

Programsko inženjerstvo

Ak. god. 2022./2023.

Company Database

Dokumentacija, Rev. 2

Grupa: *EkipaZaOcevid*

Voditelj: *Petar Hajduk*

Datum predaje: *13. 01. 2022.*

Nastavnik: *Manuela Lukić*

Sadržaj

1	Dnevnik promjena dokumentacije	3
2	Opis projektnog zadatka	5
2.1	Opis problematike	5
2.2	Postojeća rješenja	5
2.3	Opseg projektnog zadatka	6
2.4	Mogućnost prilagodbe rješenja	7
2.5	Mogućnost nadogradnje rješenja	8
3	Specifikacija programske potpore	9
3.1	Funkcionalni zahtjevi	9
3.1.1	Obrasci uporabe	12
3.1.2	Sekvencijski dijagrami	26
3.2	Ostali zahtjevi	30
4	Arhitektura i dizajn sustava	31
4.1	Baza podataka	33
4.1.1	Opis tablica	33
4.1.2	Dijagram baze podataka	37
4.2	Dijagram razreda	39
5	Implementacija i korisničko sučelje	42
5.1	Korištene tehnologije i alati	42
5.2	Ispitivanje programskog rješenja	43
5.2.1	Ispitivanje komponenti	43
5.3	Dijagram razmještaja	51
5.4	Upute za puštanje u pogon	52
5.4.1	Backend	52
5.4.2	Frontend	53
6	Zaključak i budući rad	55

Popis literature	56
Indeks slika i dijagrama	58
Dodatak: Prikaz aktivnosti grupe	59

1. Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Napravljen predložak.	Jakov Jakovac	22.10.2022.
0.2	Napisani funkcionalni zahtjevi.	Marko Čengiđ	09.11.2022.
0.3	Napisan opis projektnog zadatka.	Lovro Čunović	11.11.2022.
0.4	Napravljen opis baze podataka.	Nikola Capan	12.11.2022.
0.5.1	Napisani obrasci uporabe.	Marko Čengiđ	12.11.2022.
0.5.2	Dorada obrazaca uporabe.	Jakov Jakovac	14.11.2022.
0.6	Napisan opis arhitekture sustava.	Lovro Čunović	16.11.2022.
0.7.1	Napravljeni dijagrami obrazaca uporabe.	Marko Čengiđ	17.11.2022.
0.7.2	Napravljeni sekvencijski dijagrami.	Marko Čengiđ	17.11.2022.
0.8	Napravljeni dijagrami razreda.	Nikola Capan	17.11.2022
0.9	Priprema dokumenta za prvu potpunu verziju.	Jakov Jakovac	17.11.2022

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodatka	Autori	Datum
0.10.1	Dorada arhitekture sustava temeljem povratnih informacija.	Lovro Čunović	18.11.2022
0.10.2	Promjena aktora funkcionalnih zahtjeva temeljem povratnih informacija.	Marko Čengić	18.11.2022
0.10.3	Dorada tablica baze podataka temeljem povratnih informacija.	Jakov Jakovac	18.11.2022
0.10.4	Dorada dijagrama baze podataka temeljem povratnih informacija.	Nikola Capan	18.11.2022
0.11	Dorada opisa projektnog zadatka.	Jakov Jakovac	18.11.2022
0.12	Ispravci grešaka i posljednje pripreme dokumenta za prvu potpunu verziju.	Jakov Jakovac	18.11.2022
1.0	Verzija samo s bitnim dijelovima za 1. ciklus.	Jakov Jakovac	18.11.2022.

2. Opis projektnog zadatka

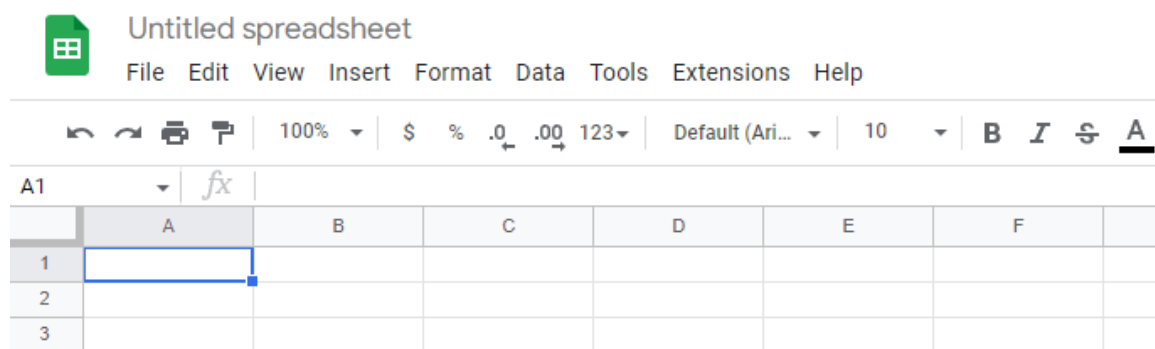
2.1 Opis problematike

Porastom broja neprofitnih organizacija iz godine u godinu, raste i potreba za kvalitetnim aplikacijama koje bi takvim organizacijama omogućile nesmetan rad i funkcioniranje. Kako takve organizacije nisu u mogućnosti samostalno postići zadovoljavajuću razinu prihoda, to im postaje najveći problem te primorane su na suradnju s kompanijama sponzorima.

Odaziv sponzora je često vrlo malen te je stoga potrebno kontaktirati velik broj kompanija. Praćenje statusa suradnji (uspješne / neuspješne) i osoba zaduženih za kontaktiranje sponzora vrlo brzo postaje kompleksno i neefikasno, a također dolazi i do mogućnosti curenja informacija. Raste potreba za aplikacijom koja će omogućiti evidenciju navedenih stavki te korisnicima dodijeliti razinu ovlasti kako bi umanjili mogućnost curenja informacija.

2.2 Postojeća rješenja

Najraširenije rješenje koje danas koriste gotovi svi jesu tablice. Bilo to u MS Excelu za jednokorisnički ili Google sheets 2.1 za višekorisnički rad, tablice su jedan od najboljih načina prikaza velikog broja podataka. Nažalost, ova rješenja imaju gore navedene probleme te stoga nisu pogodna za upotrebu na razini veće organizacije.



Slika 2.1: Prikaz alata Google sheets

2.3 Opseg projektnog zadatka

Cilj ovog projekta je razviti programsku podršku za stvaranje web aplikacije "Company Database" koja će služiti za evidenciju statusa suradnje kompanija na projektima i jednostavan uvid od strane ovlaštenih korisnika.

Korisnici će se u web aplikaciju prijavljivati koristeći Google račun koji mora biti registriran od strane drugih korisnika putem web aplikacije. Svaki je korisnik definiran sljedećim parametrima:

- ime
- prezime
- korisničko ime (nadimak)
- opis
- email za login u aplikaciju
- email za obavijesti
- najviša razina ovlasti koju posjeduje (ako ima više razina ovlasti)

Razlikovat ćemo 5 razina ovlasti, a to su redom od najviše prema najnižoj:

- Administrator
- Moderator
- Fundraising (FR) responsible
- Fundraising (FR) team member
- Observer

Svaki korisnik određene razine ovlasti ima sve mogućnosti svoje i svih nižih razina ovlasti sustava. Također ima mogućnost (kroz određene akcije) dodjeljivanja

i oduzimanja razina ovlasti korisnicima nižih razina od svoje.

Iznimke su administrator koji može dodijeliti i oduzeti i ostalim administratorima sustava te FR team member koji ne može dodijeliti niti oduzeti ovlasti jer se observer automatski dodjeljuje svim korisnicima koja nemaju višu razinu ovlasti.

Projekt je obilježen nazivom, kategorijom, tipom (interni ili eksterni), datumom početka i kraja, FR responsible-om zaduženim za navedeni projekt, popisom FR team member-a, FR ciljem (željeni prihod od projekta), prvim i drugim „pingom“ (timestamp) te popisom kompanija s kojima se surađuje na tom projektu.

Kompanija je obilježena nazivom, područjem kojim se bavi (npr. IT), ABC kategorizacijom, mjesecom planiranja budžeta, državom, poštanskim brojem, gradom, adresom, linkom na web stranicu, kratkim opisom, boolean varijablom „*kontaktirati*“ (koja označava treba li tu tvrtku u budućnosti kontaktirati) te popisom zaposlenika.

Popis kompanija moguće je uzlazno i silazno sortirati prema nazivu, području, mjesecu planiranja budžeta i linku na web stranicu te pretraživati po nazivu.

Zaposlenik kao entitet pripada kompaniji, a definiran je imenom, prezimenom, email adresom, brojem mobitela, ulogom u kompaniji (npr. CEO) te kratkim opisom.

Suradnja predstavlja poveznicu između projekta i kompanije, a obilježena je odgovornom osobom (koja je za nju zadužena), kontaktiranom osobom u kompaniji, kategorijom (financijska, materijalna ili akademska suradnja), vlastitim statusom (kontaktirano, ping, dopis, sastanak, uspješno ili neuspješno), komentarom (tj. sažetkom suradnje) te vrijednošću suradnje (koje se sumiraju i popunjavaju FR cilj projekta).

Aplikacija, sa svim navedenim funkcionalnostima i specifičnostima, bit će izvedena kao web aplikacija kojoj korisnici pristupaju pomoću Google autentifikatora. Bit će jednostavna za korištenje zahvaljujući preglednom i intuitivnom sučelju. Sustav će podržavati rad više korisnika u stvarnom vremenu, frontend će biti ostvaren u React-u, a backend će koristiti relacijsku bazu podataka i Spring boot.

2.4 Mogućnost prilagodbe rješenja

Kako bi što više ljudi moglo znati za projekt, organizatori nerijetko kontaktiraju razne medije kako zainteresirani ne bi propustili priliku. Slična aplikacija bi se

također mogla koristiti za komunikaciju s medijima. Entitet Tvrtka bi bilo bi potrebno izmijeniti u Medij zajedno s odgovarajućim atributima, dok bi u entitetu Suradnja bilo potrebno izmijeniti samo attribute.

Ovakva bi aplikacija bila od koristi ne samo neprofitnim organizacijama, već i svima koji organiziraju nešto što treba publicitet.

2.5 Mogućnost nadogradnje rješenja

Aplikacija bi se dodatno mogla nadograditi proširenjem komentara suradnje, primarno u svrhu prenošenja znanja od iskusniji osoba. Te bi osobe kroz komentar na suradnju mogli dati svoj input vezano uz neku temu. Tada bi i korisnik koji je odgovoran za suradnju dobio te obavijesti na naslovnicu.

Nadalje, aplikacija bi mogla, osim tabličnog, podržavati više načina prikaza podataka o korisnicima, projektima, tvrtkama i suradnjama. Možda bi ih se moglo grupirati temeljem nekih postojećih ili novih atributa te onda to iskoristiti.

Također bi bilo korisno uvesti dodatne mogućnosti filtriranja svih tablica i drugih načina prikaza podataka.

Ako se projekti neke organizacije ponavljaju iz godine u godinu, bilo bi korisno uvesti neki način kopiranja podataka iz tablice prošlogodišnjeg projekta. Korisno bi bilo dodati i mogućnost izvoza i uvoz podataka iz i u tablice u JSON i CSV formatima.

Kako bi se još više smanjila mogućnost curenja informacija, mogao bi se uvesti tzv. soft lock korisnika koji bi očuvao podatke o korisniku, ali mu onemogućio da se ulogira (npr. ako korisnik više nije dio organizacije).

Aplikaciju bi također bilo moguće spojiti s nekakvim internim sustavom za upravljanjem članovima organizacije.

Što se tiče UX-a, tu uvijek ima mjesta za napredak. Dodavanje animacija tokom prijelaza od jedne stranice na drugu ili važnije tokom dohvata podataka iz baze kako bi korisnik znao da se njegov zahtjev procesuirao. Moglo bi se uvesti neki način prikaza uspješnosti odrade nekog zadatka (npr. dodavanje podatka u bazu).

Mogao poboljšati i sam UI da aplikacija izgleda profesionalnije i bude ugodnija za korištenje, ali i kako bi aplikaciju mogle koristiti osobe s posebnim potrebama.

3. Specifikacija programske potpore

3.1 Funkcionalni zahtjevi

Dionici:

1. Predsjednik neprofitabilne organizacije
2. Vlasnik kompanije
3. Zaposlenik kompanije
4. Kontakt
5. Osoba zadužena za prodaju
6. Član neprofitabilne organizacije
7. Osoba odgovorna za odnose s tvrtkama
8. Osoba odgovorna za odnose s tvrtkama na projektu
9. Član tima za odnose s tvrtkama
10. Intern
11. Administrator
12. Razvojni tim

Aktori i njihovi funkcionalni zahtjevi:

1. Neprijavljeni korisnik (inicijator) može:
 - (a) Ako je dodan od strane administratora može se prijaviti svojim emailom
 - (b) U slučaju da je prvi neprijavljeni korisnik postaje administrator
2. Intern - observer (inicijator) može:
 - (a) Pregledavati projekte
 - (b) Pregledavati kompanije (naziv i područje)
 - (c) Pregledavati informacija o projektima (datum početka, datum završetka, organizator)
 - (d) Pregledavati suradnje
 - (e) Pregledati korisnike
 - (f) Pregledati podatke o korisniku

3. Član tima za odnose s tvrtkama - FR team member (inicijator) može:
- (a) Pregledavati projekte
 - (b) Pregledavati kompanije (naziv i područje)
 - (c) Pregledavati informacija o projektima (datum početka, datum završetka, FR responsible)
 - (d) Pregledavati suradnje
 - (e) Promijeniti podatke o suradnji (kategorija, status, sažetak, vrijednost)
 - (f) Pregledavati i promijeniti sve podatke za dodijeljenu kompaniju
 - (g) Pregledati korisnike
 - (h) Pregledati podatke o korisniku
4. Osoba odgovorna za odnose s tvrtkama na projektu - FR responsible (inicijator) može:
- (a) Pregledavati projekte
 - (b) Pregledavati kompanije (naziv i područje)
 - (c) Pregledavati informacija o projektima (datum početka, datum završetka, FR responsible)
 - (d) Pregledavati suradnje
 - (e) Promijeniti podatke o suradnji (kategorija, status, sažetak, vrijednost)
 - (f) Pregledavati i promijeniti sve podatke za dodijeljenu kompaniju
 - (g) Stvoriti i obrisati suradnju
 - (h) Stvoriti i obrisati vlastito stvorene kompanije
 - (i) Pregledati korisnike
 - (j) Pregledati podatke o korisniku
5. Osoba odgovorna za odnose s tvrtkama - moderator (inicijator) može:
- (a) Pregledavati projekte
 - (b) Pregledavati kompanije (naziv i područje)
 - (c) Pregledavati informacija o projektima (datum početka, datum završetka, FR responsible)
 - (d) Pregledavati suradnje
 - (e) Promijeniti podatke o suradnji (kategorija, status, sažetak, vrijednost)
 - (f) Pregledavati i promijeniti sve podatke za dodijeljenu kompaniju
 - (g) Stvoriti i obrisati suradnju
 - (h) Stvoriti i obrisati kompanije
 - (i) Stvoriti i obrisati vlastito stvorene projekte

