

Programsko inženjerstvo

Ak. god. 2023./2024.

Digitalizacija

Dokumentacija, Rev. 2

Grupa: *Kompletići*

Voditelj: *Martin Ante Rogošić*

Datum predaje: 19. 1. 2024.

Nastavnik: *Igor Stančin*

Sadržaj

1 Dnevnik promjena dokumentacije	3
2 Opis projektnog zadatka	5
2.1 Potencijalna korist ovog projekta	7
2.2 Slična rješenja	7
2.3 Mogućnost prilagodbe	7
2.4 Opseg projektnog zadatka	8
2.5 Moguće nadogradnje	8
3 Specifikacija programske potpore	9
3.1 Funkcionalni zahtjevi	9
3.1.1 Obrasci uporabe	11
3.1.2 Sekvencijski dijagrami	21
3.2 Ostali zahtjevi	24
4 Arhitektura i dizajn sustava	25
4.1 Baza podataka	26
4.1.1 Opis tablica	26
4.1.2 Dijagram baze podataka	29
4.2 Dijagram razreda	30
4.3 Dijagram stanja	33
4.4 Dijagram aktivnosti	34
4.5 Dijagram komponenti	35
5 Implementacija i korisničko sučelje	37
5.1 Korištene tehnologije i alati	37
5.2 Ispitivanje programskog rješenja	38
5.2.1 Ispitivanje komponenti	38
5.2.2 Ispitivanje sustava	38
5.3 Dijagram razmještaja	39
5.4 Upute za puštanje u pogon	40

6 Zaključak i budući rad	43
Popis literature	44
Indeks slika i dijagrama	45

1. Dnevnik promjena dokumentacije

Kontinuirano osvježavanje

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Napravljen predložak.	Martin Ante Rogošić	7.11.2023.
0.2	Dodan opis projektnog zadatka	Martin Ante Rogošić	9.11.2023.
0.3	Dodani svi obrasci uporabe.	Martin Ante Rogošić	16.11.2023.
0.4	Dodani dijagrami i arhitektura sustava.	Luka Panda, Emanuel Njegovec, Dominik Pavelić	17.11.2023.
1.0	Gotova prva revizija dokumentacije.	Svi	17.11.2023.
1.1	Dovršena arhitektura i započeta implementacija	Martin Ante Rogošić	15.1.2023.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodatka	Autori	Datum
1.2	Dovršena implementacija i zaključak	Martin Ante Rogošić i Luka Panda	18.1.2024.
2.0	Dovršena druga verzija dokumentacije.	Martin Ante Rogošić	19.1.2024.

2. Opis projektnog zadatka

Cilj ovog projekta je razvoj i implementacija aplikacije Digitaliziraj, koja će tvrtkama olakšati digitalizaciju dokumenata nužnih za poslovanje koristeći OCR (eng. Optical character recognition). Aplikacija će olakšati digitalizaciju dokumenata unutar organizacija, te osigurati da svaki radnik organizacije dobije dokumente za koje je zadužen. Time, aplikacija će povećati učinkovitost organizacije te ubrzati njeno djelovanje.

Neregistrirani korisnik mora napraviti korisnički račun kako bi mogao koristiti funkcionalnosti aplikacije. Za to su mu potrebni sljedeći podatci:

- *ime*
- *prezime*
- *email adresa*
- *lozinka*

Dodatno, ne registrirani korisnik mora odabrati na kojoj razini ovlasti želi stvoriti korisnički račun. Dostupne su 4 razine ovlasti:

- *zaposlenik*
- *revizor*
- *računovođa*
- *direktor*

Razne razine iznad su navedene od najniže do najviše. U većini slučajeva viša razina ovlasti ima sve funkcionalnosti onih ispod sebe, osim u onim slučajevima kada struktura tvrtke te razne odgovornosti zaposlenika to traže drugačije. Detalji o funkcionalnostima svake razine ovlasti nalaze se u nastavku dokumenta. Dodatno postoji još razina ovlasti direktor, koja predstavlja razinu ovlasti vlasnika tvrtke. Aplikacija sama stvara jedan korisnički račun s tom razinom ovlasti te su podatci potrebni za prijavu u taj korisnički račun dani vlasniku tvrtke.

