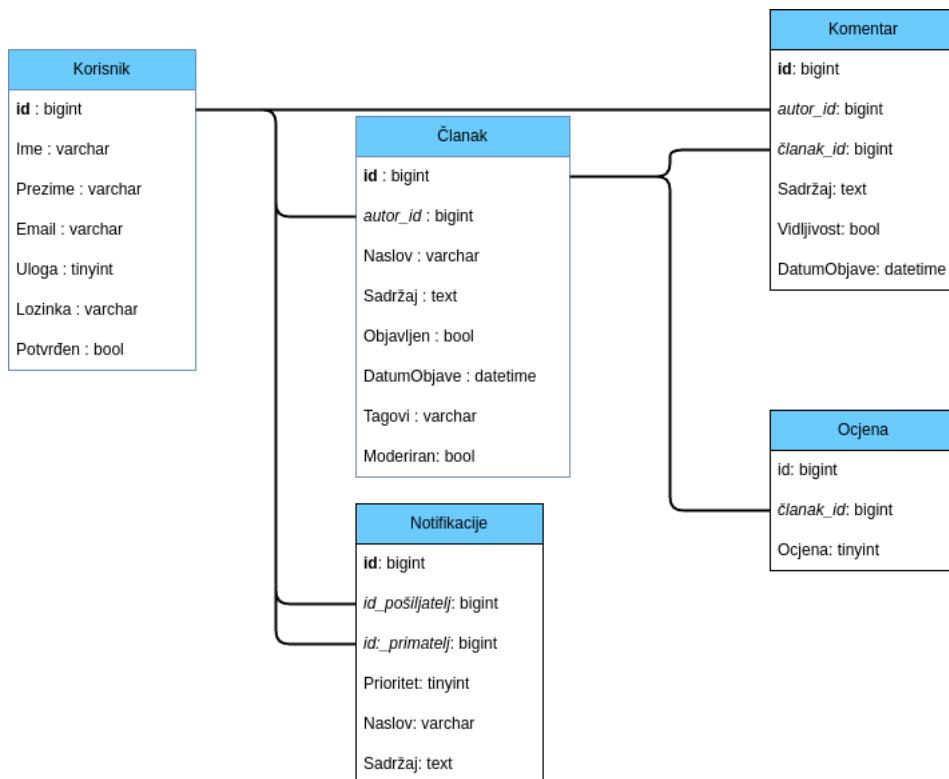
Notifikacija		
id	BIGINT	jedinstveni brojčani identifikator
idpošiljatelj	BIGINT	jedinstveni brojčani identifikator pošiljatelja (korisnik.id)
idprimatelj	BIGINT	jedinstveni brojčani identifikator primatelja (korisnik.id)

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Notifikacija		
Prioritet	TINYINT	oznaka prioriteta poruke
Naslov	VARCHAR	naslov poruke
Sadržaj	TEXT	tekst poruke