- (j) Dodijeliti i maknuti nekom korisniku mogućnosti organizatora projekta na projektu koji je stvorio
- (k) Staviti i maknuti nekog korisnika s rolom člana neprofitabilne kompanije na projekt koji je stvorio
- (l) Pregledati korisnike
- (m) Pregledati podatke o korisniku

6. Administrator (inicijator) može:

- (a) Pregledavati projekte
- (b) Pregledavati kompanije (naziv i područje)
- (c) Pregledavati informacija o projektima (datum početka, datum završetka, FR responsible)
- (d) Pregledavati suradnje
- (e) Promijeniti podatke o suradnji (kategorija, status, sažetak, vrijednost)
- (f) Pregledavati i promijeniti sve podatke za dodijeljenu kompaniju
- (g) Stvoriti i obrisati suradnju
- (h) Stvoriti i obrisati projekte
- (i) Stvoriti i obrisati kompanije
- (j) Dodijeliti nekom korisniku mogućnosti organizatora projekta na projektu koji je stvorio
- (k) Staviti nekog korisnika s rolom člana neprofitabilne kompanije na projekt koji je stvorio
- (l) Pregledati korisnike
- (m) Pregledati podatke o korisniku
- (n) Dodijeliti nekom korisniku bilo koju od prije navedenih pozicija
- (o) Registrirati i maknuti korisnika iz sustava
- (p) Registrirati sebe u sustav pri prvom pokretanju
- (q) Promijeniti osobne podatke

7. Baza podataka (sudionik) može:

- (a) Pohranjuje sve podatke o projektima
- (b) Pohranjuje sve podatke o kompanijama i njihovim kontaktima

3.1.1 Obrasci uporabe

Opis obrazaca uporabe

UC1 - Registracija prvog administratora

- **Glavni sudionik:** Neprijavljeni korisnik
- **Cilj:** Stvoriti administracijski korisnički račun za pristup aplikaciji
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Korisnik dolazi na stranicu koja ga preusmjerava na formu za registraciju prvog korisnika
 2. Korisnik ispunjava formu emailom prvog administratora
 3. Stranica preusmjerava korisnika na početnu stranicu
- **Opis mogućih odstupanja:**
 - 2.a Upis već postojećeg emaila ili upis korisničkih podatka u nedozvoljenom formatu
 1. Aplikacija obavještava korisnika o neuspjelom upisu
 2. Administrator mijenja potrebne podatke te uspješno završava unos ili odustaje od dodavanja novog korisnika

UC2 - Prijava u aplikaciju

- **Glavni sudionik:** Neprijavljeni korisnik
- **Cilj:** Dati korisniku pristup aplikaciji
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je registriran
- **Opis osnovnog tijeka:**
 1. Korisnik dolazi na stranicu za prijavu
 2. Otvara mu se prozor za prijavu s google računom
 3. Korisnik se prijavljuje google računom
 4. Korisnik dobiva pristup aplikaciji koja ga preusmjerava na naslovnu stranicu
- **Opis mogućih odstupanja:**
 - 2.a Prijavom s računom koji nije u bazi
 1. Aplikacija obavještava korisnika o neuspjeloj prijavi
 2. Aplikacija obavještava korisnika na koju email adresu se treba javiti ako korisnik smatra da je to greška

3. Korisnik se ponovo prijavljuje s drugim računom ili odustaje od prijave

UC3 - Registracija korisnika

- **Glavni sudionik:** Administrator
- **Cilj:** Stvoriti korisnički račun za pristup aplikaciji
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik (administrator) odabire opciju "Users"
 2. Aplikacija prikazuje popis korisnika
 3. Korisnik odabire opciju za dodavanje korisnika
 4. Aplikacija prikaže formu za dodavanje korisnika
 5. Korisnik popuni potrebne podatke (ime, prezime, email i razinu ovlasti)
 6. Korisnik odabere opciju "Add"
 7. Baza podataka se ažurira
 8. Aplikacija zatvori formu za dodavanje korisnika
 9. Aplikacija ažurira popis korisnika
- **Opis mogućih odstupanja:**
 - 3.a Upis postojećeg emaila ili upis korisničkih podatka u nedozvoljenom formatu
 1. Aplikacija obavještava korisnika o neuspjelom upisu
 2. Administrator mijenja potrebne podatke te uspješno završava unos ili odustaje od dodavanja novog korisnika

UC4 - Pregled projekata

- **Glavni sudionik:** Intern
- **Cilj:** Pregledati projekte
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju "Projects"
 2. Aplikacija prikazuje popis naziva projekata

UC5 - Pregled kompanija

- **Glavni sudionik:** Intern

- **Cilj:** Pregledati kompanije
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju "Companies"
 2. Aplikacija prikazuje popis kompanijama s osnovnim podacima (naziv, područje, ABC kategorizacija i link na web)

UC6 - Pregled podataka o projektu

- **Glavni sudionik:** Intern
- **Cilj:** Pregledati podatke o projektu
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju "Projects"
 2. Aplikacija prikazuje popis projekata
 3. Korisnik klikne na naziv projekta o kojem želi informacije
 4. Prikažu se detalji projekta

UC7 - Pregled podataka o korisniku

- **Glavni sudionik:** Intern
- **Cilj:** Pregledati podatke o željenom korisniku
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju "Users"
 2. Aplikacija prikazuje popis korisnika
 3. Korisnik odabire željenog korisnika iz popisa
 4. Aplikacija prikazuje podatke o korisniku

UC8 - Promjena podataka o korisniku

- **Glavni sudionik:** Administrator
- **Cilj:** Promijeniti podatke o korisniku
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen, Korisnik ima razinu ovlasti administrator
- **Opis osnovnog tijeka:**

1. Korisnik odabire opciju "Users"
 2. Aplikacija prikazuje popis korisnika
 3. Korisnik odabire opciju "Edit"
 4. Aplikacija prikaže formu za izmjenu podataka korisnika
 5. Korisnik mijenja podatke
 6. Korisnik odabire opciju spremi
 7. Baza podataka se ažurira
 8. Aplikacija zatvori formu za dodavanje korisnika
 9. Aplikacija ažurira popis korisnika
- **Opis mogućih odstupanja:**
 - 2.f Upis već postojećeg emaila ili upis korisničkih podatka u nedozvoljenom formatu
 1. Aplikacija obavještava korisnika o neuspjelom upisu
 2. Administrator mijenja potrebne podatke te uspješno završava unos ili odustaje od izmjene podataka o korisniku

UC9 - Pregled suradnji

- **Glavni sudionik:** Intern
- **Cilj:** Pregledati podatke o suradnji
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen, Korisnik je FR team member projekta
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju "Projects"
 2. Aplikacija prikazuje popis projekata
 3. Korisnik klikne na ime projekta u kojem se nalazi suradnja koja ga zanima
 4. Aplikacija prikazuje popis suradnji vezanih uz taj projekt s osnovnim podacima (naziv kompanije, odgovoru osobu, status, kategoriju, kontakt, dio komentara, akcije)

UC10 - Promjena podataka o suradnji

- **Glavni sudionik:** Član tima za odnose s tvrtkama
- **Cilj:** Promijeniti podatke o suradnji
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen, Korisnik je FR responsible projekta
- **Opis osnovnog tijeka:**

1. Korisnik odabire opciju "Projects"
 2. Aplikacija prikazuje popis projekata
 3. Korisnik klikne na ime projekta u kojem se nalazi suradnja koja ga zanima
 4. Aplikacija prikazuje popis suradnji vezanih uz taj projekt s osnovnim podacima (naziv kompanije, odgovoru osobu, status, kategoriju, kontakt, dio komentara, akcije)
 5. Korisnik odabire opciju "Edit"
 6. Aplikacija prikaže formu s podacima te suradnje
 7. Korisnik mijenja podatke o suradnji
 8. Korisnik odabire opciju "Save"
 9. Baza podataka se ažurira
 10. Aplikacija ažurira popis suradnji
- **Opis mogućih odstupanja:**
 - 7.a Korisnik promijeni podatke o suradnji na nevažeće vrijednosti i odabere opciju "Save"
 1. Aplikacija obavještava korisnika o neuspjelom upisu
 2. Administrator mijenja potrebne podatke te uspješno završava unos ili odustaje od promjene podataka

UC11 - Pregled podataka o kompaniji

- **Glavni sudionik:** Član tima za odnose s tvrtkama
- **Cilj:** Pregledati podatke o kompaniji
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen, Korisnik ima razinu ovlasti FR responsible
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju "Companies"
 2. Aplikacija prikaže popis kompanija
 3. Korisnik odabire kompaniju
 4. Aplikacija prikaže detalje o toj kompaniji

UC12 - Promjena podataka o kompaniji

- **Glavni sudionik:** Član tima za odnose s tvrtkama
- **Cilj:** Promijeniti podatke o kompaniji
- **Sudionici:** Baza podataka

- **Preduvjet:** Korisnik je prijavljen, Korisnik je odgovoran za suradnju s kompanijom
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju "Companies"
 2. Aplikacija prikaže popis kompanija
 3. Korisnik odabire kompaniju za koju je dodijeljen
 4. Aplikacija prikaže podatke o kompaniji
 5. Korisnik odabire opciju "Edit"
 6. Aplikacija prikaže formu s podacima o kompaniji
 7. Korisnik mijenja podatke o kompaniji
 8. Korisnik sprema promjene
 9. Baza podataka se ažurira
 10. Aplikacija zatvori formu s podacima o kompaniji
 11. Aplikacija ažurira podatke o kompaniji
- **Opis mogućih odstupanja:**
 - 5.b Korisnik promijeni podatke o kompaniji na nevaljane podatke i odabere opciju "Save"
 1. Aplikacija obavještava korisnika o neuspjelom upisu
 2. Korisnik mijenja potrebne podatke te uspješno završava unos ili odustaje od promjene podataka

UC13 - Stvaranje suradnje

- **Glavni sudionik:** Osoba odgovorna za odnose s tvrtkama na projektu
- **Cilj:** Stvoriti suradnju za dodijeljen projekt
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju "Projects"
 2. Aplikacija prikazuje popis projekata
 3. Korisnik klikne na ime projekta u kojem želi stvoriti suradnju
 4. Aplikacija prikazuje popis suradnji vezanih uz taj projekt s osnovnim podacima (naziv kompanije, odgovoru osobu, status, kategoriju, kontakt, komentar, akcije)
 5. Korisnik odabere opciju "New collaboration"
 6. Aplikacija prikaže formu za stvaranje nove suradnje
 7. Korisnik popunjava podatke o suradnji

8. Korisnik sprema promjene
 9. Baza podataka se ažurira
 10. Aplikacija zatvara formu za stvaranje nove suradnje
 11. Aplikacija ažurira popis suradnji
- **Opis mogućih odstupanja:**
 - 8.c Korisnik unese nevaljane podatke o suradnji i odabere opciju "Save"
 1. Aplikacija obavještava korisnika o neuspjelom upisu
 2. Korisnik mijenja potrebne podatke te uspješno završava unos ili odustaje od dodavanja suradnje

UC14 - Brisanje suradnje

- **Glavni sudionik:** Osoba odgovorna za odnose s tvrtkama na projektu
- **Cilj:** Maknuti suradnju s dodijeljenog projekta
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju "Projects"
 2. Aplikacija prikazuje popis projekata
 3. Korisnik klikne na ime projekta u kojem se nalazi suradnja koja ga zanima
 4. Aplikacija prikazuje popis suradnji vezanih uz taj projekt s osnovnim podacima (naziv kompanije, odgovoru osobu, status, kategoriju, kontakt, dio komentara, akcije)
 5. Korisnik odabere opciju za brisanje suradnje
 6. Aplikacija pita korisnika je li siguran želi li izbrisati suradnju
 7. Korisnik odabire potvrdnu opciju
 8. Baza podataka se ažurira
 9. Aplikacija ažurira popis suradnji

UC15 - Stvaranje projekta

- **Glavni sudionik:** Osoba odgovorna za odnose s tvrtkama
- **Cilj:** Stvoriti projekt
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju "Projects"

2. Aplikacija prikazuje popis projekata
 3. Korisnik odabire opciju "Create project"
 4. Aplikacija prikazuje formu za stvaranje novog projekta
 5. Korisnik popuni podatke za stvaranje novog projekta
 6. Korisnik sprema promjene
 7. Baza podataka se ažurira
 8. Aplikacija ažurira popis projekata
- **Opis mogućih odstupanja:**
 - 7.b Korisnik unese nevaljane podatke o projektu i odabere opciju "Save"
 1. Aplikacija obavještava korisnika o neuspjelom upisu
 2. Korisnik mijenja potrebne podatke te uspješno završava unos ili odustaje od dodavanja projekta

UC16 - Brisanje projekta

- **Glavni sudionik:** Osoba odgovorna za odnose s tvrtkama
- **Cilj:** Izbrisati projekt
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju "Projects"
 2. Aplikacija prikazuje popis projekata
 3. Korisnik odabire neki od projekata
 4. Aplikacija prikazuje detalje o tom projektu
 5. Korisnik odabire opciju za brisanje projekta
 6. Aplikacija prikaže modal koji pita korisnika je li siguran
 7. Korisnik odabere opciju da je siguran
 8. Baza podataka se ažurira
 9. Aplikacija ažurira popis projekata

UC17 - Dodavanje korisnika na projekt

- **Glavni sudionik:** Osoba odgovorna za odnose s tvrtkama
- **Cilj:** Dodati člana neprofitabilne organizacije na projekt
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju "Projects"

