

Programsko inženjerstvo

Ak. god. 2021./2022.

## Radno vrijeme

Dokumentacija, Rev. 0.5.1

Grupa: *Dugonogi Progi*

Voditelj: *Bernard Kazazić*

Datum predaje: *14. siječnja 2022.*

Nastavnik: *Miljenko Krhen*

# Sadržaj

<b>1 Dnevnik promjena dokumentacije</b>	<b>3</b>
<b>2 Opis projektnog zadatka</b>	<b>4</b>
<b>3 Specifikacija programske potpore</b>	<b>8</b>
3.1 Funkcionalni zahtjevi . . . . .	8
3.1.1 Obrasci uporabe . . . . .	10
3.1.2 Sekvencijski dijagrami . . . . .	18
3.2 Ostali zahtjevi . . . . .	20
<b>4 Arhitektura i dizajn sustava</b>	<b>21</b>
4.1 Baza podataka . . . . .	22
4.1.1 Opis tablica . . . . .	22
4.1.2 Dijagram baze podataka . . . . .	26
4.2 Dijagram razreda . . . . .	28
4.3 Dijagram stanja . . . . .	38
4.4 Dijagram aktivnosti . . . . .	39
4.5 Dijagram komponenti . . . . .	40
<b>5 Implementacija i korisničko sučelje</b>	<b>41</b>
5.1 Korištene tehnologije i alati . . . . .	41
5.2 Ispitivanje programskog rješenja . . . . .	42
5.2.1 Ispitivanje komponenti . . . . .	42
5.2.2 Ispitivanje sustava . . . . .	42
5.3 Dijagram razmještaja . . . . .	43
5.4 Upute za puštanje u pogon . . . . .	44
<b>6 Zaključak i budući rad</b>	<b>45</b>
<b>Popis literature</b>	<b>46</b>
<b>Indeks slika i dijagonama</b>	<b>47</b>

**Dodatak: Prikaz aktivnosti grupe**

**48**

# 1. Dnevnik promjena dokumentacije

<b>Rev.</b>	<b>Opis promjene/dodatka</b>	<b>Autori</b>	<b>Datum</b>
0.1	Napravljen predložak.	Raspudić	14.10.2021.
0.1.1	Dopisani dnevnički sastajanja.	Raspudić	15.10.2021.
0.2	Dodani nefunkcionalni zahtjevi.	Raspudić	16.10.2021.
0.3	Dodani funkcionalni zahtjevi.	Pašalić, Sušac	17.10.2021.
0.4	Dodan opis projekta.	Pavlović, Raspudić	21.10.2021.
0.5	(1) Dodani obrasci uporabe i sekvencijski dijagrami  (2) Dodana arhitektura i dizajn sustava	(1) M.Erlić, A.Pašalić, V.Žunar  (2) B.Kazazić, P.Sušac	24.10.2021.
0.5.1	Manje gramatičke izmjene.	Raspudić	04.11.2021.
0.6	Napravljeni dijagrami razreda.	M. Erlić, B. Pavlović, B. Kazazić, V. Žunar	18.11.2021.
0.6.1	Opis dijagrama razreda	M. Erlić, B. Pavlović, B. Kazazić, V. Žunar	19.11.2021.

## 2. Opis projektnog zadatka

Cilj ovog projekta je razviti programsku podršku za web aplikaciju "Radno vrijeme" koja će omogućiti djelatnicima poduzeća „Mi puno radimo“ praćenje realizacije pojedinih djelatnosti i radnih sati na razini svakog djelatnika, grupe i poduzeća, kao i praćenje zauzeća/raspoloživosti pojedinih djelatnika. Na taj će se način olakšati organizacija istovremenog obavljanja mnogih uslužnih djelatnosti.

Korisnike sustava možemo podijeliti u 4 grupe:

- vlasnika sustava
- voditelja grupa
- zaposlenike
- neregistrirane korisnike

Neregistrirani korisnici mogu samo vidjeti popis i opis djelatnosti koje poduzeće ima u svojem portfoliju. Direktoru je omogućeno registriranje korisnika u sustav kreiranjem novog računa. Za kreiranje novog računa potrebni su sljedeći podaci:

- korisničko ime
- lozinka
- ime
- prezime
- OIB
- adresa e-pošte

Svim djelatnicima kasnije korisničko ime i lozinka služe za prijavu u sustav, a adresa e-pošte kako bi ih se moglo kontaktirati.

Naknadno se djelatnicima, ako već nisu, mogu još i dodijeliti prava voditelja grupe. Ukoliko radnici idu na intervencije izvan lokacije sjedišta poduzeća, potrebno je za svakoga prikazati mjesto na karti za lokaciju na koju je bio upućen.

Djelatnik poduzeća kroz aplikaciju može vidjeti podatke samo o sebi. Njemu se prikazuju grupe u koje je raspodijeljen te zadatke koji su mu dodijeljeni. Jedan djelatnik može biti raspoređen i na nekoliko radnih zadataka unutar jedne djelatnosti, a i između više njih.

Ukoliko je djelatniku dodijeljen zadatak na nekoj lokaciji različitoj od sjedišta poduzeća, on tu promjenu u lokaciji mora unijeti na karti prije odlaska, ako nije voditelj grupe već unio u sustav.

Voditelji grupe mogu kroz aplikaciju dobiti podatke o sebi, ali i o članovima svoje grupe. Oni razrađuju plan rada svoje grupe te određuju zadatke i pridjeljuju ih zaposlenicima. Za svaki zadatak voditelj određuje očekivani broj potrebnih radnih sati te cijenu radnog sata na razini pojedine djelatnosti i/ili zadatka.

Vlasnik može stvarati grupe, pri čemu zaposlenika postavlja kao voditelja te grupi dodjeljuje djelatnost i članove, a također ima mogućnost brisanja grupe.

On također može definirati nove uslužne djelatnosti kojima se poduzeće bavi te će te promjene biti vidljive u popisu djelatnosti.

Vlasniku sustava se kroz aplikaciju prikazuju zauzetost i realizacija (stvarna i materijalna) za sve djelatnike te odnos planiranih i realiziranih troškova/dobiti. Informacija o zauzetosti djelatnika pomaže mu pri stvaranju grupa kako ne bi preopteretio zaposlenike prevelikom količinom zadataka te na taj način osigurao veću količinu i kvalitetu održenog posla.

Ukoliko postoje aktivnosti izvan sjedišta poduzeća, direktor u svakom trenutku mora moći vidjeti na karti gdje se nalazi (ili se nalazio) koji djelatnik. Podaci o adresi intervencije moraju biti prethodno uneseni u sustav.

Na kraju svakog radnog dana svaki pojedini djelatnik (uključujući i voditelje

grupa) upisuju broj odrđenih radnih sati taj dan.

Sustav mora omogućiti istovremeni rad svih korisnika sustava i mora omogućiti unos hrvatskih dijakritičkih znakova.

Sustav također mora biti prilagođen i u potpunosti poštivati sva prava radnika, uključujući radno vrijeme (početak i kraj), duljinu radnog vremena, slobodne dana i ostala prava koja radnik ima temeljem općih i posebnih uvjeta definiranih zakonom.

Gledajući dostupne programe koji bi omogućili praćenje produktivnosti zaposlenika možemo vidjeti razne kategorije. Od najpopularinijih, najskupljih, najjednostavnijih pa sve do najjeftinijih. Ujedno je problem što je toliko veliki izbor između svih programa. To otežava voditeljima poduzeća posao odmjeravanja jakosti i slabosti svakog programa. Pošto su svi ti programi namjenjeni široj publici, svaki od tih programa će doći sa svojim opcijama koje nisu nužne niti potrebne poduzeću "Mi puno radimo". Ukoliko odaberu neki program, tu dolazi još jedan problem. Učenje korištenja tog programa (engl. *learning curve*). Potrebno je vrijeme i trud prije nego što počnu koristiti taj program optimizirano.

Tu dolazimo mi. Mi dajemo priliku poduzeću "Mi puno radimo" sa krojenim programom prema njihovim potrebama. Ne trebaju plaćati za opcije koje neće koristiti, ne trebaju odustati od svojih zahtjeva kako bi koristili jeftinija i/ili lošija rješenja. Dajemo im priliku da napravimo program koji će oni sami intuitivno znati koristiti jer će sudjelovati u svakom koraku korisničkog sučelja i njegovih funkcionalnosti.

Projekt je uvijek moguće nadograditi. Ovaj projekt je savršen za male, srednje i velike tvrtke. Možemo uz dogovor s klijentima povezati program s njihovim drugim servisima za koje drugi programi nemaju mogućnost, a niti im je u planu jer većina ne bi koristila. Trenutni cilj projekta nije specifičan da ga može samo jedno poduzeće koristiti. Na primjer, trenutna ideja je da zaposlenici sami unesu broj radnih sata za pojedini dan, moguće je nadograditi da se to automatski računa uz pametne kartice, optionalno koje bismo mi enkriptirali. Također, ukoliko neke tvrtke imaju već brojače sati za zaposlenike, ali im treba većina naših opcija, uz

dogovor možemo proširiti naš program da dobije podatke iz njihovog brojača te samo proširimo program za te servise koje oni traže.

Prednost našeg rješenja je što ne zahtjeva instalaciju na svaki uređaj koji treba pristupiti servisu. Svatko može sa svojeg pametnog mobitela ili računala doći do svojih zadataka, preinaka i novih informacija. Samo trebaju znati svoje korisničko ime i lozinku. Iako ova prednost može doći kao i mana, tu opet dolazimo do naše prethodne snage. Mi možemo, ukoliko je potrebno, ojačati sigurnost programa. Na način da program dozvoli pristup samo ovlaštenim računalima/mobitelima. Na primjer, zabrana pristupa osim dozvoljenim MAC adresama, dozvola pristupa računalima i mobitelima sa instaliranim certifikatom koji mi napišemo i sl. Naša tvrtka kroji proizvod svojim klijentima.

