

# Programsko inženjerstvo

Ak. god. 2021./2022.

## Digitalizacija

Dokumentacija, Rev. 1

Grupa: *ZelenaLipa*

Voditelj: *Matej Lopotar*

Datum predaje: *19.11.2021.*

Nastavnik: *Igor Stančin*

# Sadržaj

<b>1 Dnevnik promjena dokumentacije</b>	<b>3</b>
<b>2 Opis projektnog zadatka</b>	<b>5</b>
<b>3 Specifikacija programske potpore</b>	<b>7</b>
3.1 Funkcionalni zahtjevi . . . . .	7
3.1.1 Obrasci uporabe . . . . .	9
3.1.2 Sekvencijski dijagrami . . . . .	19
3.2 Ostali zahtjevi . . . . .	23
<b>4 Arhitektura i dizajn sustava</b>	<b>24</b>
4.1 Baza podataka . . . . .	25
4.1.1 Opis tablica . . . . .	25
4.1.2 Dijagram baze podataka . . . . .	28
4.2 Dijagram razreda . . . . .	29
4.3 Dijagram stanja . . . . .	31
4.4 Dijagram aktivnosti . . . . .	32
4.5 Dijagram komponenti . . . . .	33
<b>5 Implementacija i korisničko sučelje</b>	<b>34</b>
5.1 Korištene tehnologije i alati . . . . .	34
5.2 Ispitivanje programskog rješenja . . . . .	35
5.2.1 Ispitivanje komponenti . . . . .	35
5.2.2 Ispitivanje sustava . . . . .	35
5.3 Dijagram razmještaja . . . . .	36
5.4 Upute za puštanje u pogon . . . . .	37
<b>6 Zaključak i budući rad</b>	<b>47</b>
<b>Popis literature</b>	<b>48</b>
<b>Indeks slika i dijagrama</b>	<b>49</b>

**Dodatak: Prikaz aktivnosti grupe**

**50**

# 1. Dnevnik promjena dokumentacije

## *Kontinuirano osvježavanje*

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Napravljen predložak.	Matej Lopotar	29.10.2021.
0.2	Opis projektnog zadatka.	Antonio Babić i Matej Lopotar	31.10.2021.
0.3	Napisani UC-ovi	Iwan Ćulemović, Josip Hanak Ana Marija Pavičić	7.11.2021.
0.4	ER i REL sheme	Matej Lopotar i Iwan Ćulemović	9.11.2021.
0.5	Sekvencijski dijagrami	Matej Lopotar, Andrej Pogačić, Antonio Kuran	16.11.2021.
0.6	Dodane slike u dokumentaciju	Matej Lopotar	16.11.2021.
0.7	Ispravak UC-ova	Matej Lopotar	16.11.2021.
0.8	Arhitektura sustava, Dijagram razreda, Obrasci uporabe	Matej Lopotar i Antonio Babić	19.11.2021.
1.1	Uređivanje teksta – funkcionalni i nefunkcionalni zahtjevi	* *	14.09.2013.
1.2	Manje izmjene:Timer - Brojilo vremena	*	15.09.2013.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodatka	Autori	Datum
1.3	Popravljeni dijagrami obrazaca uporabe	*	15.09.2013.
1.5	Generalna revizija strukture dokumenta	*	19.09.2013.
1.5.1	Manja revizija (dijagram razmještaja)	*	20.09.2013.
2.0	Konačni tekst predloška dokumentacije	*	28.09.2013.

*Moraju postojati glavne revizije dokumenata 1.0 i 2.0 na kraju prvog i drugog ciklusa. Između tih revizija mogu postojati manje revizije već prema tome kako se dokument bude nadopunjavao. Očekuje se da nakon svake značajnije promjene (dodatka, izmjene, uklanjanja dijelova teksta i popratnih grafičkih sadržaja) dokumenta se to zabilježi kao revizija. Npr., revizije unutar prvog ciklusa će imati oznake 0.1, 0.2, ..., 0.9, 0.10, 0.11.. sve do konačne revizije prvog ciklusa 1.0. U drugom ciklusu se nastavlja s revizijama 1.1, 1.2, itd.*

## 2. Opis projektnog zadatka

Cilj ovog projekta je izraditi web aplikaciju koja će računovodstvenim tvrtkama ubrzati digitalizaciju. Glavna funkcionalnost aplikacije je detekcija dokumenta s učitanih slika i izrada OCR-a (optical character recognition) detektiranog teksta. Učitana slika mora biti slikana iz dobrog kuta te mora imati približno pravokutan oblik. Moguće je istovremeno učitati do 50 uslikanih dokumenata.

Pristupanjem na aplikaciju korisniku se prikazuju opcije prijave ili registracije ovisno o tome ima li profil. Za prijavu je potrebna email adresa i šifra, a za registraciju potrebno je upisati:

- ime
- prezime
- email adresa
- identifikacijski broj (dobiva se pri zaposlenju u tvrtku)
- željena šifra

Nakon prijave ovisno o ulozi se korisniku dodjeljuju prava. Svaki korisnik ima mogućnost učitati dokument. Nakon što je napravljen OCR dokumenta, korisniku se prikazuje sažetak dokumenta. Korisnik može skenirani dokument označiti kao točno skenirani ili kao krivo skenirani. Što god korisnik odabrao, dokument i korisnikov odabir spremaju se u bazu. Korisnici aplikacije su zaposlenik, revizor, računovođa, direktor i administrator.

*Zaposlenik* ulaskom u aplikaciju odabire jednu od dvije mogućnosti može aplikacijom skenirati dokumente i može vidjeti povijest svih svojih skeniranja (datum i skenirani dokument). Nakon skeniranja željenog broja dokumenata, zaposleniku je prikazan sažetak svakog od priloženih dokumenata, te on provjerava ispravnost svakog pojedinačnog dokumenta. Odobreni dokumenti šalju se revizoru.

*Revizor* dobiva dokumente koje mu šalju zaposlenici te ih provjerava sve kako bi svaki dokument preusmjerio do ispravnog računovođe zaduženog za taj tip dokumenata. Ako revizor skenira dokumente aplikacija će automatski iz dobivenog teksta odrediti kojem računovođi se šalje dokument.

*Računovođa* dobivene dokumente arhivira. Aplikacija prilikom arhiviranja dokumentu dodjeljuje jedinstveni broj arhiva. Također, računovođa ima opciju slanja

dokumenata direktoru na potpis prije arhiviranja. Računovođa može arhivirati poslane dokumente tek kada dobije potvrdu da je direktor potpisao traženi dokument.

*Direktor* može vidjeti povijest svih dokumenata te povijest i statistike svih zaposlenika. Direktor ima mogućnost promaknuti članove tvrtke nakon čega će administratoru poslati obavijest u kojem se traži da se zaposleniku daju veće ovlasti u aplikaciji. Također, potpisuje dokumente koje mu šalje računovođa te ih prosljeđuje natrag nakon potpisa.

Postoje tri tipa dokumenata – računi, ponude i interni dokumenti. Računi će u svom tekstu nakon OCR-a imati oznaku računa koja je veliko slovo R te šest znamenaka, oznaka ponude će imati veliko slovo P i devet znamenaka, a oznaka internog dokumenta „INT“ i četiri znamenke. Računi osim oznake sadrže ime klijenta, artikle s cijenama i ukupnu cijenu. Ponude su kao računi, ali ne sadrže ime klijenta. Interni dokumenti sadrže samo nestrukturirani tekst.

## 3. Specifikacija programske potpore

### 3.1 Funkcionalni zahtjevi

#### Dionici:

1. Korisnici
2. Admini sustava
3. Razvojni tim

#### Aktori i njihovi funkcionalni zahtjevi:

1. Zaposlenik (inicijator) može:
  - (a) učitavati slike
  - (b) skenirati dokumente
  - (c) pregledati povijest skeniranih dokumenata
  - (d) slati dokumente revizoru
2. Revizor (sudionik) može:
  - (a) skenirati dokument
  - (b) provjeriti jeli dokumenat ispravan i poslati ga računovođi
3. Računovođa (sudionik) može:
  - (a) dodjeljivati jedinstveni broj arhivu
  - (b) arhivirati dokument
  - (c) slati obavijest direktoru da se potpiše dokument
4. Direktor (sudionik) može:
  - (a) potpisati dokumente i poslati obavijest da je dokument potpisan
  - (b) pregledati povijest svih dokumenata
  - (c) pregledati statistiku svih zaposlenika
5. Baza podataka (sudionik) može:
  - (a) dodati nove dokumente i arhive u bazu



- (b) vraćati povijest svih dokumenata
- (c) vraćati statistiku svih zaposlenika

### 3.1.1 Obrasci uporabe

#### Opis obrazaca uporabe

##### UC1 - Prijava korisnika

- **Glavni sudionik:** Zaposlenik
- **Cilj:** Omogućiti prijavu postojećim korisnicima ili odabrati registraciju novih
- **Sudionici:** Baza podataka
- **Preduvjet:** U bazi podataka mora postojati račun korisnika, odnosno direktor mora odobriti registraciju
- **Opis osnovnog tijeka:**
  1. Korisnik unosi svoje korisničke detalje i nakon provjere ulazi u home page
  2. Nepostojeći korisnici imaju opciju preusmjerenja na registraciju
- **Opis mogućih odstupanja:**
  - 2.a Korisnik je unio korisničke detalje koji su krivi ili nepostojani
    1. Ako je korisničko ime poznato ali korisnička šifra kriva korisnik mora ponovno unijeti šifru dok ne bude ispravna
    2. Ako je korisničko ime nepostojano unutar baze podataka korisnika se navodi na registraciju

##### UC2 - Odjava korisnika

- **Glavni sudionik:** Zaposlenik
- **Cilj:** Omogućiti odjavu zaposlenika
- **Sudionici:** -
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Zaposlenik odabire opciju odjave
  2. Web aplikacija završava sesiju

##### UC3 - Prilaganje dokumenta

- **Glavni sudionik:** Zaposlenik
- **Cilj:** Priložiti dokument
- **Sudionici:** Revizor, baza podataka
- **Preduvjet:** Ulogirani verificirani zaposlenik, funkcionalna kamera
- **Opis osnovnog tijeka:**

1. Zaposlenik klikće na označeno mjesto za prilaganje slika
  2. Zaposlenik označuje slike koje će se priložiti i prilaže ih
  3. Zaposlenik inicira OCR test
  4. Web aplikacija vraća dokumente u skeniranom obliku
  5. Zaposlenik označava jeli dokument krivo ili točno skeniran
  6. Zaposlenik šalje revizoru dokument
- **Opis mogućih odstupanja:**
    1. Korisnik je priložio više od 50 slika
      1. Javiti korisniku grešku i onemogućiti slanje

#### UC4 - Verifikacija dokumenata

- **Glavni sudionik:** Revizor
- **Cilj:** Verificirati dokumente i poslati računovođi
- **Sudionici:** Računovođa
- **Preduvjet:** Zaposlenik mora nešto poslati
- **Opis osnovnog tijeka:**
  1. Revizor dobiva poslani skenirani dokument
  2. Revizor provjerava dokument i sam ga šalje ga web aplikaciji
  3. Web aplikacija proslijeđuje dokument računovođi kojeg je revizor oda-  
brao
  4. Ako je revizor skenirao dokument, web aplikacija će sama odrediti ko-  
jem se računovođi šalje

#### UC5 - Pregled povijesti skeniranih dokumenata

- **Glavni sudionik:** Zaposlenik
- **Cilj:** Dobiti listu povijesti skeniranih dokumenata
- **Sudionici:** Baza podataka
- **Preduvjet:** Ulogirani verificirani zaposlenik
- **Opis osnovnog tijeka:**
  1. Zaposlenik klikće na mjesto za prikaz povijesti dokumenata
  2. Web aplikacija šalje upit bazi podataka koji sadrži podatke o zaposle-  
niku
  3. Baza podataka vraća listu skeniranih dokumenata

**UC6 - Arhiviranje dokumenata**

- **Glavni sudionik:** Računovođa
- **Cilj:** Arhiviranje dokumenata
- **Sudionici:** Baza podataka, direktor
- **Preduvjet:** Ako je dokument poslan direktoru on mora biti odobren
- **Opis osnovnog tijeka:**
  1. Računovođa odabire dokumente te ih odmah arhivira ili ih šalje direktoru te čeka odgovor
  - 2.a Direktor nije odobrio dokument
    1. Računovođa je obaviješten
    2. Računovođa ima opciju brisanja dokumenata