2. Aplikacija prikazuje popis projekata
 3. Korisnik odabire neki od projekata
 4. Aplikacija prikazuje detalje o tom projektu
 5. Korisnik odabire opciju za dodavanje korisnika na projekt
 6. Aplikacija prikaže formu
 7. Korisnik upiše ime, prezime, nadimak ili email člana kojeg želi dodati
 8. Korisnik odabire opciju "Add"
 9. Baza podataka se ažurira
 10. Aplikacija ažurira popis korisnika na projektu
- **Opis mogućih odstupanja:**
 - 7.b Korisnik unese nevaljane podatke o korisniku i odabere opciju "Add"
 1. Aplikacija obavještava korisnika o neuspjelom upisu
 2. Korisnik mijenja potrebne podatke te uspješno završava unos ili odustaje od dodavanja korisnika na projekt

UC18 - Micanje korisnika s projekta

- **Glavni sudionik:** Osoba odgovorna za odnose s tvrtkama
- **Cilj:** Maknuti člana neprofitabilne organizacije s projekta
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju "Projects"
 2. Aplikacija prikazuje popis projekata
 3. Korisnik odabire neki od projekata
 4. Aplikacija prikazuje detalje o tom projektu
 5. Korisnik odabire opciju za micanje korisnika s projekta
 6. Aplikacija prikaže modal koji pita korisnika je li siguran
 7. Korisnik odabere opciju da je siguran
 8. Baza podataka se ažurira
 9. Aplikacija ažurira popis korisnika na projektu

UC19 - Pregled korisnika

- **Glavni sudionik:** Intern
- **Cilj:** Pregledati korisnike
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen

- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju "Users"
 2. Aplikacija prikazuje popis korisnika

UC20 - Promjena razine ovlasti korisnika

- **Glavni sudionik:** Administrator
- **Cilj:** Promijeniti rolu korisnika
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju "Users"
 2. Aplikacija prikazuje popis korisnika
 3. Korisnik odabere opciju za izmjenu podataka o korisniku
 4. Aplikacija prikaže formu za izmjenu podataka o korisniku
 5. Korisnik odabire jednu od razina ovlasti
 6. Korisnik odabire opciju spremi
 7. Baza podataka se ažurira
 8. Aplikacija ažurira popis korisnika
- **Opis mogućih odstupanja:**
 - 3.a Korisnik kojem se mijenja rola je administrator i ne postoji nijedan drugi administrator
 1. Sustav obavještava korisnika da mora postojati barem jedan administrator, pa je ova akcija zabranjena

UC21 - Brisanje korisnika iz sustava

- **Glavni sudionik:** Administrator
- **Cilj:** Maknuti korisnika iz sustava
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju "Users"
 2. Aplikacija prikazuje popis korisnika
 3. Korisnik odabere opciju brisanja željenog korisnika
 4. Aplikacija pita korisnika je li siguran da želi provesti ovu akciju
 5. Korisnik odabire opciju da je siguran
 6. Baza podataka se ažurira

7. Aplikacija ažurira popis korisnika

- **Opis mogućih odstupanja:**

3.a Korisnik kojem se miče iz sustava je administrator i ne postoji nijedan drugi administrator

1. Sustav obavještava korisnika da mora postojati barem jedan administrator, pa je ova akcija zabranjena

UC22 - Filtriranje korisnika

- **Glavni sudionik:** Intern

- **Cilj:** Filtrirati korisnike

- **Sudionici:** Baza podataka

- **Preduvjet:** Korisnik je prijavljen

- **Opis osnovnog tijeka:**

1. Korisnik odabire opciju "Users"

2. Aplikacija prikazuje popis korisnika

3. Korisnik upiše tekst u polje za pretragu

4. Aplikacija ažurira popis korisnika tako da prikazuje samo korisnike koji imaju taj tekst u imenu, prezimenu, nadimku ili emailu

UC23 - Stvaranje kompanije

- **Glavni sudionik:** Osoba odgovorna za odnose s tvrtkama na projektu

- **Cilj:** Stvoriti kompaniju

- **Sudionici:** Baza podataka

- **Preduvjet:** Korisnik je prijavljen

- **Opis osnovnog tijeka:**

1. Korisnik odabire opciju "Companies"

2. Aplikacija prikazuje popis kompanija

3. Korisnik odabire opciju "Create company"

4. Aplikacija prikazuje formu za stvaranje nove kompanije

5. Korisnik popuni podatke za stvaranje nove kompanije

6. Korisnik sprema promjene

7. Baza podataka se ažurira

8. Aplikacija ažurira popis kompanija

- **Opis mogućih odstupanja:**

7.b Korisnik unese nevaljane podatke o kompaniji i odabere opciju "Save"

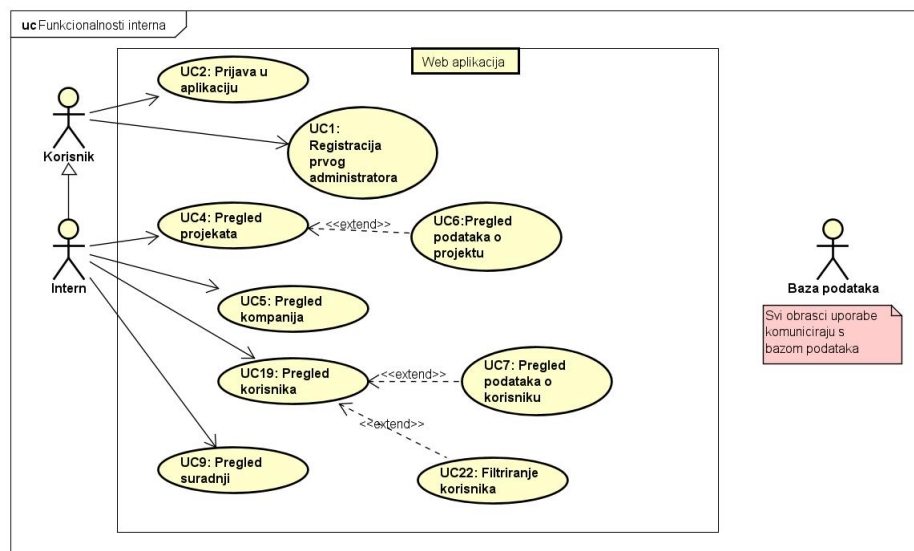
1. Aplikacija obavještava korisnika o neuspjelom upisu

2. Korisnik mijenja potrebne podatke te uspješno završava unos ili odustaje od dodavanja kompanije

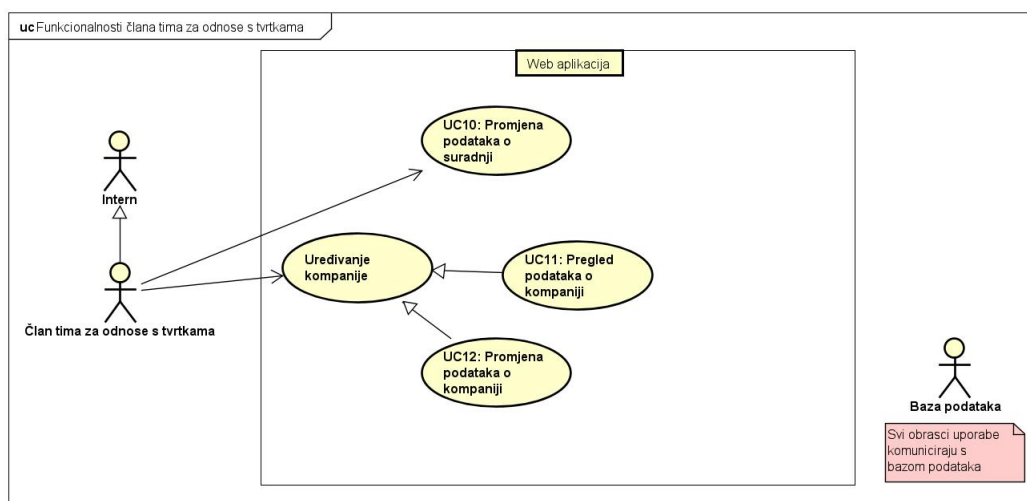
UC24 - Brisanje kompanije

- **Glavni sudionik:** Osoba odgovorna za odnose s tvrtkama na projektu
- **Cilj:** Izbrisati kompaniju
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju "Companies"
 2. Aplikacija prikazuje popis kompanija
 3. Korisnik odabire gumb za detalje pored neke od kompanija
 4. Aplikacija prikazuje detalje o toj kompaniji
 5. Korisnik odabire opciju za brisanje kompanije
 6. Aplikacija prikaže modal koji pita korisnika je li siguran
 7. Korisnik odabere opciju da je siguran
 8. Baza podataka se ažurira
 9. Aplikacija ažurira popis projekata

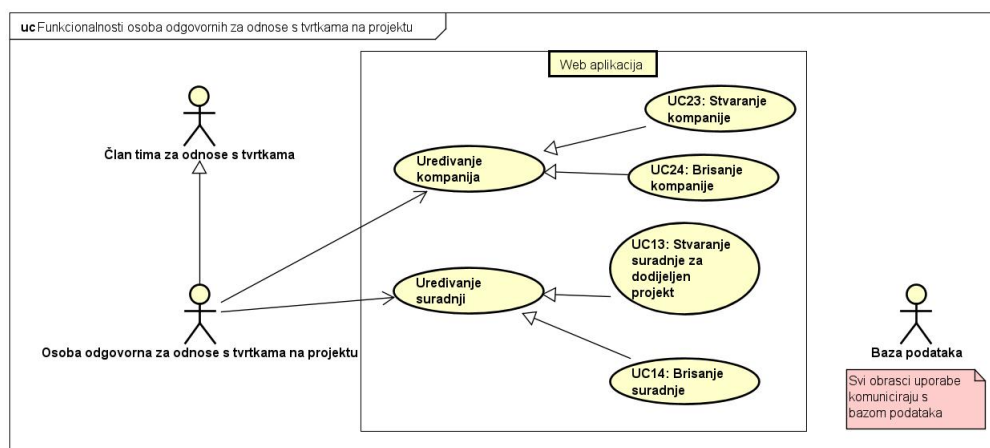
Dijagrami obrazaca uporabe



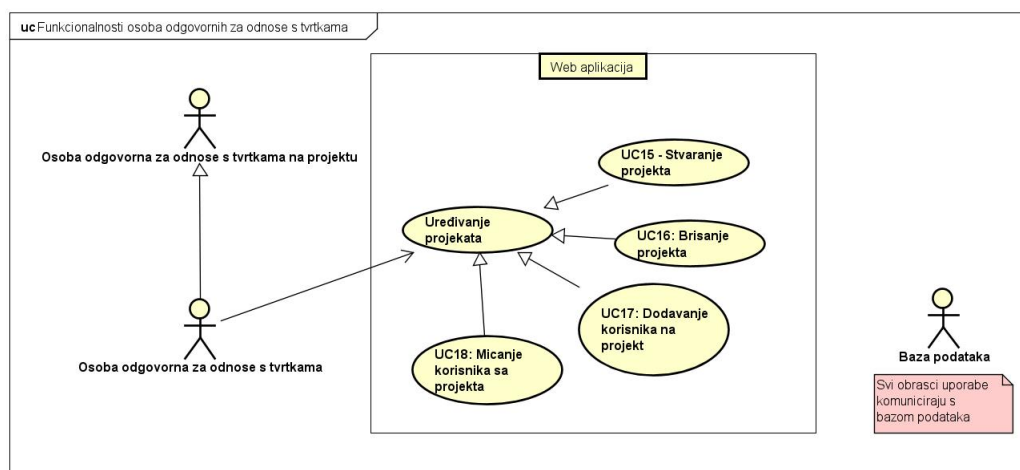
Slika 3.1: Funkcionalnosti observera



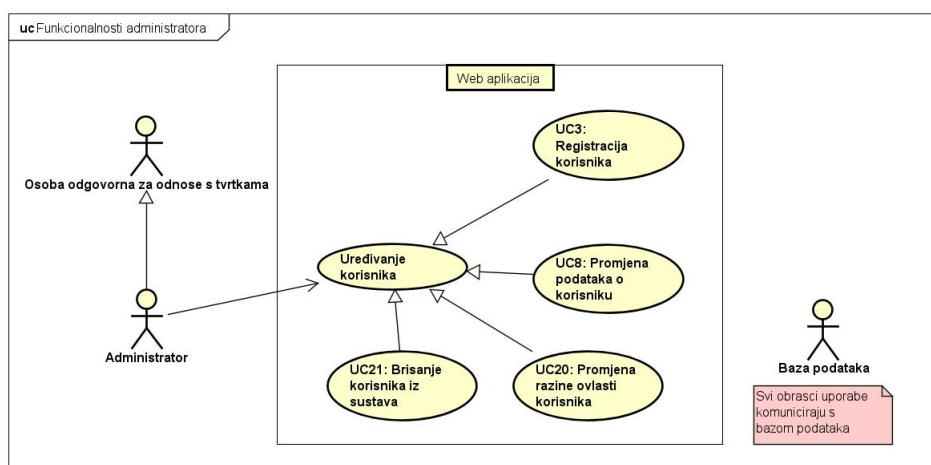
Slika 3.2: Funkcionalnosti člana tima za odnose s tvrtkama