# 3. Specifikacija programske potpore

## 3.1 Funkcionalni zahtjevi

**Dionici:**

1. Vlasnik sustava (direktor poduzeća/naručitelj)
2. Djelatnici poduzeća
  - Voditelji grupa
  - Ostali djelatnici
3. Razvojni tim

**Aktori i njihovi funkcionalni zahtjevi:**

1. Vlasnik sustava (direktor poduzeća) može:
  - (a) definirati uslužne djelatnosti koje će poduzeće raditi
  - (b) dodijeliti djelatnosti voditeljima grupa
  - (c) registrirati novog djelatnika (uključujući i voditelja grupa) i pritom mu dodijeliti ulogu
  - (d) vidjeti zauzetost i realizaciju za sve djelatnike
  - (e) vidjeti trenutno i prošlu poticiju svih djelatnika koji su izašli na intervencije na karti
2. Voditelji grupa mogu:
  - (a) definirati zadatke i dodjeljivati ih djelatnicima
  - (b) zabilježiti procjenu radnih sati potrebnih za zadatak
  - (c) odrediti cijenu sata rada ovisno o djelatnosti ili zadatku
  - (d) pregledati podatke za sebe i svoju grupu
  - (e) upisati broj odrađenih sati za svaki dan

3. Ostali djelatnici mogu:

- (a) vidjeti koji su mu zadaci dodijeljeni i u kojim se grupama nalazi
- (b) pregledati vlastite podatke
- (c) upisati broj odrađenih radnih sati za svaki dan

4. Neregistrirani korisnik može:

- (a) vidjeti popis i opis djelatnosti poduzeća

### 3.1.1 Obrasci uporabe

#### Opis obrazaca uporabe

##### UC1 - Pregled djelatnosti

- **Glavni sudionik:** Korisnik
- **Cilj:** Pregledati djelatnosti koje poduzeće obavlja
- **Sudionici:** baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Korisnik u aplikaciji odabire opciju "Prikaz djelatnosti"
  2. Prikaže se popis svih djelatnosti poduzeća

##### UC2 - Pregled opisa djelatnosti

- **Glavni sudionik:** Korisnik
- **Cilj:** Pregledati opis djelatnosti koje poduzeće obavlja
- **Sudionici:** baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Korisnik odabire djelatnost za koju želi vidjeti informacije
  2. Prikaže se stranica s opisom odabrane djelatnosti

##### UC3 - Registracija korisnika

- **Glavni sudionik:** Direktor
- **Cilj:** Stvoriti korisnički račun za pristup sustavu
- **Sudionici:** baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Direktor odabire opciju za registraciju novih korisnika
  2. Direktor unosi sve potrebne korisničke podatke
  3. Direktor novom korisniku dodjeljuje ulogu
  4. Direktor prima obavijest o uspješnoj registraciji
- **Opis mogućih odstupanja:**
  - 2.a Odabir već zauzetog korisničkog imena i/ili elektroničke pošte, unos podataka u nedozvoljenom formatu ili upis neispravnog imena elektroničke pošte

1. Sustav obavljaštava direktora o neuspjelom upisu i vraća ga na stranicu za registraciju
2. Direktor mijenja potrebne podatke te završava unos ili odustaje od registracije

#### UC4 - Prijava u sustav

- **Glavni sudionik:** Djetatnik
- **Cilj:** Dobiti pristup korisničkom sučelju
- **Sudionici:** baza podataka
- **Preduvjet:** Registracija
- **Opis osnovnog tijeka:**
  1. Djetatnik odabire opciju za prijavu u sustav
  2. Djetatnik upisuje korisničko ime i lozinku
  3. Sustav potvrđuje ispravnost unesenih podataka
  4. Djetatnik dobiva pristup korisničkim funkcijama
- **Opis mogućih odstupanja:**
  - 2.a Unos neispravnog imena i/ili lozinke
    1. Sustav obavljaštava korisnika o neispravnom upisu podataka i neuspjeloj prijavi te ga vraća na stranicu za prijavu

#### UC5 - Pregled svih grupa

- **Glavni sudionik:** Vlasnik sustava (direktor poduzeća)
- **Cilj:** Pregledati sve postojeće grupe u poduzeću
- **Sudionici:** baza podataka
- **Preduvjet:** vlasnik je prijavljen
- **Opis osnovnog tijeka:**
  1. Vlasnik odabire opciju "Pregled grupa"
  2. Vlasniku se prikazuje popis grupa

#### UC6 - Stvaranje grupe

- **Glavni sudionik:** Vlasnik sustava (direktor poduzeća)
- **Cilj:** Stvoriti grupu djetatnika
- **Sudionici:** Baza podataka
- **Preduvjet:** Vlasnik je prijavljen
- **Opis osnovnog tijeka:**
  1. Vlasnik odabire opciju "Pregled grupa"

2. Vlasniku se prikazuje popis grupa
3. Vlasnik odabire opciju "Stvori novu grupu"
4. Vlasniku se prikazuje popis svih djelatnika
5. Vlasnik odabire djelatnike koji će biti raspodijeljeni u grupu
6. Vlasnik odabire opciju "Odaberi voditelja"
7. Vlasniku odabire jednoga od članova grupe s popisa kojega postavlja za voditelja
8. Vlasnik odabire opciju "Potvrđujem odabir"
9. Sustav javlja vlasniku da je grupa uspješno stvorena

### **UC7 - Brisanje grupe**

- **Glavni sudionik:** Vlasnik
- **Cilj:** Obrisati grupu
- **Sudionici:** Baza podataka
- **Preduvjet:** Vlasnik je prijavljen
- **Opis osnovnog tijeka:**
  1. Vlasnik odabire opciju "Pregled grupe"
  2. Vlasniku se prikazuje popis grupa
  3. Vlasnik odabire grupu koju želi obrisati
  4. Vlasnik odabire opciju "Brisanje grupe"
  5. Vlasniku se prikazuje poruka "Jeste li sigurni da želite obrisati grupu?"
  6. Vlasnik odabire opciju "Da"

### **UC8 - Definiranje uslužne djelatnosti**

- **Glavni sudionik:** Vlasnik
- **Cilj:** Definirati uslužnu djelatnost
- **Sudionici:** baza podataka
- **Preduvjet:** Vlasnik je prijavljen
- **Opis osnovnog tijeka:**
  1. Vlasnik odabire opciju "Pregled djelatnosti"
  2. Vlasniku se prikazuje popis djelatnosti
  3. Vlasnik odabire opciju "Stvori djelatnost"
  4. Vlasnik ispuni obrazac za stvaranje djelatnosti
  5. Vlasnik odabire opciju "Potvrdi unos"
- **Opis mogućih odstupanja:**
  - 4.a Neispunjena sva polja obrasca

1. Sustav šalje poruku "Molimo vas ispunite sva polja obrasca."
2. Vlasnik ispunjava obrazac do kraja i ponovno ga potvrđuje

### **UC9 - Dodjela zadataka (voditelji djelatnicima)**

- **Glavni sudionik:** voditelj grupe
- **Cilj:** Dodijeliti djelatnost Razrada plana rada i dodjela zadataka pojedinim djelatnicima
- **Sudionici:** baza podataka
- **Preduvjet:** Korisnik je registriran i dodijeljena mu je uloga "voditelj grupe"
- **Opis osnovnog tijeka:**
  1. Voditelj grupe odabire djelatnika koji će obavljati zadatak.
  2. Voditelj grupe dodjeljuje zadatak odabranom djelatniku.
  3. Upisani podaci se spremaju pritiskom na opciju "Spremi"
  4. Zadaci postaju vidljivi pojedinim djelatnicima.
- **Opis mogućih odstupanja:**
  - 2.a Zauzetost djelatnika
    1. U trenutku dodjele zadatka odabranom djelatniku, ukoliko je već u tom vremenskom periodu istom zadan neki zadatak, sustav šalje poruku upozorenja o preklapanju.
    2. Voditelj grupe zatim zadaje drugi termin djelatniku.

### **UC10 - Upis odrđenih sati**

- **Glavni sudionik:** djelatnici, voditelji grupa
- **Cilj:** Praćenje odrđenih radnih sati u danu za svakog djelatnika.
- **Sudionici:** baza podataka
- **Preduvjet:** Prijava korisnika u sustav.
- **Opis osnovnog tijeka:**
  1. Svaki djelatnik i voditelj grupa nakon radnog dana upiše odrđeno vrijeme (sati i minute) za taj dan.
  2. Uneseni podaci pohranjuju se u bazu klikom na gumb "Spremi".

### **UC11 - Pregled dodijeljenih zadataka**

- **Glavni sudionik: djelatnik**
- **Cilj:** Pojedini djelatnik može vidjeti koji su mu zadaci dodijeljeni.
- **Sudionici:** baza podataka
- **Preduvjet:** Korisnik mora biti prijavljen u sustav i imati ulogu "djelatnika".

- **Opis osnovnog tijeka:**

1. Djelatnik pristupa stranici dodijeljenih zadataka.
2. S obzirom da je djelatnik prijavljen u sustav, poznati su nam njegovi identifikacijski podaci te na temelju njih prikazuje se raspored samo za tog djelatnika.

#### UC12 - Prikaz dodijeljenih grupa

- **Glavni sudionik:** djelatnik

- **Cilj:** Pojedini djelatnik može vidjeti u kojim se grupama nalazi

- **Sudionici:** baza podataka

- **Preduvjet:** Korisnik mora biti prijavljen u sustav i imati ulogu "djelatnika".

- **Opis osnovnog tijeka:**

1. Djelatnik odabire stranicu dodijeljenih grupa.
2. Na temelju njegovih identifikacijskih oznaka prikazuju mu se dodijeljene grupe.

#### UC13 - Pregled zauzetosti djelatnika

- **Glavni sudionik:** djelatnik, voditelj grupe, direktor

- **Cilj:** Korisnik može vidjeti predviđenu zauzetost djelatnika ovisno o ulozi. Direktor ima pristup zauzetosti svih djelatnika, voditelji grupe imaju pristup podacima za sebe i svoju grupu, a djelatnik može vidjeti samo podatke za sebe.

- **Sudionici:** baza podataka

- **Preduvjet:** Korisnik mora biti prijavljen u sustav.

- **Opis osnovnog tijeka:**

1. Korisnik, ako je direktor ili voditelj grupe odabire djelatnika čiju zauzetost želi provjeriti.
2. Odabire se vrijeme za koje se želi provjeriti zauzetost.
3. Na temelju identifikacijskih podataka odabranog djelatnika ili prijavljenog korisnika, ako je njegova uloga "djelatnik" ili "voditelj grupe", prikazuje se zauzetost u prethodno navedenom vremenu.

#### UC14 - Pregled odnosa planiranih i realiziranih troškova

- **Glavni sudionik:** Direktor

- **Cilj:** Evidencija finansijskih rashoda.

- **Sudionici:** baza podataka

- **Preduvjet:** Korisnik mora biti prijavljen u sustav i imati ulogu "direktor".
- **Opis osnovnog tijeka:**
  1. Direktor unosi planirane troškove u bazu.
  2. Tijekom obavljanja poslova, redovito se upisuju realizirani troškovi.
  3. Sustav na temelju unesenih podataka računa razliku.
  4. Prikazuje se odnos planiranih I realiziranih troškova.

#### **UC15 - Pregled odnosa planiranih I realiziranih dobiti**

- **Glavni sudsionik:** direktor
- **Cilj:** Evidencija finansijskih prihoda.
- **Sudsionici:** baza podataka
- **Preduvjet:** Korisnik prijavljen u sustav i ima ulogu "direktor"
- **Opis osnovnog tijeka:**
  1. Direktor unosi planiranu dobit u bazu.
  2. Tijekom obavljanja poslova, redovito se upisuju realizirane dobiti.
  3. Sustav na temelju unesenih podataka računa razliku.
  4. Prikazuje se odnos planiranih I realiziranih dobiti.