**UC7 - Direktorski potpis**

- **Glavni sudionik:** Direktor
- **Cilj:** Potpisati dokument elektroničkim potpisom
- **Sudionici:** Računovođa, baza podataka
- **Preduvjet:** Računovođa mora poslati nearhivirane dokumente za potpis
- **Opis osnovnog tijeka:**
  1. Računovođa šalje web aplikaciji nepotpisani nearhivirani dokument
  2. Web aplikacija ga prosljeđuje direktoru
  3. Direktor dobiva obavijest u inboxu i poslani nearhivirani dokument
  4. Direktor potpisuje dokument i šalje potpisani dokument web aplikaciji
  5. Web aplikacija šalje upit za arhiviranje potpisanog dokumenta u bazi podataka

**UC8 - Pregled potpisanih dokumenata**

- **Glavni sudionik:** Direktor
- **Cilj:** Dohvatiti listu svih potpisanih dokumenata
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Direktor šalje upit web aplikaciji za za potpisane dokumente
  2. Web aplikacija šalje upit bazi podataka
  3. Nakon što baza podataka vrati listu dokumenata, web aplikacija je prosljeđuje direktoru
  4. Direktor odabрати iz liste potpisani dokument i pregledati ga

**UC9 - Pregled podataka o zaposlenicima**

- **Glavni sudionik:** Direktor
- **Cilj:** Dohvatiti podatke o zaposlenicima
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Direktor šalje upit web aplikaciji o statistici o zaposlenicima
  2. Web aplikacija šalje upit bazi podataka
  3. Nakon što baza podataka vrati listu, web aplikacija je prosljeđuje direktoru
  4. Direktor odabirati iz liste zaposlenika i pregledati sve o njemu

**UC10 - Dodavanje komentara dokumentu**

- **Glavni sudionik:** Zaposlenik
- **Cilj:** Dodavanje kratkog opisa svakom priloženom dokumentu
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Zaposlenik odabire dokument te ima opciju nadodati kratak opis cijelog dokumenta

**UC11 - Stvaranje korisničkog računa**

- **Glavni sudionik:** Zaposlenik
- **Cilj:** Registrirati nove zaposlenike
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju registracije te unosi svoje osnovne podatke, jedinstvenu šifru što je dobio pri zaposlenju te aktivira svoj račun
  2. Web aplikacija dobiva obrazac te provjerava postoji li zaposlena osoba sa odgovarajućom jedinstvenom šifrom, imenom i prezimenom
  3. Ako postoji zaposlena osoba, u bazi podataka će se umetnuti novi korisnički račun
- **Opis mogućih odstupanja:**
  1. Zaposlenik je unio nepostojeću šifru u bazi podataka (ne postoji zaposlenik s tom šifrom)

## 1. Odbiti registraciju

**UC12 - Deaktivacija korisničkog računa**

- **Glavni sudionik:** Zaposlenik
- **Cilj:** Deaktivirati račun korisnika
- **Sudionici:** Baze podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Zaposlenik odabire opciju deaktivacije računa, potvrđuje deaktivaciju te upisuje zaporku
  2. Web aplikacija će dobiti obrazac te će deaktivirati korisnički račun slanjem upita bazi podataka za deaktivacijom korisničkog računa
  3. Baza podataka će postaviti atribut aktivnog računa na false

**UC13 - Potpuno brisanje korisničkog računa**

- **Glavni sudionik:** Direktor
- **Cilj:** Obrisati račun korisnika
- **Sudionici:** Baze podataka
- **Preduvjet:** Zaposlenik mora dobit otkaz
- **Opis osnovnog tijeka:**
  1. Ako je zaposlenik dobio otkaz nema više pravo koristiti aplikaciju te baza podataka uklanja račune onima koji su dobili otkaz (ON DELETE)

**UC14 - Pregled arhiviranih dokumenata**

- **Glavni sudionik:** Računovođa
- **Cilj:** Omogućiti pregled svih arhiviranih dokumenata
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Računovođa bira pregled arhiviranih dokumenata
  2. Web aplikacija šalje upit bazi podataka za listu sa arhiviranim dokumentima
  3. Nakon što baza vrati listu, web aplikacija je prosljeđuje računovođi
  4. Računovođa su prikazani svi arhivirani dokumenti te ih može pojedinačno odabrati i pregledati

**UC15 - Pregled hijerarhija zaposlenika**

- **Glavni sudionik:** Zaposlenik
- **Cilj:** Omogućiti prikaz hijerarhije zaposlenika unutar firme
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Zaposlenik odabire pregled hijerarhije
  2. Web aplikacija šalje upit bazi podataka za listu sa hijerarhijom zaposlenih
  3. Nakon što baza vrati listu, web aplikacija je prosljeđuje korisniku

**UC16 - Pregled novijih registracija**

- **Glavni sudionik:** Direktor
- **Cilj:** Omogućiti pregled novih registracija
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Direktor šalje upit web aplikaciji sa odabranim periodom zadnjih registracija npr. 1 dan, 1 tjedan, 3 mjeseca... (filtar)
  2. Web aplikacija šalje upit bazi podataka za listu sa uvjetom perioda
  3. Nakon što baza vrati listu, web aplikacija je prosljeđuje direktoru
  4. Direktor može pregledati listu registriranih u tom periodu

**UC17 - Dodjela pozicije zaposleniku**

- **Glavni sudionik:** Direktor
- **Cilj:** Dodjeliti poziciju zaposlenika u tvrtci
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Direktor šalje upit web aplikaciji sa ili šifrom ili imenom ili prezimenom zaposlenika (filtar)
  2. Web aplikacija šalje upit bazi podataka za listu sa ili šifrom ili imenom ili prezimenom
  3. Nakon što baza vrati listu, web aplikacija je prosljeđuje direktoru
  4. Direktor iz liste može odabrati traženog zaposlenika i kliknuti na mjesto postavljanje uloge zaposlenika

5. Šalje se uputa web aplikaciji za ažuriranje uloge zaposlenika specifične jedinstvene šifre

#### UC18 - Dodati zaposlenika u tablicu zaposlenih

- **Glavni sudionik:** Direktor
- **Cilj:** Zaposliti osobu
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Direktor upisuje sve podatke o osobi u zato predviđena mjesta te šalje obrazac web aplikaciji
  2. Web aplikacija generira jedinstvenu šifru te sve zajedno sprema u bazu podataka, a također još tu šifru vraća direktoru

#### UC19 - Ukloniti zaposlenika iz tablice zaposlenih

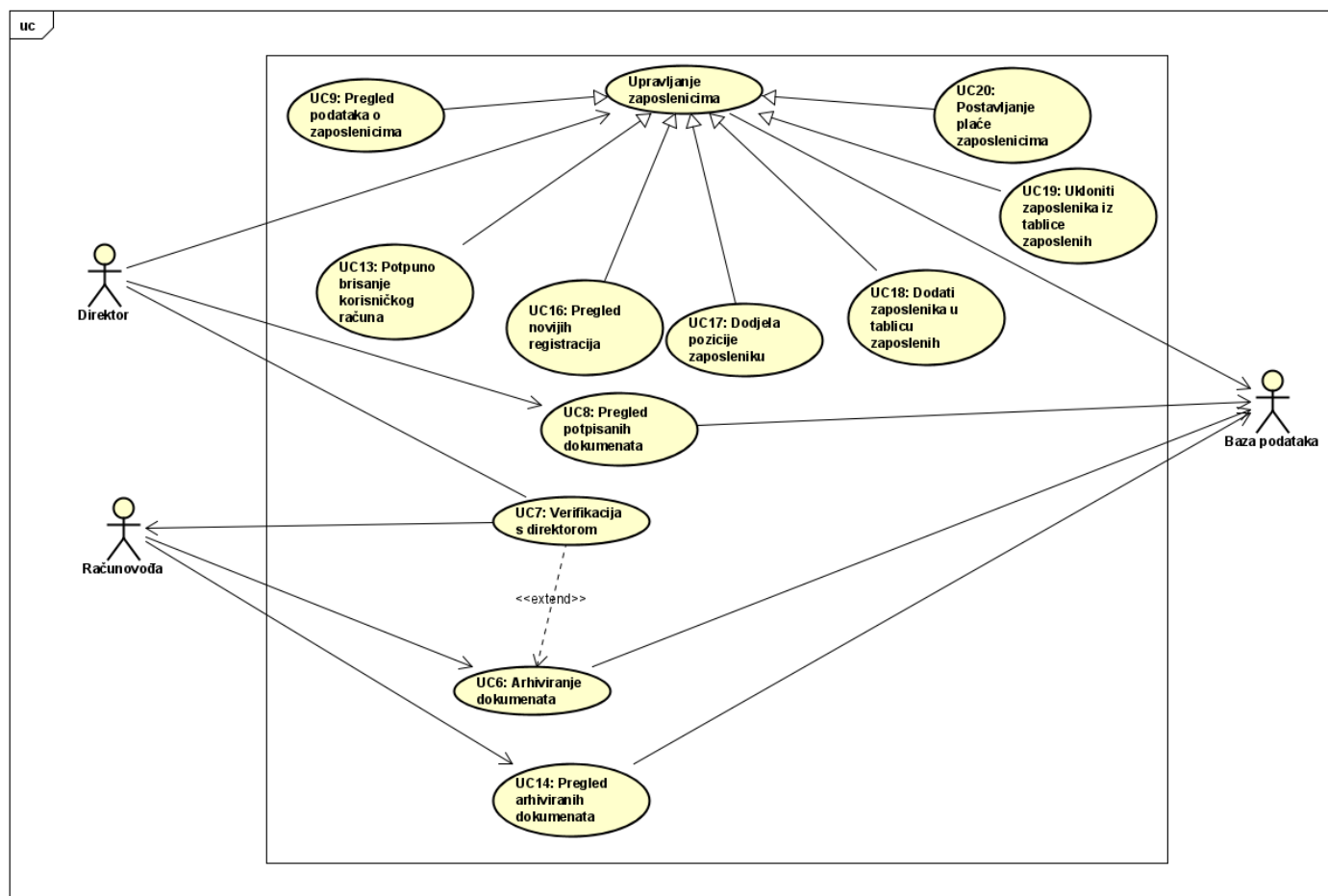
- **Glavni sudionik:** Direktor
- **Cilj:** Dati otkaz zaposlenom
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Direktor šalje upit web aplikaciji sa ili šifrom ili imenom ili prezimenom zaposlenika (filtar)
  2. Web aplikacija šalje upit bazi podatak za listu sa ili šifrom ili imenom ili prezimenom
  3. Nakon što baza vrati listu, web aplikacija je prosljeđuje direktoru
  4. Direktor iz liste može odabrati traženog zaposlenika i kliknuti na gumb za davanje otkaza
  5. Šalje se uputa web aplikaciji za uklanjanje zaposlenika specifične jedinstvene šifre te web aplikacija uklanja zaposlenika te se njegova jedinstvena šifra deaktivira (također se miče iz baze podataka)
  6. račun zaposlenika (ako postoji) se također automatski uklanja iz baze podataka