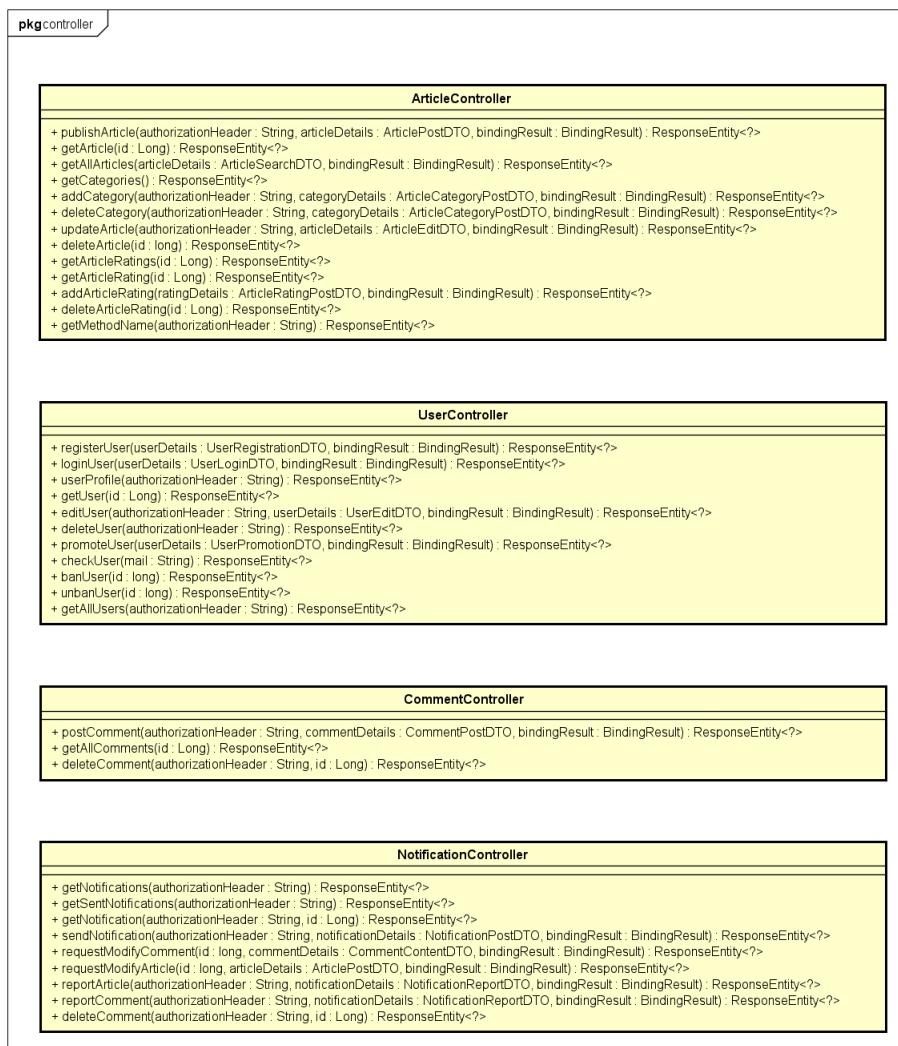
4.1.2 Dijagram baze podataka



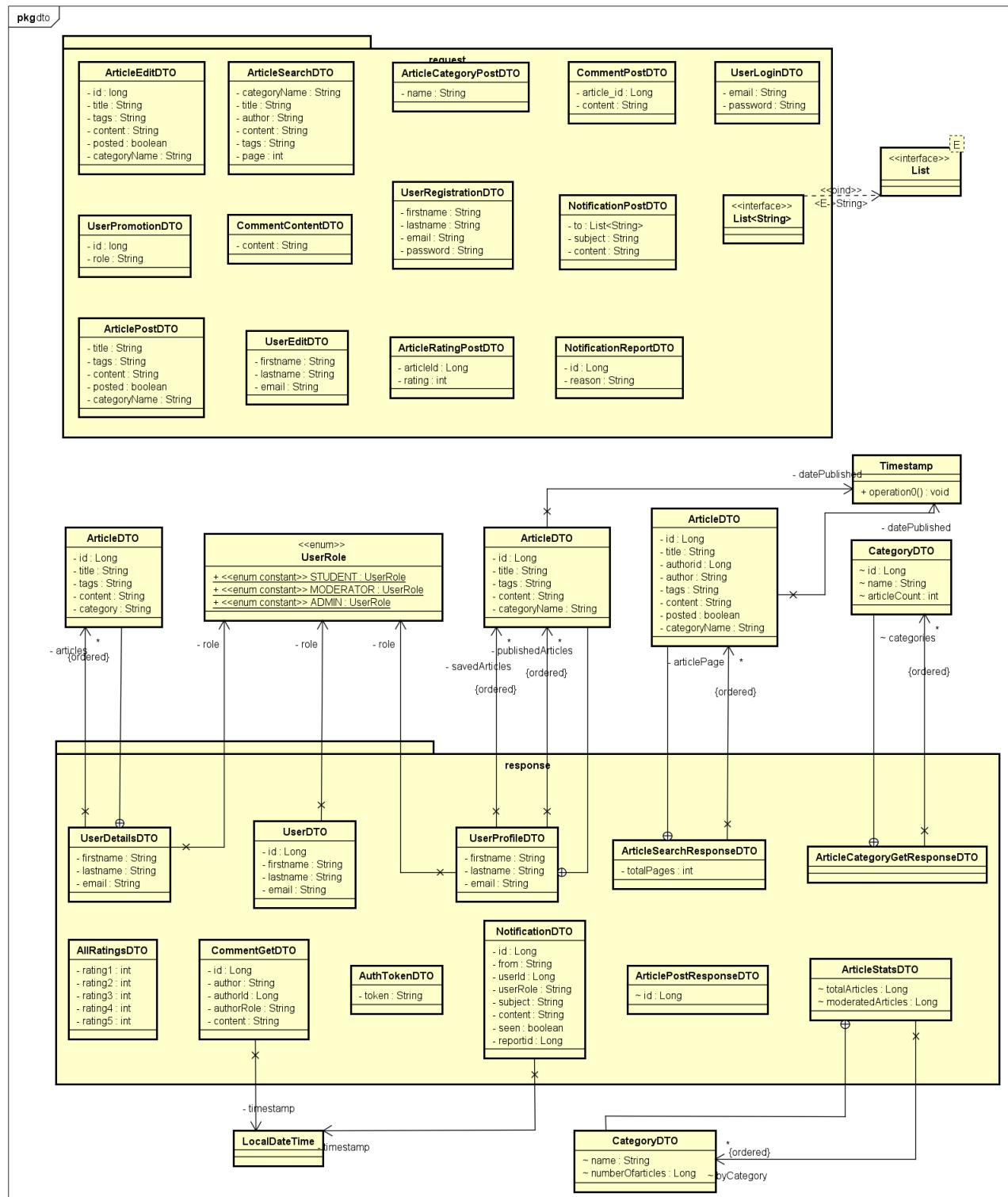
Slika 4.3: E-R dijagram baze podataka

4.2 Dijagram razreda

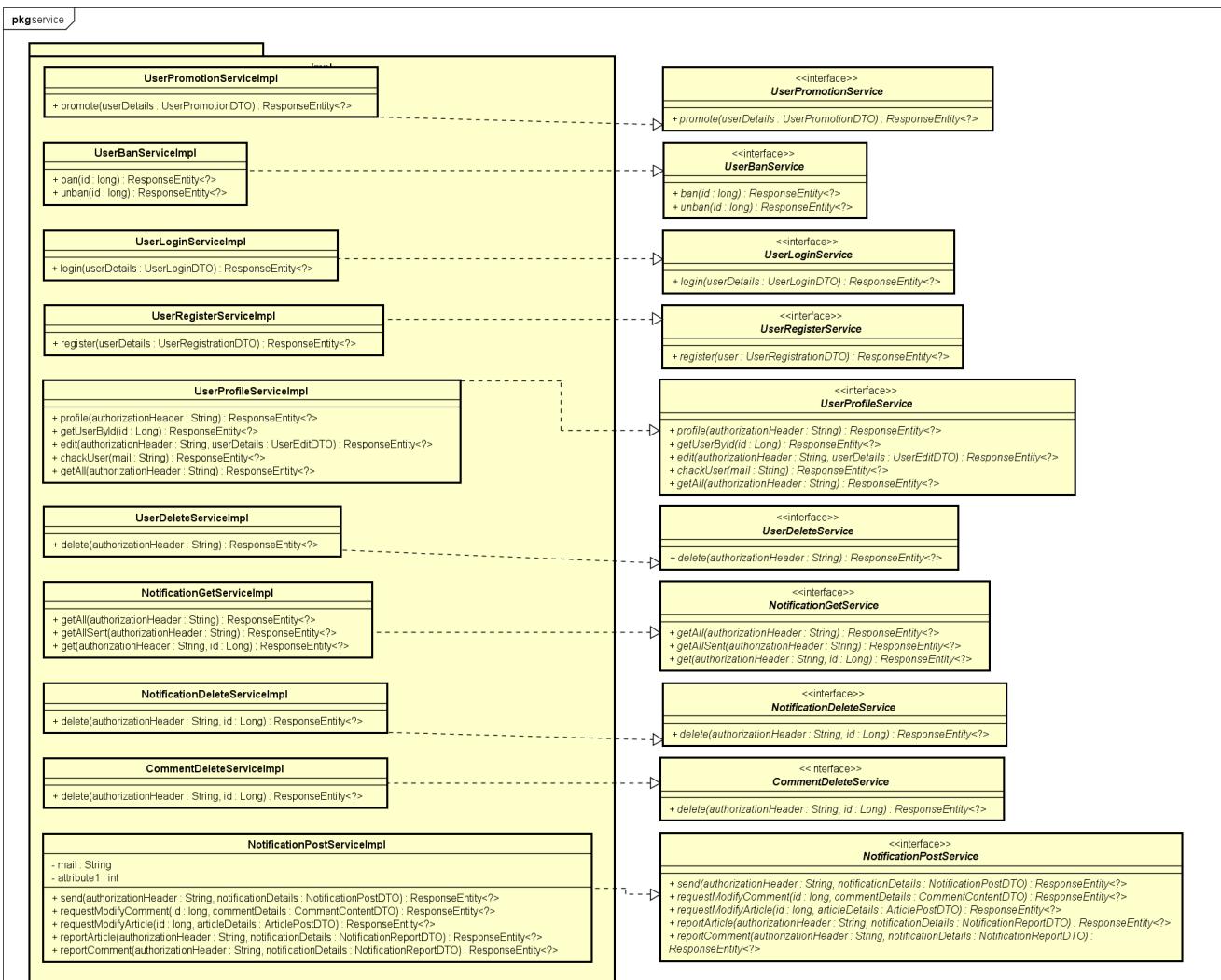
Slike od 4.4 - 4.9 prikazuju razrede koji pripadaju backend dijelu MVC arhitekture. Slika 4.4 prikazuje controller razrede koji definiraju pristupne točke i ponašanje za svaku. Osim toga manipuliraju DTO(Data transfer object), koji su definirani u paketu dto prikazan na slici 4.5 koji je podijeljen u request i response podpakete. Metode implementirane u controller razredima vraćaju JSON u tijelu te odgovarajući HTTP status kod. Logika ponašanja za svaku pristupnu točku definirana je u service razredima koje implementiraju pripadna service sučelja, a prikazani su na slici 4.6. Zbog lakše organizacije service razredi su podijeljeni po funkcionalnosti. Oni pomoću repository sučelja koja nasljeđuju JpaRepository (prikazana na slici 4.7) komuniciraju s bazom podataka. JpaRepository repository parametriziran je odgovarajućom klasom iz paketa enttiy (Slika 4.8) koje opisuju model baze. Za autentifikaciju korisnika koriste se metode implementirane u razredima paketa jwt prikazane na slici 4.9.



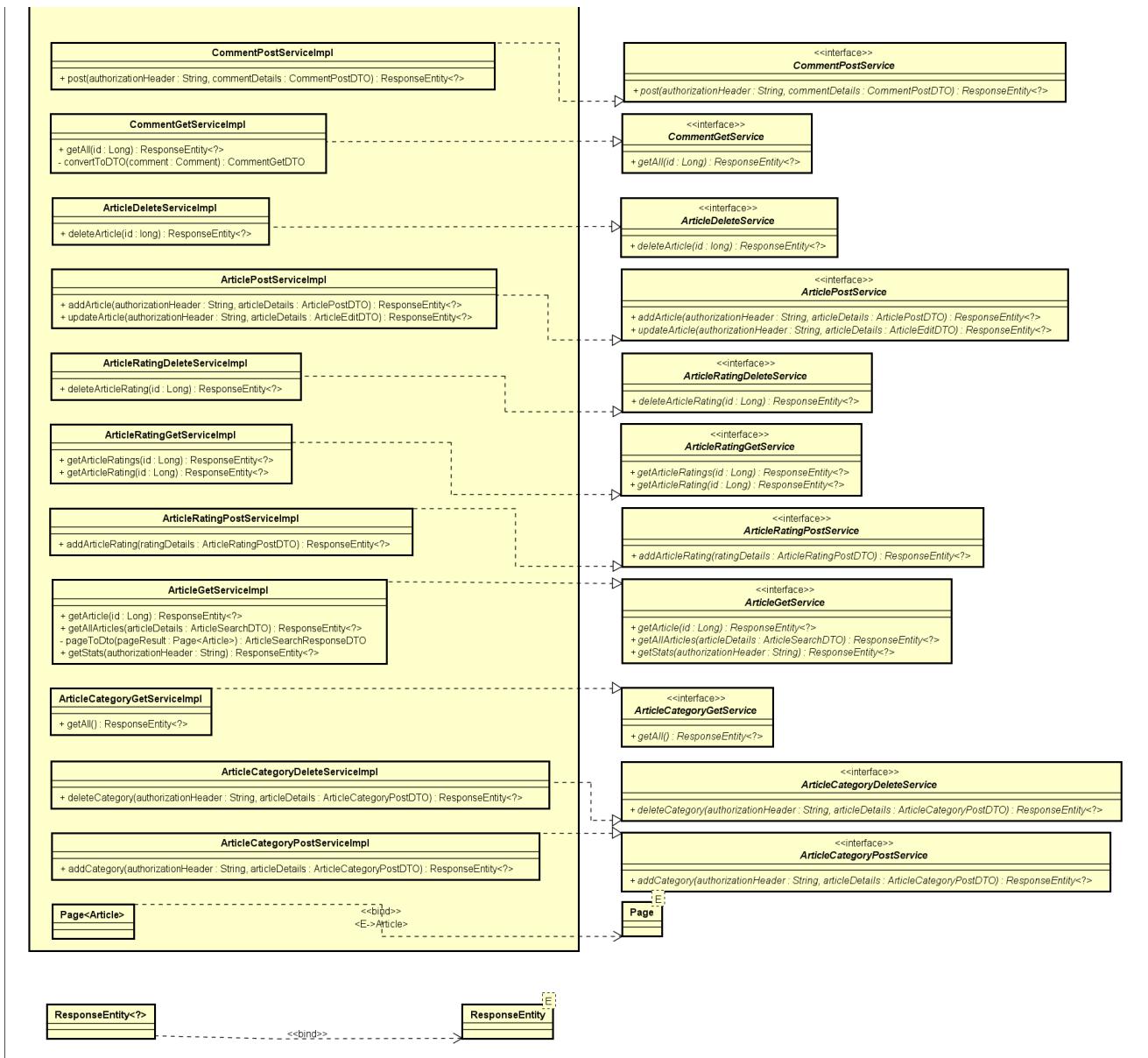
Slika 4.4: Dijagram razreda - dio Controllers