#### **UC16 - Pregled podređene grupe (voditelj)**

- **Glavni sudsionik:** Voditelj grupe
- **Cilj:** Voditelj treba moći vidjeti članove grupe kojoj je on nadređen. Može vidjeti članove grupe, njihove podatke i dodijeljene zadatke.
- **Sudsionici:** baza podataka
- **Preduvjet:** korisnik je prijavljen u sustav i ima ulogu "voditelj"
- **Opis osnovnog tijeka:**
  1. Voditelj odabire opciju "Moje podređene grupe".
  2. Prikazuje se popis svih grupa kojima je taj voditelj nadređen.
  3. Voditelj odabire grupu za koju želi vidjeti podatke.
  4. Prikazuju se članovi grupe i djelatnost koja joj je dodijeljena. Odabirom pojedinog djelatnika prikazuju se njegovi podaci i dodijeljeni zadaci.

#### **UC17 - Pregled vlastitih podataka**

- **Glavni sudsionik:** direktor, voditelj grupe, djelatnik
- **Cilj:** Svi djelatnici (uključujući i direktora i voditelje grupe) mogu vidjeti svoje osobne podatke, dodijeljene zadatke i grupe u kojima se nalazi.
- **Sudsionici:** baza podataka

- **Preduvjet:** Korisnik je prijavljen u sustav
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju "Moji podaci".
  2. Prikazuju se korisnikovi osobni podaci, njegovi dodijeljeni zadaci te grupe u kojima se nalazi.

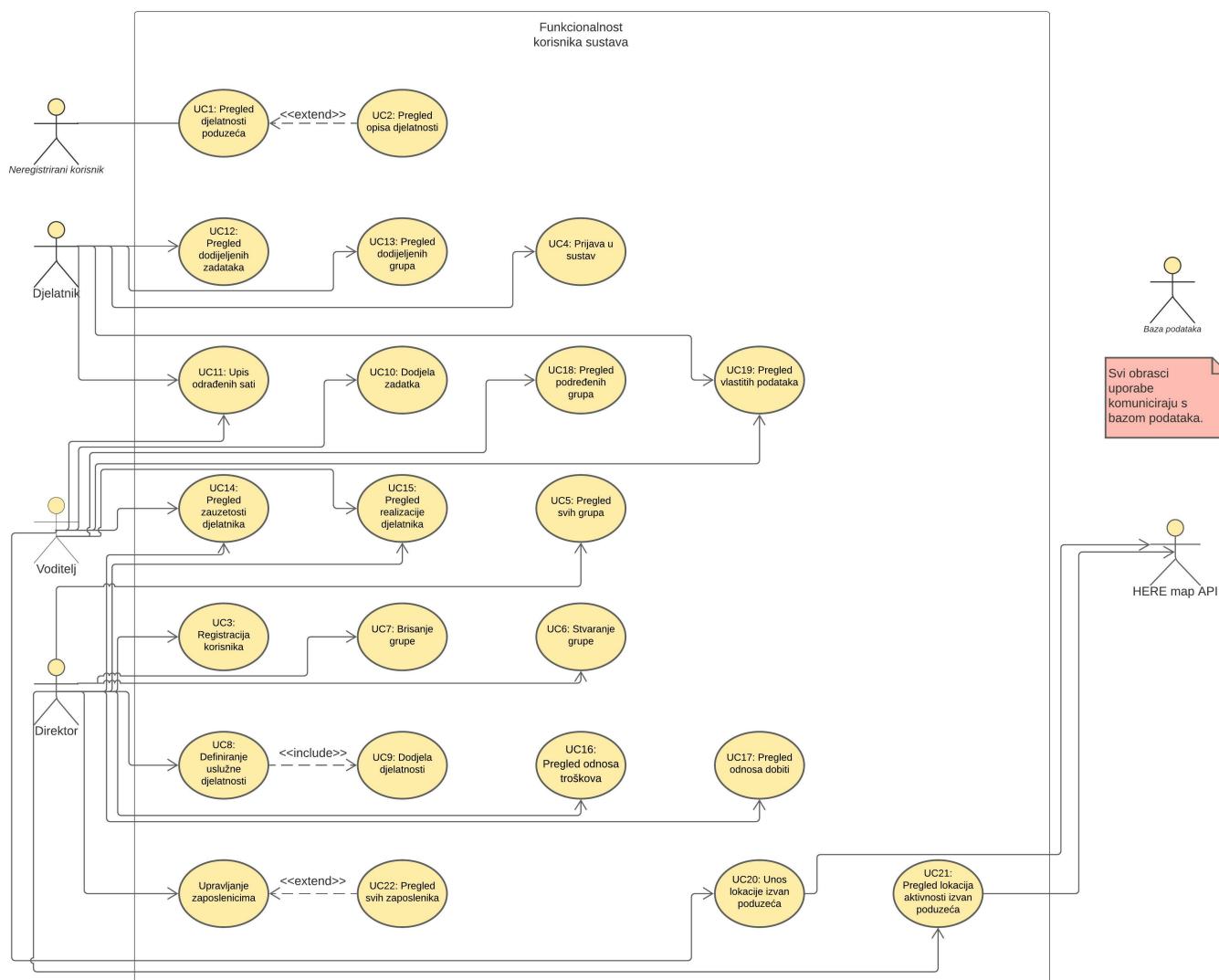
### UC18 - Unos lokacije aktivnosti izvan poduzeća

- **Glavni sudionik:** voditelj grupe
- **Cilj:** Voditelj treba stvoriti zadatak koji se obavlja izvan poduzeća
- **Sudionici:** baza podataka
- **Preduvjet:** Voditelj je prijavljen u sustav
- **Opis osnovnog tijeka:**
  1. Voditelj grupe odabire djelatnika koji će obavljati zadatak.
  2. Voditelj grupe dodjeljuje zadatak izvan poduzeca odabranom djelatniku.
  3. Upisani podaci se spremaju pritiskom na opciju "Spremi"
  4. Zadaci postaju vidljivi pojedinim djelatnicima.
- **Opis mogućih odstupanja:**
  - 2.a Zauzetost djelatnika
    1. U trenutku dodjele zadatka odabranom djelatniku, ukoliko je već u tom vremenskom periodu istom zadan neki zadatak, sustav šalje poruku upozorenja o preklapanju.
    2. Voditelj grupe zatim zadaje drugi termin djelatniku.

### UC19 - Pregled lokacija aktivnosti izvan poduzeća

- **Glavni sudionik:** direktor
- **Cilj:** Moguće je da djelatnici imaju aktivnosti koje moraju obaviti na nekoj lokaciji izvan poduzeća. U tom slučaju direktor mora moći vidjeti gdje su se nalazili.
- **Sudionici:** baza podataka
- **Preduvjet:** korisnik je prijavljen u sustav i ima ulogu "direktor"
- **Opis osnovnog tijeka:**
  1. Direktor odabire opciju "Pregledajte aktivnosti izvan poduzeća".
  2. Prikazuje se karta s označenim lokacijama aktivnosti.
  3. Odabirom pojedine lokacije prikazuju se podaci o aktivnosti koja se obavljala na toj lokaciji I djelatniku koji je tu aktivnost obavljao.