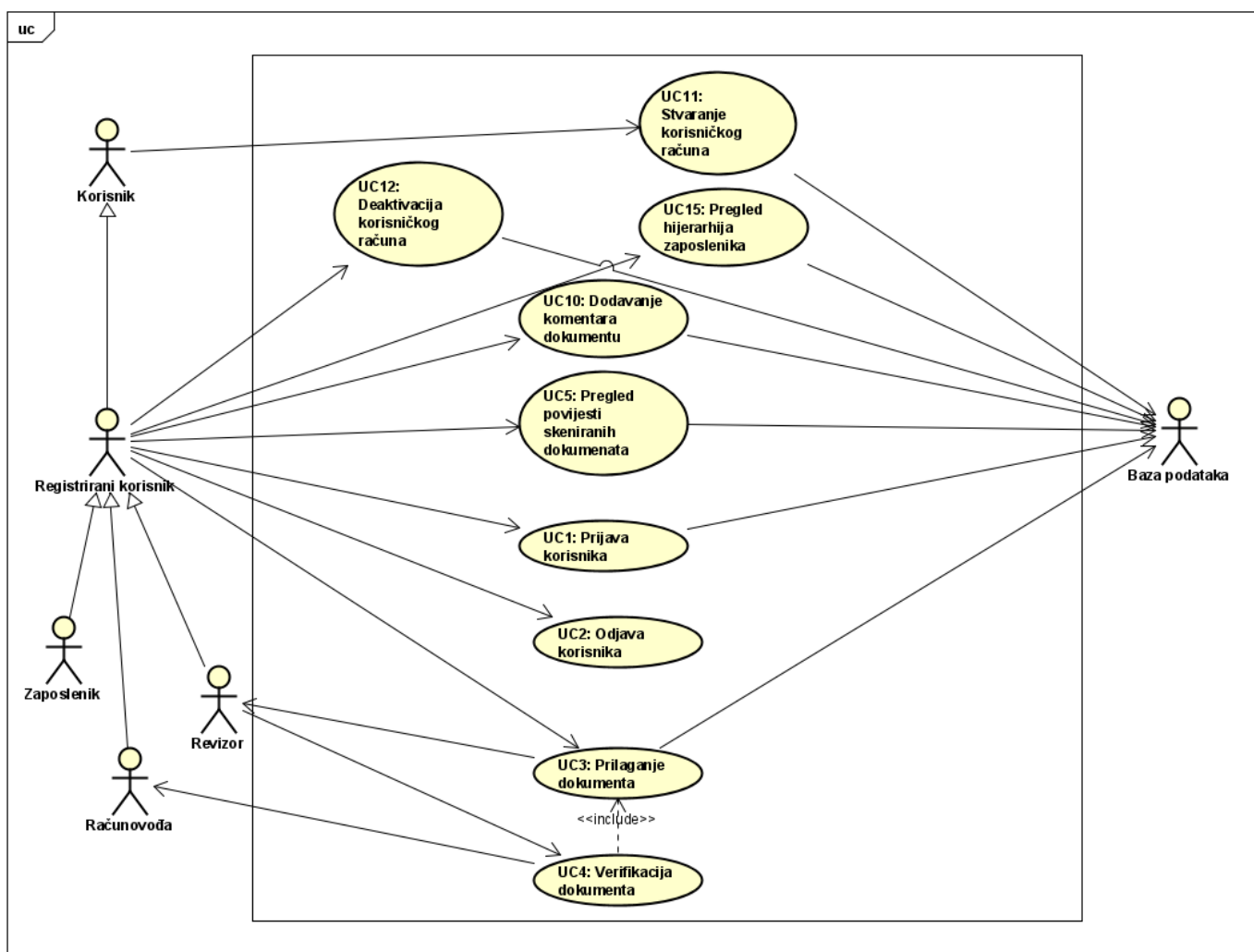
**UC20 - Postavljanje plaće zaposlenicima**

- **Glavni sudionik:** Direktor
- **Cilj:** Dati otkaz radniku
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Direktor šalje upit web aplikaciji sa ili šifrom ili imenom ili prezimenom radnika (filtar)
  2. Web aplikacija šalje upit bazi podatak za listu sa ili šifrom ili imenom ili prezimenom
  3. Nakon što baza vrati listu, web aplikacija je prosljeđuje direktoru
  4. Direktor iz liste može odabrati traženog radnika i kliknuti na mjesto postavljanje plaće
  5. Šalje se uputa web aplikaciji za ažuriranje plaće radnika specifične jedinstvene šifre

## Dijagrami obrazaca uporabe



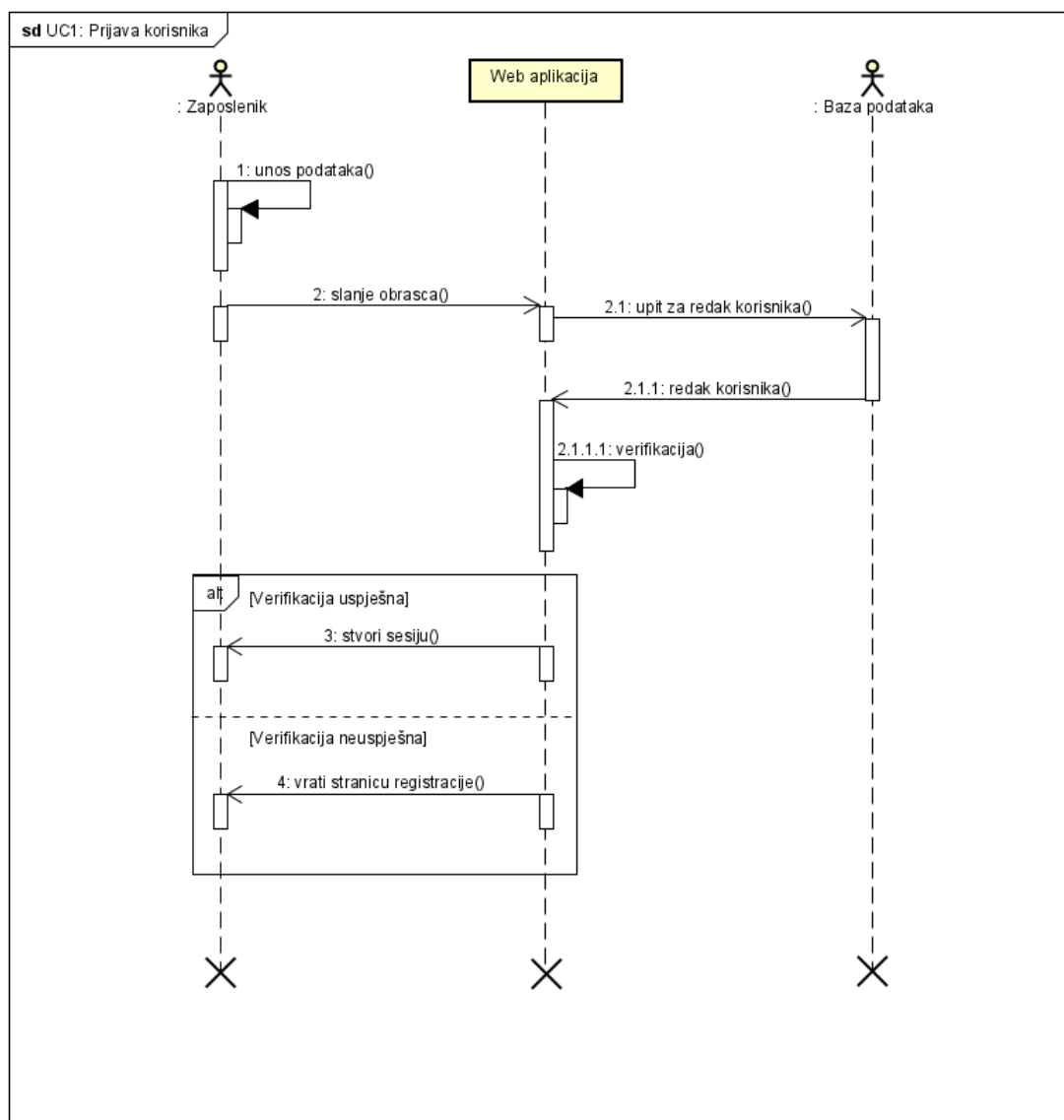
Slika 3.1: Dijagram obrasca uporabe, funkcionalnost direktora i računovođe



Slika 3.2: Dijagram obrasca uporabe, funkcionalnost registriranih korisnika

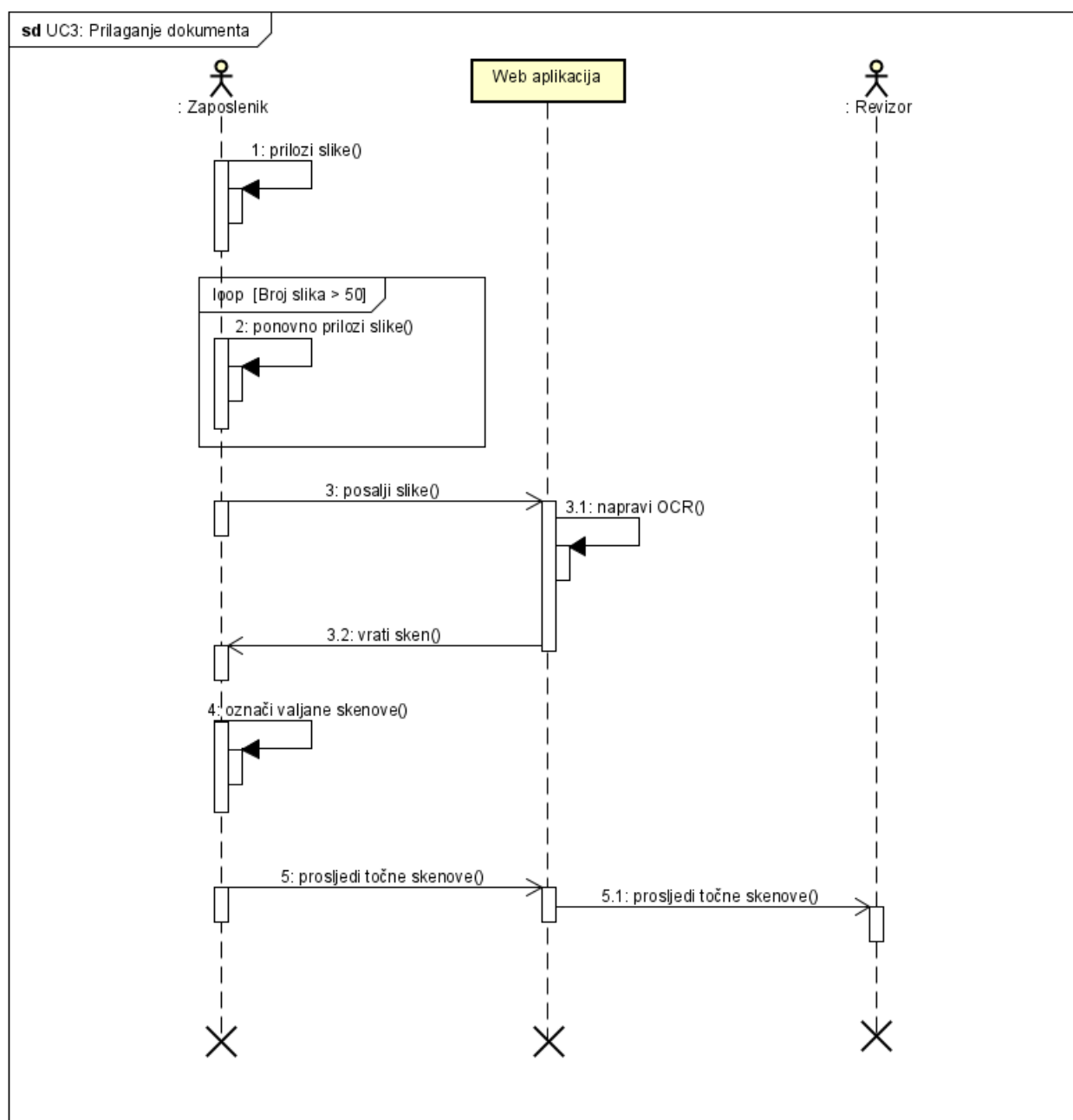
### 3.1.2 Sekvencijski dijagrami

Zaposlenik unosi svoje podatke kako bi se mogao prijaviti i ući u aplikaciju. Poslužitelj iz baze podataka dohvaća sve zaposlenike unutar tvrtke te traži postoji li zaposlenik s upisanim podacima. Ukoliko zaposlenik postoji u bazi podataka, zaposlenik se uspješno prijavio, a ukoliko ne postoji, poslužitelj ga šalje na stranicu registracije.