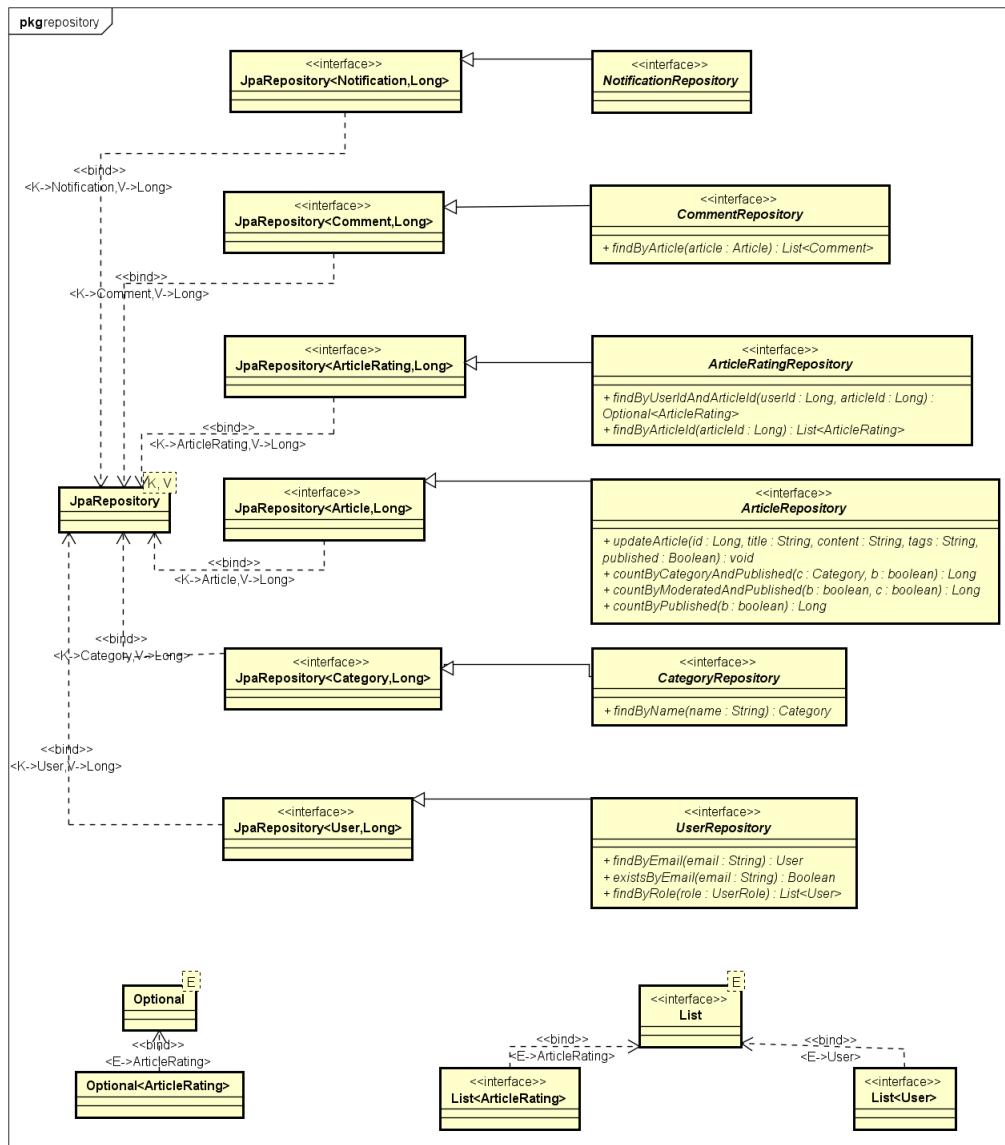
Slika 4.5: Dijagram razreda - dio DTO



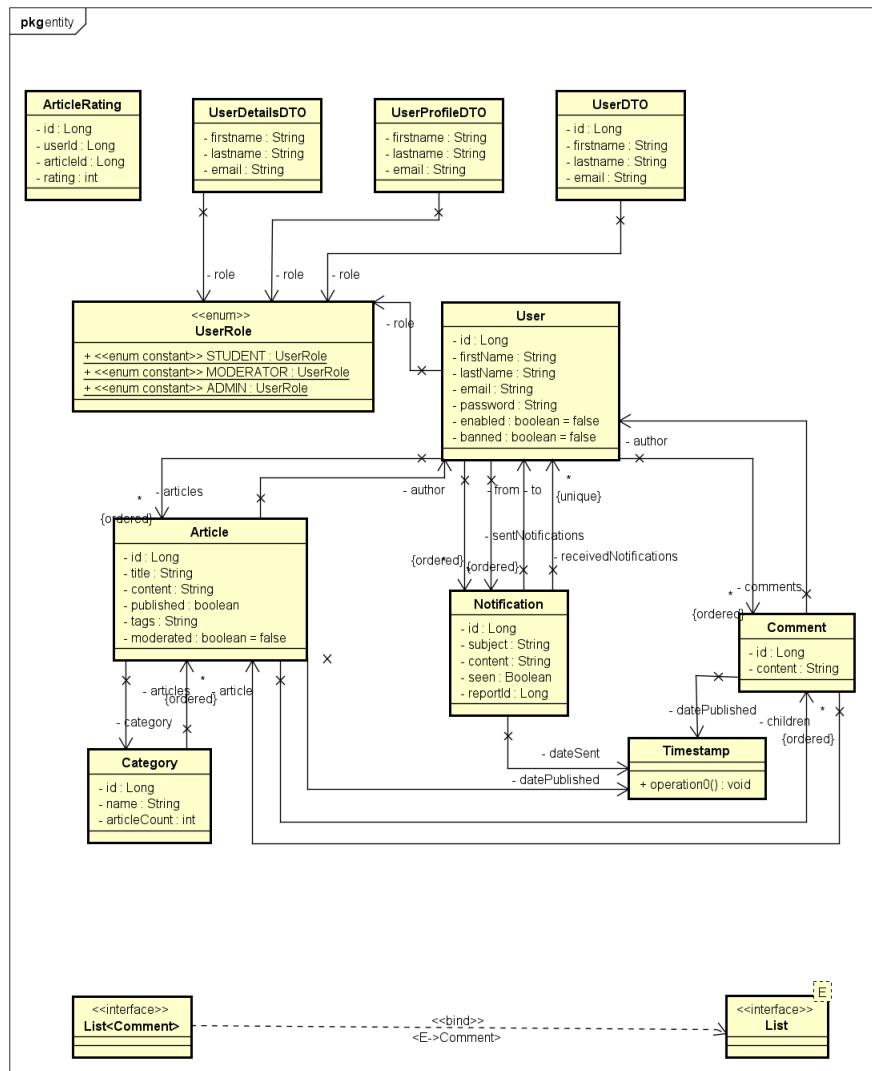
Slika 4.6: Dijagram razreda - Service dio 1



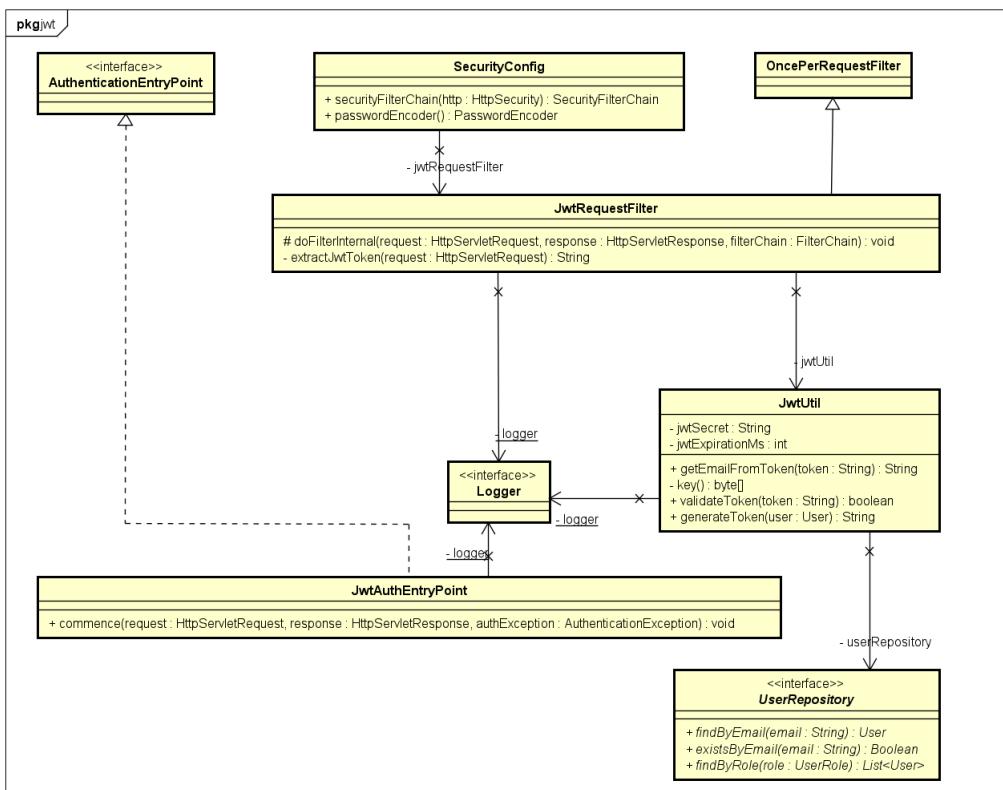
Slika 4.7: Dijagram razreda - Service dio 2