## Dijagrami obrazaca uporabe

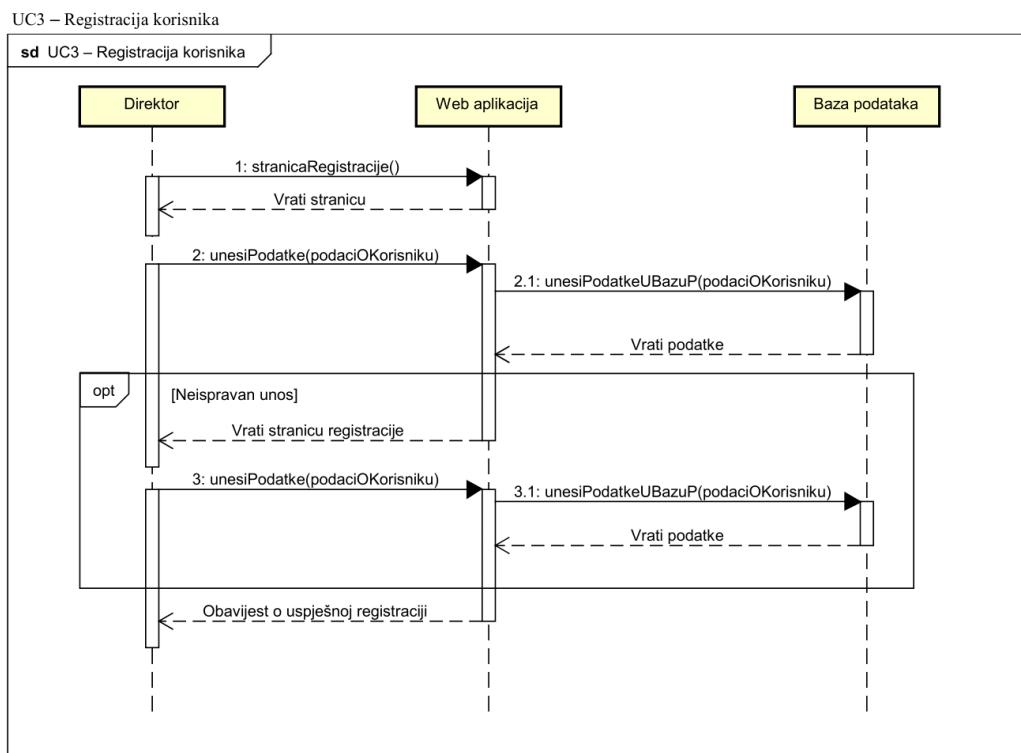


Slika 3.1 Funkcionalnosti korisnika sustava

### 3.1.2 Sekvencijski dijagrami

#### UC3 – Registracija korisnika

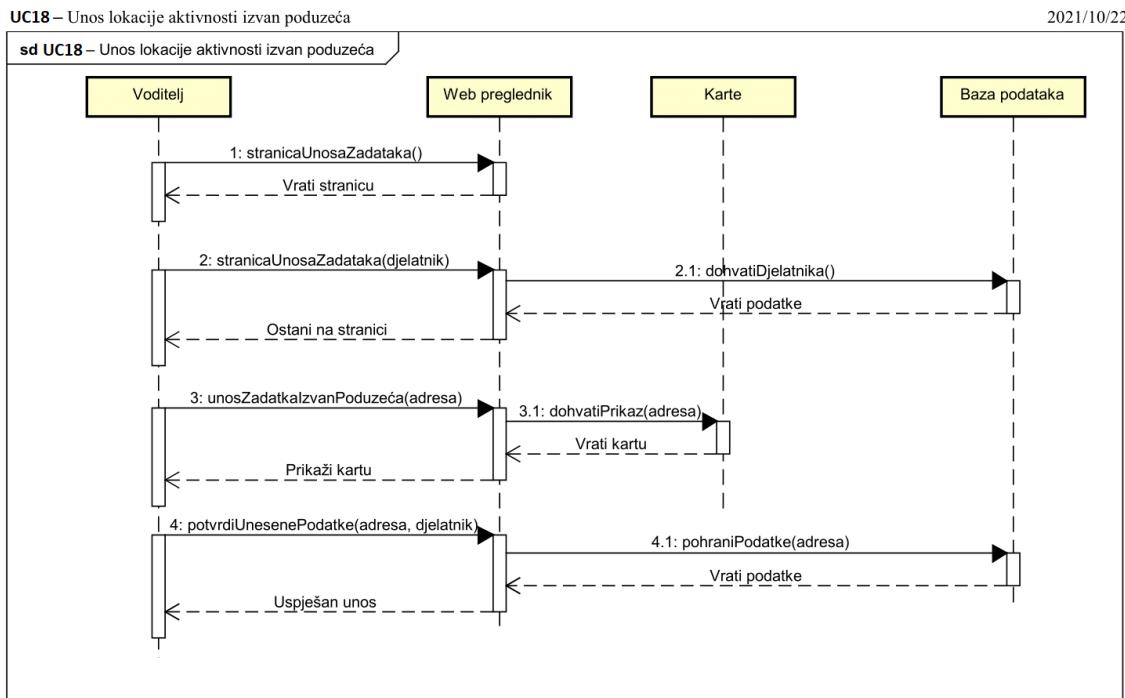
Direktor otvara stranicu za registraciju djelatnika. Obavljanje zahtjeva obavlja se na poslužitelju koji kao odgovor vraća zatraženu stranicu. Direktor unosi sve potrebne korisničke podatke za stvaranje potpuno novog profila djelatnika. Prilikom registracije direktor također dodjeljuje ulogu djelatniku. Tijekom unosa moguće su sljedeća odstupanja: već zauzeto korisničko ime i/ili elektronička pošta, unos podataka u neispravnom formatu (ilegalni znakovi, predugačak unos ili unos krivog tipa podataka). Ukoliko dođe do pogreške prilikom unosa, poslužitelj obavještava direktora o neuspjelom zahtjevu i vraća ga na stranicu za registraciju. U takvom slučaju direktor ispravlja unos te ponovno pokušava izvršiti unos ili odustaje od registracije. Svi uneseni podaci šalju se na poslužitelj te se spremaju u bazu podataka. Direktor prima obavijest o uspješnoj registraciji.



Slika 3.2 Registracija korisnika

### UC18 – Unos lokacije aktivnosti izvan poduzeća

Djelatnik šalje poslužitelju zahtjev za unos aktivnosti izvan poduzeća. Na web pregledniku otvara se obrazac za upis adrese na kojoj je djelatnik obavljao aktivnost. Podaci se šalju pregledniku te se dohvaća karta s prikazom navedene adrese. Karta se prikazuje djelatniku. Djelatnik potvrđuje lokaciju pritiskom na gumb “Potvrди” te se adresa upisuje u bazu podataka.



Slika 3.5 Unos lokacije aktivnosti izvan poduzeća

### 3.2 Ostali zahtjevi

Sustav treba:

- omogućiti korištenje hrvatskih dijakritičkih znakova pri unosu i prikazu tekstuallnog sadržaja
- podržati višekorisnički rad u realnom vremenu
- dati odgovor na traženi upit unutar nekoliko sekundi kada se dohvaćaju podaci iz baze podataka
- imati intuitivno i jednostavno za korištenje korisničko sučelje
- biti implementiran kao web aplikacija koristeći objektno-orientirane jezike
- neispravno korištenje korisničkog sučelja ne smije narušiti rad sustava
- osigurati sigurnu, brzu i otpornu na vanjske greške vezu s bazom podataka

## 4. Arhitektura i dizajn sustava

Arhitektura se može podijeliti na tri podsustava:

- Web poslužitelj
- Web aplikacija
- Baza podataka

Web preglednik je program koji korisniku omogućuje pregled web stranica i multimedijalnih sadržaja vezanih uz njih. Svaki internetski preglednik je prevoditelj. Dakle, stranica je pisana u kodu koji preglednik nakon toga interpretira kako nešto svakome razumljivo. Korisnik putem web preglednika šalje zahtjev web poslužitelju.

Web poslužitelj osnova je rada web aplikacije. Njegova primarna zadaća je komunikacija klijenta s aplikacijom. Komunikacija se odvija preko HTTP (engl. Hyper Text Transfer Protocol) protokola, što je protokol u prijenosu informacija na webu. Poslužitelj je onaj koji pokreće web aplikaciju te joj prosljeđuje zahtjev.

Korisnik koristi web aplikaciju za obrađivanje željenih zahtjeva. Web aplikacija obrađuje zahtjev te ovisno o zahtjevu, pristupa bazi podataka nakon čega preko poslužitelja vraća korisniku odgovor u obliku HTML dokumenta vidljivog u web pregledniku. HTML elementi su stvoreni koristeći JavaScript biblioteku React. Odabранo razvojno okruženje je Visual Studio Code.

Programski jezik kojeg smo odabrali za izradu naše web aplikacije je Java Spring Boot. Odabrana razvojna okruženja su Eclipse IDE i IntelliJ IDEA. Arhitektura sustava temeljiti će se na MVC (Model-View-Controller) konceptu.

Karakteristika MVC koncepta je nezavisan razvoj pojedinih dijelova aplikacije što za posljedicu ima jednostavnije ispitivanje kao i jednostavnije razvijanje i dodavanje novih svojstava u sustav.

MVC koncept sastoji se od:

- Model - Središnja komponenta sustava. Predstavlja dinamičke strukture podataka, neovisne o korisničkom sučelju. Izravno upravlja podacima, logikom i pravilima aplikacije. Također prima ulazne podatke od Controllera

- View - Bilo kakav prikaz podataka, poput grafa. Mogući su različiti prikazi iste informacije poput grafičkog ili tabličnog prikaza podatak.
- Controller – Prima ulaze i prilagođava ih za proslijedivanje Model-u ili View-u. Upravlja korisničkim zahtjevima i na temelju njih izvodi daljnju interakciju s ostalim elementima sustava.

## 4.1 Baza podataka

*dio 1. revizije*

Baza podataka je predstavljena pomoću ER te relacijskog dijagrama. Od glavnih komponenti, primarno se pojavljuju:

- Employee
- Role
- Group
- EmployeeGroup
- Job
- Task
- EmployeeTask
- Location
- WorkHoursInput

### 4.1.1 Opis tablica

#### Employee

Sadrži podatke o djelatniku, uključujući osobne podatke i podatke za prijavu (korisničko ime i lozinka).

Employee		
pid	CHAR	Osobni identifikacijski broj djelatnika.

<b>Employee</b>		
name	VARCHAR	Ime djelatnika.
surname	VARCHAR	Prezime djelatnika.
email	VARCHAR	E-mail adresa.
username	VARCHAR	Jedinstveno korisničko ime djelatnika.
password	VARCHAR	Lozinka djelatnika.
idRole	INTEGER	Identifikator uloge djelatnika.

## Role

Sadrži podatke o ulozi korisnika. Uloga može biti: direktor, vođa tima ili djelatnik.

<b>Role</b>		
idRole	INTEGER	Identifikator uloge.
name	VARCHAR	Naziv uloge (direktor, vođa tima, djelatnik).

## Group

Djelatnici rade u grupama. Grupa ima svog voditelja - djelatnika kojem je direktor dodijelio ulogu voditelja. Grupa može biti stvorena bez članova, ali ne može biti bez voditelja. Svaka se grupa bavi nekom djelatnosti.

<b>Group</b>		
idGroup	INTEGER	Identifikator grupe.
name	VARCHAR	Naziv grupe.
idLeader	INTEGER	Identifikator voditelja grupe.
idJob	INTEGER	Identifikator djelatnosti kojom se grupa bavi.

## EmployeeGroup

Tablica koja ostvaruje vezu više-na-više između tablica Employee i Group, tj. asocira djelatnike s grupama čiji su članovi. Djelatnik može istovremeno raditi u više grupa.

<b>EmployeeGroup</b>		
idEmployee	CHAR	Identifikator djelatnika.
idGroup	INTEGER	Identifikator grupe.

## Job

Sadrži podatke o djelatnosti kojima se tvrtka bavi. Svakoj djelatnosti pridružen je naziv i tekstualni opis.

Job		
idJob	INTEGER	Identifikator djelatnosti.
name	VARCHAR	Naziv djelatnosti.
description	TEXT	Opis djelatnosti.

## Task

Zadaci na kojima djelatnici rade, a zadaje ih voditelj grupe za djelatnike kojima je nadležan. Svaki zadatak ima naziv i opis, datum i vrijeme planiranog početka i završetka, te procjenu broja radnih sati koji će biti potrebni da bi se zadatak dovršio. Voditelj grupe može i pridružiti zadatak nekoj djelatnosti. Zadatak se može obavljati na nekoj lokaciji. Pri stvaranju zadatka, voditelj određuje planirani trošak i prihod. Kada djelatnik dovrši zadatak, stvarni trošak i prihod se upisuju u tablicu.

Task		
idTask	INTEGER	Identifikator zadatka.
name	VARCHAR	Naziv zadatka.
description	TEXT	Tekstualni opis zadatka.
dateTimeStart	TIMESTAMP	Datum i vrijeme početka zadatka. Mora biti manje vrijednosti od polja datumVrijemeZavrsetka.
dateTimeEnd	TIMESTAMP	Datum i vrijeme završetka zadatka. Mora biti veće vrijednosti od polja datumVrijemePocetka.
hoursNeededEstimate	SMALLINT	Procjena broja radnih sati potrebnih za dovršetak zadatka. Procjenu određuje voditelj tima koji je zadao zadatak. Procjena može, ali ne mora odgovarati razlici između dateTimeEnd i dateTimeStart.
plannedCost	FLOAT	Planirani trošak.

Task		
realizedCost	FLOAT	Realizirani trošak.
plannedProfit	FLOAT	Planirana dobit.
realizedProfit	FLOAT	Realizirana dobit.
idJob	INTEGER	Identifikator djelatnosti kojoj zadatak pripada.
idLocation	INTEGER	Identifikator lokacije na kojoj se zadatak obavlja.

### EmployeeTask

Tablica koja ostvaruje vezu više-na-više između tablica Employee i Task, tj. određuje koji djelatnik radi na kojim zadacima. Zapisuje se i realizacija zadatka koja predstavlja postotak dovršenog posla. Na početku iznosi 0 i povećava se kako djelatnik napreduje s poslom.