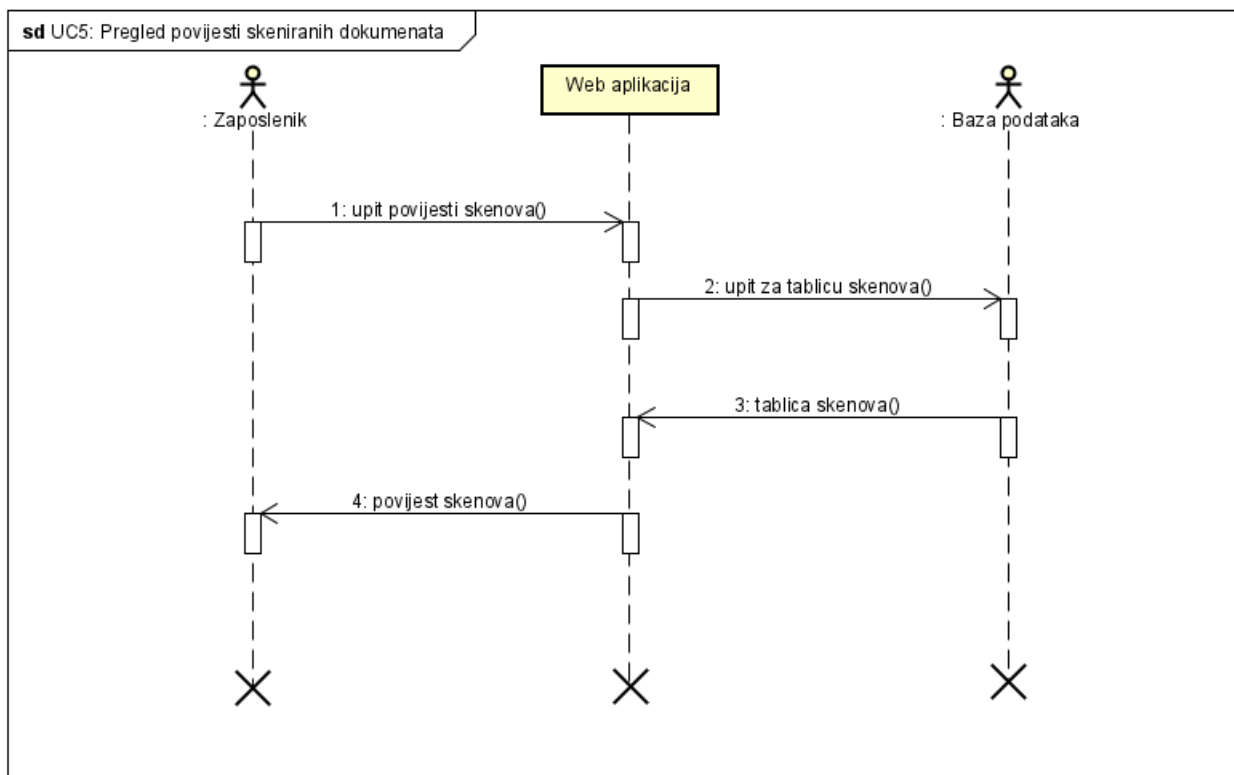
Slika 3.3: UC1 - Prijava korisnika

Zaposlenik priloži (engl. uploada) sliku. Ima mogućnost priložiti do 50 slika. Ukoliko priloži više od 50 slika, ponavlja postupak sve dok ne priloži dozvoljeni broj. Nakon što uspješno priloži slike, slike se šalju poslužitelju koji radi OCR te zaposleniku vraća skenirani dokument. Zaposlenik ima mogućnost pogledati sve skenirane dokumente te mora označiti koji skenirani dokumenti su ispravni te ih šalje na poslužitelj koji ih prosljeđuje revizoru na daljnju obradu.



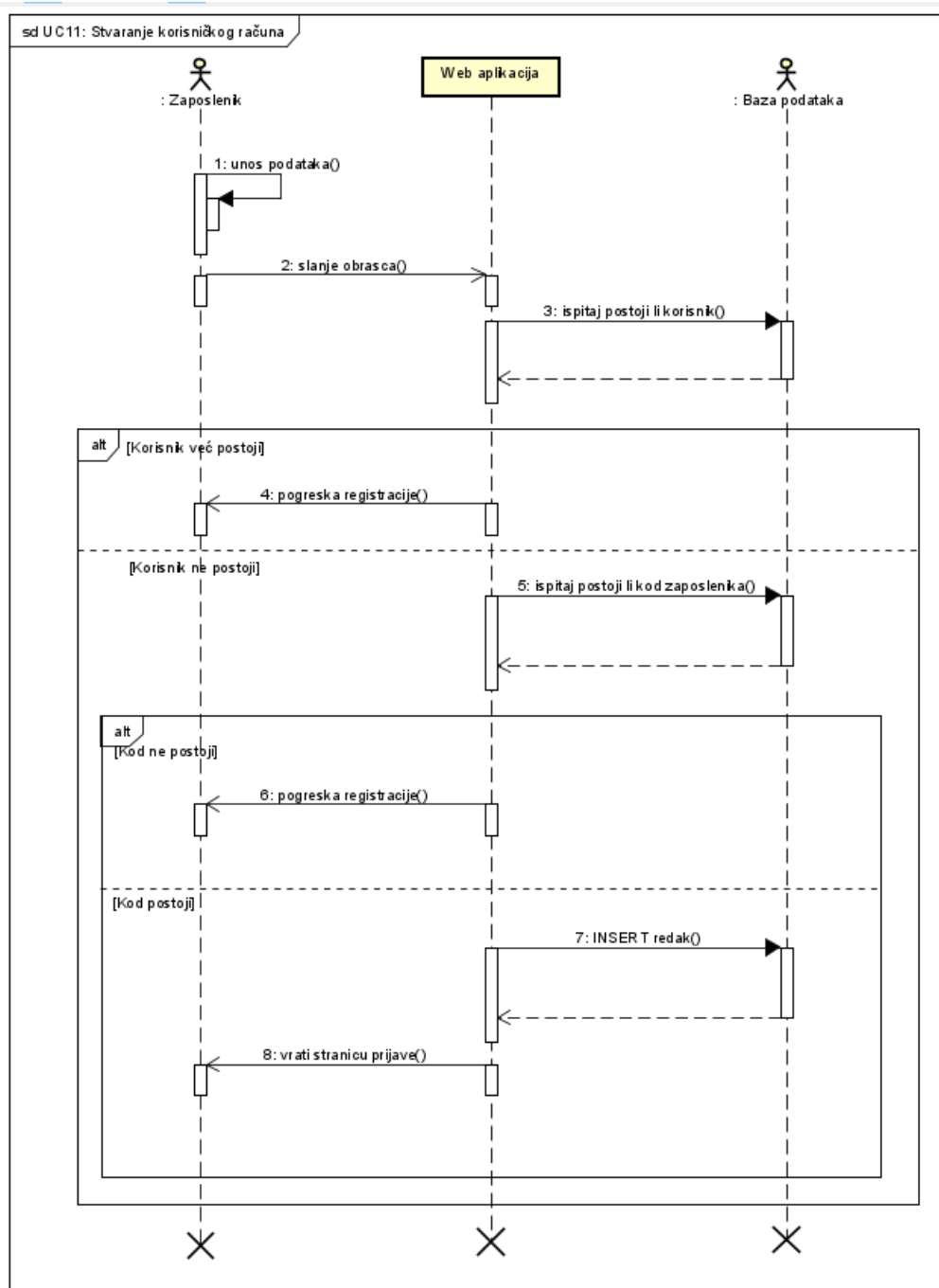
Slika 3.4: UC3 - Prilaganje dokumenta

Zaposlenik šalje upit poslužitelju za povijest skeniranih dokumenata. Poslužitelj upit prosljeđuje bazi podataka. Baza podataka iz tablice uzima zatražene dokumente te ih šalje poslužitelju koji ih proslijedi zaposleniku.



Slika 3.5: UC5 Pregled - povijesti skeniranih dokumenata

Zaposlenik unosi svoje podatke. Podaci se šalju poslužitelju koji pita bazu postoji li ovaj korisnik. Ako korisnik postoji, poslužitelj korisniku javlja da je registracija pogrešna, a ako korisnik ne postoji, web poslužitelj šalje upit bazi je li zaposlenik upisao dobar jedinstveni kod. Ukoliko je kod neispravan, poslužitelj korisniku javlja da je registracija pogrešna, a ako je kod ispravan podaci se zapisuju u bazu podataka te se korisnik prosljeđuje na stranicu za prijavu.



Slika 3.6: UC11 - Registracija računa

## 3.2 Ostali zahtjevi

- Sustav treba omogućiti rad više korisnika u isto vrijeme
- Korisničko sučelje za prijavu i registraciju moraju podržavati hrvatsku abecedu
- Pristup sustavu mora biti omogućen iz javne mreže pomoću HTTPS (heroku deployment)
- Ukoliko se korisnik krivo služi aplikacijom, aplikacija mora i dalje biti funkcionalna
- Pristup bazi podataka ne smije trajati dulje od 5 sekundi
- Sustav treba biti implementiran kao web aplikacija koristeći objektno-orijentirane jezike
- OCR treba biti funkcionalan



## 4. Arhitektura i dizajn sustava

Arhitektura se može podijeliti na tri podsustava:

- Web poslužitelj
- Web aplikacija
- Baza podataka

*Web preglednik* je program (Google Chrome) koji korisniku omogućuje pregled web-stranice, a ujedno je i prevoditelj. Dakle, stranica je pisana u kodu kojeg preglednik razumije te ga nakon toga interpretira kao nešto razumljivo korisniku.

*Web poslužitelj* ima zadaću omogućiti komunikaciju klijenta s aplikacijom. Komunikacija se odvija preko HTTP (engl. Hyper Text Transfer Protocol) protokola. Poslužitelj je onaj koji pokreće web aplikaciju te joj prosljeđuje zahtjeve.

*Web aplikacija* obrađuje zahtjeve koje korisnik pošalje te je u komunikaciji s bazom podataka s kojom preko poslužitelja vraća korisniku odgovor u obliku HTML dokumenta vidljivog web pregledniku.

Programski jezik korišten za izradu aplikacije je Java i Javascript. Za baze podataka korišten je pgAdmin. Aplikacija je puštena u pogon preko Heroku-a. Arhitektura sustava temeljit će se na MVC (Model-View-Controller) konceptu. MVC koncept smo koristili jer je podržan u JavaSpringBoot-u kojeg koristimo te je pogodan za moguće kohezije. MVC model odvaja korisničko sučelje od ostatka sustava.

- Model sadrži razrede čiji objekti se obrađuju
- Pogled sadrži razrede čiji objekti služe za prikaz podataka
- Nadglednik (Controller) sadrži razrede koji upravljaju i rukuju korisničkom interakcijom s pogledom i modelom

## 4.1 Baza podataka

U našoj web aplikaciji koristili smo relacijsku bazu podataka. Baza se sastoji od tablica koje imaju svoje atribute. Baza podataka je jako bitna da bi web aplikacija radila jer su na njoj pohranjeni svi podaci o zaposlenicima i svi skenirani dokumenti.

Entiteti baze podataka su sljedeći:

- Dokument
- Zaposlenik
- Korisnički račun
- Uloga
- Racun
- Ponuda
- Interni

### 4.1.1 Opis tablica

**Zaposlenik** Ovaj entitet sadrži osnovne informacije o zaposleniku. Sadrži sljedeće atribute: genID, PID, Name, Surname, Residence, Salary i roleID. genID je alternativni ključ dok je PID primarni ključ. Entitet je povezan s entitetom **UserAccount** preko genID i **Role** preko roleID.

Employee		
genID	VARCHAR	kod koji zaposlenik dobije od direktora
PID	VARCHAR	OIB zaposlenika
Name	VARCHAR	Ime zaposlenika
Surname	VARCHAR	Prezime zaposlenika
Residence	VARCHAR	Mjesto stanovanja
Salary	INT	Plaća zaposlenika
roleID	INT	jedinstveni kod pozicije (uloge)

**Role** Ovaj entitet sadržava roleID te RoleName što su ujedno i atributi. Povezan je sa entitetom **Employee** preko roleID.

Role		
roleID	INT	jedinstveni kod uloge
RoleName	VARCHAR	ime uloge

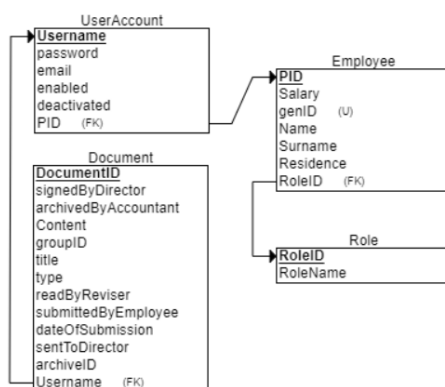
**Document** Ovaj entitet sadrži osnovne informacije o dokumentu. Sadrži sljedeće attribute: documentID, username, title, type, content, readbyreviser, archivedbyaccountant, signedbydirector, submittedbyemployee, dateofsubmission, senttodirector, archived. Entitet je povezan s entitetom **UserAccount** preko atributa Username.