Zaposlenik ima najnižu razinu ovlasti unutar sustava. Korisnik s razinom ovlasti zaposlenik može učitavati slike u sustav te provjeriti je li sustav točno odradio konverziju slike u dokument. Jednom kad zaposlenik potvrdi da je pretvorba iz slike u dokument odrađena ispravno, generiran dokument se šalje jednom od revizora tvrtke na pregled. U svrhu ubrzanja procesa aplikacija omogućuje učitavanje do 50 slika istovremeno (ova funkcionalnost omogućena je i za sve više razine ovlasti). Pri tom zaposlenik i dalje mora svaku od konverzija potvrditi kao točnu prije nego što je dokument proslijeđen revizoru na pregled. Zaposlenik ima pristup povijesti svih dokumenata koje je skenirao.

Revizor je druga najniža razina ovlasti unutar sustava. Posao revizora jest kontrola rada zaposlenika te preusmjeravanje dokumenata odgovarajućem računovođi (ovisno o vrsti dokumenta). Kao takav revizor ima pristup povijesti svih dokumenata nad kojima je vršio kontrolu. U slučaju da revizor skenira dokument aplikacija će automatski detektirati računovođu kojem vrsta skeniranog dokumenta treba biti poslana, te će nakon što revizor potvrdi ispravnost konverzije taj dokument biti automatski poslan odgovarajućem računovođi.

Računovođa je najviša razina ovlasti u sustavu, izuzev direktora. Svaki računovođa je zadužen za obradu jedne od 3 vrste dokumenata koje aplikacija raspoznaje. To su računi, ponude i interni dokumenti. Računovođa ima pristup povijesti svih dokumenata one vrste za koju je zadužen. Posao računovođe je arhivirati dokumente. Pri arhiviranju dokumentu se dodjeljuje jedinstven broj arhiva. Računovođa također po potrebi može poslati dokument direktoru na potpisivanje. U tom slučaju direktor prima obavijest o tome. Nakon što direktor potpiše dokument računovođa dobiva obavijest da je dokument potpisan te ga onda može arhivirati. U slučaju da računovođa učitava sliku dokumenta one vrste za koju nije zadužen aplikacija će mu automatski ponuditi slanje dokumenta odgovarajućem računovođi.

Direktor je najviša razina ovlasti unutar sustava, te predstavlja najodgovorniju osobu unutar organizacije koja koristi aplikaciju za svoje poslovanje. Direktor ima uvid u povijest svih dokumenata te u statistike svih zaposlenika. U slučaju da mu računovođa pošalje zahtjev za potpis može potpisati dokument. U slučaju da direktor sam učitava slike u sustav te vrši konverziju, odmah mu se nudi mogućnost potpisivanja dokumenta te aplikacija automatski određuje računovođu kojem treba proslijediti

učitan dokument. Dodatno za promotivne svrhe, direktoru je omogućena objava dokumenata na sljedećim društvenim mrežama:

- Facebook
- X
- Instagram

2.1 Potencijalna korist ovog projekta

Svojom strukturom i funkcionalnošću aplikacija ostvaruje brz i učinkovit sustav digitalizacije i distribucije dokumenata unutar organizacije. Njenom primjenom može se osigurati učinkovito poslovanje te raspodjela odgovornosti među zaposlenicima što pospješuje rad organizacije u mnogim aspektima. S obzirom na njenu općenitost aplikacija bi mogla biti od interesa svim organizacijama koje se moraju baviti papirologijom. To uključuje: neprofitne humanitarne organizacije, tvrtke koje se natječu na tržištu, vladine agencije...

2.2 Slična rješenja

Na tržištu postoje razni sustavi za distribuciju dokumenata, također postoje i mnoge implementacije OCR-a. No vrlo je malen broj alata koji integriraju te dvije tehnologije u jedan sustav. Time se zaobilazi potreba da se odvojeni alati koriste za digitalizaciju dokumenata i njihovu distribuciju.

2.3 Mogućnost prilagodbe

Implementacija sustava u velikoj mjeri ovisi o strukturi organizacije i njenim potrebama za distribuciju dokumenata. Dodatno ako je klasifikacija dokumenata koju organizacija koristi složena, potrebno je doraditi sustav automatskog prosljeđivanja dokumenata. Međutim implementacija dijelova sustava, koji se ne moraju mijenjati ovisno o potrebi klijenta, mogu se izvesti na način da je ta promjena relativno jednostavna.