EmployeeTask		
idEmployee	CHAR	Identifikator djelatnika koji radi na zadatku.
idTask	INTEGER	Identifikator zadatka na kojem djelatnik radi.
realized	SMALLINT	Postotak posla kojeg je djelatnik obavio na zadatku u odnosu na ukupan posao. Može biti u intervalu [0, 100]. Početna vrijednost je 0.

### Location

Lokacije na kojima se obavljaju poslovi. Za svaku lokaciju zapisuje se adresa, mjesto i geografska širina i dužina radi prikaza na karti.

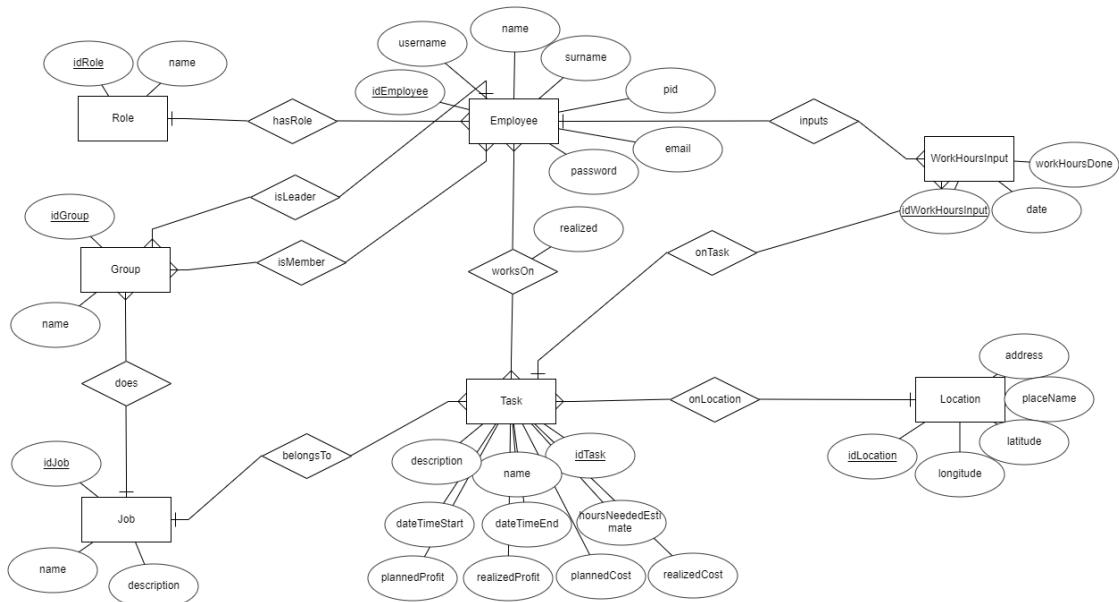
Location		
idLocation	INTEGER	Identifikator lokacije.
address	VARCHAR	Adresa na kojoj se lokacija nalazi.
placeName	VARCHAR	Mjesto u kojem se lokacija nalazi.
latitude	DOUBLE	Geografska širina lokacije.
longitude	DOUBLE	Geografska dužina lokacije.

## WorkHoursInput

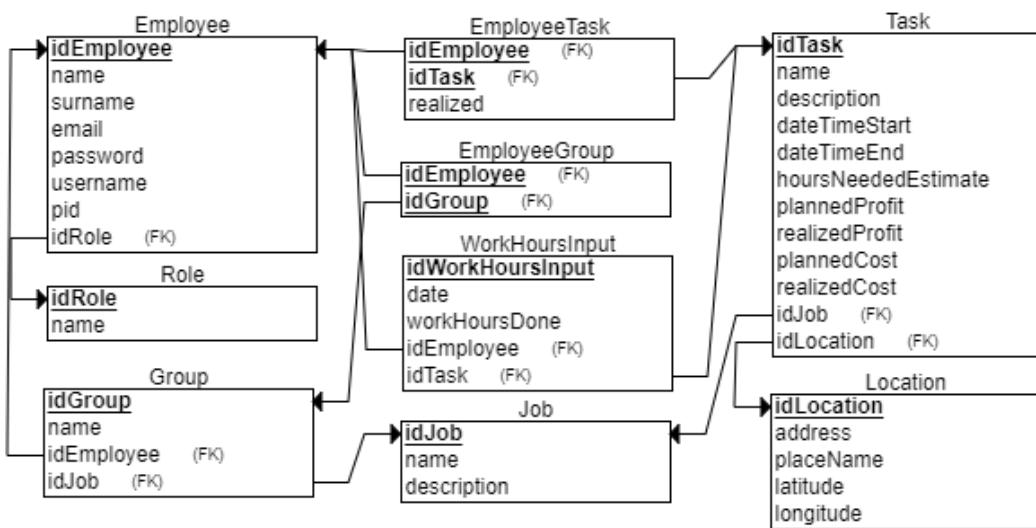
Tablica koja predstavlja evidenciju radnih sati djelatnika. Svaki djelatnik upisuje koliko je sati odradio na kojem zadatku svaki radni dan.

WorkHoursInput		
idWorkHoursInput	INTEGER	Identifikator unosa.
date	DATE	Datum unosa.
workHoursDone	SMALLINT	Broj radnih sati koje je djelatnik odradio na zadatku tog dana.
idEmployee	CHAR	Identifikator djelatnika koji obavlja unos.
idTask	INTEGER	Identifikator zadatka na kojeg se unos odnosi.

### 4.1.2 Dijagram baze podataka



Slika 4.1: ER dijagram baze podataka



Slika 4.2: Relacijski dijagram baze podataka

## 4.2 Dijagram razreda

Na slikama su prikazani razredi koji pripadaju backend dijelu aplikacije. Dijagramme sačinjavaju razredi te među njima uočavamo dijelove poslužiteljskog sloja, odnosno entitete, objekte za prijenos podataka (engl. Data Transfer Object), rezervorije, servise i kontrolere.

Entiteti su klase koje predstavljaju tablice spremljene u bazi podataka. Svaka instanca entiteta predstavlja jedan redak u toj tablici.

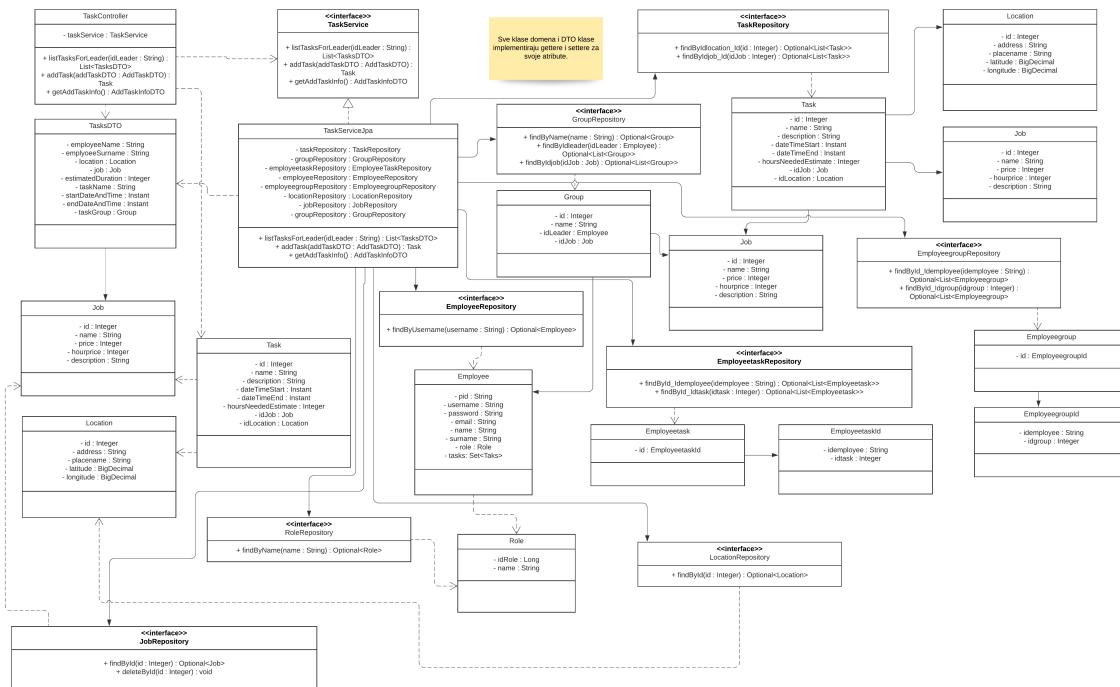
Objekti za prijenos podataka su objekti koji se koriste za komunikaciju između poslužiteljskog i klijentskog dijela aplikacije. Korisni su jer pomoću njih prenosimo samo ono što je bitno, isključivo nužne podatke o određenim entitetima, no ne i cijele entitete.

Rezervoriji su Java sučelja koje upravljaju tokom podataka između poslužitelja i baze podataka. Klase rezervorija služe kao mehanizam za enkapsulaciju pothrane, dohvatanja i pretraživanja podataka.