Slika 4.8: Dijagram razreda - dio Repository

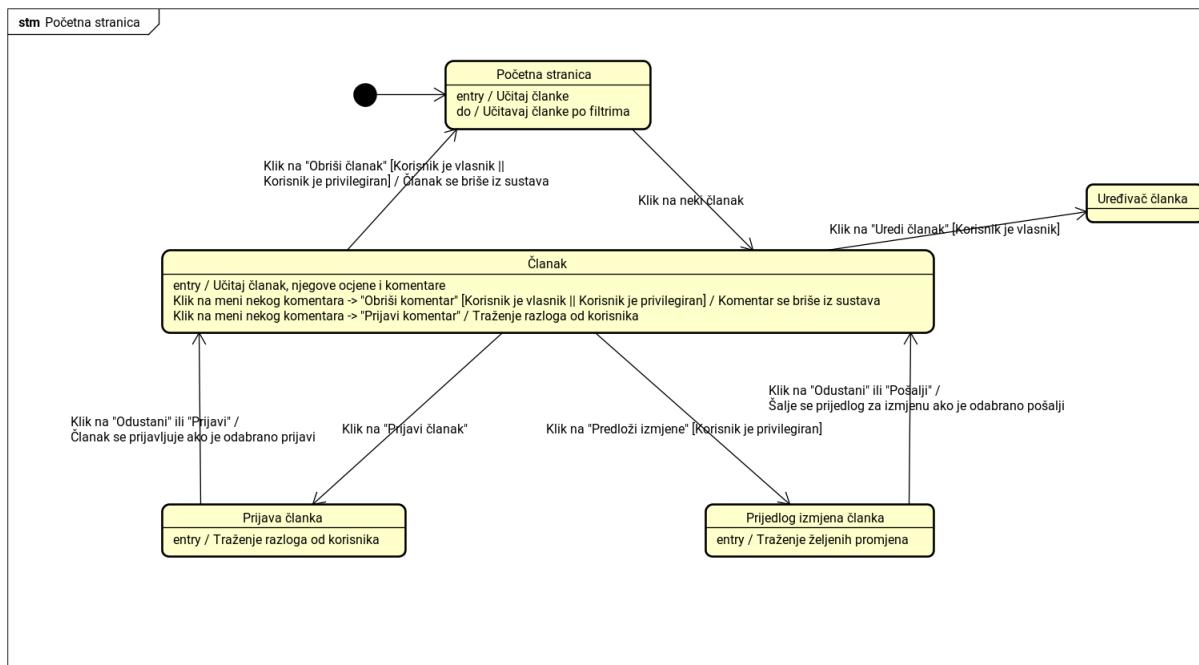


Slika 4.9: Dijagram razreda - dio Entity

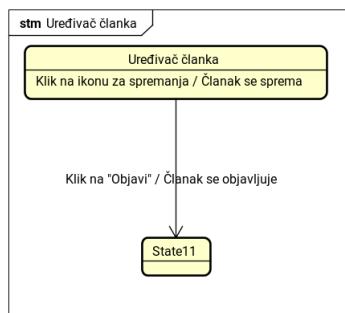


Slika 4.10: Dijagram razreda - dio JWT

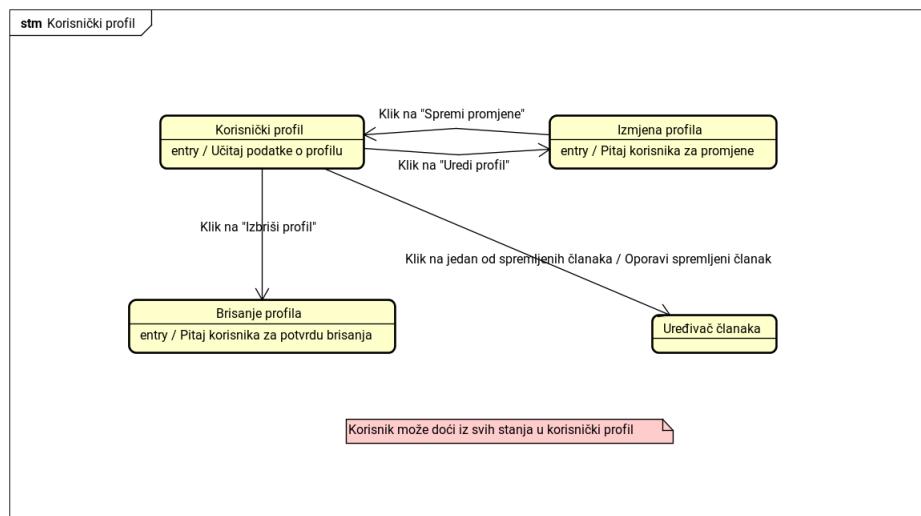
4.3 Dijagram stanja



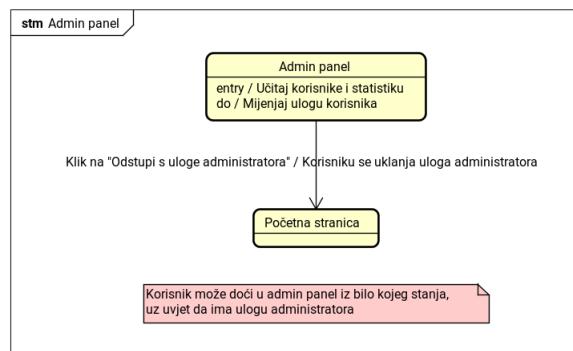
Slika 4.11: Dijagram stanja - Početna stranica