Slika 3.3: Funkcionalnosti osobe odgovorne za odnose s tvrtkama na projektu



Slika 3.4: Funkcionalnosti osobe odgovorne za odnose s tvrtkama

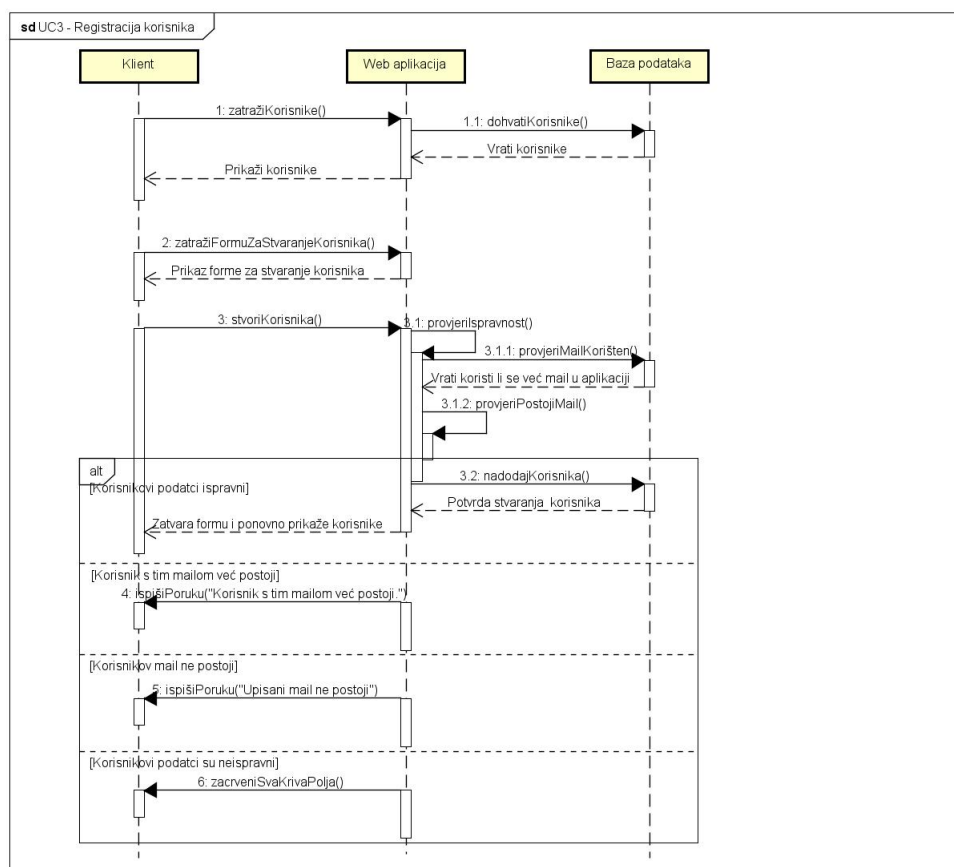


Slika 3.5: Funkcionalnosti administratora

3.1.2 Sekvencijski dijagrami

Obrazac uporabe UC3 - Registracija korisnika

Korisnik (administrator) šalje zahtjev za prikaz korisnika kako bi mogao stvoriti novog. Aplikacija dohvaća trenutne korisnike i prikazuje ih. Korisnik šalje zahtjev za formu za stvaranje novog korisnika. Aplikacija prikaže formu za stvaranje novog korisnika. Da bi uspješno stvorio novog korisnika trenutni korisnik šalje zahtjev za stvaranje novog korisnika koji sadrži ime, prezime, email i razinu ovlasti. Aplikacija provjerava je su li svi poslani podatci točni. Ako su ime ili prezime prazni aplikacija zacrveni ta polja. Ako već postoji korisnik s tim emailom, aplikacija to javlja trenutnom korisniku. U slučaju uspješne provjere baza podataka se ažurira s novim korisnikom, a korisnik se vraća na stranicu s trenutnim korisnicima.

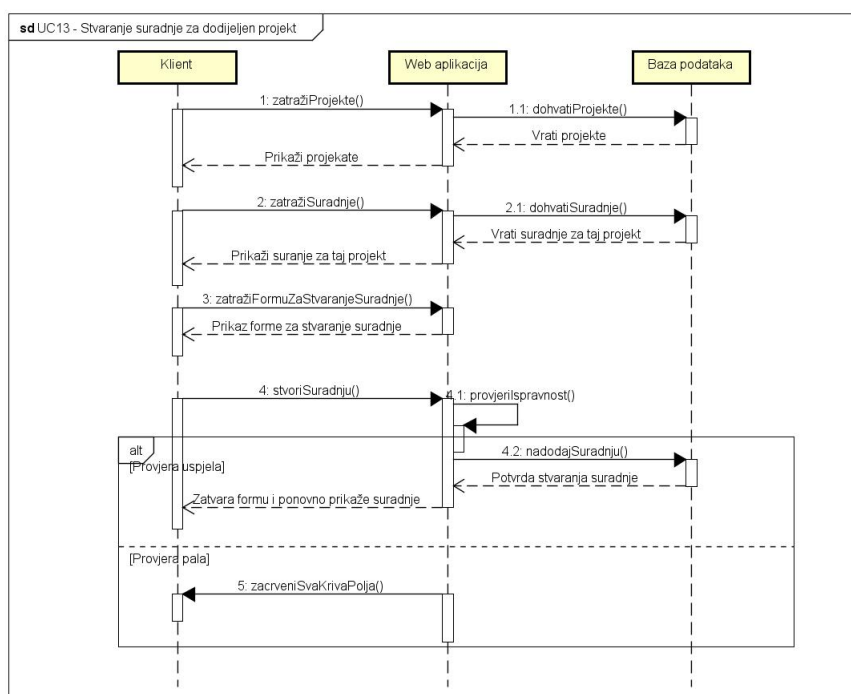


Slika 3.6: Sekvencijski dijagram za UC3 - Registracija korisnika

Obrazac uporabe UC13 - Stvaranje suradnje

Korisnik šalje zahtjev za prikaz projekata kako bi mogao stvoriti novu surad-

nju. Aplikacija dohvaća trenutne projekte i prikazuje ih. Korisnik šalje zahtjev za prikaz detalja projekta koji uključuju i suradnje na tom projektu. Aplikacija prikazuje popis suradnji vezanih uz taj projekt s osnovnim podacima (naziv kompanije, odgovoru osobu, status, kategoriju, kontakt, komentar, akcije). Korisnik šalje zahtjev za formu za novu suradnju. Aplikacija prikaže formu za stvaranje nove suradnje. Da bi uspješno stvorio novu suradnju korisnik šalje zahtjev za stvaranje nove suradnje koji sadrži naziv, odgovornu osobu, status, kategoriju i komentar. Aplikacija provjerava je su li svi poslani podatci točni. Ako su neki podatci krivi aplikacija zacrveni ta polja. U slučaju uspješne provjere baza podataka se ažurira s novom suradnjom u tom projektu, a korisnik se vraća na stranicu sa suradnjama na projektu.

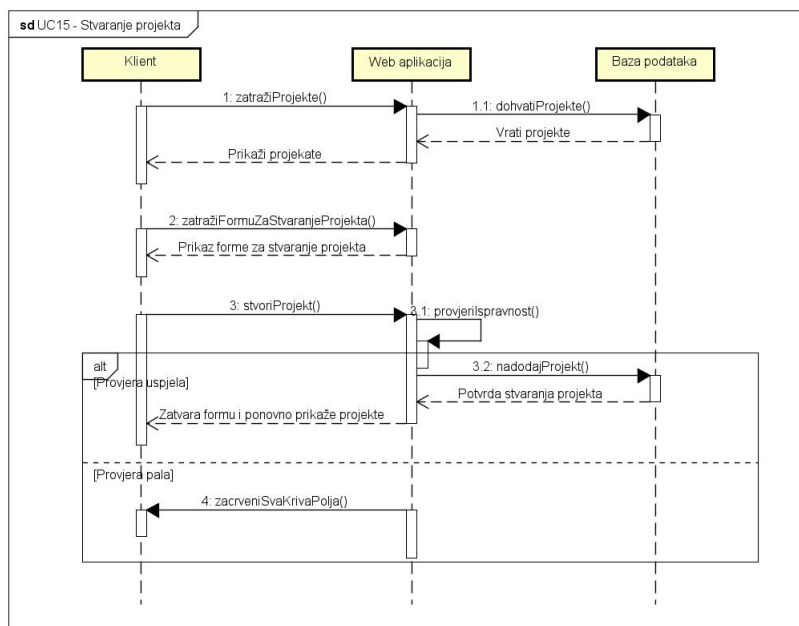


Slika 3.7: Sekvencijski dijagram za UC13 - Stvaranje suradnje

Obrazac uporabe UC15 - Stvaranje projekta

Korisnik šalje zahtjev za prikaz projekata kako bi mogao stvoriti novi. Aplikacija dohvaća trenutne projekte i prikazuje ih. Korisnik šalje zahtjev za formu za stvaranje novog projekta. Aplikacija prikaže formu za stvaranje nove suradnje. Da bi uspješno stvorio novu suradnju korisnik šalje zahtjev za stvaranje novog projekta koji sadrži naziv, kategoriju, tip, timestamp početka i novčani cilj projekta. Aplikacija provjerava je su li svi poslani podatci točni. Ako su neki podatci krivi

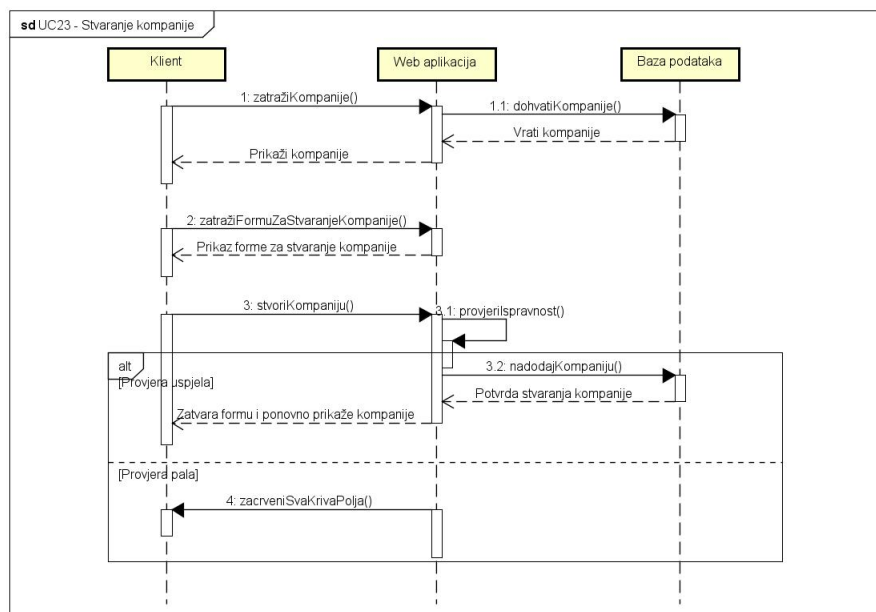
aplikacija zacrveni ta polja. U slučaju uspješne provjere baza podataka se ažurira s novim projektom, a korisnik se vraća na stranicu s projektima.



Slika 3.8: Sekvencijski dijagram za UC15 - Stvaranje projekta

Obrazac uporabe UC23 - Stvaranje kompanije

Korisnik šalje zahtjev za prikaz kompanija kako bi mogao stvoriti novu. Aplikacija dohvaća trenutne kompanije i prikazuje ih. Korisnik šalje zahtjev za formu za stvaranje nove kompanije. Aplikacija prikaže formu za stvaranje nove kompanije. Da bi uspješno stvorio novu kompaniju korisnik šalje zahtjev za stvaranje nove kompanije koji sadrži naziv, kategoriju, područje kojim se bavi, državu i poštanski broj, grad, adresu i boolean kontaktirati. Aplikacija provjerava je su li svi poslani podatci točni. Ako su neki podatci krivi aplikacija zacrveni ta polja. U slučaju uspješne provjere baza podataka se ažurira s novim projektom, a korisnik se vraća na stranicu s kompanijama.



Slika 3.9: Sekvencijski dijagram za UC23 - Stvaranje kompanije

3.2 Ostali zahtjevi

- Sustav treba omogućiti rad više korisnika u stvarnom vremenu.
- Korisničko sučelje i sustav moraju podržavati hrvatsku abecedu (dijakritičke znakove) pri unosu i prikazu tekstualnoga sadržaja.
- Izvršavanje dijela programa u kojem se pristupa bazi podataka ne smije trajati duže od nekoliko sekundi.
- Sustav treba biti implementiran kao web aplikacija koristeći objektno - orijentirane jezike.
- Neispravno korištenje korisničkog sučelja ne smije narušiti funkcionalnost i rad sustava.
- Sustav treba biti jednostavan za korištenje, korisnici se moraju znati koristiti sučeljem bez opširnih uputa.
- Veza s bazom podataka mora biti kvalitetno zaštićena, brza i otporna na vanjske greške.
- Pristup sustavu mora biti omogućen iz javne mreže pomoću HTTPS
- Prikazivanje stranice korisniku ne smije trajati dulje od desetak sekundi, bez obzira na količinu podataka koju prikazuje.

4. Arhitektura i dizajn sustava

Arhitekturu možemo podijeliti na 3 podsustava:

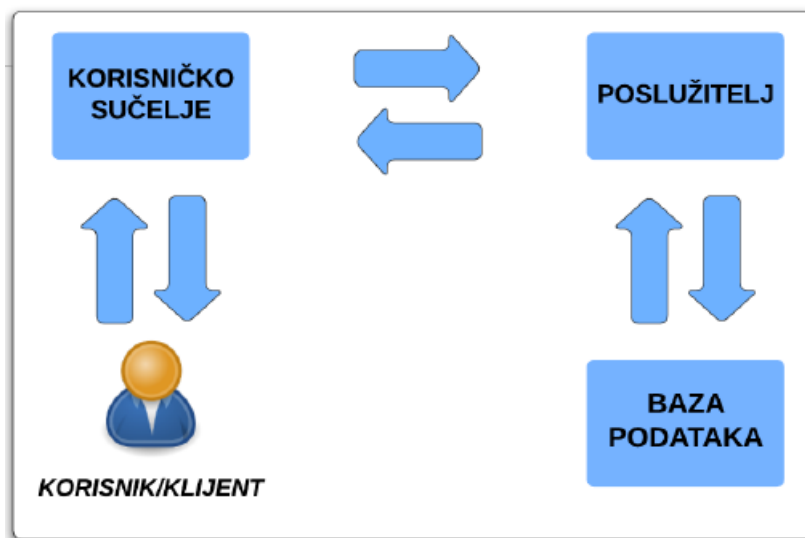
- Korisničko sučelje (frontend)
- Poslužitelj (backend)
- Baza podataka

Web preglednik (*Internetski preglednik, Web browser, Internet browser*) je program koji korisniku omogućuje pregled web-stranica i multimedijalnih sadržaja vezanih uz njih. Svaki internetski preglednik je interpreter (program koji u realnom vremenu izvršava izvorni kod napisan u nekom programskom jeziku, umjesto da ga, prije izvršavanja cijelog prevede u strojni jezik, što inače radi jezični prevoditelj). Dakle, stranica je pisana u kodu koji preglednik interpretira u čitljiv sadržaj. Korisnici putem web preglednika šalju zahtjeve poslužitelju.

Web poslužitelj je temelj svake Web aplikacije jer on služi za komunikaciju klijenta s aplikacijom. Takva vrsta komunikacije odvija se uz pomoć HTTP (*Hyper Text Transfer Protocol*) protokola koji predstavlja glavnu i najčešću metodu prijenosa informacija na Webu. Osnovna namjena ovog protokola je omogućavanje objavljivanja i prezentacije HTML dokumenata, tj. Web stranica. Poslužitelj nam dakle služi za prosljeđivanje zahtjeva Web aplikaciji te je zadužen za njezino pokretanje.