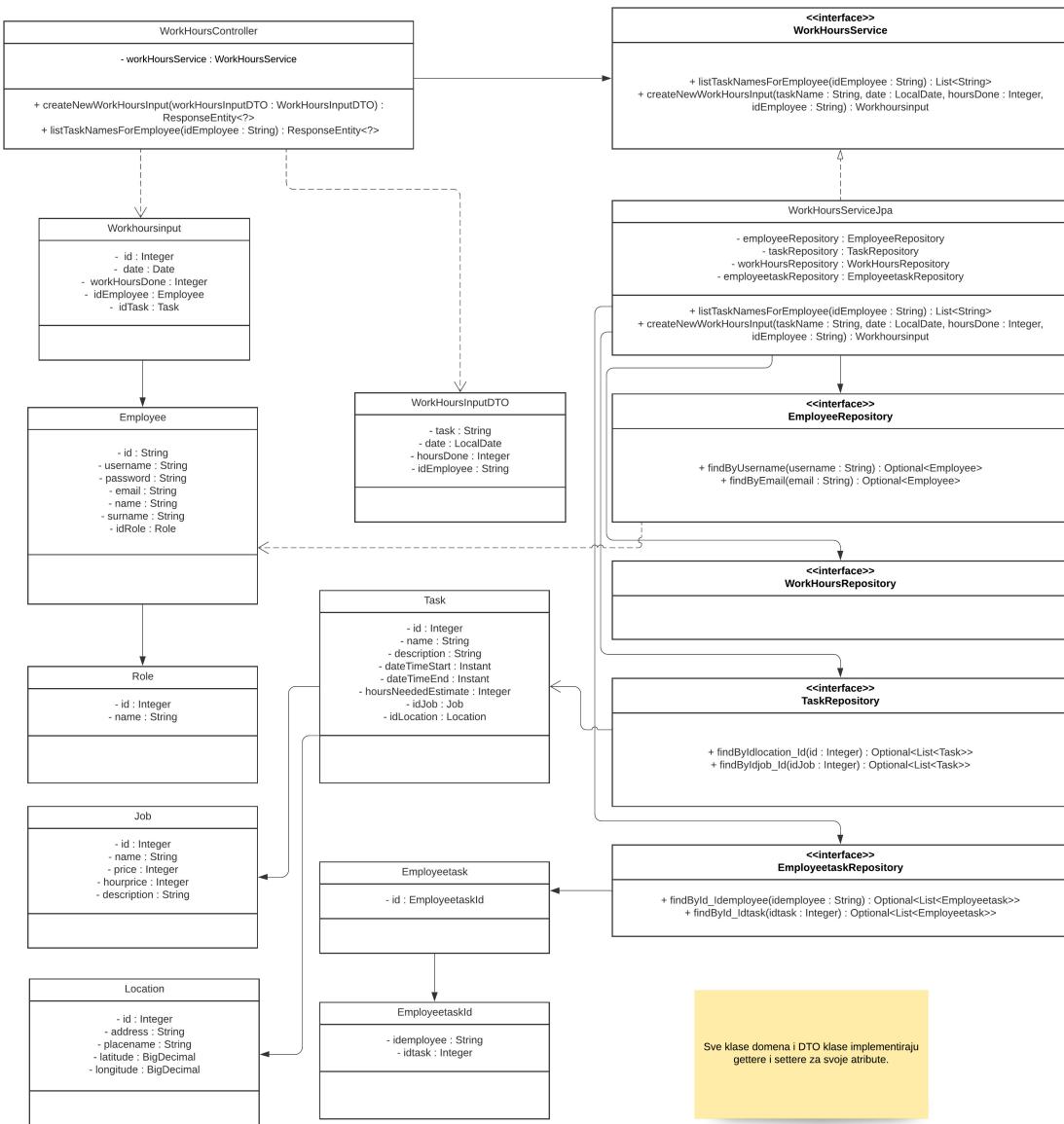
Servisi su klase koje sadrže poslovnu logiku aplikacije. U njima se nalaze metode čija je zadaća izvršavanje operacija nad bazom podataka, odnosno pozivanje metoda definiranih u rezervorijima.

Kontroleri, odnosno REST (engl. Representational State Transfer) kontroleri presreću dolazeće zahtjeve, pretvaraju podatke iz zahtjeva u unutarnju strukturu podataka, šalju podatke modelu na daljnju obradu te dobivaju određene podatke iz modela i prosljeđuju ih u klijentski dio aplikacije kako bi se prikazali korisniku.

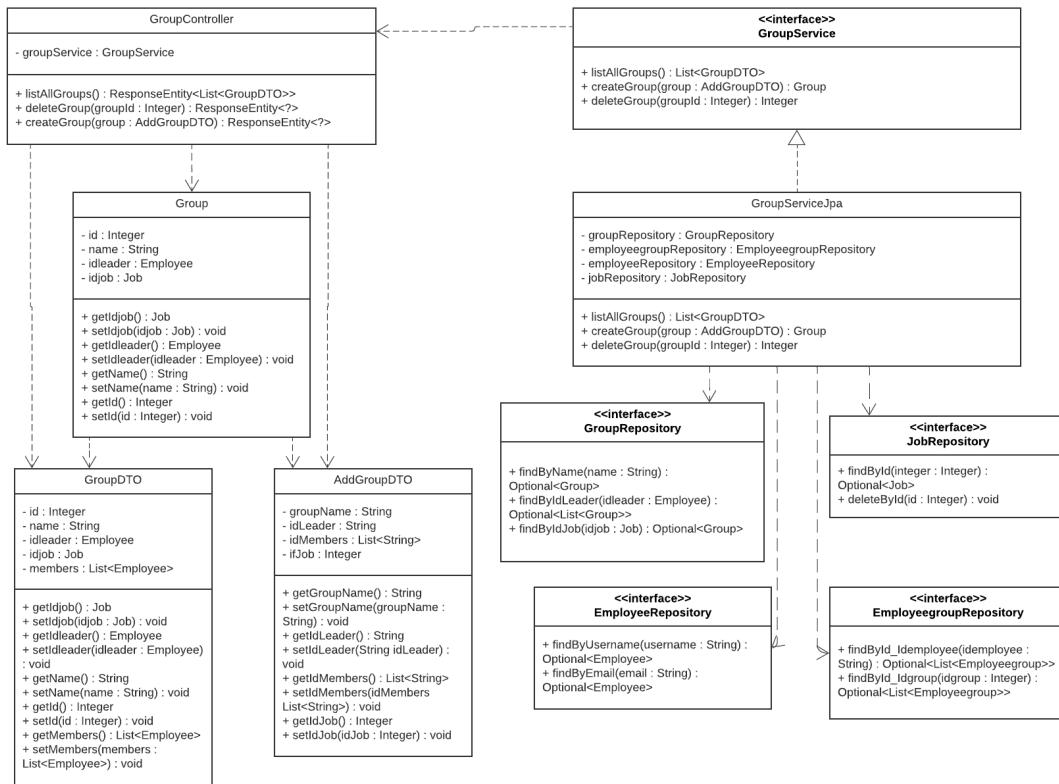
Iz naziva i tipova atributa u razredima može se zaključiti vrsta ovisnosti među različitim razredima prikazanima na priloženim dijagramima.



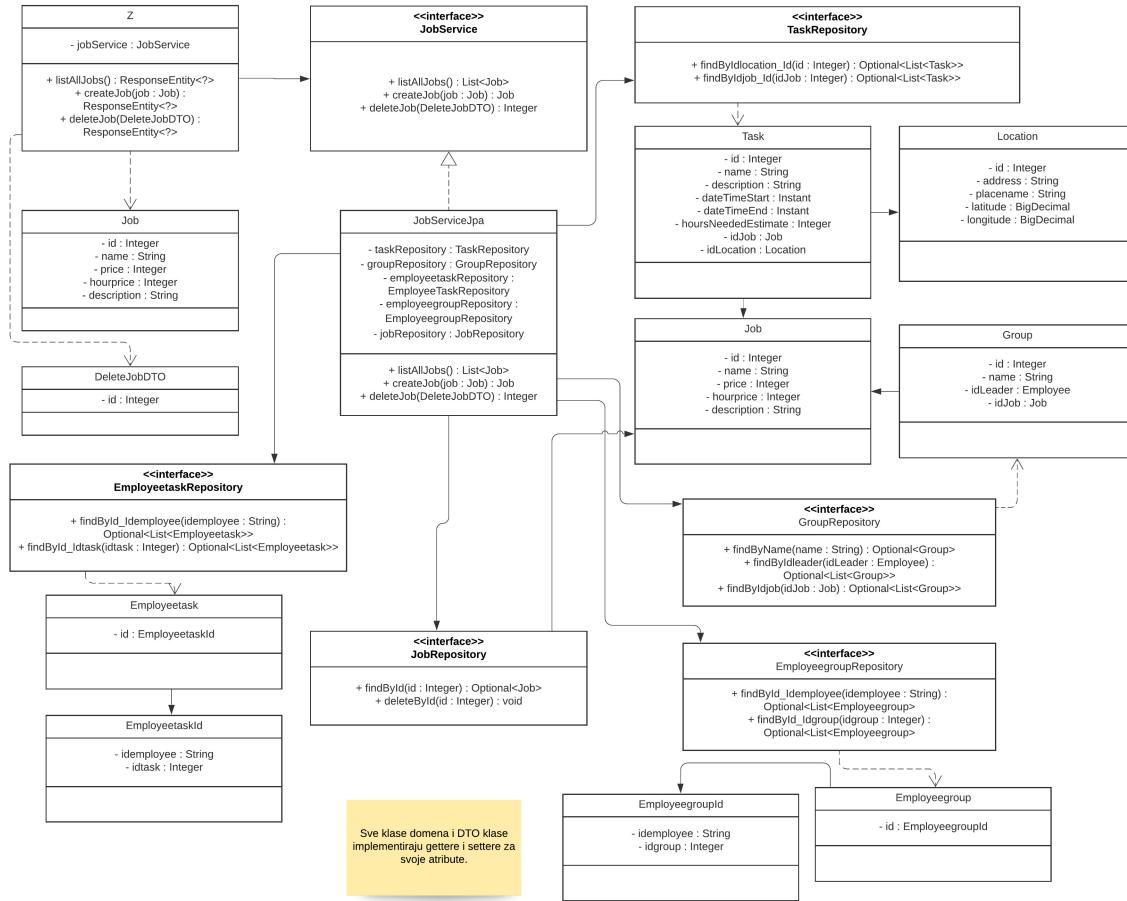
Slika 4.3: Dijagram razreda TaskController



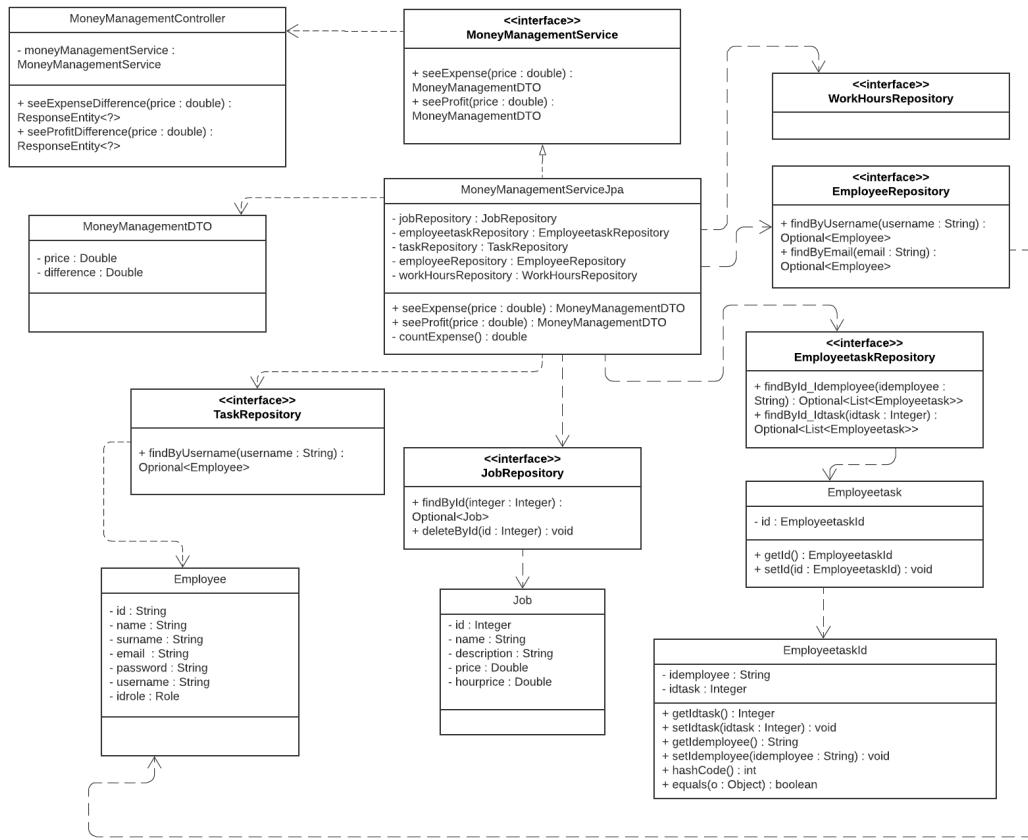
Slika 4.4: Dijagram razreda WorkHoursController