Slika 4.12: Dijagram stanja - Uređivač članka

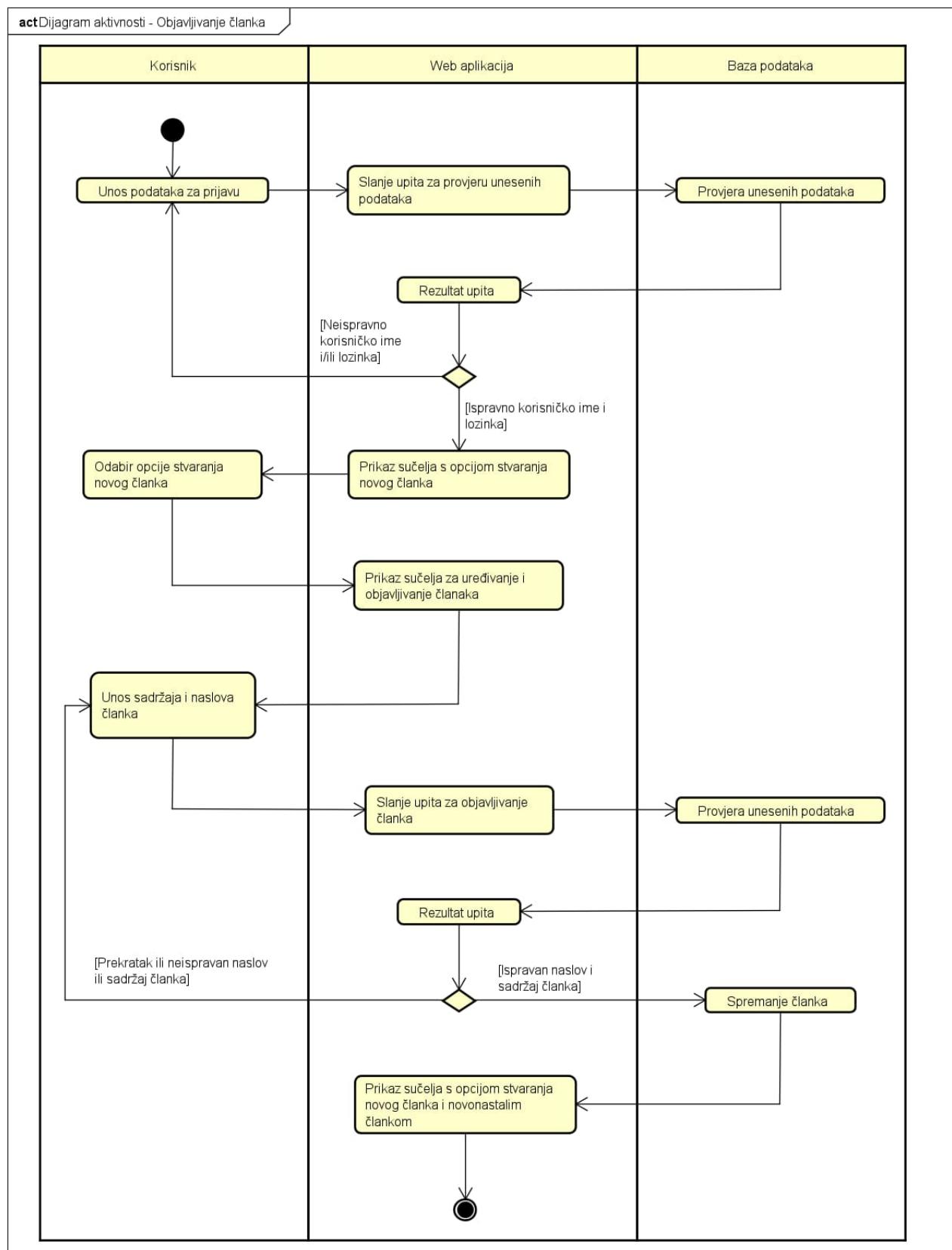


Slika 4.13: Dijagram stanja - Korisnički profil



Slika 4.14: Dijagram stanja - Admin panel

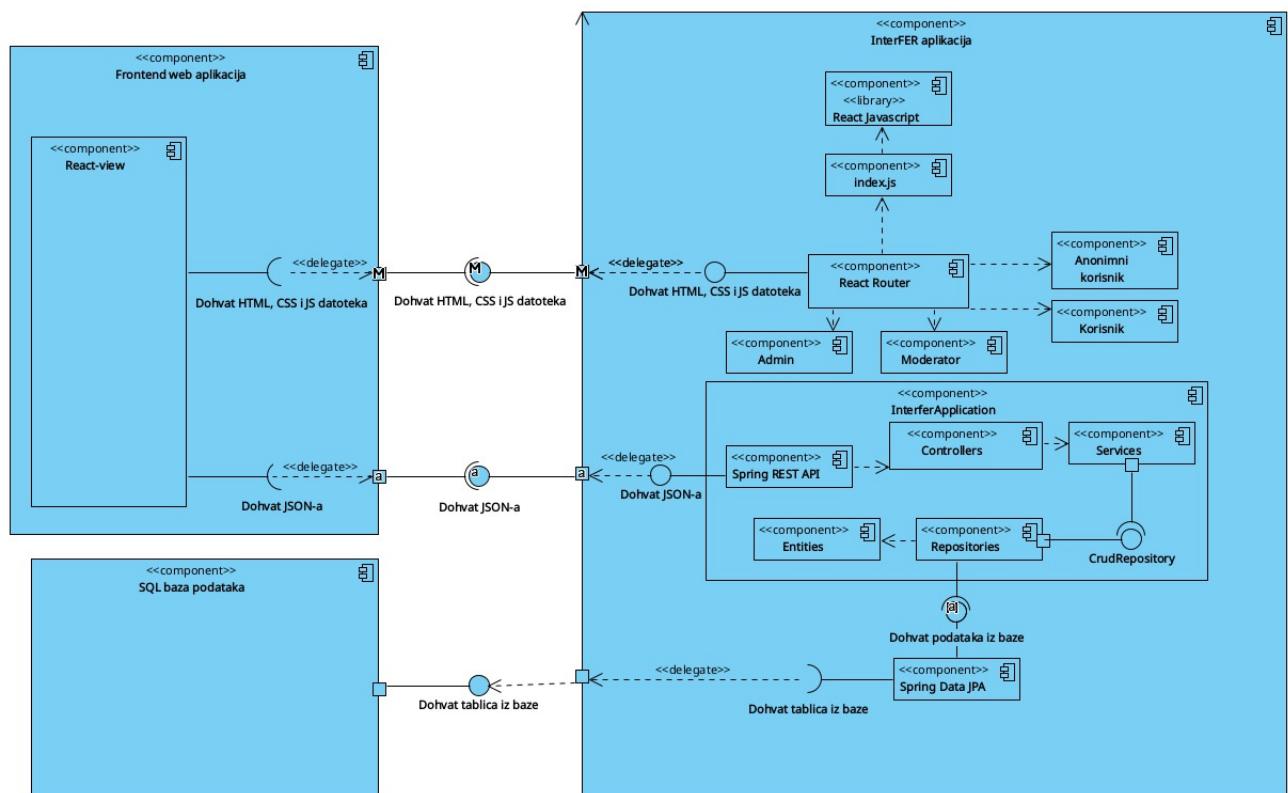
4.4 Dijagram aktivnosti



Primjena dijagrama aktivnosti odnosi se na opis modela toka upravljanja ili toka podataka opisivanog sustava. U modeliranju toka upravljanja bitno je da svaki sljedeći korak slijedi isključivo nakon završetka prethodnog, s naglaskom na jednostavnom i lako razumljivom opisu sustava. Navedeni dijagram prikazuje proces kreiranja novog članka. Nakon prijave u sustav korisnik ima opciju napraviti novi članak kojem zadaje naslov i sadržaj. Kada korisnik odabere opciju za objavu članka, sustav provjeri nisu li uneseni prekratki ili nevaljani podaci. Ako su podaci neispravni, korisnik dobije poruku s opisom neispravnosti podataka te može nastaviti uređivati članak. U slučaju da korisnik odluči objaviti članak te je unio ispravne podatke, isti se objavi te se korisnika prebací na početnu stranicu gdje može vidjeti svoj uspješno objavljeni članak na vrhu stranice.

4.5 Dijagram komponenti

Dijagram komponenti prikazan na slici opisuje međuvisnost komponenti te organizaciju i internu strukturu sustava. Pristup frontendu omogućen je preko sučelja za dohvat HTML,CSS i JS datoteka, a pristup backendu preko sučelja za dohvat JSON-a. Spring REST API zatim podatke iz JSON-a proslijeđuje kontrolerima, koji ih dalje proslijeđuju servisima na obradu. JPA Repository zadužen je za dohvat podataka iz baze. Frontend se sastoji od niza JavaScript datoteka koje ovise o React biblioteci.



Slika 4.16: Dijagram komponenti

5. Implementacija i korisničko sučelje

5.1 Korištene tehnologije i alati

5.1.1 Komunikacijske platforme

Tim je ostvarivao komunikaciju putem različitih platformi kako bi osigurao učinkovitu suradnju i komunikaciju među članovima. Komunikacija unutar tima

WhatsApp

WhatsApp poslužio je kao glavna platforma za svakodnevnu komunikaciju unutar tima. Ova aplikacija omogućila je brzu razmjenu poruka, datoteka i informacija , čime je podržala fluidnost timskih interakcija.

Microsoft Teams

Microsoft Teams postao je ključna platforma za komunikaciju s profesorima i asistentom. Integrirana s drugim alatima iz Microsoft ekosustava, Teams omogućava organizaciju sastanaka, dijeljenje dokumenata te olakšava suradnju na projektu.

Razvojno okruženje

IntelliJ IDEA

IntelliJ IDEA je integrirano razvojno okruženje (IDE) koje pruža napredne značajke za Java, Kotlin, Groovy i drugih programske jezike. Poznat po svojoj inteligentnoj podršci za refaktoriranje, automatsko dovršavanje koda te integraciji s popularnim alatima poput Mavena i Gradlea.

Neovim

Neovim je unaprijeđeni klon klasičnog Vim uređivača teksta. Pruža napredne mogućnosti uređivanja teksta, podršku za različite programerske jezike i proširivost putem raznih dodataka. Idealno rješenje za korisnike koji cijene efikasnost u radu s tekstualnim datotekama.

Visual Studio Code

Visual Studio Code je uređivač teksta koji je vrlo brzo postao favorit među programerima. Danas je gotovo sveprisutan, a svoju popularnost uveliko duguje moćnoj proširivosti, sa proširenjima za bilo koju potrebu koja se može zamisliti.

5.1.2 Razvoj aplikacije

Aplikacija je izgrađena koristeći sljedeće tehnologije:

Backend

Spring: Framework za Java aplikacije koji omogućuje brz i efikasan razvoj. Koristi se za izgradnju robustnih backend komponenti.

Frontend

React: Biblioteka za izgradnju korisničkih sučelja u JavaScriptu. Omogućuje razvoj visoko modularnih i brzih aplikacija.

JavaScript: Programska jezika koji se koristi u razvoju frontend dijela aplikacije.

Baza podataka

PostgreSQL: Sustav upravljanja bazama podataka otvorenog koda, sa jakim fokusom na ACID principe, i pridržavanje SQL standarda.

5.2 Ispitivanje programskog rješenja

5.2.1 Ispitivanje komponenti

Za ispitivanje komponenti, koristili smo JUnit. Svi testovi su napravljeni za UserRepository klasu, koja je zadužena za sučelje s bazom podataka.

Testovi su napravljeni po AAA (Arrange, Act, Assert) uzorku, čiji je princip vrlo jednostavan. Prvo treba napraviti pripreme (kao što su varijable koje ćemo koristiti), nakon čega pozivamo kod čiju funkcionalnost želimo testirati. Na posljetku provjeravamo je li nam taj kod vratio očekivan rezultat.

Zbog mogućnosti reproduciranja testova, u razredu u kojem se nalaze testovi se koristi in-memory baza podataka, te test neće smetati podacima u "stvarnom svijetu", i obrnuto.

```
@Test
public void testSave() {
    var user = User
        .builder()
        .firstName("John")
        .lastName("Doe")
        .email("john.doe@fer.hr")
        .password("password")
        .role(UserRole.STUDENT)
        .build();

    var savedUser = userRepository.save(user);

    assertNotNull(savedUser);
    assertTrue(savedUser.getId() > 0);
}
```

Slika 5.1: Ispitivanje spremanja korisnika u bazu podataka

Test počinje stvaranjem korisnika, koji se potom sprema ugrađenom metodom u klasi JpaRepository.

Nakon toga provjerava da korisnik nije null, te da je njegov id veći od 0, što nam potvrđuje da je korisnik spremlijen u bazu podataka.

```
@Test
public void testFindAll() {
    var user = User
        .builder()
        .firstName("John")
        .lastName("Doe")
        .email("john.doe@fer.hr")
        .password("password")
        .role(UserRole.STUDENT)
        .build();
    var user2 = User
        .builder()
        .firstName("John")
        .lastName("Doe")
        .email("john.doe2@fer.hr")
        .password("password")
        .role(UserRole.STUDENT)
        .build();

    userRepository.save(user);
    userRepository.save(user2);

    List<User> userList = userRepository.findAll();

    assertNotNull(userList);
    assertEquals(2, userList.size());
}
```

Slika 5.2: Ispitivanje spremanja dva korisnika u bazu podataka

Test počinje stvaranjem dva korisnika, koje spremamo u bazu podataka. Nakon toga dohvaćamo sve korisnike iz baze podataka.

Za dohvaćene podatke provjeravamo da je lista dohvaćenih korisnika duljine 2, što odgovara broju korisnika koji smo spremili u prošlom koraku.

```
@Test
public void testDuplicateEmail() {
    var user = User
        .builder()
        .firstName("John")
        .lastName("Doe")
        .email("john.doe@fer.hr")
        .password("password")
        .role(UserRole.STUDENT)
        .build();
    var user2 = User
        .builder()
        .firstName("John")
        .lastName("Doe")
        .email("john.doe@fer.hr")
        .password("password")
        .role(UserRole.STUDENT)
        .build();

    userRepository.save(user);

    var thrown = false;

    try {
        userRepository.save(user2);
    } catch (DataIntegrityViolationException e) {
        thrown = true;
    }

    assertTrue(thrown);
}
```

Slika 5.3: Ispitivanje jedinstvenosti email adrese

Test, kao i zadnji počinje stvaranjem i spremanjem dva korisnika, međutim, korisnici imaju identičnu email adresu.