Od ostalih protokola koji služe za komunikaciju i prijenos informacija i podataka između klijenta i poslužitelja treba napomenuti: IP (*Internet Protocol*) koji služi za prijenos podataka u blokovima (paketi, datagrami), komunikacijski protokol ARP (*Address Resolution Protocol*) te ICMP (*Internet Control Message Protocol*) koji je ugrađen u svaki IP modul kako bi omogućio računalima slanje kontrolnih poruka o greškama. ICMP prijavljuje greške, ali nije zadužen za njihovo ispravljanje.

Klijent koristi korisničko sučelje, tj. Web aplikaciju, prikazanu u pregledniku, te preko nje šalje zahtjeve poslužitelju. Nakon obrade podataka, poslužitelj vraća odgovor aplikaciji koja ga potom prikazuje korisniku.



Slika 4.1: Prikaz arhitekture sustava

Programski i korisnički jezici koje smo koristili su: HTML, CSS i JavaScript za Frontend te Java za Backend. Radni okviri koje smo odabrali su React (Frontend) i Spring Boot (Backend). Od razvojnih okruženja koristili smo IntelliJ u okviru izrade Backenda te Visual Studio Code za izradu Frontenda.

U okviru Backenda korišten je REST (*Representational State Transfer*) API (*Application programming Interface*). REST je softverska arhitektura koja nameće uvjete o tome kako API treba raditi, a API definira pravila koja morate slijediti za komunikaciju s drugim softverskim sustavima.

REST API je sučelje koje dva računalna sustava koriste za sigurnu razmjenu informacija putem interneta. Većina Web aplikacija mora komunicirati s drugim internim aplikacijama i aplikacijama trećih strana kako bi izvršile razne zadatke, a REST API podržava ovu razmjenu informacija jer slijedi sigurne, pouzdane i učinkovite softverske komunikacijske standarde.

4.1 Baza podataka

Podaci u aplikaciji će se spremati u relacijsku bazu podataka koristeći Postgres kao DBMS. Baza će se sastojati od sljedećih entiteta:

- Korisnik
- Mjesto
- Tvrtka
- Zaposlenik
- Projekt
- ProjectFRTeamMembers
- Suradnja

4.1.1 Opis tablica

Korisnik		
IDKorisnik	INT	ID korisnika
Ime	VARCHAR(40)	Ime korisnika
Prezime	VARCHAR(40)	Prezime korisnika
Nadimak	VARCHAR(40)	Nadimak korisnika
LoginEmail	VARCHAR(60)	Email adresa pomoću kojeg se korisnik logira
NotificationEmail	VARCHAR(60)	Email adresa na koju korisnik prima obavijesti
MaxRazinaOvlasti	INT	Maksimalna razina ovlasti koju korisnik posjeduje (ENUM)
Opis	VARCHAR(480)	Opis korisnika

Mjesto		
PBr	INT	Pošanski broj mjesta
Naziv	VARCHAR(120)	Naziv mjesta

Tvrtka		
IDTvrtka	INT	Id tvrtke
Naziv	VARCHAR(60)	Naziv tvrtke
Podrucje	VARCHAR(40)	Područje kojim se tvrtka bavi
ABCKategorizacija	CHAR	Kategorija tvrtke (ENUM: A, B ili C)
MjesecPlaniranjaBudzeta	INT	Mjesec u kojem tvrtka planira budžet
Drzava	VARCHAR(60)	Država u kojoj tvrtka posluje
PBr	INT	Pošanski broj mjesta tvrtke
Adresa	VARCHAR(120)	Adresa tvrtke
WebURL	VARCHAR(60)	Adresa web stranice tvrtke
Kontaktirati	BOOLEAN	Treba li ubuduće tu tvrtku kontaktirati

Zaposlenik		
IDZaposlenik	INT	Id zaposlenika
IDTvrtka	INT	Id tvrtke za koju zaposlenik radi
Ime	VARCHAR(40)	Ime zaposlenika
Prezime	VARCHAR(40)	Prezime zaposlenika
Email	VARCHAR(60)	Email adresa zaposlenika
Tel	VARCHAR(20)	Broj telefona zaposlenika

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

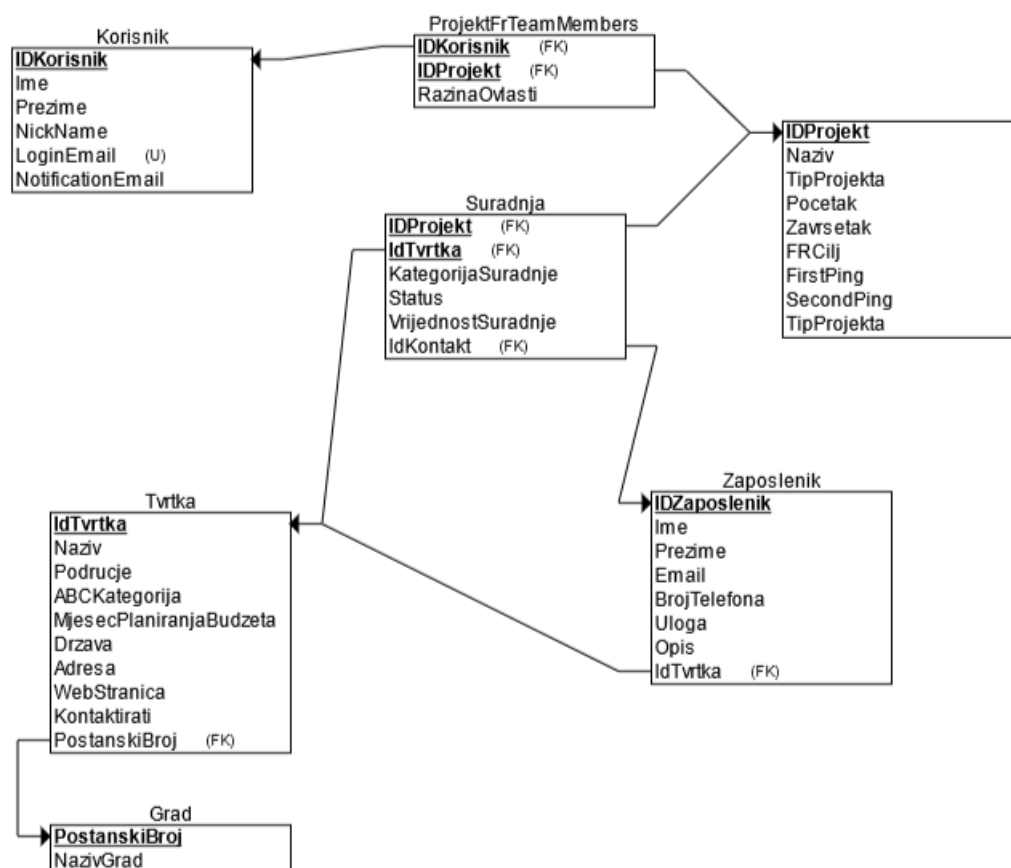
Zaposlenik		
Uloga	VARCHAR(40)	Pozicija zaposlenika u tvrtki (npr. PR)
Opis	VARCHAR(60)	Opis zaposlenika (npr. glavni kontakt)

Projekt		
IDProjekt	INT	ID projekta
Naziv	VARCHAR(40)	Naziv projekta
Kategorija	INT	Kategorija u koju spada projekt
Tip	INT	Tip projekta (ENUM: eksterni ili interni)
ProjektPocetak	TIMESTAMP	Datum početka projekta
ProjektZavrsetak	TIMESTAMP	Datum završetka projekta
IDFRResp	INT	ID korisnika odgovornog za odnose s tvrtkama projekta
FRCilj	INT	Količina novca koja se želi prikupiti za projekt
PrviFRPing	TIMESTAMP	Datum prvog "ping"-a
DrugiFRPing	TIMESTAMP	Datum drugog "ping"-a

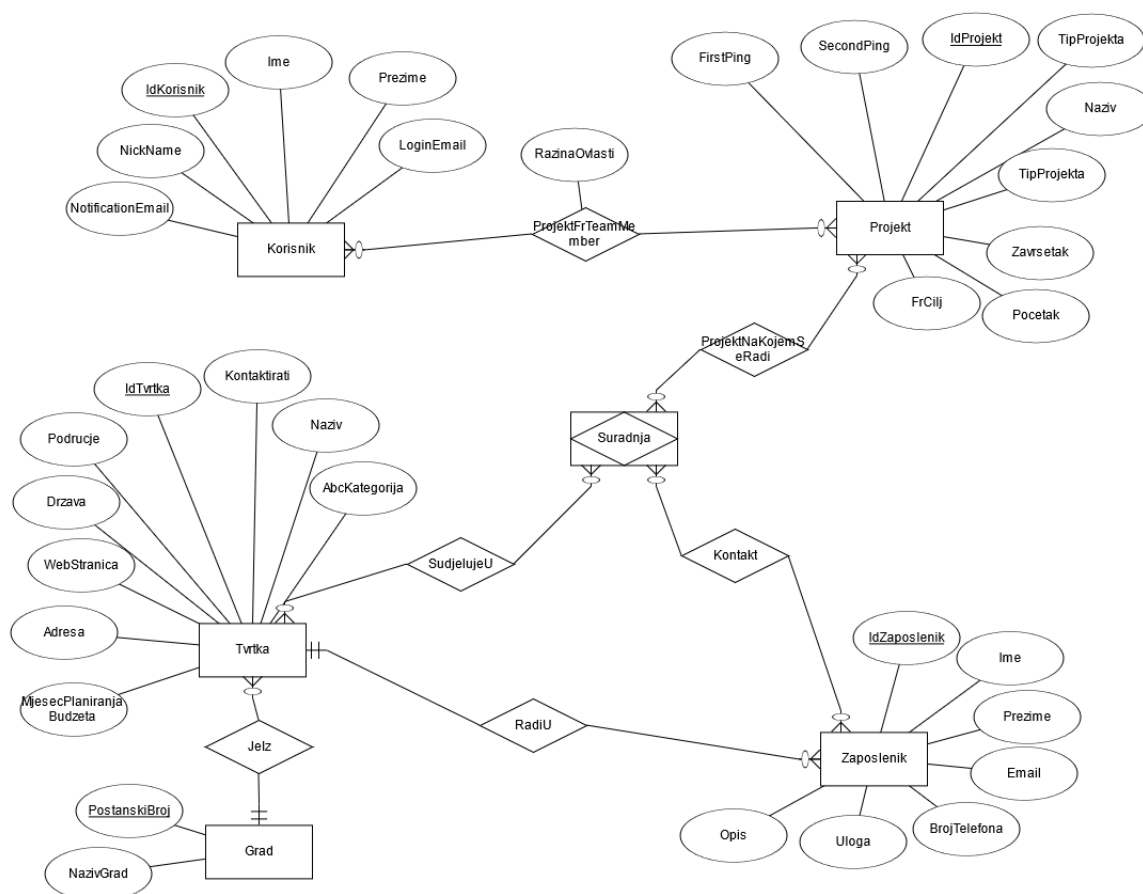
ProjectFRMembers		
IDKorisnik	INT	Id korisnika koji je FR team member projekta
IDProjekt	INT	Id projekta

Suradnja		
IDProjekt	INT	Id projekta
IDTvrтка	INT	Id tvrtke
IDOdgovoran	INT	Id korisnika odgovornog za suradnju s tvrtkom
IDKontakt	INT	Id kontakt osobe u tvrtci
Kategorija	VARCHAR(20)	Kategorija suradnje (ENUM: financijska, materijalna ili akademska)
Status	VARCHAR(20)	Kontaktirano, ping, dopis, sastanak, uspješno ili neuspješno
Komentar	VARCHAR(480)	Komentar na suradnju
Vrijednost	INT	Ostvarena novčana vrijednost suradnje