2.4 Opseg projektnog zadatka

Konkretna implementacija izvedena u sklopu ovog projekta koristi relativno jednostavnu strukturu organizacije opisanu iznad, te razlikuje 3 različite vrste dokumenata i razina ovlasti. To je čini relativno jednostavnom za implementaciju no ujedno i lakom za integrirati u radni tok bilo koje organizacije koja koristi aplikaciju.

2.5 Moguće nadogradnje

Mnoge nadogradnje su moguće na sustav. Integracija s alatima za preuređivanje digitalnih dokumenata, kako bi sustav postao znatno općenitije sredstvo za rukovanje dokumentima. Dodatno mogu se implementirati značajke koje bi omogućile komunikaciju među zaposlenicima, generaciju raznih rasporeda i slično. Time bi aplikacija postala univerzalno sredstvo za upravljanje radom organizacije.

3. Specifikacija programske potpore

3.1 Funkcionalni zahtjevi

Dionici:

1. Neregistrirani korisnici
2. Zaposlenici
3. Revizori
4. Računovođe
5. Direktor
6. Razvojni tim

Aktori i njihovi funkcionalni zahtjevi:

1. Neregistrirani korisnik (inicijator/sudionik) može:
 - (a) Registrirati se u sustav
2. Zaposlenik (inicijator/sudionik) može:
 - (a) Prijaviti se u sustav
 - (b) Učitati fotografiju u sustav te izvršiti konverziju te slike u dokument
 - (c) Učitati više fotografija od jednom u sustav te izvršiti njihovu konverziju u dokumente od jednom
 - (d) Potvrditi ili odbiti točnost konverzije dokumenata u sustav
 - (e) Poslati dokument revizoru na pregled
 - (f) Pregledavati povijest svih dokumenata koje je zaposlenik unio u sustav
3. Revizor (inicijator/sudionik) može:
 - (a) Provjeriti valjanost dokumenta kojeg mu je poslao zaposlenik
 - (b) Učitati fotografiju u sustav te izvršiti konverziju te slike u dokument
 - (c) Učitati više fotografija od jednom u sustav te izvršiti njihovu konverziju u dokumente od jednom
 - (d) Potvrditi ili odbiti točnost konverzije dokumenata učitanih u sustav
 - (e) Potvrditi ili ispraviti automatsku kategorizaciju dokumenta

- (f) Proslijediti dokument računovođi odgovornom za vrstu dokumenta kojoj dokument pripada
- (g) Pregledati povijest svih dokumenata koje je revizor provjerio ili unio u sustav

4. Računovođa (inicijator/sudionik) može:

- (a) Računovođa može učitati jednu ili više slika u sustav te izvršiti njihovu konverziju u dokumente
- (b) Potvrditi ili ispraviti automatsku kategorizaciju dokumenta
- (c) Poslati dokumente direktoru na potpisivanje
- (d) Arhivirati dokumente
- (e) Pregledati povijest svih dokumenata one vrste za koju je računovođa zadužen

5. Direktor (inicijator/sudionik) može:

- (a) Potpisati dokument kojeg mu je poslao računovođa na potpisivanje
- (b) Učitati jednu ili više fotografija u sustav te izvršiti njihovu konverziju u dokumente
- (c) Potvrditi ili odbiti točnost konverzije dokumenata unesenih u sustav
- (d) Potvrditi ili ispraviti automatsku kategorizaciju dokumenta
- (e) Proslijediti dokument računovođi odgovornom za vrstu dokumenta kojoj dokument pripada

3.1.1 Obrasci uporabe

Opis obrazaca uporabe

UC1 -Registracija

- **Glavni sudionik:** Neregistrirani korisnik
- **Cilj:** Registracija korisnika u sustav
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Neregistrirani korisnik odabire opciju za registraciju
 2. Neregistrirani korisnik unosi podatke za registraciju
 3. Korisnik prima obavijest o uspješnoj registraciji
- **Opis mogućih odstupanja:**
 - 2.a Korisnik je odabrao već zauzetu ili neispravnu e-mail adresu, ili je dao ne dovoljno sigurnu lozinku
 1. Neregistriranog korisnika se vraća na stranicu za registraciju te ga se obavještava o neuspješnoj registraciji
 2. Korisnik mijenja nepravilne podatke ili odustaje od registracije