Ovdje bi kod trebao baciti iznimku, na osnovu greške u integritetu baze podataka, te za to koristimo try-catch blok, koji, u slučaju da je iznimka uhvaćena, namješta varijablu thrown na true, što kasnije provjeravamo u assertTrue metodi.

```
@Test
public void testFindById() {
    var user = User
        .builder()
        .firstName("John")
        .lastName("Doe")
        .email("john.doe@fer.hr")
        .password("password")
        .role(UserRole.STUDENT)
        .build();

    userRepository.save(user);

    var userFetched = userRepository.findById(user.getId());
    userFetched.orElseThrow();
}
```

Slika 5.4: Ispitivanje dohvatanja korisnika po id-u

Test počinje stvaranjem korisnika, i spremanjem istog u bazu podataka. Nakon toga dohvaćamo korisnika iz baze podataka na osnovu id-a upravo spremlijenog korisnika.

Kod bi trebao dohvatiti korisnika, jer znamo da je taj korisnik upravo spremljen.

```
@Test
void testExistsByEmail() {
    var email = "john.doe@fer.hr";

    var user = User
        .builder()
        .firstName("John")
        .lastName("Doe")
        .email(email)
        .password("password")
        .role(UserRole.STUDENT)
        .build();

    userRepository.save(user);

    assertTrue(userRepository.existsByEmail(email));
}
```

Slika 5.5: Ispitivanje postojanja korisnika po email adresi

Test počinje stvaranjem korisnika, i spremanjem istog u bazu podataka. Nakon toga pomoću vlastite metode provjeravamo je li korisnik sa tom email adresom postoji u bazi podataka.

```
@Test
void testFindByRole() {
    var role = UserRole.STUDENT;

    var user = User
        .builder()
        .firstName("John")
        .lastName("Doe")
        .email("john.doe@fer.hr")
        .password("password")
        .role(role)
        .build();

    userRepository.save(user);

    var userList = userRepository.findByRole(role);
    assertFalse(userList.isEmpty());
}
```

Slika 5.6: Ispitivanje dohvatanja korisnika po ulozi

Test počinje stvaranjem korisnika, i spremanjem istog u bazu podataka. Nakon toga pomoću vlastite metode provjeravamo je dohvачen korisnik sa tom ulogom.

5.2.2 Ispitivanje sustava

Ispitivanje sustava provedeno je koristeći Selenium WebDriver¹ unutar JUnit testova. Testovima se ispituje funkcionalnost prijave u sustav. U svakom testu se u formu za prijavu unose e-mail i lozinka.

Ispitni slučaj 1: Pokušaj prijave u sustav s neispravnim e-mail adresom Test u formu za prijavu unosi e-mail adresu koja nije iz akademske zajednice (npr. @fer.hr).

Kod 5.1

```
@Test
public void loginTest_neispravan_mail() {

    setUp();
    driver.get(url);

    WebElement element = driver.findElement(By.cssSelector("input[type='email']"));
    element.sendKeys("krivimail@gmail.com");
```

¹<https://www.selenium.dev/documentation/webdriver/>

```
element= driver.findElement(By.cssSelector("input[type='password']"));
element.sendKeys("password");

driver.findElement(By.cssSelector("button")).click();

assertEquals(true, driver.findElement(By.cssSelector("div[class='MuiAlert-
message css-1xsto0d']")).getText().equals("Molimo vas da koristite svoj sveučilišni
driver.quit();
}
```

Ispitni slučaj 2: Pokušaj prijave u sustav s lozinkom koja ne sadrži barem 1 numerički znak

Kod 5.2

@Test

```
public void loginTest_neispravna_lozinka() {

setUp();
driver.get(url);

WebElement element = driver.findElement(By.cssSelector("input[type='email']"));
element.sendKeys("krivimail@fer.hr");

element= driver.findElement(By.cssSelector("input[type='password']"));
element.sendKeys("password");

driver.findElement(By.cssSelector("button")).click();

assertEquals(true, driver.findElement(By.cssSelector("div[class='MuiAlert-
message css-1xsto0d']")).getText().equals("Lozinka mora sadržavati barem jednu znamku
driver.quit();
}
```

Ispitni slučaj 3: Pokušaj prijave u sustav s e-mail adresom koja nije registrirana u sustavu

Kod 5.3

```
@Test
public void loginTest_nepostojeci_korisnik() {

    setUp();
    driver.get(url);

    WebElement element = driver.findElement(By.cssSelector("input[type='email']"));
    element.sendKeys("nepostojeci@fer.hr");

    element= driver.findElement(By.cssSelector("input[type='password']"));
    element.sendKeys("password123");

    driver.findElement(By.cssSelector("button")).click();

    assertEquals(true, driver.findElement(By.cssSelector("div[class='MuiAlert-message css-1xsto0d']")).getText().equals("Pogrešni podatci."));
    driver.quit();
}
```

Ispitni slučaj 4: Pokušaj prijave u sustav s krivom lozinkom Test u formu za prijavu unosi e-mail adresu korisnika koji je registriran u sustavu i krivu lozinku.

Kod 5.4

```
@Test
public void loginTest_kriva_lozinka() {

    setUp();
    driver.get(url);

    WebElement element = driver.findElement(By.cssSelector("input[type='email']"));
    element.sendKeys("1d@fer.hr");

    element= driver.findElement(By.cssSelector("input[type='password']"));
    element.sendKeys("krivalozinka123");

    driver.findElement(By.cssSelector("button")).click();
```

```
assertEquals(true, driver.findElement(By.cssSelector("div[class='MuiAlert-message css-1xsto0d']")).getText().equals("Pogrešni podatci."));  
driver.quit();  
}
```

Ispitni slučaj 5: Pokušaj prijave u sustav s ispravnim podacima

Kod 5.5

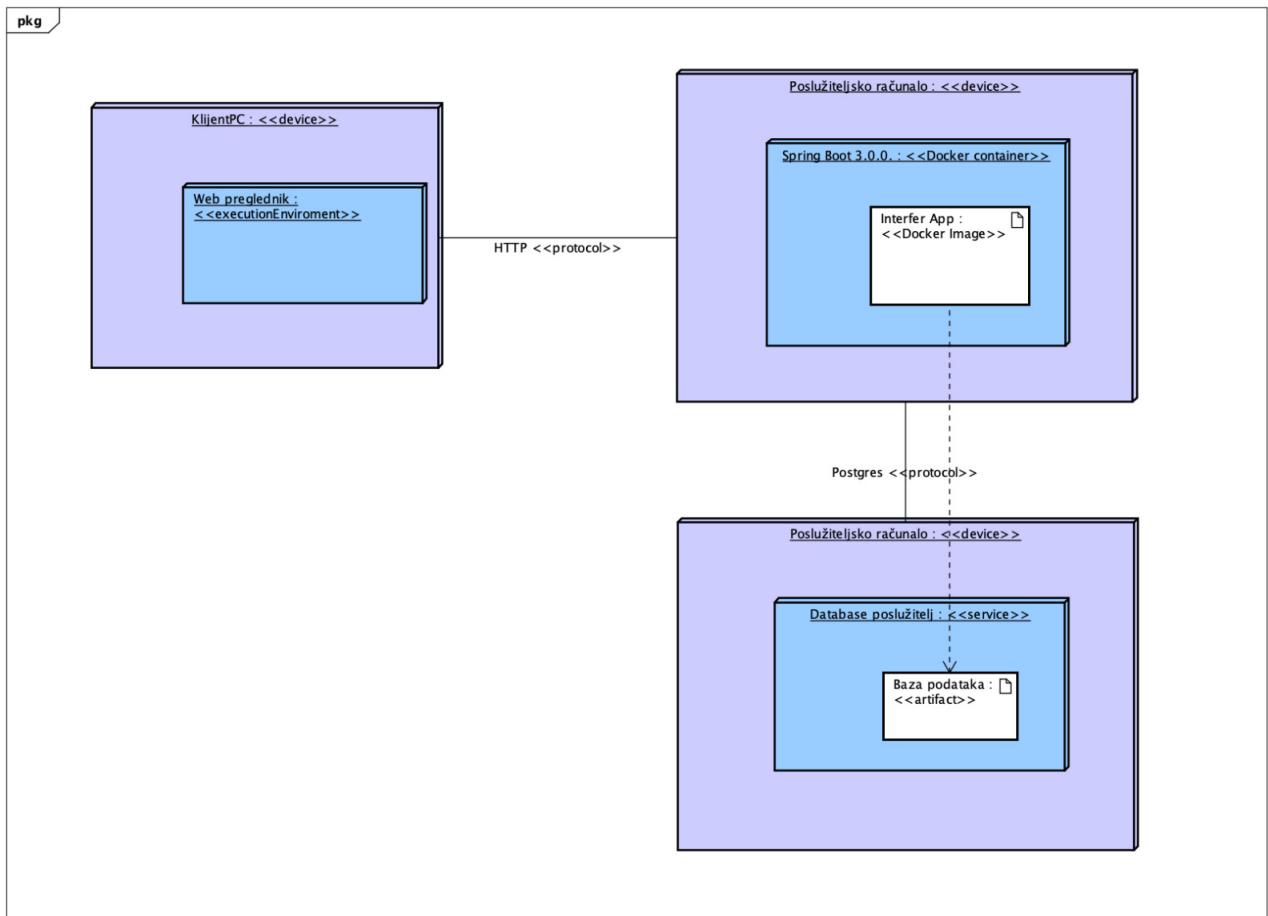
```
@Test  
public void loginTest_ispravan() {  
  
    setUp();  
    driver.get(url);  
  
    WebElement element = driver.findElement(By.cssSelector("input[type='email']"));  
    element.sendKeys("1d@fer.hr");  
  
    element= driver.findElement(By.cssSelector("input[type='password']"));  
    element.sendKeys("Lozinka123");  
  
    driver.findElement(By.cssSelector("button")).click();  
  
    WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(10)); // seconds  
    wait.until(ExpectedConditions.visibilityOfElementLocated(By.cssSelector("svg")));  
    String redirURL = driver.getCurrentUrl();  
  
    assertEquals(true, !redirURL.contains("login"));  
    driver.quit();  
}
```

```

✓ ⓘ interfer 12.6s
  ✓ ⓘ hr.fer.progi.interfer 12.6s
    ✓ ⓘ InterferApplicationTests 12.6s
      ⓘ loginTest_neispravan_mail() 1.7s
      ⓘ loginTest_neispravna_lozinka() 2.3s
      ⓘ loginTest_nepostojeci_korisnik() 3.2s
      ⓘ loginTest_kriva_lozinka() 2.7s
      ⓘ loginTest_ispravan() 2.7s
  