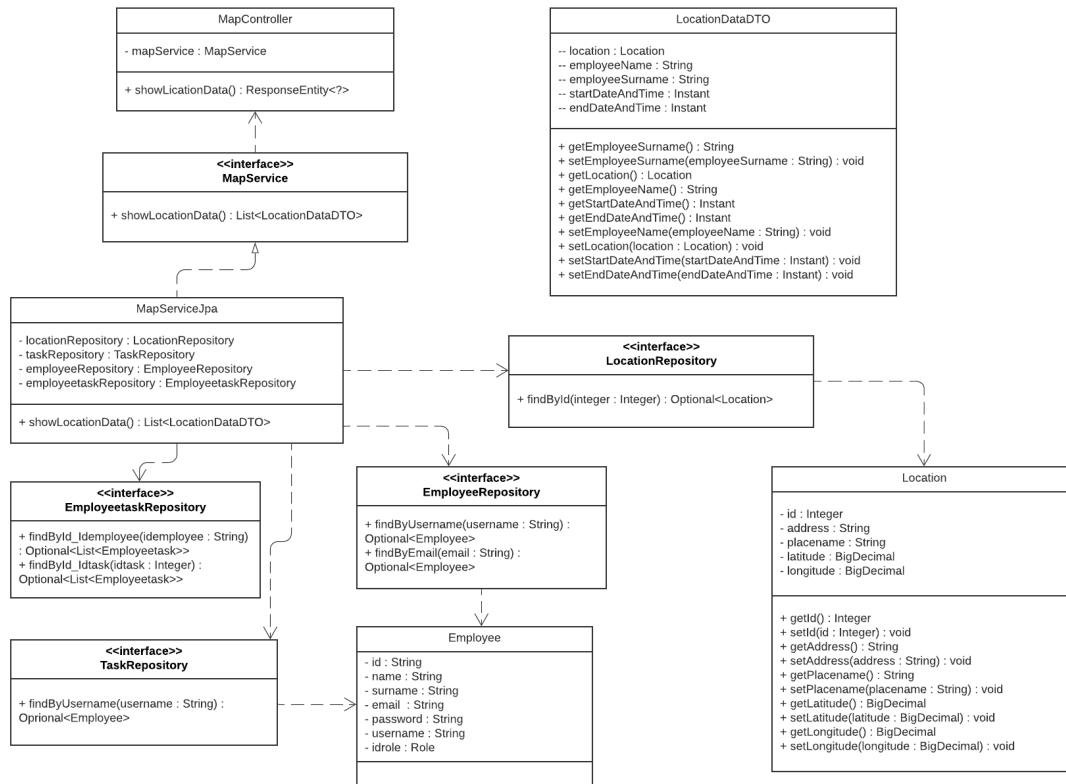
Slika 4.5: Dijagram razreda GroupController



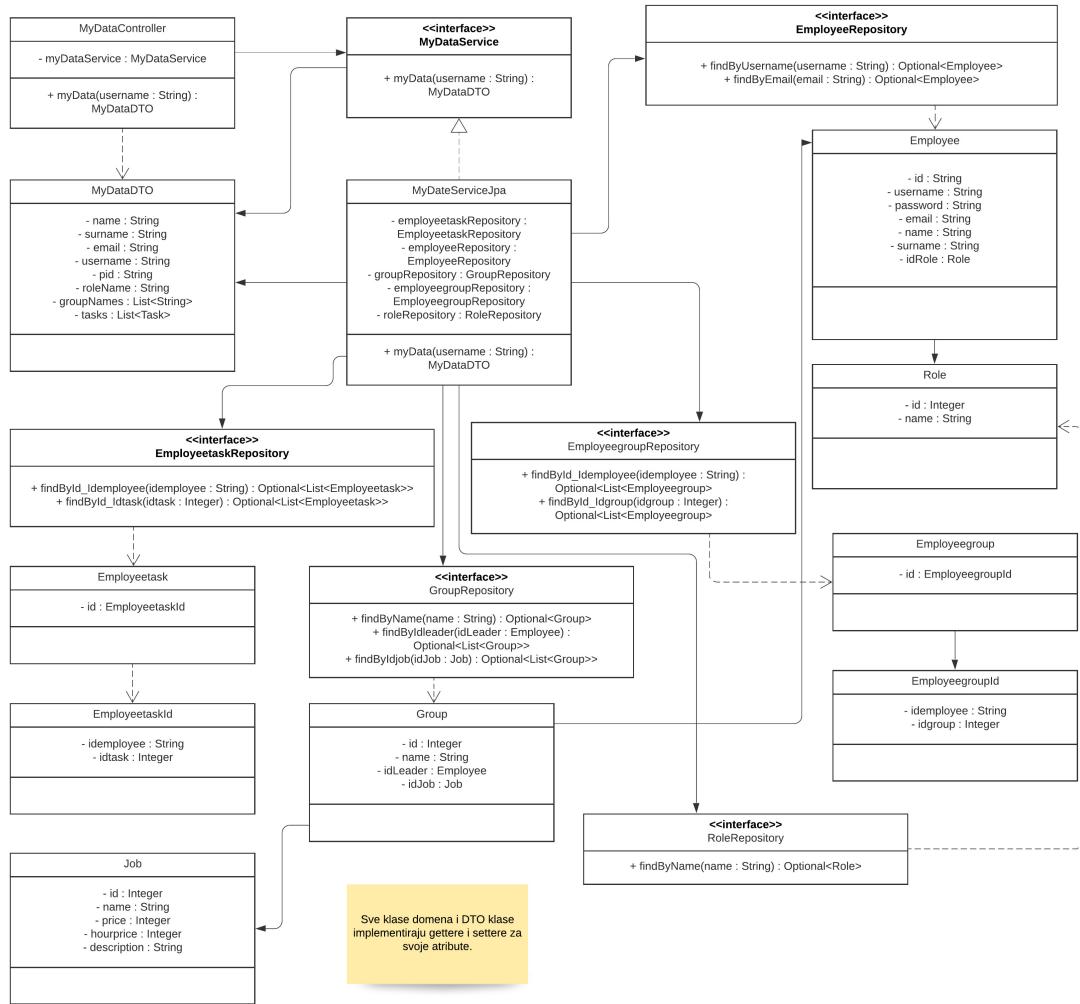
Slika 4.6: Dijagram razreda JobController



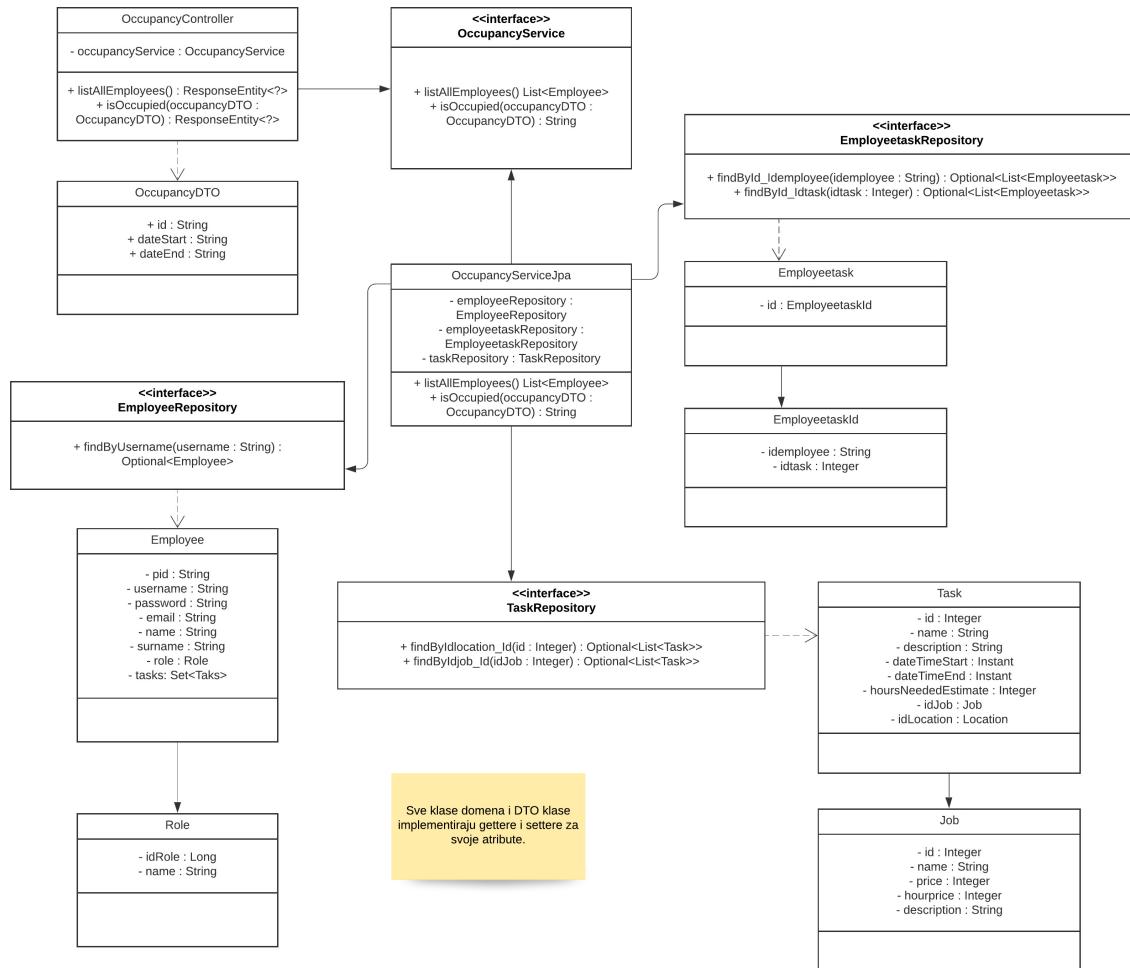
Slika 4.7: Dijagram razreda MoneyManagementController