4.1.2 Dijagram baze podataka



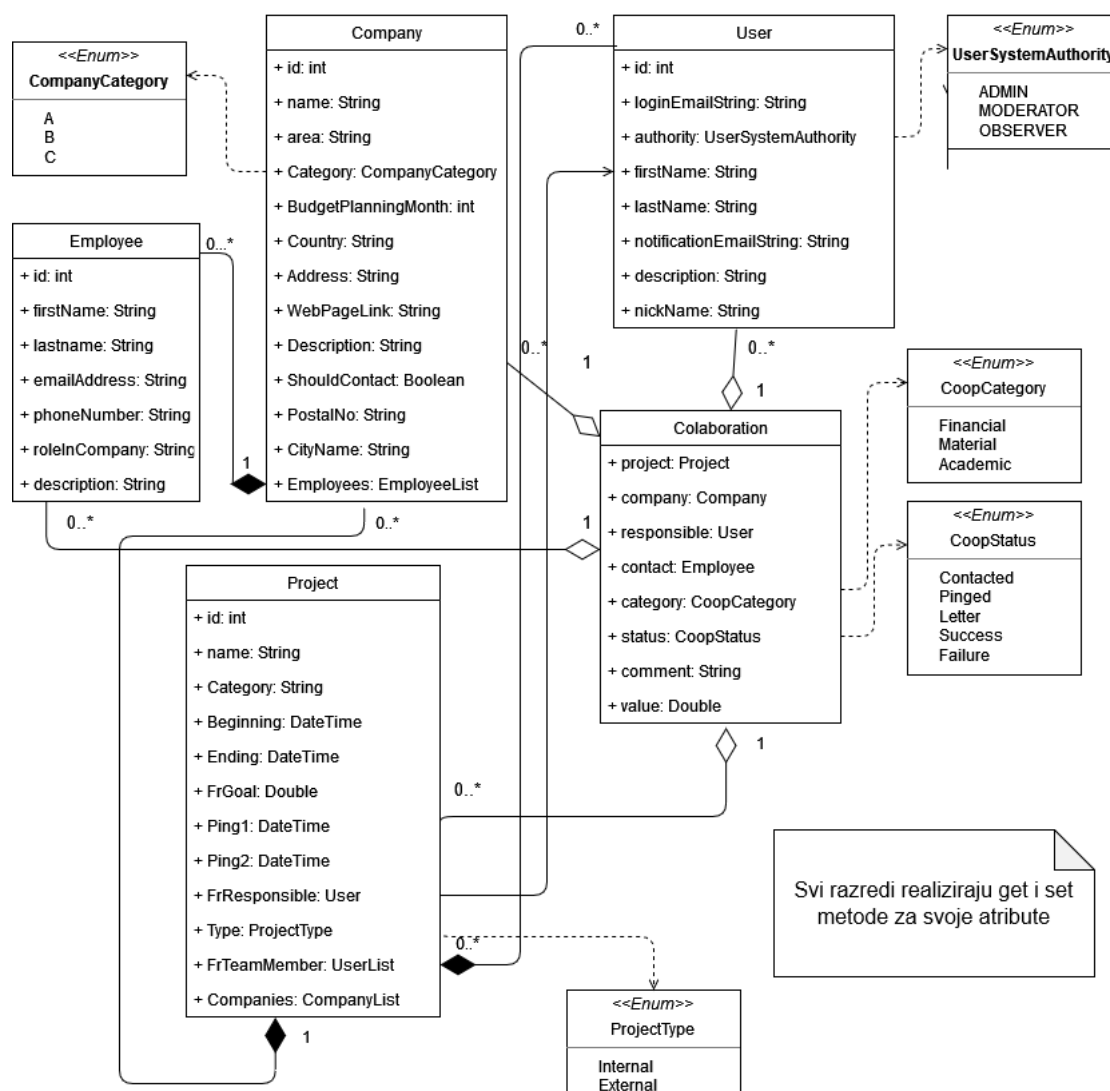
Slika 4.2: Dijagram baze podataka



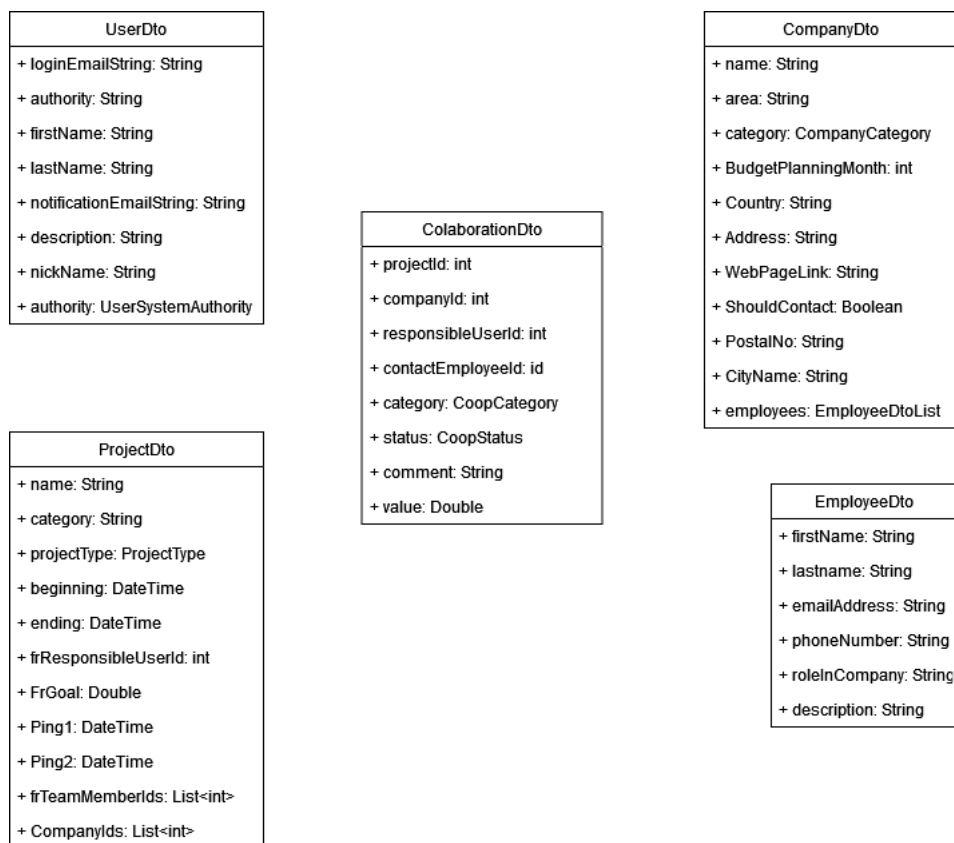
Slika 4.3: Entity-relationship dijagram

4.2 Dijagram razreda

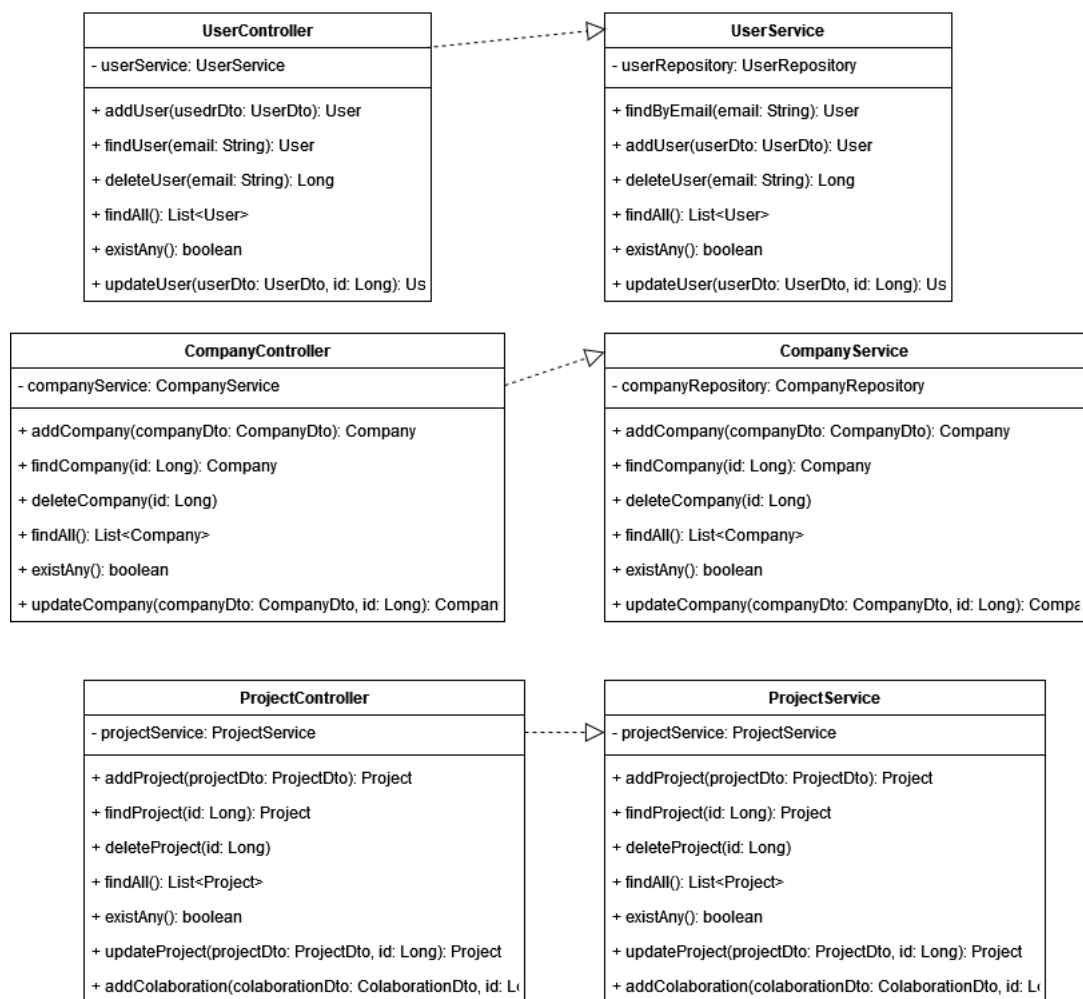
Na sljedećim slikama prikazani su UML dijagrami razreda koji se koriste u aplikaciji. Podijeljeni su u 3 dijela. Prvi dijagram prikazuje klase kojima smo modelirali entitete aplikacije, drugi prikazuje DTO-ove (*Data transfer object*), a treći prikazuje kontrolere. Može se uočiti da postoje veze između klasa iz različitih dijagrama (npr. UserDto sadrži UserSystemAuthority), ali su one izostavljene radi preglednosti.



Slika 4.4: UML dijagram razreda



Slika 4.5: UML dijagram Data Transfer Object-a



Slika 4.6: UML dijagram kontrolera i servisa

5. Implementacija i korisničko sučelje

5.1 Korištene tehnologije i alati

Većinski dio komunikacije održan je putem WhatsApp¹ komunikacijskog kanala, dok je drugi dio održan ili uživo ili preko platforme Discord². Prilikom izrade dijagrama korišten je alat Astah³, dok je za pisanje dokumentacije korišten online editor Overleaf⁴.

U svrhu verzioniranja koda od strane svih članova grupe korišten je alat Git⁵, dok je kao udaljeni repozitorij korištena platforma GitLab⁶.

Za razvoj Backend dijela aplikacije korišten je Java JDK⁷, gradle⁸ i IntelliJ Idea⁹ razvojno okruženje za razvoj Spring boot¹⁰ aplikacije. Za Frontend dio aplikacije korišten je Node.js¹¹ kao program za pokretanje i React¹², također poznat kao React.js ili ReactJS, kao Javascript¹³ biblioteka za izradu sučelja. Kao razvojno okruženje za React korišten je VS Code¹⁴ IDE.

Za upravljanje bazom podataka korišten je PostgreSQL¹⁵ instaliran na Ubuntu serveru¹⁶ verzije 20.04.

¹<https://www.whatsapp.com/>

²<https://discord.com/>

³<https://astah.net/>

⁴<https://www.overleaf.com/>

⁵<https://git-scm.com/>

⁶<https://gitlab.com/>

⁷<https://www.oracle.com/java/technologies/downloads/>

⁸<https://gradle.org/>

⁹<https://www.jetbrains.com/idea/>

¹⁰<https://spring.io/projects/spring-boot>

¹¹<https://nodejs.org/en/>

¹²<https://reactjs.org/>

¹³<https://www.javascript.com/>

¹⁴<https://code.visualstudio.com/>

¹⁵<https://www.postgresql.org/>

¹⁶<https://ubuntu.com/download/server>

5.2 Ispitivanje programskog rješenja

5.2.1 Ispitivanje komponenti

Svi endpointovi na backend-u su ručno testirani, a ispitivanje jedinica (engl. unit testing) proveli smo nad osnovnim funkcionalnostima servisnog sloja koristeći biblioteku junit. Prilikom testiranja pojedinog razreda, servise i repozitorije čije metode taj razred poziva smo *mock-ali* koristeći biblioteku mockito, pa smo samo u svakoj test metodi zadali što će pojedina pozvana metoda vraćati.

1. UserServiceTests

U UserServiceTests testnoj klasi ispitali smo funkcionalnost dodavanja novog korisnika u bazu podataka.

Naredbom `Mockito.when(userRepository.save(Mockito.any())).thenReturn(test);` smo rekli mockanom userRepository-ju kako da se ponaša prilikom poziva save metode.

```
@Test
public void addUserTest() {
    UserDTO userDTO = new UserDTO(
        loginEmailString: "peroperic@gmail.com",
        AUTHORITY.ADMIN,
        firstName: "pero",
        lastName: "peric",
        notificationEmailString: "nest@nesto.com",
        description: "opis",
        nickname: "perica"
    );
    AppUser test = new AppUser(
        loginEmailString: "peroperic@gmail.com",
        AUTHORITY.ADMIN,
        firstName: "pero",
        lastName: "peric",
        notificationEmailString: "nest@nesto.com",
        description: "opis",
        nickname: "perica"
    );

    Mockito.when(userRepository.save(Mockito.any())).thenReturn(test);

    AppUser user = userService.addUser(userDTO);

    assertThat(user).isSameAs(test);
}
```

Slika 5.1: Add user test

```
✓ Tests passed: 1 of 1 test – 855 ms

Starting Gradle Daemon...
Gradle Daemon started in 8 s 370 ms

> Task :compileJava UP-TO-DATE
> Task :processResources UP-TO-DATE
> Task :classes UP-TO-DATE
> Task :compileTestJava UP-TO-DATE
> Task :processTestResources NO-SOURCE
> Task :testClasses UP-TO-DATE
> Task :test
BUILD SUCCESSFUL in 23s
4 actionable tasks: 1 executed, 3 up-to-date
12:02:29 AM: Execution finished ':test --tests "com.example.backend.UserServiceTests.addUserTest"'.

```

Slika 5.2: Add user test result

2. ProjectServiceTests

U ProjectServiceTests klasi smo ispitali funkcionalnosti dohvata projekta iz baze podataka te dodavanje FR team membera na projekt.

```
@Test
public void findById_ReturnExpected(){
    Project project = mockProject();
    Mockito.when(projectRepository.findById(Mockito.any())).thenReturn(Optional.of(project));
    Project result = projectService.findById(Long.valueOf(1));
    assertThat(result).isEqualTo(project);
}

```

Slika 5.3: Find project by Id test

```
✓ Tests passed: 1 of 1 test – 877 ms

> Task :compileJava UP-TO-DATE
> Task :processResources UP-TO-DATE
> Task :classes UP-TO-DATE
> Task :compileTestJava UP-TO-DATE
> Task :processTestResources NO-SOURCE
> Task :testClasses UP-TO-DATE
> Task :test
BUILD SUCCESSFUL in 3s
4 actionable tasks: 1 executed, 3 up-to-date
12:06:50 AM: Execution finished ':test --tests "com.example.backend.ProjectServiceTests.findById_ReturnExpected"'.

```

Slika 5.4: Find project by Id test result

```
@Test
public void addFrTeamMember_AddsTeamMember(){
    Project project = mockProject();
    AppUser user = mockUser();
    Mockito.when(projectRepository.findById(Mockito.any())).thenReturn(Optional.of(project));
    Mockito.when(userRepository.findById(Mockito.any())).thenReturn(Optional.of(user));
    projectService.addFrTeamMember(Long.valueOf(1), Long.valueOf(1));
    assertThat(project.getFrteammembers().contains(user);
    assertThat(project.getFrteammembers().size()).isEqualTo( expected: 1);
}
```

Slika 5.5: Add FR team member to project test

```
✓ Tests passed: 1 of 1 test - 821 ms

> Task :compileJava UP-TO-DATE
> Task :processResources UP-TO-DATE
> Task :classes UP-TO-DATE
> Task :compileTestJava UP-TO-DATE
> Task :processTestResources NO-SOURCE
> Task :testClasses UP-TO-DATE
> Task :test
BUILD SUCCESSFUL in 2s
4 actionable tasks: 1 executed, 3 up-to-date
12:08:54 AM: Execution finished ':test --tests "com.example.backend.ProjectServiceTests.addFrTeamMember_AddsTeamMember"'.