Document		
documentID	INT	jedinstveni kod dokumenta
Archived	INT	je li dokument arhiviran
groupID	INT	šesteroznamenkasti brojevi
title	VARCHAR	naslov dokumenta
type	VARCHAR	tip dokumenta
content	VARCHAR	sadržaj dokumenta
readByRevise	BOOLEAN	je li revizor dobio dokument
archivedByAccountant	BOOLEAN	je li računovođa arhivirao dokument
signedByDirector	BOOLEAN	je li direktor potpisao dokument
submittedByEmployee	BOOLEAN	je li zaposlenik skenirao i poslao dokument
sentToDirector	BOOLEAN	je li dokument poslan direktoru
dateOfSubmission	TIMESTAMP	datum skeniranja
Username	VARCHAR	korisničko ime osobe koja je skenirala dokument

**UserAccount** Ovaj entitet sadrži podatke o korisničkim računima. Sadrži sljedeće attribute: Username što je ujedno i primarni key, password, email, enabled, deactivated i genID što je strani ključ. Entitet je povezan s entitetom **Document** preko atributa Username i s entitetom **Employee** preko atributa genID.

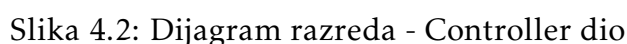
UserAccount		
Username	VARCHAR	korisničko ime
password	VARCHAR	šifra
email	VARCHAR	mail adresa
enabled	BOOLEAN	jel li account u funkciji
deactivated	BOOLEAN	je li account deaktiviran
genID	VARCHAR	oib zaposlenika

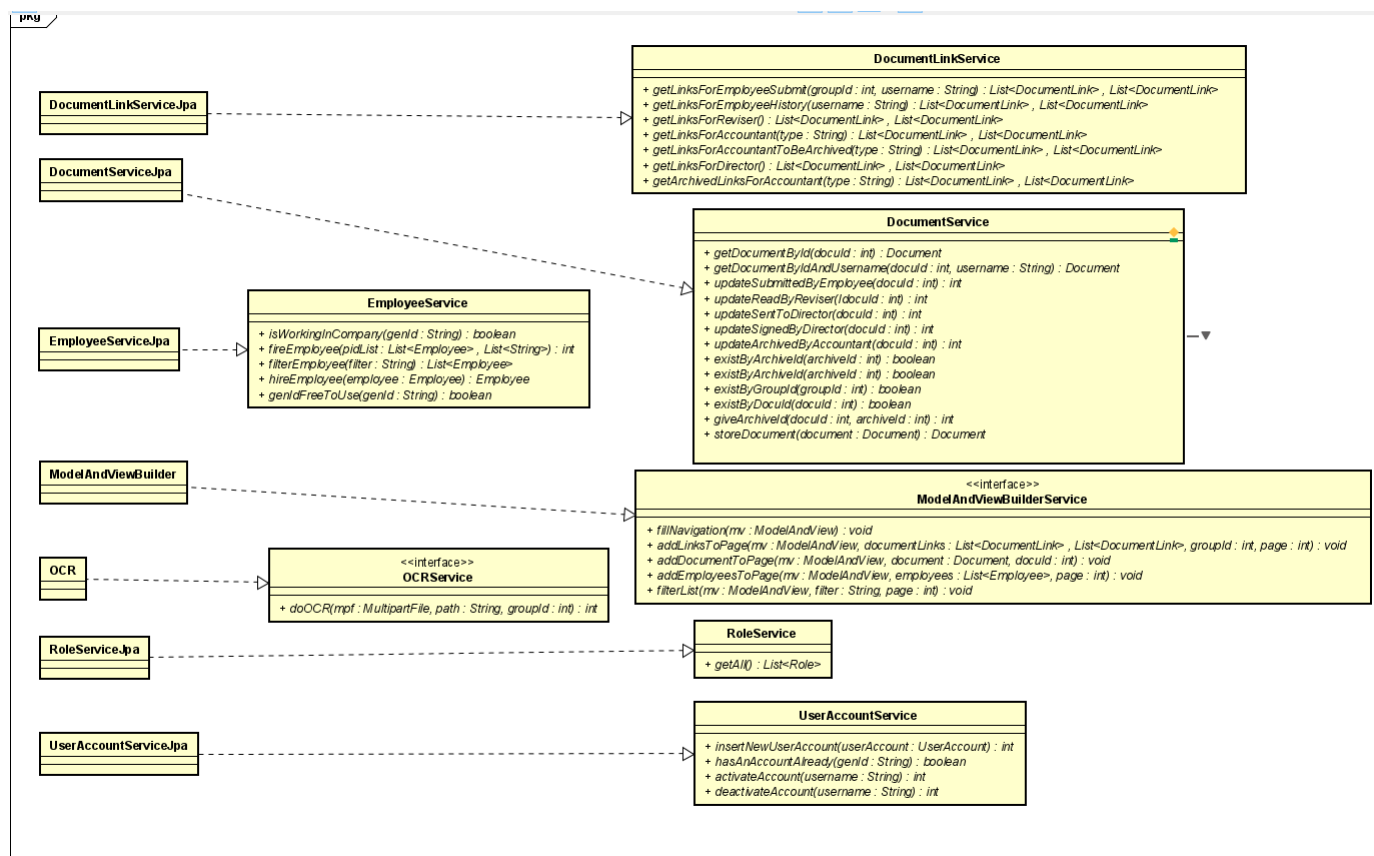
### 4.1.2 Dijagram baze podataka



Slika 4.1: Relacijski model baze podataka

Na sljedeće dvije slike prikazani su dijagrami razreda koji su prisutni u našoj aplikaciji. Slika 4.2 prikazuje dio s Controller dijelom MVC arhitekture dok slika 4.3 prikazuje sučelja i ostale klase koje su nam bile potrebne za implementaciju.

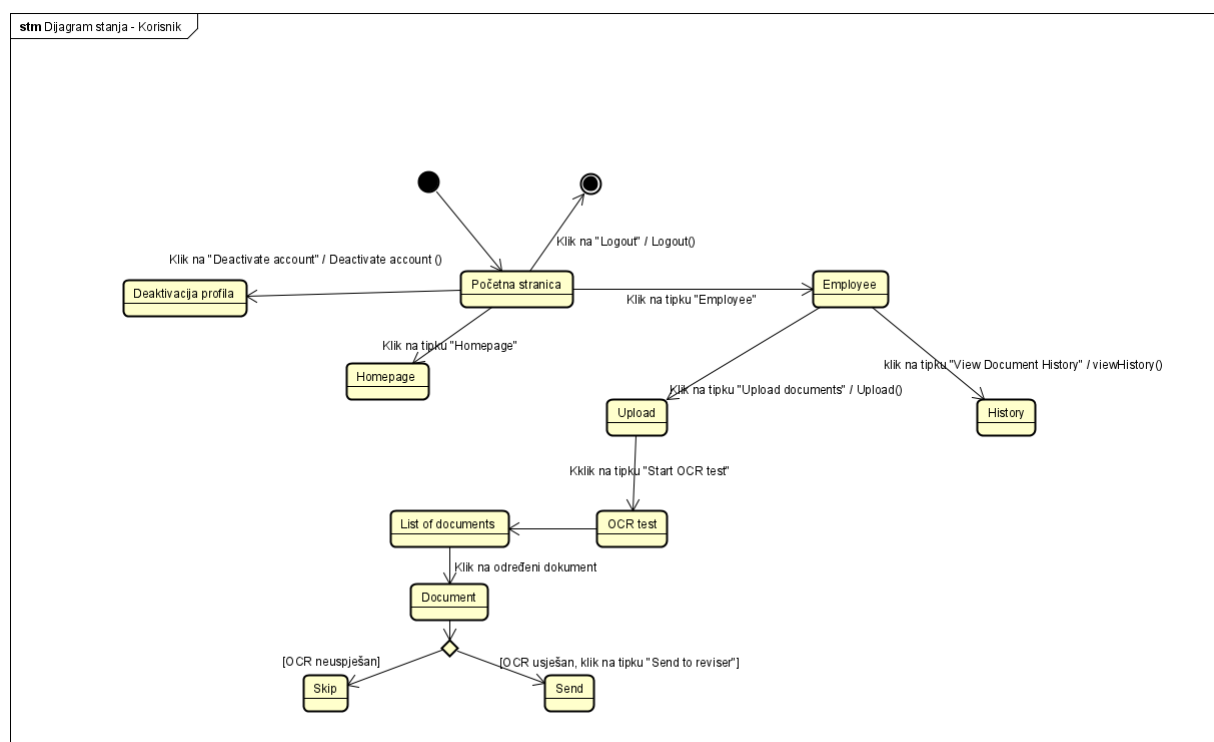




Slika 4.3: Dijagram razreda

### 4.3 Dijagram stanja

Dijagram stanja prikazuje prijelaze iz jednog stanja u drugo na temelju različitih događaja. U prvom planu je zaposlenik koji ima mogućnost odjaviti se sa profila, deaktivirati profil, vidjeti popis svih dokumenata koje je skenirao te upload dokumenata. Ukoliko uploada dokumente, mora ih skenirati pomoću OCR-a. Nakon toga dobiva listu skenova te gleda kako su skenovi ispali. Dokumenti kod kojih je OCR uspješno izveden šalju se revizoru na daljnju obradu, a ukoliko OCR nije uspješan dokument se ne šalje nego se preskoči.