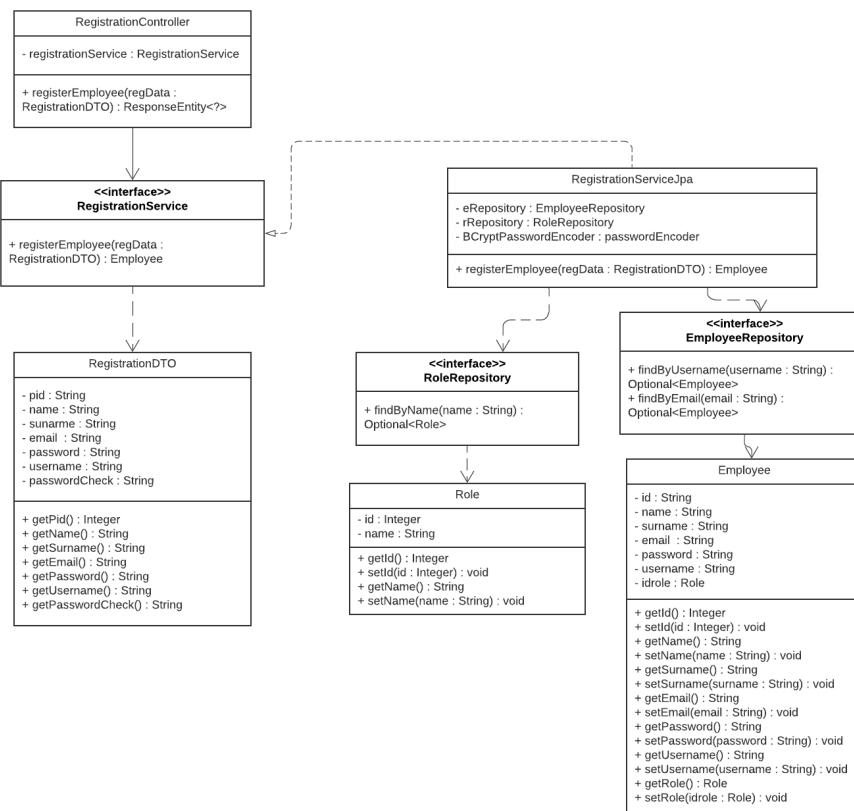
Slika 4.8: Dijagram razreda MapController



Slika 4.9: Dijagram razreda MyDataController



Slika 4.10: Dijagram razreda OccupancyController



Slika 4.11: Dijagram razreda RegistrationController

## 4.3 Dijagram stanja

*dio 2. revizije*

Potrebno je priložiti dijagram stanja i opisati ga. Dovoljan je jedan dijagram stanja koji prikazuje **značajan dio funkcionalnosti** sustava. Na primjer, stanja korisničkog sučelja i tijek korištenja neke ključne funkcionalnosti jesu značajan dio sustava, a registracija i prijava nisu.

## 4.4 Dijagram aktivnosti

*dio 2. revizije*

*Potrebno je priložiti dijagram aktivnosti s pripadajućim opisom. Dijagram aktivnosti treba prikazivati značajan dio sustava.*

## 4.5 Dijagram komponenti

*dio 2. revizije*

*Potrebno je priložiti dijagram komponenti s pripadajućim opisom. Dijagram komponenti treba prikazivati strukturu cijele aplikacije.*

## 5. Implementacija i korisničko sučelje

### 5.1 Korištene tehnologije i alati

#### *dio 2. revizije*

*Detaljno navesti sve tehnologije i alate koji su primijenjeni pri izradi dokumentacije i aplikacije. Ukratko ih opisati, te navesti njihovo značenje i mjesto primjene. Za svaki navedeni alat i tehnologiju je potrebno navesti internet poveznicu gdje se mogu preuzeti ili više saznati o njima.*

## 5.2 Ispitivanje programskog rješenja

### dio 2. revizije

U ovom poglavlju je potrebno opisati provedbu ispitivanja implementiranih funkcionalnosti na razini komponenti i na razini cijelog sustava s prikazom odabranih ispitnih slučajeva. Studenti trebaju ispitati temeljnu funkcionalnost i rubne uvjete.

#### 5.2.1 Ispitivanje komponenti

Potrebno je provesti ispitivanje jedinica (engl. unit testing) nad razredima koji implementiraju temeljne funkcionalnosti. Razraditi **minimalno 6 ispitnih slučajeva** u kojima će se ispitati redovni slučajevi, rubni uvjeti te izazivanje pogreške (engl. exception throwing). Poželjno je stvoriti i ispitni slučaj koji koristi funkcionalnosti koje nisu implementirane. Potrebno je priložiti izvorni kôd svih ispitnih slučajeva te prikaz rezultata izvođenja ispita u razvojnem okruženju (prolaz/pad ispita).

#### 5.2.2 Ispitivanje sustava

Potrebno je provesti i opisati ispitivanje sustava koristeći radni okvir Selenium<sup>1</sup>. Razraditi **minimalno 4 ispitna slučaja** u kojima će se ispitati redovni slučajevi, rubni uvjeti te poziv funkcionalnosti koja nije implementirana/izaziva pogrešku kako bi se vidjelo na koji način sustav reagira kada nešto nije u potpunosti ostvareno. Ispitni slučaj se treba sastojati od ulaza (npr. korisničko ime i lozinka), očekivanog izlaza ili rezultata, koraka ispitivanja i dobivenog izlaza ili rezultata.

Izradu ispitnih slučajeva pomoću radnog okvira Selenium moguće je provesti pomoću jednog od sljedeća dva alata:

- dodatak za preglednik **Selenium IDE** - snimanje korisnikovih akcija radi automatskog ponavljanja ispita
- **Selenium WebDriver** - podrška za pisanje ispita u jezicima Java, C#, PHP koristeći posebno programsко sučelje.

Detalji o korištenju alata Selenium bit će prikazani na posebnom predavanju tijekom semestra.

---

<sup>1</sup><https://www.seleniumhq.org/>

## 5.3 Dijagram razmještaja

### *dio 2. revizije*

Potrebno je umetnuti **specifikacijski dijagram razmještaja** i opisati ga. Moguće je umjesto specifikacijskog dijagrama razmještaja umetnuti dijagram razmještaja instanci, pod uvjetom da taj dijagram bolje opisuje neki važniji dio sustava.

## 5.4 Upute za puštanje u pogon

*dio 2. revizije*

*U ovom poglavlju potrebno je dati upute za puštanje u pogon (engl. deployment) ostvarene aplikacije. Na primjer, za web aplikacije, opisati postupak kojim se od izvornog kôda dolazi do potpuno postavljene baze podataka i poslužitelja koji odgovara na upite korisnika. Za mobilnu aplikaciju, postupak kojim se aplikacija izgradi, te postavi na neku od trgovina. Za stolnu (engl. desktop) aplikaciju, postupak kojim se aplikacija instalira na računalo. Ukoliko mobilne i stolne aplikacije komuniciraju s poslužiteljem i/ili bazom podataka, opisati i postupak njihovog postavljanja. Pri izradi uputa preporučuje se naglasiti korake instalacije uporabom natuknica te koristiti što je više moguće slike ekrana (engl. screenshots) kako bi upute bile jasne i jednostavne za slijediti.*

*Dovršenu aplikaciju potrebno je pokrenuti na javno dostupnom poslužitelju. Studentima se preporuča korištenje neke od sljedećih besplatnih usluga: Amazon AWS, Microsoft Azure ili Heroku. Mobilne aplikacije trebaju biti objavljene na F-Droid, Google Play ili Amazon App trgovini.*

## 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

## **Kontinuirano osvježavanje**

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

1. Programsко 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

4.1	ER dijagram baze podataka . . . . .	26
4.2	Relacijski dijagram baze podataka . . . . .	27
4.3	Dijagram razreda TaskController . . . . .	29
4.4	Dijagram razreda WorkHoursController . . . . .	30
4.5	Dijagram razreda GroupController . . . . .	31
4.6	Dijagram razreda JobController . . . . .	32
4.7	Dijagram razreda MoneyManagementController . . . . .	33
4.8	Dijagram razreda MapController . . . . .	34
4.9	Dijagram razreda MyDataController . . . . .	35
4.10	Dijagram razreda OccupancyController . . . . .	36
4.11	Dijagram razreda RegistrationController . . . . .	37

# Dodatak: Prikaz aktivnosti grupe

## Dnevnik sastajanja

### 1. sastanak

- Datum: 13. listopada 2021.
- Prisustvovali: B.Kazazić, M.Erlić, B.Pavlović, A.Pašalić, L.Raspudić, P.Sušac, V.Žunar
- Teme sastanka:
  - 1. sastanak s asistentom i demonstratorom
  - raščišćavanje nejasnoća

### 2. sastanak

- Datum: 14. listopada 2021.
- Prisustvovali: B.Kazazić, M.Erlić, B.Pavlović, A.Pašalić, L.Raspudić, V.Žunar
- Teme sastanka:
  - ažuriranje LaTeX dokumentacije
  - dodjela zadatka članovima tima
  - platforme za komunikaciju

### 3. sastanak

- Datum: 18. listopada 2021.
- Prisustvovali: B.Kazazić, M.Erlić, B.Pavlović, L.Raspudić, P.Sušac, V.Žunar
- Teme sastanka:
  - raspodjela zadataka

### 4. sastanak

- Datum: 22. listopada 2021.
- Prisustvovali: B.Kazazić, M.Erlić, B.Pavlović, A.Pašalić, P.Sušac, V.Žunar
- Teme sastanka:
  - dodjela zadatka članovima tima

## Tablica aktivnosti

### *Kontinuirano osvježavanje*

Napomena: Doprinose u aktivnostima treba navesti u satima po članovima grupe po aktivnosti.

	Bernard Kazazić	Marijan Erlić	Barbara Pavlović	Ante Pašalić	Luka Raspuđić	Petar Sušac	Veronika Žunar
Upravljanje projektom							
Opis projektnog zadatka							
Funkcionalni zahtjevi							
Opis pojedinih obrazaca							
Dijagram obrazaca							
Sekvencijski dijagrami							
Opis ostalih zahtjeva					60		
Arhitektura i dizajn sustava							
Baza podataka							
Dijagram razreda							
Dijagram stanja							
Dijagram aktivnosti							
Dijagram komponenti							
Korištene tehnologije i alati							
Ispitivanje programskog rješenja							
Dijagram razmještaja							
Upute za puštanje u pogon							
Dnevnik sastajanja					120		
Zaključak i budući rad							
Popis literature							

	Bernard Kazazić						
	Marijan Erlić						
	Barbara Pavlović						
	Ante Pašalić						
	Luka Raspudić						
	Petar Sušac						
	Veronika Žunar						
Dodatne stavke kako ste podijelili izradu aplikacije npr. izrada početne stranice izrada baze podataka spajanje s bazom podataka back end							

## Dijagrami pregleda promjena

### *dio 2. revizije*

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