```

Slika 5.6: Add FR team member result

3. CompanyServiceTests

U CompanyServiceTests klasi smo ispitali sljedeće funkcionalnosti:

Kreiranje nove kompanije

Dohvaćanje svih kompanije

Bacanje EntityNotFoundException-a prilikom pokušaja brisanja nepostojeće kompanije

Bacanje AuthenticationException-a prilikom pokušaja dohvata podataka od kompanije od strane Usera koji je samo Observer

```
@Test
public void createCompany_CreatesCompany() throws AuthenticationException
{
    CompanyDto companyDto = new CompanyDto(
        name: "name",
        domain: "domain",
        abcCategory: 'a',
        budgetPlanningMonth: 2,
        country: "Croatia",
        zipCode: 10000,
        address: "Address",
        webUrl: "www.example.com",
        contactInFuture: false);

    Company company = new Company(
        name: "name",
        domain: "domain",
        abcCategory: 'a',
        budgetPlanningMonth: 2,
        country: "Croatia",
        zipCode: 10000,
        address: "Address",
        webUrl: "www.example.com",
        contactInFuture: false);

    Mockito.when(companyRepository.save(Mockito.any())).thenReturn(company);
    Company result = companyService.createCompany(mockUser(AUTHORITY.ADMIN), companyDto);
    assertThat(result).isSameAs(company);
}
```

Slika 5.7: Create company test

```
✓ Tests passed: 1 of 1 test - 805 ms

> Task :compileJava UP-TO-DATE
> Task :processResources UP-TO-DATE
> Task :classes UP-TO-DATE
> Task :compileTestJava UP-TO-DATE
> Task :processTestResources NO-SOURCE
> Task :testClasses UP-TO-DATE
> Task :test
BUILD SUCCESSFUL in 2s
4 actionable tasks: 1 executed, 3 up-to-date
12:16:18 AM: Execution finished ':test --tests "com.example.backend.CompanyServiceTests.createCompany_CreatesCompany"'.
```

Slika 5.8: Create company test result


```
@Test
public void getAllCompanies_ReturnsCompanies() throws AuthenticationException
{
    List<Company> companies = List.of(
        new Company(Long.valueOf(1), name: "name", domain: "domain", abcCategory: "a", budgetPlanningMonth: 3, country: "Croatia", zipCode: 10000, address: "Address", webUrl: "example.com", contactInfoFuture: false),
        new Company(Long.valueOf(2), name: "name", domain: "domain", abcCategory: "a", budgetPlanningMonth: 3, country: "Croatia", zipCode: 10000, address: "Address", webUrl: "example.com", contactInfoFuture: false),
        new Company(Long.valueOf(3), name: "name", domain: "domain", abcCategory: "a", budgetPlanningMonth: 3, country: "Croatia", zipCode: 10000, address: "Address", webUrl: "example.com", contactInfoFuture: false));
    Mockito.when(companyRepository.findAll()).thenReturn(companies);

    List<Company> result = companyService.getAllCompanies(mockUser(AUTHORITY.ADMIN));
    assertThat(result).isEqualTo(companies);
}
```

Slika 5.9: Get all companies test

```
✓ Tests passed: 1 of 1 test - 852 ms

> Task :compileJava UP-TO-DATE
> Task :processResources UP-TO-DATE
> Task :classes UP-TO-DATE
> Task :compileTestJava
> Task :processTestResources NO-SOURCE
> Task :testClasses
> Task :test
BUILD SUCCESSFUL in 5s
4 actionable tasks: 2 executed, 2 up-to-date
12:12:22 AM: Execution finished ':test --tests "com.example.backend.CompanyServiceTests.getAllCompanies_ReturnsCompanies"'.
```

Slika 5.10: Get all companies test result

```
@Test
public void deleteCompany_WhenCompanyDoesntExist_ThrowsEntityNotFoundException(){
    Mockito.when(companyRepository.findById(Mockito.any())).thenReturn( value: Optional.empty());
    Assertions.assertThrows(EntityNotFoundException.class, () -> {
        companyService.deleteCompany(mockUser(AUTHORITY.ADMIN), Long.valueOf(1));
    });
}
```

Slika 5.11: Delete company which does not exist test

```
✓ Tests passed: 1 of 1 test - 784 ms

> Task :compileJava UP-TO-DATE
> Task :processResources UP-TO-DATE
> Task :classes UP-TO-DATE
> Task :compileTestJava UP-TO-DATE
> Task :processTestResources NO-SOURCE
> Task :testClasses UP-TO-DATE
> Task :test
BUILD SUCCESSFUL in 2s
4 actionable tasks: 1 executed, 3 up-to-date
12:17:25 AM: Execution finished ':test --tests "com.example.backend.CompanyServiceTests.deleteCompany_WhenCompanyDoesntExist_ThrowsEntityNotFoundException"'.
```

Slika 5.12: Delete company which does not exist tes result

```
@Test
public void getCompany_IfUserIsObserver_ThrowsAuthenticationException() throws AuthenticationException
{
    Assertions.assertThrows(AuthenticationException.class, () -> {
        companyService.getCompany(mockUser(AUTHORITY.OBSERVER), Long.valueOf(1));
    });
}
```

Slika 5.13: Get company info by observer test

```
✓ Tests passed: 1 of 1 test - 745 ms

> Task :compileJava UP-TO-DATE
> Task :processResources UP-TO-DATE
> Task :classes UP-TO-DATE
> Task :compileTestJava UP-TO-DATE
> Task :processTestResources NO-SOURCE
> Task :testClasses UP-TO-DATE
> Task :test
BUILD SUCCESSFUL in 2s
4 actionable tasks: 1 executed, 3 up-to-date
12:14:50 AM: Execution finished ':test --tests "com.example.backend.CompanyServiceTests.getCompany_IfUserIsObserver_ThrowsAuthenticationException"'.

```

Slika 5.14: Get company info by observer test result

4. CollaborationServiceTests

U CollaborationServiceTests testnoj klasi smo ispitali funkcionalnost dodavanja suradnje u bazu podataka.

```
@Test
public void addCollaboration_AddsCollaboration(){
    Project project = mockProject();
    Company company = mockCompany();
    Contact contact = mockContact();
    AppUser responsible = mockUser();

    Mockito.when(projectRepository.findById(Mockito.any())).thenReturn(Optional.of(project));
    Mockito.when(companyRepository.findById(Mockito.any())).thenReturn(Optional.of(company));
    Mockito.when(contactRepository.findById(Mockito.any())).thenReturn(Optional.of(contact));
    Mockito.when(userRepository.findById(Mockito.any())).thenReturn(Optional.of(responsible));

    Collaboration collaboration = new Collaboration(
        new CollaborationId(project, company),
        contact,
        responsible,
        priority: false,
        Category.ACADEMIC,
        Status.LETTER,
        comment: "Comment",
        Long.valueOf(12000)
    );

    Mockito.when(collaborationsRepository.save(Mockito.any())).thenReturn(collaboration);
    CollaborationDTO collaborationDTO = new CollaborationDTO(
        Long.valueOf(1),
        Long.valueOf(1),
        Long.valueOf(1),
        priority: false,
        Category.ACADEMIC,
        Status.LETTER,
        comment: "Comment",
        Long.valueOf(12000)
    );

    Collaboration result = collaborationsService.addCollaboration(Long.valueOf(1), collaborationDTO);
    assertThat(result).isSameAs(collaboration);
}
```

Slika 5.15: Add collaboration test

```
✓ Tests passed: 1 of 1 test - 1 sec 51 ms

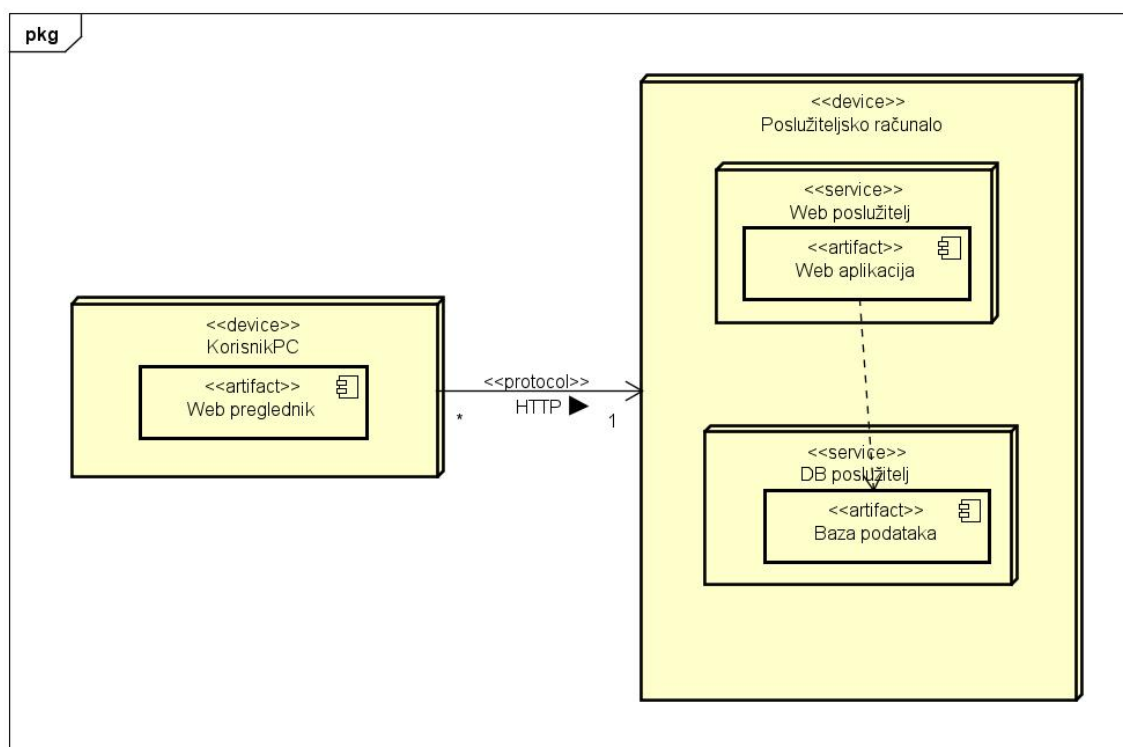
> Task :compileJava UP-TO-DATE
> Task :processResources UP-TO-DATE
> Task :classes UP-TO-DATE
> Task :compileTestJava UP-TO-DATE
> Task :processTestResources NO-SOURCE
> Task :testClasses UP-TO-DATE
> Task :test
BUILD SUCCESSFUL in 3s
4 actionable tasks: 1 executed, 3 up-to-date
12:26:22 AM: Execution finished ':test --tests "com.example.backend.CollaborationsServiceTests.addCollaboration_AddsCollaboration"'.

```

Slika 5.16: Add collaboration test result

5.3 Dijagram razmještaja

Na poslužiteljskom računalu se nalaze web poslužitelj i poslužitelj baze podataka. Klijenti koriste web preglednik kako bi pristupili web aplikaciji. Sustav je baziran na arhitekturi "klijent – poslužitelj", a komunikacija između računala korisnika (klijent, zaposlenik, vlasnik, administrator) i poslužitelja odvija se preko HTTP veze.



Slika 5.17: Dijagram razmještaja

5.4 Upute za puštanje u pogon

Kako se aplikacija sastoji od dva distinktivna dijela, backend-a i frontend-a, svaki dio funkcionira samo za sebe te se zasebno pušta u pogon.

5.4.1 Backend

Backend se sastoji od koda u obliku gradle projekta u rađenom u Spring boot okruženju. Kako Spring boot samostalno radi tablice i ažurira relacije u bazi podataka, potrebno je samo na poslužitelju na kojem će se nalaziti kompajliran pokrenut kod stvoriti bazu podataka s nazivom *cdb*.

Za početak je potrebno otvoriti virtualni stroj s ubuntu serverom na poslužitelju po izboru nakon čega je potrebno unesti sljedeće naredbe koristeći command prompt kako bi se ulogirali i postavili server:

- `ssh root@ipv4AdresaServera`
- `sudo apt update`
- `sudo apt upgrade`
- `sudo apt install fail2ban`
- `sudo systemctl restart fail2ban.service`

Sada kada je server postavljen, potrebno je instalirati postgresql i podignuti bazu naziva *cdb*:

- `sudo apt-get install postgresql postgresql-contrib`
- `sudo systemctl start postgresql.service`
- `sudo systemctl enable postgresql.service`
- `sudo -u postgres psql`
- `\password postgres`
- `CREATE DATABASE cdb`
- `\c cdb`
- `\q`

Nakon što je baza podataka na serveru stvorena, potrebno je kompajlirati kod za backend aplikacije te ga tako kompajliranog prebaciti na server. Sljedeće naredbe potrebno je izvršiti u matičnog folderu koda preuzetnog na vlastito računalo:

- `cd Backend`

- `gradle bootJar`
- `scp build/libs/backend-0.0.1-SNAPSHOT.jar root@ipv4AdresaServera:/var/www`

Nakon što je kompajlirani kod na serveru, potrebno je instalirati javu kako bi ga mogli pokrenuti:

- `sudo apt install openjdk-17-jdk`

Kako bi osigurali da aplikacija radi i kada na serveru ne postoji ulogirani korisnik, potrebno je kompajliran kod pokrenuti kao servis:

- `cd /usr/lib/systemd/system`
- `nano runSpringServer.service`
- *Sljedeći kod upiši u novootvoreni text editor:*
- [Unit]
- Description=webserver Daemon
- [Service]
- ExecStart=/usr/bin/java -jar /var/www/backend-0.0.1-SNAPSHOT.jar
- User=root
- [Install]
- WantedBy=multi-user.target
- *Zatvori text editor koristeći kombinacije tipki CTRL+S pa CTRL+X.*
- `sudo systemctl start runSpringServer.service`
- `sudo systemctl enable runSpringServer.service`

5.4.2 Frontend

Frontend je aplikacija rađena u React frameworku za čije je pokretanje i izgradnju potreban *Node.js* koji je moguće preuzeti s adrese <https://nodejs.org/en/download/>. Nakon instalacije *Node.js*-a na vlastito računalo, u matičnom folderu koda preuzetog na vlastito računalo potrebno je izvršiti sljedeće naredbe kako bi kompajlirali frontend dio aplikacije:

- `cd Frontend`
- `npm install`
- `npm run build`

Nakon izvršenja ovih naredbi, u mapi Frontend će se stvoriti nova mapa naziva *Build*. Ovu mapa sadrži kompajlirani kod Frontend aplikacije koji će biti potrebno prenesti na poslužitelj.