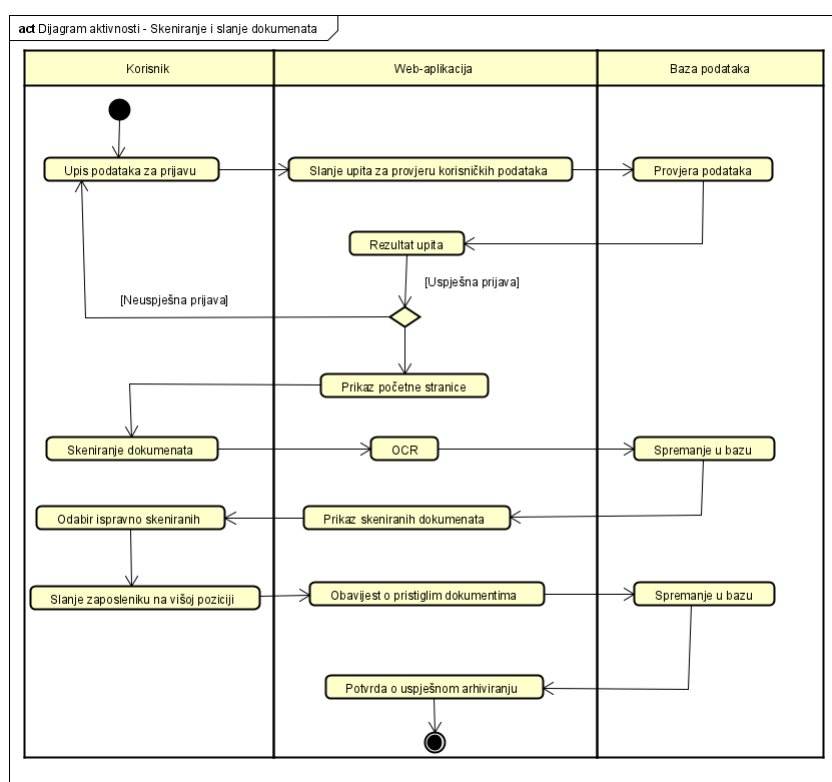


Slika 4.4: Dijagram stanja



## 4.4 Dijagram aktivnosti

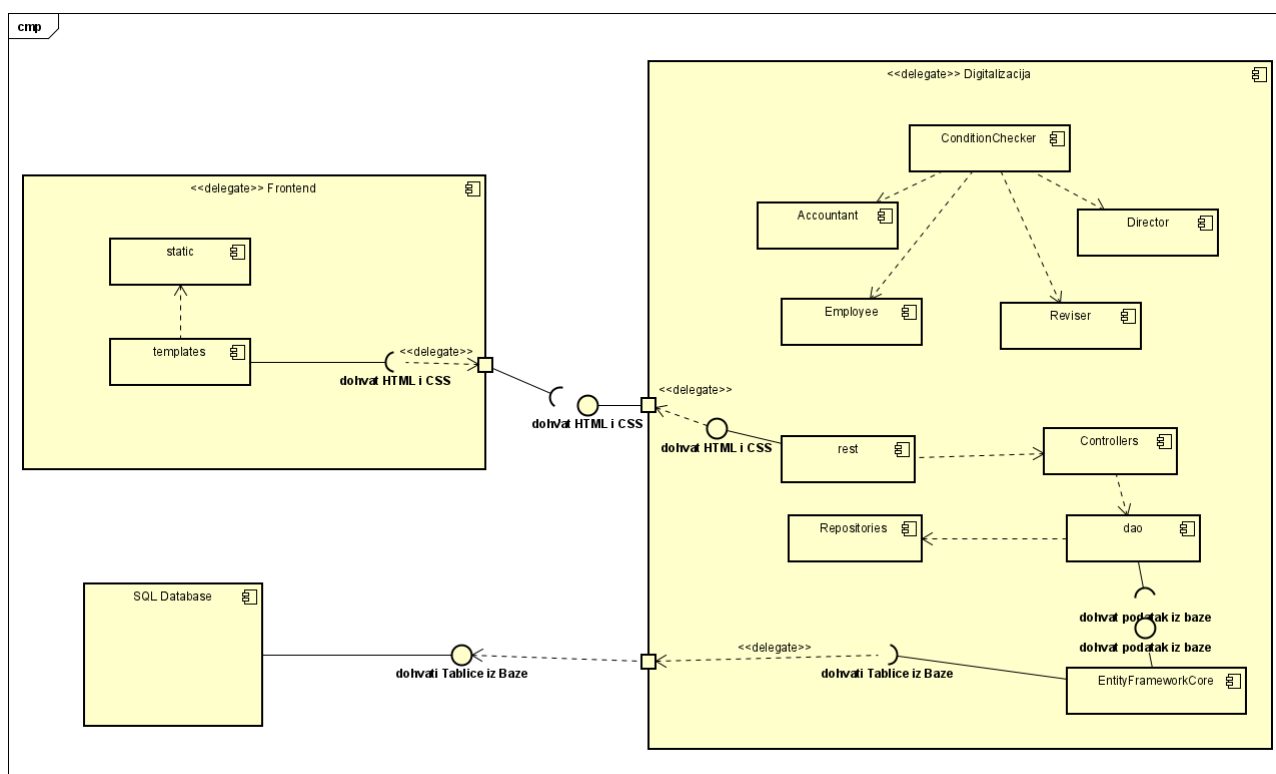
Dijagram aktivnosti odnosi se na korisnika, njegovo uploadanje dokumenata, skeniranje dokumenata i slanje dalje drugim zaposlenicima. Dakle, korisnik se prvo mora prijaviti u sustav. U bazi se provjeravaju podaci i ako je sve točno, prijava je uspješna. Ukoliko korisnik nije u bazi podataka, ima priliku opet se prijaviti. Nakon što se korisnik prijavio ima mogućnost da skenira dokument. Web aplikacija provede OCR, skenirani dokument se spremi u bazu podataka. Korisnik vidi kako izgledaju skenirani dokumenti te bira one koji su dobri. Ukoliko se radi o zaposleniku, on šalje skenirane dokumente revizoru. Revizor dokument prosljeđuje računovođi koji prosljeđuje direktoru na potpis. Korisnik ima obavijest kada mu netko pošalje dokument. Na kraju se dokument arhivira čime je proces završen.



Slika 4.5: Dijagram aktivnosti

## 4.5 Dijagram komponenti

Dijagram komponenti prikazuje organizaciju i međuovisnost komponenti te internu strukturu aplikacije. Frontend je pisan u HTML-u i CSS-u te je on povezan s aplikacijom. Baza podataka također komunicira sa aplikacijom te se iz nje uzimaju potrebni podaci pomoću SQL upita. ConditionChecker za svaku ulogu provjerava o kome se radi. Rest poslužuje aplikaciju te povezuje frontend s backendom. Uz to, komunicira s Controllers koji je dio MVC strukture. Potrebni podaci se šalju u bazu podataka, a za to je zaslužan "dao". Baza podataka se tako cijelo vrijeme ažurira prilikom ukoliko se nešto mijenja za vrijeme rada aplikacije.



Slika 4.6: Dijagram komponenti

## 5. Implementacija i korisničko sučelje

### 5.1 Korištene tehnologije i alati

Komunikacija u timu realizirana je korištenjem aplikacija Whatsapp<sup>1</sup> i Discord<sup>2</sup>. Pri izradu raznih dijagrama kao što su dijagram razreda, dijagram komponenti, dijagram razmještaja, sekvenvijski dijagram itd. korišten je alat Astah-UML<sup>3</sup>. Za upravljanje izvornim kodom korišten je Git<sup>4</sup>, a udaljeni repozitorij projekta nalazi se na platformi GitLab<sup>5</sup>. Kao razvojno okruženje korišten je Visual Studio Code<sup>6</sup> koji je napravio Microsoft. VSC se koristi pri izradi web-aplikacija, web-stranica, ali i mobilnih aplikacija. *Backend* naše web-aplikacije napisan je u programskom jeziku Java<sup>7</sup> u Intellij<sup>8</sup> razvojnom okruženju dok je *frontend* napisan u VSC-u pomoću HTML-a i CSS-a. Baza podataka nalazi se na PostgreSQL-u<sup>9</sup>, tj. korišten je pgAdmin<sup>10</sup> koji je dio PostgreSQL-a. Za *deploy* aplikacije korišten je servis Heroku<sup>11</sup> koji podržava Javu kao programski jezik. Tesseract OCR<sup>12</sup> smo koristili kako bi implementirali OCR.

---

<sup>1</sup><https://www.whatsapp.com/>

<sup>2</sup><https://discord.com/>

<sup>3</sup><https://astah.net/products/astah-uml/>

<sup>4</sup><https://git-scm.com/>

<sup>5</sup><https://about.gitlab.com/>

<sup>6</sup><https://code.visualstudio.com/>

<sup>7</sup><https://www.java.com/en/>

<sup>8</sup><https://www.jetbrains.com/idea/>

<sup>9</sup><https://www.postgresql.org/>

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

<sup>11</sup><https://www.heroku.com/>

<sup>12</sup><https://github.com/tesseract-ocr/tesseract>

## 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 razvojnom okruženju (prolaz/pad ispita).*

### 5.2.2 Ispitivanje sustava

*Potrebno je provesti i opisati ispitivanje sustava koristeći radni okvir Selenium<sup>13</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 programsko sučelje.*

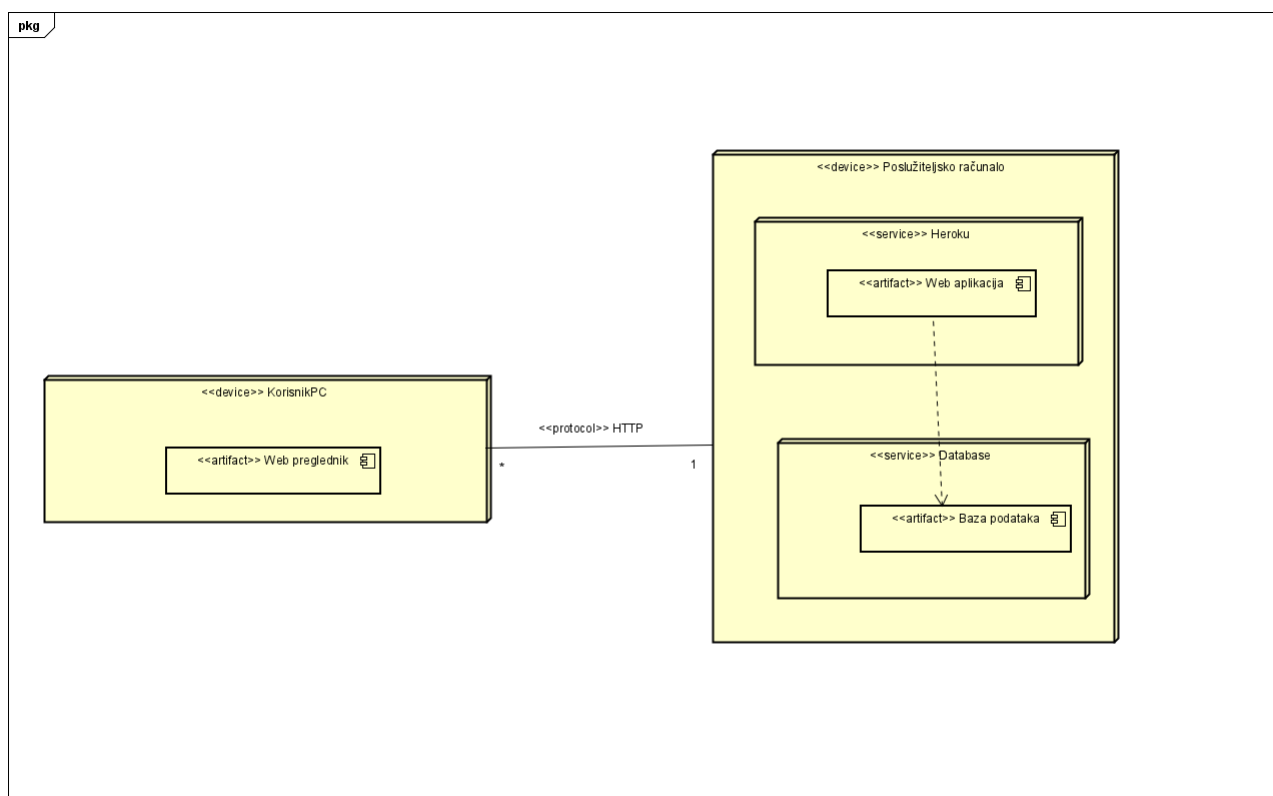
*Detalji o korištenju alata Selenium bit će prikazani na posebnom predavanju tijekom semestra.*

---

<sup>13</sup><https://www.seleniumhq.org/>

## 5.3 Dijagram razmještaja

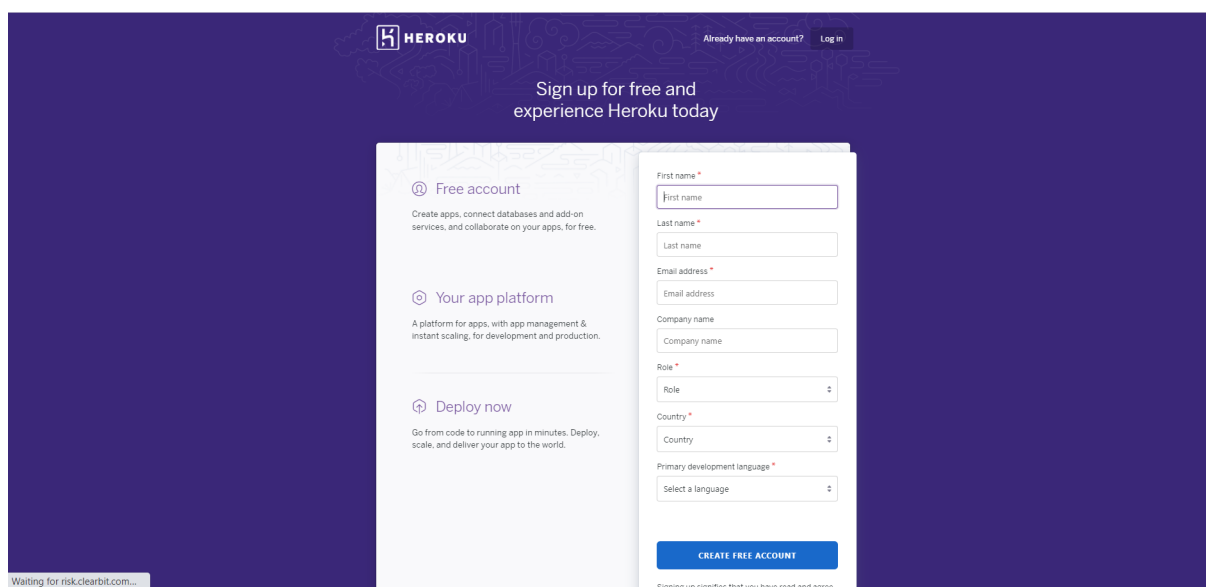
Dijagram razmještaja opisuje topologiju sklopovlja i programsku potporu koja se koristila u našoj aplikaciji. Postoji poslužiteljsko računalo na kojem se nalazi web poslužitelj i baza podataka koji su međusobno povezani. Korisnik sa HTTP protokolom preko svojeg računala sa nekim web preglednikom pristupa aplikaciji.