```

Slika 5.7: Prikaz uspješnog izvršenja testova

5.3 Dijagram razmještaja



Slika 5.8: Dijagram Razmještaja

Opis UML dijagrama razmještaja za sustav komunikacije između dva poslužiteljska računala, jednog s Spring Boot 3.0.0. poslužiteljem i drugog s poslužiteljem baze

podataka, pruža uvid u organizaciju elemenata sustava. Ovaj dijagram naglašava strukturu i komunikacijske veze u arhitekturi "klijent - poslužitelj".

Na prvom poslužiteljskom računalu smješten je Spring Boot 3.0.0. poslužitelj, čija je uloga centralna u obradi i upravljanju poslovnim logikama sustava. Ovaj poslužitelj komunicira s drugim poslužiteljem, gdje se nalazi baza podataka Postgres, čime se uspostavlja stabilna veza za pohranu i dohvaćanje podataka. Ova jasna podjela omogućuje učinkovito upravljanje podacima unutar sustava.

Klijenti, bilo da su korisnici, zaposlenici, vlasnici ili administratori, pristupaju sustavu putem web preglednika. U arhitekturi "klijent - poslužitelj", komunikacija između računala korisnika i poslužitelja odvija se putem HTTP veze. Ova veza osigurava siguran prijenos podataka, omogućujući interakciju s web aplikacijom na intuitivan način. Kroz korištenje HTTP veze, pruža se skalabilnost i pristupačnost sustava, što je ključno za korisničko iskustvo.

Dijagram razmještaja također naglašava važnost organizacije komponenti sustava, pružajući uvid u strukturu i međusobne veze poslužitelja, baze podataka i klijentskih računala. Ova organizacija ključna je za održavanje performansi i dostupnosti sustava, pridonoseći time ukupnoj stabilnosti i funkcionalnosti.

Sustav temeljen na ovakvoj arhitekturi omogućuje učinkovitu komunikaciju između svih elemenata, pridonošenje sigurnosti podataka putem HTTP veze, te osigurava optimalnu izvedbu sustava. Osim toga, organizacija elemenata na razmještajnom dijagramu omogućuje lako praćenje i upravljanje svim dijelovima sustava, čime se olakšava održavanje i daljnji razvoj.

5.4 Upute za puštanje u pogon

U nastavku su navedeni koraci za puštanje u pogon web aplikacije koja uključuje : PostgreSQL bazu podataka, Java Spring backend, i React.js frontend na javni poslužitelj Render. (<https://render.com>)

Render omogućuje besplatno posluživanje PostgreSQL baze podataka i web servisa koji će se koristiti u nastavku. Osim toga spajanjem na GitHub nudi mogućnost CI/CD modela posluživanja.

5.4.1 Stvaranje baze na javnom poslužitelju

Potrebno je otici na dashboard.render.com/new/database te popuniti formu. Potrebno je navesti jedinstveno ime instance PosrgreSQL-a, ime baze te ime korisnika kao npr:

The screenshot shows the 'New PostgreSQL' creation form on the Render dashboard. The 'Name' field is set to 'interfer'. The 'Database' field is set to 'springdb'. The 'User' field is set to 'root'. The 'Region' dropdown is set to 'Frankfurt (EU Central)'. The 'PostgreSQL Version' dropdown is set to '15'.

Slika 5.9: Render - Nova PostgreSQL baza podataka

Render će stvoriti bazu podataka a podacima ju popunjava sama aplikacija. Podatke za spajanje s bazom moguće je pronaći na dashboard-u.

The screenshot shows the 'Connection Details' page for the PostgreSQL instance. It displays the following information:
Hostname: dpg-crmluglvqz9ns73dr820-a
Port: 5432
Database: springdb_yrwe
Username: root
Password: [REDACTED]
Internal Database URL: [REDACTED]
External Database URL: [REDACTED]
PSQL Command: [REDACTED]

Slika 5.10: Render - Baza podataka

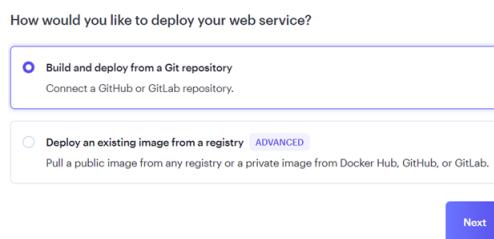
5.4.2 Konfiguriranje backenda

Kako bi se spring aplikacija mogla spojiti na bazu potrebno je dodati dependency org.postgresql.postgresql u pom.xml, te zadati podatke za spajanje na bazu u IzvorniKod/backend/src/main/resources/application.properties. Za navedeni primjer potrebno je zadati:

```
spring.datasource.url=jdbc:postgresql://dpg-cm9ug1vqd2ns73drl820-a.frankfurt-
postgres.render.com/springdb_yrwe
spring.datasource.username=root
spring.datasource.password=;zaporka;
spring.jpa.properties.hibernate.jdbc.lob.non_contextual_creation = true
spring.jpa.hibernate.ddl-auto=update
```

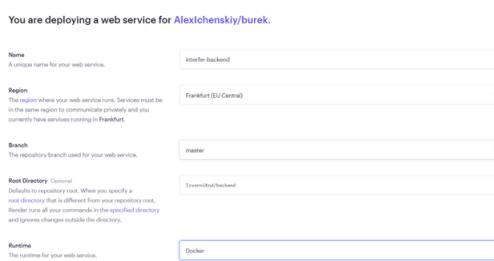
5.4.3 Posluživanje backenda

Potrebno je na Renderu napraviti instancu web servisa. Render za to nudi dvije mogućnosti:



Slika 5.11: Render - Stvaranje web servisa

U ovim uputama opisat ćemo kako stvoriti servis iz git repozitorija (za to je potrebna prijava putem GitHub računa). U sljedećem koraku potrebno je odabrati repozitorij te postaviti konfiguracije:



Slika 5.12: Render - konfiguriranje web servisa

Poslužitelj gradi Docker sliku iz Dockerfile-a u navedenom direktoriju te ju poslužuje.

5.4.4 Konfiguriranje frontenda

Potrebno je u IzvorniKod/frontend/src/assets/constants.js zadati APIURL nakojem je poslužen backend.

5.4.5 Posluživanje frontenda

Frontend se poslužuje na isti način kao backend, jedino je u postavkama umjesto IzvorniKod/backend potrebno za root directory zadati IzvorniKod/ frontend.

6. Zaključak i budući rad

Cilj našeg tima bio je izrada web platforme za online razmjenu članaka između studenata uz odgovarajuću moderaciju i druge funkcionalnosti. Iako je taj proces bio pun prepreka i poteškoća, zajedničkim radom kroz dvije faze vremenski podijeljene na dva ciklusa smo ostvarili inicijalnu ideju.

Prva faza u trajanju od sedam tjedana je bila obilježena formiranjem tima, jasnim dokumentiranjem (i osmišljavanjem) zahtjeva te implementacijom osnovnih funkcionalnosti aplikacije. Kako je studentima inače lakše izrađivati projekte ad-hoc, važnost kvalitetne dokumentacije je na početku bila podcijenjena što je izazivalo probleme na kraju prve faze. U tom trenutku je tim krenuo u izradu detaljne i čitljive dokumentacije sa jasnim UML dijagramima, što je bilo intenzivno korišteno i u većoj je mjeri olakšalo izradu dalnjih funkcionalnosti te podjelu posla između članova tima i podtimova, zaduženih za izradu kako frontend, tako i backend dijelova aplikacije.

Druga faza projekta u trajanju od šest tjedana bila je nešto jednostavnija od prve zbog stečenog iskustva članova i bolje prirodne organizacije i podjele tima. Iako su se skoro svi članovi trebali suočiti sa novim tehnologijama i alatima, uz podršku i mentoriranje iskusnijih članova tima su se brzo uklopili u razvojni proces. Privremeno dodavanje jednostavnog CI/CD procesa je uvelike ubrzalo cijeli razvojni proces te ubrzalo testiranje uz smanjenu međuvisnost backend i frontend podtimova. Prikladna definicija svih zahtjeva aplikacije te stalna komunikacija između članova tima uštedila je mnogo truda i vremena koje bi inače trebalo utrošiti na organizaciju.