Kako bi korisnici kroz preglednik mogli pristupiti našoj aplikaciji potrebno ju je servirati na poslužitelju. Poslužitelj s kojeg će biti servirana naša aplikacija zove se *Netlify*. Prvo je potrebno otići na adresu <https://app.netlify.com/> i ulogirati se s već postojećim Github računom. Nakon toga potrebno je odabrati opciju *Add new site* pa *Deploy manually*. Prateći ove upute biti će ponuđen ekran na koji je potrebno ispustiti *Build* mapu napravljenu u prethodnom koraku.

Nakon nekoliko minuta Frontend aplikacija će se poslužiti te će nam Netlify dati IP adresu preko koje možemo pristupiti svojoj aplikaciji (npr. <https://darling-lokum-558ee5.netlify.app>).

Po želji, preteći Netlify dokumentaciju moguće je promijeniti IP adresu aplikacije kao i mnoge druge postavke koje Netlify nudi, no koji nisu potrebne za funkcioniranje aplikacije.

6. Zaključak i budući rad

dio 2. revizije

U ovom poglavlju potrebno je napisati osvrt na vrijeme izrade projektnog zadatka, koji su tehnički izazovi prepoznati, jesu li riješeni ili kako bi mogli biti riješeni, koja su znanja stečena pri izradi projekta, koja bi znanja bila posebno potrebna za brže i kvalitetnije ostvarenje projekta i koje bi bile perspektive za nastavak rada u projektnoj grupi.

Potrebno je točno popisati funkcionalnosti koje nisu implementirane u ostvarenoj aplikaciji.

Popis literature

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

Indeks slika i dijagrama

2.1	Prikaz alata Google sheets	6
3.1	Funkcionalnosti observera	23
3.2	Funkcionalnosti člana tima za odnose s tvrtkama	24
3.3	Funkcionalnosti osobe odgovorne za odnose s tvrtkama na projektu	24
3.4	Funkcionalnosti osobe odgovorne za odnose s tvrtkama	25
3.5	Funkcionalnosti administratora	25
3.6	Sekvencijski dijagram za UC3 - Registracija korisnika	26
3.7	Sekvencijski dijagram za UC13 - Stvaranje suradnje	27
3.8	Sekvencijski dijagram za UC15 - Stvaranje projekta	28
3.9	Sekvencijski dijagram za UC23 - Stvaranje kompanije	29
4.1	Prikaz arhitekture sustava	32
4.2	Dijagram baze podataka	37
4.3	Entity-relationship dijagram	38
4.4	UML dijagram razreda	39
4.5	UML dijagram Data Transfer Object-a	40
4.6	UML dijagram kontrolera i servisa	41
5.1	Add user test	44
5.2	Add user test result	45
5.3	Find project by Id test	45
5.4	Find project by Id test result	45
5.5	Add FR team member to project test	46
5.6	Add FR team member result	46
5.7	Create company test	47
5.8	Create company test result	47
5.9	Get all companies test	48
5.10	Get all companies test result	48
5.11	Delete company which does not exist test	48
5.12	Delete company which does not exist tes result	48

5.13 Get company info by observer test	49
5.14 Get company info by observer test result	49
5.15 Add collaboration test	50
5.16 Add collaboration test result	50
5.17 Dijagram razmještaja	51
6.1 Prikaz aktivnosti na repozitoriju za granu develop	62
6.2 Prikaz aktivnosti na repozitoriju za granu devdoc	63

Dodatak: Prikaz aktivnosti grupe

Dnevnik sastajanja

1. sastanak

- Datum: 17. listopada 2022.
- Prisustvovali: P. Hajduk., M. Čengić, N. Capan, J. Jakovac
- Teme sastanka:
 - Predstavljanje nove teme za prijedlog
 - Rasprava o novoj temi za prijedlog
 - Rasprava o mogućoj podjeli rada
 - Teambuilding

2. sastanak

- Datum: 14. studenoga 2022.
- Prisustvovali: P. Hajduk., M. Čengić, N. Capan, J. Jakovac, M. Balog, I. Baričević, L. Čunović
- Teme sastanka:
 - Rasprava o implementacijskim detaljima
 - Rasprava o daljnjim koracima

3. sastanak

- Datum: 04. prosinca 2022.
- Prisustvovali: P. Hajduk., M. Čengić, N. Capan, J. Jakovac, M. Balog, I. Baričević
- Teme sastanka:
 - Rekapitulacija detalja napravljenog
 - Rasprava o bazičnim stavkama funkcionalnosti aplikacije

Tablica aktivnosti

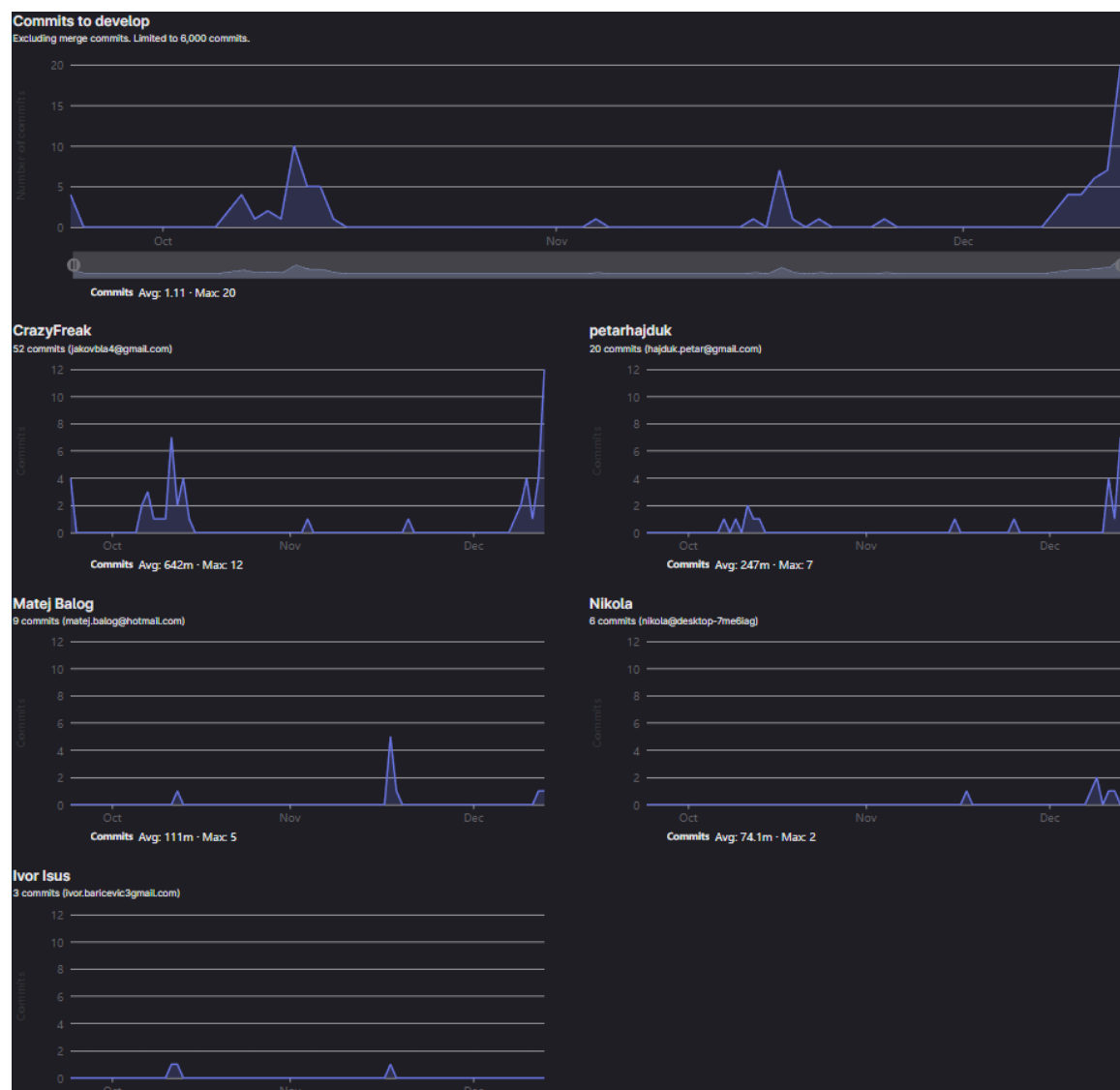
	Petar Hajduk	Jakov Jakovac	Matej Balog	Ivor Baričević	Marko Čengić	Nikola Capan	Lovro Čunović
Upravljanje projektom	0	18	0	0	2	0	0
Opis projektnog zadatka	0	10	0	0	0	0	2
Funkcionalni zahtjevi	0	1	0	0	2	0	0
Opis pojedinih obrazaca	0	3	0	0	8	0	0
Dijagram obrazaca	0	0	0	0	4	0	0
Sekvencijski dijagrami	0	0	0	0	4	0	0
Opis ostalih zahtjeva	0	0	0	0	1	0	0
Arhitektura i dizajn sustava	0	3	0	0	0	0	2
Baza podataka	2	2	0	0	0	7	0
Dijagram razreda	0	0	0	0	0	8	0
Dijagram stanja	0	0	0	0	2	0	0
Dijagram aktivnosti	0	0	0	0	2	0	0
Dijagram komponenti	0	0	0	0	2	0	0
Korištene tehnologije i alati	0	2	0	0	0	0	0
Ispitivanje programskog rješenja	3	2	0	0	0	2	0
Dijagram razmještaja	0	0	0	0	2	2	0
Upute za puštanje u pogon	0	2	0	0	0	0	0
Dnevnik sastajanja	0	1	0	0	0	0	0
Zaključak i budući rad	0	0	0	0	1	0	0
Popis literature	0	0	0	0	1	0	0

Nastavljeno na idućoj stranici

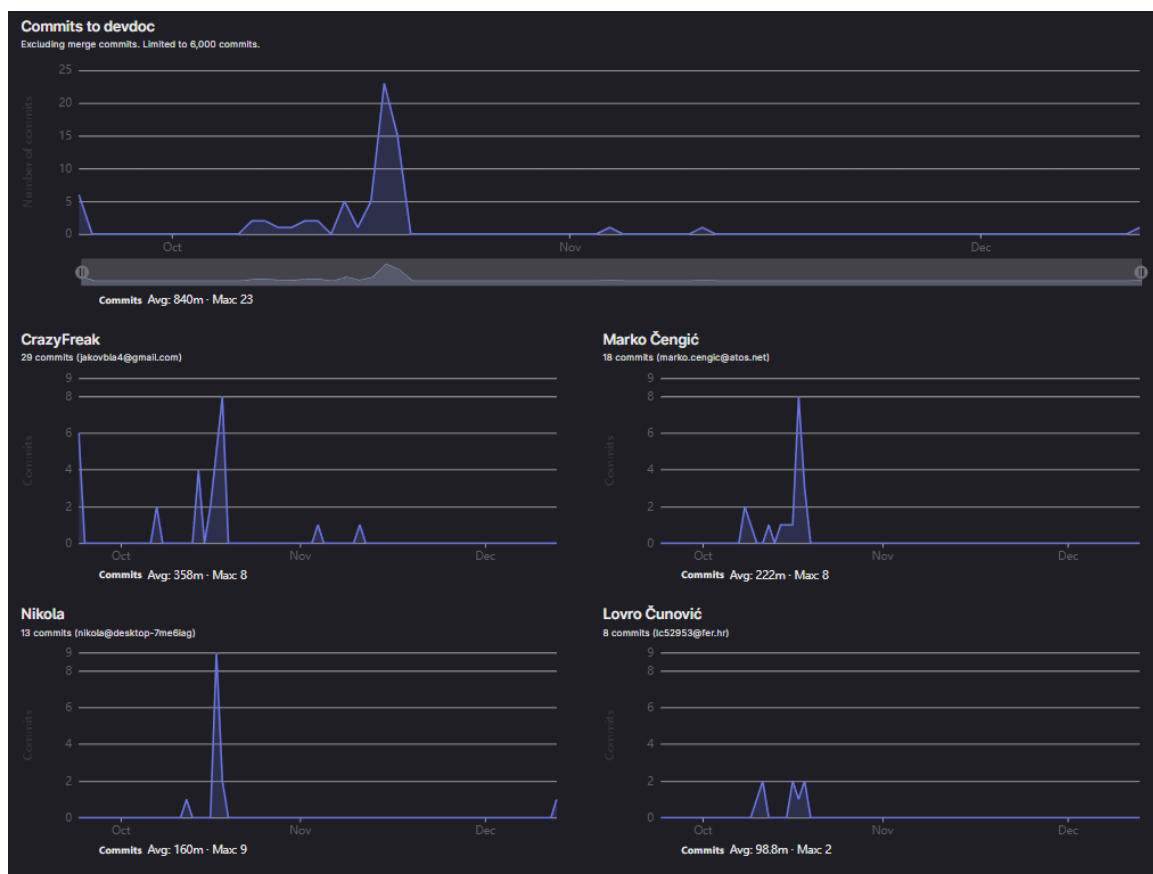
Nastavljeno od prethodne stranice

	Petar Hajduk	Jakov Jakovac	Matej Balog	Ivor Baričević	Marko Čengić	Nikola Capan	Lovro Čunović
Dizajn aplikacije	0	13	1	1	0	0	0
Rad na responzivnosti aplikacije	0	0	0	0	0	0	0
Implementacija Login/Logout	0	7	0	0	0	0	0
Implementacija Razina ovlasti	0	0	0	0	0	0	0
Izrada headera	0	8	0	2	0	0	0
Izrada naslovnice	0	1	0	0	0	0	0
Izrada users stranice	0	1	4	3	0	0	0
Izrada user forme	0	4	0	2	0	0	0
Izrada delete forme	0	0	0	1	0	0	0
Izrada projects stranice	0	3	5	0	0	0	0
Izrada project forme	0	0	4	0	0	0	0
Izrada companies stranice	0	2	4	0	0	0	0
Izrada company forme	0	0	3	0	0	0	0
Izrada collaborations komponente	0	4	2	3	0	0	0
Izrada spring backenda	17	1	0	2	0	11	0
Deploy aplikacije	0	7	0	0	0	0	0

Dijagrami pregleda promjena



Slika 6.1: Prikaz aktivnosti na repozitoriju za granu develop



Slika 6.2: Prikaz aktivnosti na repozitoriju za granu devdoc