Slika 5.1: Dijagram razmještaja

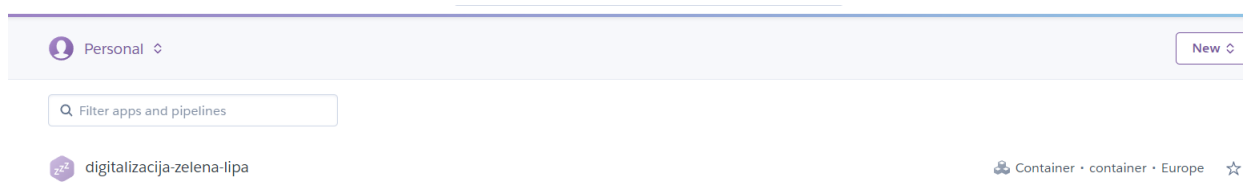
## 5.4 Upute za puštanje u pogon

Kada je gotov izvorni kod, može se prijeći na puštanje aplikacije u pogon. Aplikaciju smo pustili u pogon preko besplatne platforme za implementaciju aplikacije **Heroku**. Prvo je potrebno napraviti korisnički račun na web stranici : <https://www.heroku.com/>.



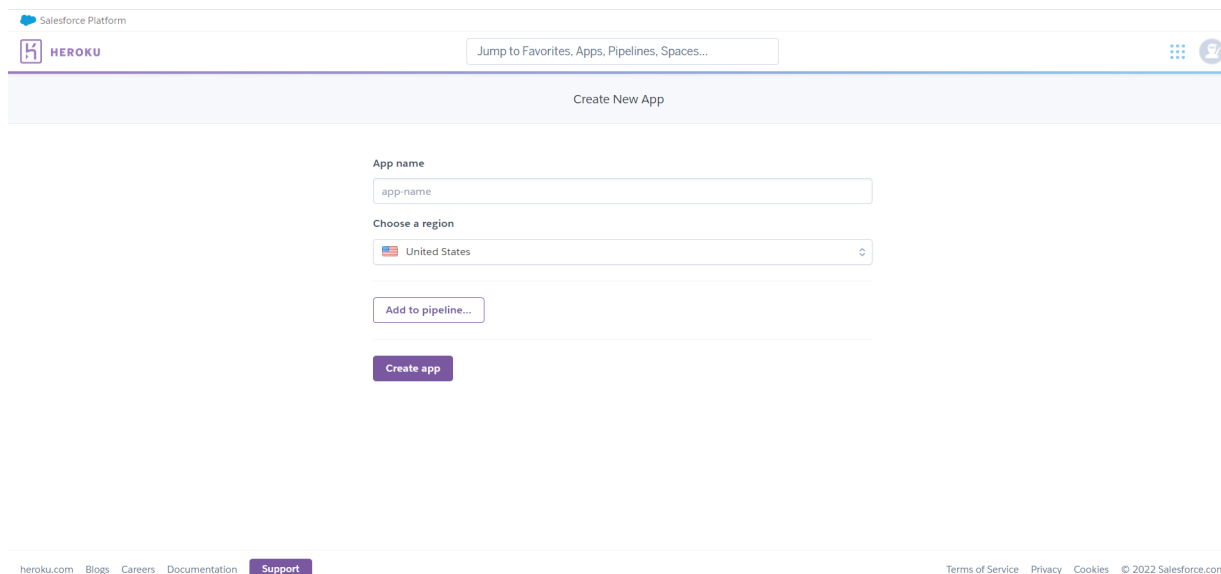
Slika 5.2: Sign up in Heroku

Nakon uspješnog kreiranja računa ulazi se u svoj račun, klikom odaberemo u gornjem desnom kutu tipku "New" iz koje će nam se ponuditi u izborniku "New app" i "New pipeline", mi odabiremo "New app".



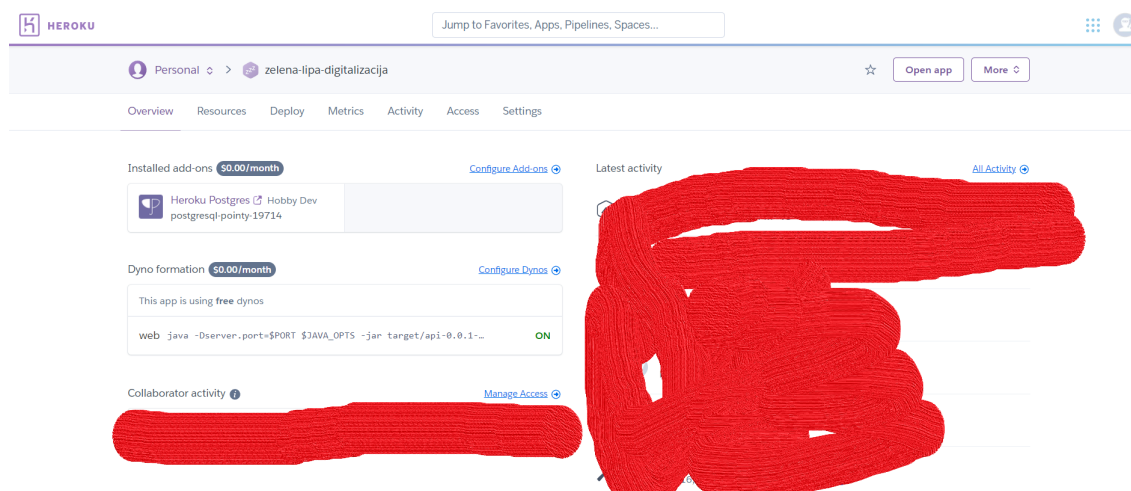
Slika 5.3: Početna stranica/ Odabir aplikacije

Nakon toga se na novom prozoru upisuje ime aplikacije i odabire se regija (preporučujemo Europa.)



Slika 5.4: Opis aplikacije

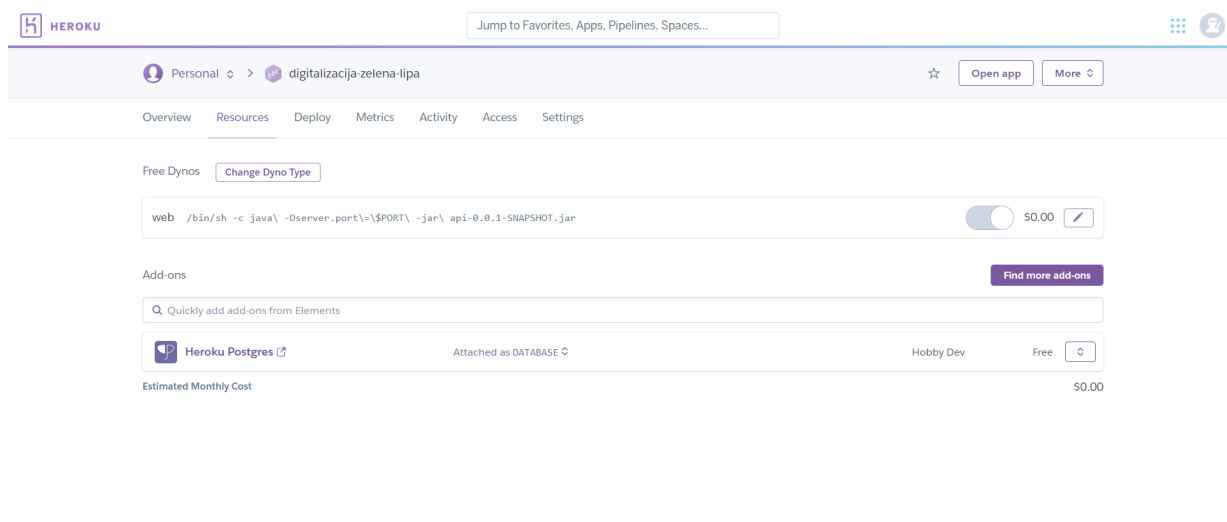
Nakon upisivanja potrebnih informacija, dolazi se na početni prozor.



Slika 5.5: Overview.

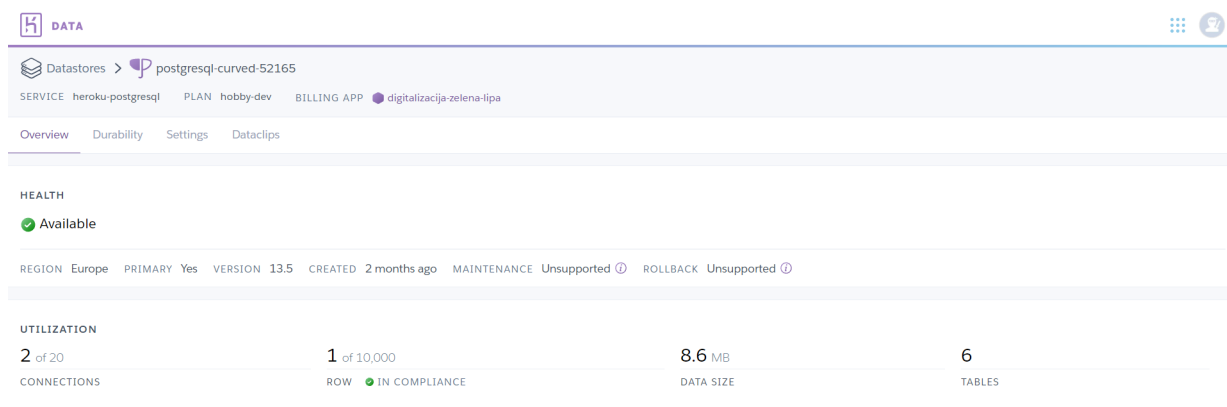
Ovdje se pruža puno mogućnosti, a trenutno ćemo se fokusirati na Add-ons i dodat ćemo novi (besplatni) add-on Heroku Postgre. Sada imamo bazu podataka za našu aplikaciju.

Sljedeće, na izborniku odabiremo Resources te pristupamo pojednostima naše baze podataka klikom na Heroku Postgres.



Slika 5.6: Resources.

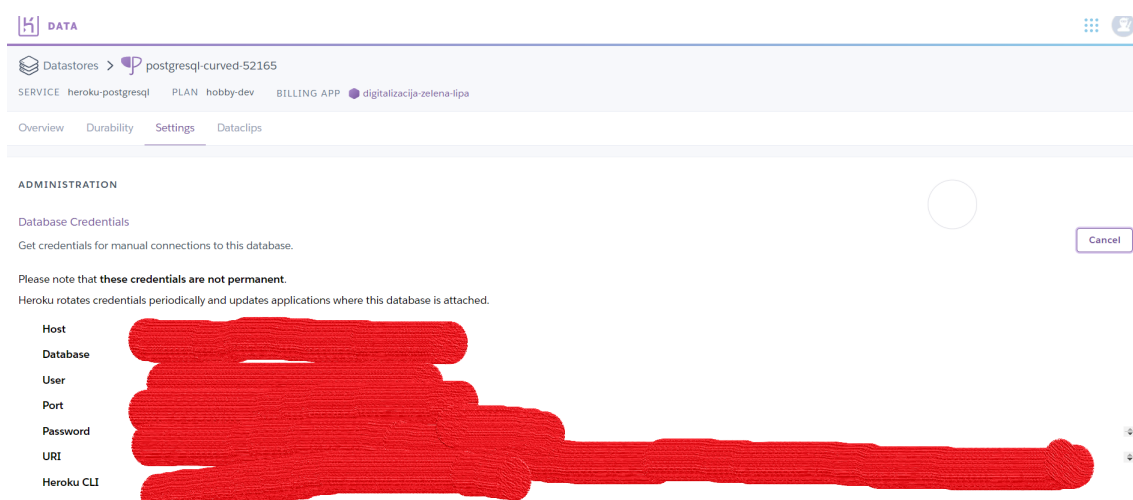
Zatim dolazimo do sljedećeg prozora:



Slika 5.7: Database overview.

Odavde idemo na Settings.



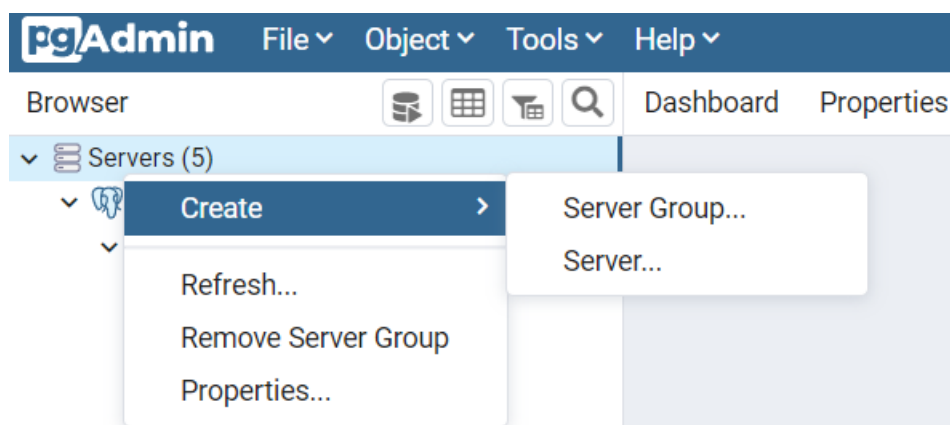


Slika 5.8: Database credentials.