UC2 - Prijava u sustav

- **Glavni sudionik:** Korisnik sustava
- **Cilj:** Prijava u sustav te pristup korisničkom sučelju
- **Sudionici:** Baza podataka
- **Preduvjet:** Registracija
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju za prijavu
 2. Korisnik unosi potrebne korisničke podatke
 3. Korisnik dobiva pristup korisničkom sučelju
- **Opis mogućih odstupanja:**
 - 2.a Korisnik je upisao nepostojeće ili pogrešne korisničke podatke
 1. Korisnika se vraća na stranicu za prijavu te ga se obavještava o neuspjeloj prijavi

UC3 -Promjena lozinke

- **Glavni sudionik:** Korisnik

- **Cilj:** Promjena lozinke koja se koristi pri prijavi u sustav
- **Sudionici:** Baza podataka
- **Preduvjet:** Registracija
- **Opis osnovnog tijeka:**
 1. Korisnik bira opciju za promjenu lozinke
 2. Unosi staru lozinku kao potvrdu svog identiteta
 3. Bira novu lozinku te je upisuje dva puta
 4. Korisnik je obaviješten o uspješnoj promjeni lozinke
- **Opis mogućih odstupanja:**
 - 2.a Korisnik je unio pogrešnu staru lozinku
 1. Korisnika se obavještava o pogrešnom unosu lozinke te ga se vraća na stranicu za promjenu lozinke
 2. Korisnik ponovno unosi staru lozinku ili odustaje od promjene lozinke

UC4 - Skeniranje jedne fotografije

- **Glavni sudionik:** Korisnik
- **Cilj:** Unijeti fotografiju u sustav te izvršiti njenu konverziju u dokument
- **Sudionici:** Baza podataka
- **Preduvjet:** Prijava u sustav
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju za učitavanje jedne fotografije
 2. Korisnik odabire fotografiju iz datotečnog sustava svog računala
 3. Korisnik dobiva obavijest o provedenoj konverziji
- **Opis mogućih odstupanja:**
 - 2.a Korisnik je odabrao datoteku koja ne postoji na njegovom računalu
 1. Sustav obavještava korisnika o nastaloj pogrešci te ga preusmjerava na korisničko sučelje

UC5 -Skeniranje više fotografija

- **Glavni sudionik:** Korisnik
- **Cilj:** Unijeti više fotografija u sustav te izvršiti njihovu konverziju u dokumente
- **Sudionici:** Baza podataka
- **Preduvjet:** Prijava u sustav
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju za učitavanje više dokumenata
 2. Korisnik odabire više datoteka iz svog datotečnog sustava

3. Korisnik dobiva obavijest o provedenim konverzijama
- **Opis mogućih odstupanja:**
 - 2.a Jedna od fotografija koje je korisnik unio ne postoji u datotečnom sustavu
 1. Sustav provodi konverziju pronađenih dokumenata
 2. Sustav šalje korisniku obavijest o ne pronađenim dokumentima

UC6 -Potvrda ispravnosti konverzije

- **Glavni sudionik:** Korisnik
- **Cilj:** Potvrditi točnost konverzije dokumenta unesenih u sustav
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen te je učitao jednu ili više fotografija u sustav
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju za potvrdu ispravnosti konverzije
 2. Korisniku se prikazuju fotografija i dokument generiran iz slike paralelno na ekranu, te mu se nudi opcija za prihvatanje ili odbijanje točnosti konverzije

UC7 -Prosljeđivanje dokumenata

- **Glavni sudionik:** Korisnik
- **Cilj:** Proslijediti dokument potvrđene točnosti drugom korisniku s prikladnom razinom ovlasti za daljnju obradu dokumenta
- **Sudionici:** Baza podataka i drugi korisnici s višom razinom ovlasti
- **Preduvjet:** Korisnik je unio u sustav fotografiju, izvršio konverziju i potvrdio njenu točnost
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju za slanje dokumenta
 2. Sustav prosljeđuje dokument odgovarajućem korisniku
 3. Korisnik dobiva obavijest o prosljeđenosti dokumenta

UC8 -Pregled vlastite povijesti

- **Glavni sudionik:** Zaposlenik
- **Cilj:** Pregled svih dokumenata koje je zaposlenik unio u sustav
- **Sudionici:** Baza podataka
- **Preduvjet:** Prijava
- **Opis osnovnog tijeka:**
 1. Zaposlenik odabire opciju za pregled povijesti

2. Aplikacija prikazuje korisniku listu svih dokumenata koje je unio u sustav
3. Zaposlenik odabire dokument iz liste
4. Aplikacija korisniku prikazuje dokument i sliku iz koje je dokument generiran