Iako je platforma u svojoj trenutnoj verziji već spremna za puštanje u pogon i korištenje od strane drugih ljudi, postoji jako velik prostor za poboljšanje uz dodavanje internacionalizacije, izradu mobilne i desktop aplikacije za jednostavnije korištenje od strane korisnika na bilo kojoj platformi te dodavanje brojnih novih funkcionalnosti poput osobnih poruka i integracija sa informacijskim sustavima sveučilišta.

Iako je ovaj projekt bio veliki izazov, svim članovima projekta će poslužiti kao neprocjenjivo iskustvo rada u timu na srednje složenoj aplikaciji za veći broj ljudi.

Bez obzira na sve poteškoće kako u implementaciji tako i u komunikaciji, ostali smo zadovoljni postignutim rezultatima i implementiranim funkcionalnostima koji će služiti motivacijom za daljnji razvoj i rad na drugim, ne manje ambicioznim, projektima.

Popis literature

1. Programsko inženjerstvo, FER ZEMRIS, <http://www.fer.hr/predmet/proinzh>
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>
7. React dokumentacija, <https://react.dev/learn>
8. Spring Boot dokumentacija, <https://docs.spring.io/>
9. Docker dokumentacija, <https://docs.docker.com/>
10. MDN web docs, <https://developer.mozilla.org/>

Indeks slika i dijagrama

3.1 Dijagram obrasca uporabe, funkcionalnost anonimnog i prijavljenog korisnika	23
3.2 Dijagram obrasca uporabe, funkcionalnost moderatora sustava	24
3.3 Dijagram obrasca uporabe, funkcionalnost administratora sustava	24
3.4 Sekvencijski dijagram za UC3	25
3.5 Sekvencijski dijagram za UC9	26
3.6 Sekvencijski dijagram za UC14	28
3.7 Sekvencijski dijagram za UC26	30
4.1 Arhitektura sustava	32
4.2 MVC Obrazac	34
4.3 E-R dijagram baze podataka	38
4.4 Dijagram razreda - dio Controllers	40
4.5 Dijagram razreda - dio DTO	41
4.6 Dijagram razreda - Service dio 1	42
4.7 Dijagram razreda - Service dio 2	43
4.8 Dijagram razreda - dio Repository	44
4.9 Dijagram razreda - dio Entity	45
4.10 Dijagram razreda - dio JWT	46
4.11 Dijagram stanja - Početna stranica	47
4.12 Dijagram stanja - Uređivač članka	47
4.13 Dijagram stanja - Korisnički profil	48
4.14 Dijagram stanja - Admin panel	48
4.15 Dijagram stanja - Admin panel	50
4.16 Dijagram komponenti	52
5.1 Ispitivanje spremanja korisnika u bazu podataka	55
5.2 Ispitivanje spremanja dva korisnika u bazu podataka	56
5.3 Ispitivanje jedinstvenosti email adrese	57
5.4 Ispitivanje dohvaćanja korisnika po id-u	58
5.5 Ispitivanje postojanja korisnika po email adresi	58

5.6 Ispitivanje dohvaćanja korisnika po ulozi	59
5.7 Prikaz uspješnog izvršenja testova	63
5.8 Dijagram Razmještaja	63
5.9 Render - Nova PostgreSQL baza podataka	65
5.10 Render - Baza podataka	65
5.11 Render - Stvaranje web servisa	66
5.12 Render - konfiguriranje web servisa	66
6.1 Prikaz aktivnosti na repozitoriju - 1. dio	76
6.2 Prikaz aktivnosti na repozitoriju - 2. dio	77

Dodatak: Prikaz aktivnosti grupe

Dnevnik sastajanja

1. sastanak

- Datum: 10. listopada 2023.
- Prisustvovali: Oleksandr Ichenskyi, Anton Ladan, Antonia Šarčević, Karlo Španović
- Teme sastanka:
 - Pisanje prijedloga zadatka

2. sastanak

- Datum: 17. listopada 2023.
- Prisustvovali: Svi
- Teme sastanka:
 - Proučavanje uputa za rad na projektu
 - Stvaranje GitHub repozitorija

3. sastanak

- Datum: 31. listopada 2023.
- Prisustvovali: Svi
- Teme sastanka:
 - Contribution guidelines
 - Kostur projekta

4. sastanak

- Datum: 7. studenog 2023.
- Prisustvovali: Svi
- Teme sastanka:
 - Funkcionalni zahtjevi

5. sastanak

- Datum: 14. studenog 2023.
- Prisustvovali: Anton Ladan, Antonia Šarčević, Karlo Španović
- Teme sastanka:

- Pisanje i testiranje backend-a

Tablica aktivnosti

	Oleksandr Ichenskyi	Nikola Antolović	Anton Ladan	Antonia Šarčević	Karlo Španović
Upravljanje projektom	20		4		
Opis projektnog zadatka	4	7	3	3	6
Funkcionalni zahtjevi			18		
Opis pojedinih obrazaca	4				
Dijagram obrazaca					
Sekvencijski dijagrami				2	
Opis ostalih zahtjeva		1			
Arhitektura i dizajn sustava	4				
Baza podataka			2	3	5
Dijagram razreda				7	
Dijagram stanja			3		
Dijagram aktivnosti	2				
Dijagram komponenti					1
Korištene tehnologije i alati	1				
Ispitivanje programskog rješenja			3		6
Dijagram razmještaja		4			
Upute za puštanje u pogon					
Dnevnik sastajanja			1		
Zaključak i budući rad	1				
Popis literature	1				

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

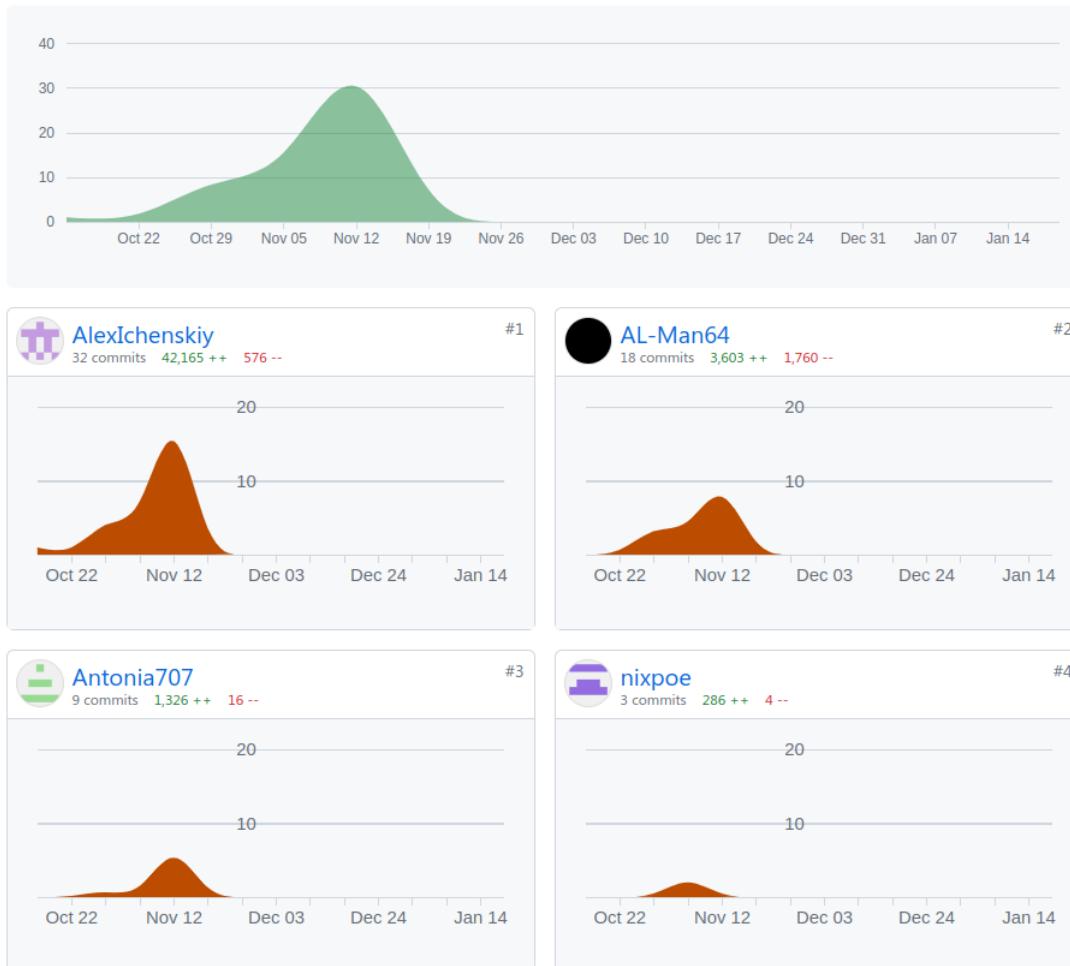
	Oleksandr Ichenskyi	Nikola Antolović	Anton Ladan	Antonia Šarčević	Karlo Španović
Izrada backenda	2		35	35	45
Izrada frontenda	32	29			
Puštanje u pogon			4	6	
Testiranje funkcionalnosti	6	5	6	4	19

Dijagrami pregleda promjena

Oct 15, 2023 – Jan 19, 2024

Contributions: Commits ▾

Contributions to master, excluding merge commits



Slika 6.1: Prikaz aktivnosti na repozitoriju - 1. dio



Slika 6.2: Prikaz aktivnosti na repozitoriju - 2. dio