U Database Credentials nam je prikazan user, pass, port, itd. Sve to će nam biti potrebno da bismo tu bazu mogli urediti i pripremiti za našu aplikaciju.

Sada nam je potreban **PGAdmin**, koji ćemo koristiti za pristupanje bazi podataka te unosenje. Instalacijski program se može preuzeti na ovom linku: <https://www.postgresql.org/download/> Potrebno je odabrati odgovarajuću verziju za OS vlastitog računala. Nakon što se preuzme instalacijski program, potrebno ga je pokrenuti. Sada treba slijediti korake instalacije, odabir instalacijskog direktorija, odabrati komponente (preporučujemo da se označe sve 4), direktorij za podatke, lozinku (po želji), port (ostavite predložena 5432) i jezik. Možete pokrenuti instalaciju. Pri prvom pokretanju će se tražiti lozinka, koju je korisnik upisao pri instalaciji. Nakon verifikacije i ulaska u aplikaciju se već nalazi jedan napravljen server koji je lokalni te korisnik može graditi svoje baze podataka na njemu. Sada ćemo pažnju posvetiti tome kako se pristupa serveru koji drži našu bazu podataka za aplikaciju.

Desnim klikom na padajući izbornik "Servers" se prikaže:



Slika 5.9: Kreiranje novog servera.

Odabiremo Server i prelazimo na sljedeći prozor.

Dolazimo do novog prozora "General" u kojem je potrebno proizvoljno nazvati server (u našem primjeru Server-alfa)

**Create - Server** [X]

General | Connection | SSL | SSH Tunnel | Advanced

**Name** [Warning Icon]

Server group:

Background: ☐

Foreground: ☐

Connect now?: ☒

Comments:

[Warning Icon] Name must be specified. [X]

[i] [?] [Cancel] [Reset] [Save]

Slika 5.10: Server-General.

Nakon toga prelazimo na "Connection".

**Create - Server** [X]

General **Connection** SSL SSH Tunnel Advanced

Host name/address [Red warning icon] [Empty field]

Port [5432]

Maintenance database [postgres]

Username [postgres]

Password [Empty field]

Save password? ☐

Role [Empty field]

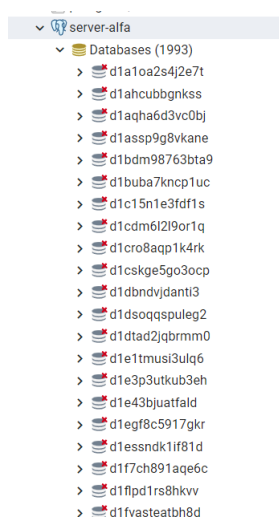
Service [Empty field]

[Red warning icon] Name must be specified. [Red X]

[i] [?] [X Cancel] [Reset] [Save]

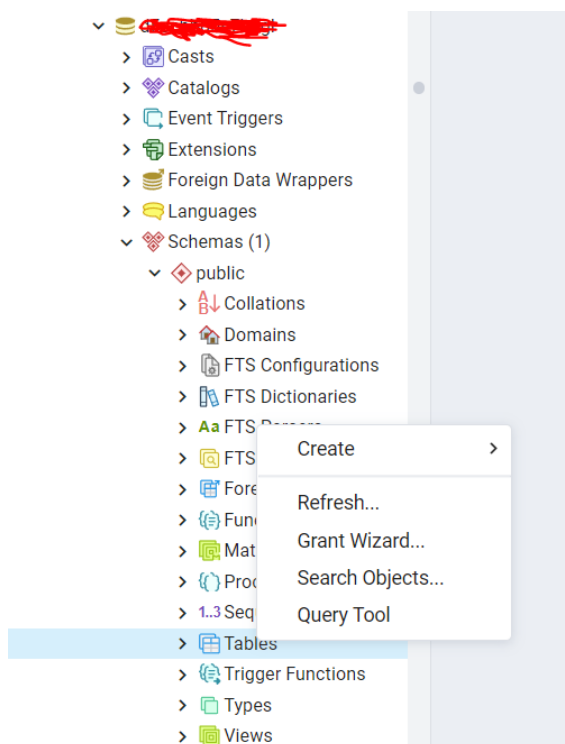
Slika 5.11: Server-Connection.

U novom prozoru je potrebno upisati sve što nam je ponuđeno na slici "Database credentials". Pritiskom na gumb "Save" stvorili smo server na našem PGAdminu te sada imamo pristup bazi podataka za našu heroku aplikaciju.



Slika 5.12: Izbornik bazi podataka.

Ponudeno je puno bazi podataka, ali pristup je dozvoljen samo bazi za našu aplikaciju.



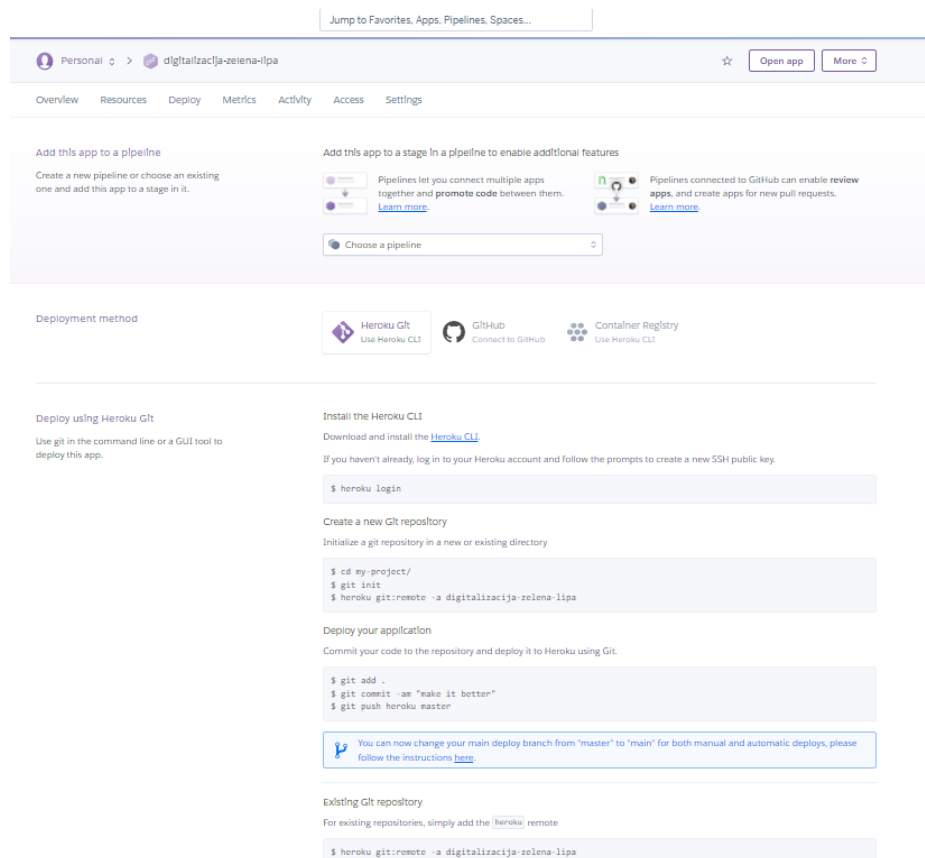
Slika 5.13: Izbornik naše baze podataka.

Odaberemo sad Query tool kao što je prikazano na Slici. Otvorit će se prostor za uređivanje teksta u kojem se upisuje kod za stvaranje tablica, umetanje podataka u te tablice i sve drugo što je potrebno za uređivanje baze. Pokretanjem koda se

gradi baza i spremna je za korištenje.

Sada kada smo pripremili bazu potrebno je završni kod staviti na git od naše heroku aplikacije, vraćamo se ponovno na naš račun.

Odabiremo Deploy.



Slika 5.14: Deploy.

Ovdje se nalazi uputa od preuzimanju Heroku CLI i opis kako klonirati git repozitorij te kako staviti završni kod na repozitorij. Naredbom git push heroku master vaš kod se stavlja na repozitorij i automatski pokreće te će vam u terminalu pisati link na kojem je vaša web aplikacija, kojoj možete pristupiti sa svog browsera i početi koristiti.

**Dodatak:**

U našoj aplikaciji, heroku nije mogao podržati funkcionalnost OCR (Tesseract) dokumenata. Pokušali smo s build-packageom za Tesseract koji je ponuđen u Heroku, što nije pomoglo. Iako bi sve funkcioniralo lokalno, nikako nismo mogli namjestiti da heroku pronađe knjižnice (engl. library) za Tesseract OCR. Da bi ipak omogućili našoj aplikaciji da izvodi OCR morali smo dodati docker container (<https://devcenter.heroku.com/articles/build-docker-images-heroku-yml>) koji je osigurao da su potrebne knjižnice za OCR instalirane te da im naša aplikacija može pristupiti.

## 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. 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

3.1	Dijagram obrasca uporabe, funkcionalnost direktora i računovođe .	17
3.2	Dijagram obrasca uporabe, funkcionalnost registriranih korisnika . .	18
3.3	UC1 - Prijava korisnika . . . . .	19
3.4	UC3 - Prilaganje dokumenta . . . . .	20
3.5	UC5 Pregled - povijesti skeniranih dokumenata . . . . .	21
3.6	UC11 - Registracija računa . . . . .	22
4.1	Relacijski model baze podataka . . . . .	28
4.2	Dijagram razreda - Controller dio . . . . .	29
4.3	Dijagram razreda . . . . .	30
4.4	Dijagram stanja . . . . .	31
4.5	Dijagram aktivnosti . . . . .	32
4.6	Dijagram komponenti . . . . .	33
5.1	Dijagram razmještaja . . . . .	36
5.2	Sign up in Heroku . . . . .	37
5.3	Početna stranica/ Odabir aplikacije . . . . .	37
5.4	Opis aplikacije . . . . .	38
5.5	Overview. . . . .	38
5.6	Resources. . . . .	39
5.7	Database overview. . . . .	39
5.8	Database credentials. . . . .	40
5.9	Kreiranje novog servera. . . . .	41
5.10	Server-General. . . . .	42
5.11	Server-Connection. . . . .	43
5.12	Izbornik bazi podataka. . . . .	44
5.13	Izbornik naše baze podataka. . . . .	44
5.14	Deploy. . . . .	45

# Dodatak: Prikaz aktivnosti grupe

## Dnevnik sastajanja

### *Kontinuirano osvježavanje*

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

#### 1. sastanak

- Datum: u ovom formatu: 14. siječnja 2022.
- Prisustvovali: I.Prezime, I.Prezime
- Teme sastanka:
  - opis prve teme
  - opis druge teme

#### 2. sastanak

- Datum: u ovom formatu: 14. siječnja 2022.
- Prisustvovali: I.Prezime, I.Prezime
- Teme sastanka:
  - opis prve teme
  - opis druge teme

## Tablica aktivnosti

### *Kontinuirano osvježavanje*

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

	Ime Prezime voditelja	Ime Prezime	Ime Prezime	Ime Prezime	Ime Prezime	Ime Prezime	Ime Prezime
Upravljanje projektom							
Opis projektnog zadatka							
Funkcionalni zahtjevi							
Opis pojedinih obrazaca							
Dijagram obrazaca							
Sekvencijski dijagrami							
Opis ostalih zahtjeva							
Arhitektura i dizajn sustava							
Baza podataka							
Dijagram razreda							
Dijagram stanja							
Dijagram aktivnosti							
Dijagram komponenti							
Korištene tehnologije i alati							
Ispitivanje programskog rješenja							

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

	Ime Prezime voditelja	Ime Prezime	Ime Prezime	Ime Prezime	Ime Prezime	Ime Prezime	Ime Prezime
Dijagram razmještaja							
Upute za puštanje u pogon							
Dnevnik sastajanja							
Zaključak i budući rad							
Popis literature							
<i>Dodatne stavke kako ste podijelili izradu aplikacije</i>							
<i>npr. izrada početne stranice</i>							
<i>izrada baze podataka</i>							
<i>spajanje s bazom podataka</i>							
<i>back end</i>							

## Dijagrami pregleda promjena

### *dio 2. revizije*

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