UC9 -Verifikacija točnosti konverzije

- **Glavni sudionik:** Korisnik
- **Cilj:** Potvrditi točnost generiranog dokumenta temeljem slike iz koje je dokument generiran
- **Sudionici:** Baza podataka
- **Preduvjet:** Skeniranje jedne ili više fotografija
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju za potvrdu konverzije
 2. Aplikacija korisniku prikazuje listu dokumenata čiju je točnost konverzije potrebno potvrditi
 3. Korisnik odabire dokument s liste, te mu aplikacija prikazuje dokument i sliku iz koje je dokument generiran. Dodatno aplikacija nudi mogućnost potvrde ili odbijanja točnosti konverzije
 4. Odabir korisnika o točnosti (ili netočnosti) konverzije pohranjuje se u bazu podataka

UC10 -Automatska kategorizacija dokumenata

- **Glavni sudionik:** Revizor ili Računovođa ili Direktor
- **Cilj:** Automatsko prepoznavanje vrste dokumenta pri konverziji dokumenta iz slike
- **Sudionici:** Baza podataka
- **Preduvjet:** Skeniranje jedne ili više fotografija
- **Opis osnovnog tijeka:**
 1. Revizor ili Računovođa ili Direktor unose jednu ili više fotografija u sustav
 2. Sustav nakon konverzije automatski određuje kategoriju dokumenta
 3. Pri verifikaciji točnosti konverzije korisnicima se omogućuje i verifikacija točnosti automatske kategorizacije
 4. Nakon što je točnost automatske kategorizacije dokumenta potvrđena, kategorija dokumenta pohranjuje se u bazu podataka

UC11 -Slanje dokumenata na potpis

- **Glavni sudionik:** Računovođa
- **Cilj:** Slanje dokumenta Direktor na potpis
- **Sudionici:** Baza podataka, Direktor
- **Preduvjet:** Prisutnost jednog ili više dokumenata koje treba obraditi u sustavu
- **Opis osnovnog tijeka:**
 1. Računovođa bira jedan dokument iz liste dokumenata koje treba obraditi
 2. Računovođa odabire opciju za slanje dokumenta za potpis
 3. Dokument je privremeno uklonjen iz liste dokumenata koje treba obraditi dok direktor ne potpiše dokument

UC12 -Arhiviranje

- **Glavni sudionik:** Računovođa
- **Cilj:** Arhiviranje dokumenata
- **Sudionici:** Baza podataka
- **Preduvjet:** Prisutnost jednog ili više dokumenata koji su spremni za arhiviranje
- **Opis osnovnog tijeka:**
 1. Računovođa bira jedan ili više dokumenata iz liste dokumenata koje treba obraditi
 2. Aplikacija nudi mogućnost za arhiviranje jednog ili više dokumenata
 3. Odabirom opcije za arhiviranje svi odabrani dokumenti dobivaju jedinstven broj arhiva
 4. Baza podataka pohranjuje jedinstven broj arhiva za svaki arhiviran dokument

UC13 -Pregled svih dokumenata određene kategorije

- **Glavni sudionik:** Računovođa
- **Cilj:** Pregled svih dokumenata određene kategorije koje su svi korisnici unijeli u sustav
- **Sudionici:** Baza podataka
- **Preduvjet:** Prisutnost jednog ili više dokumenata pretraživane kategorije u sustavu
- **Opis osnovnog tijeka:**
 1. Računovođa odabire opciju za pregled povijesti svih dokumenata određene kategorije
 2. Aplikacije prikazuje računovođi listu svih dokumenata određene kategorije u sustavu

3. Računovođa odabire dokument iz liste
4. Aplikacija prikazuje računovođi dokument i sliku iz koje je dokument generiran

UC14 -Potpisivanje dokumenta

- **Glavni sudionik:** Direktor
- **Cilj:** Potpisivanje određenog dokumenta
- **Sudionici:** Baza podataka
- **Preduvjet:** Prisutnost jednog ili više dokumenata u sustavu za koje je zatražen potpis
- **Opis osnovnog tijeka:**
 1. Direktor odabire opciju za potpisivanje dokumenata
 2. Aplikacija direktoru prikazuje listu dokumenata za koje je zatražen potpis
 3. Odabirom dokumenta aplikacija nudi direktoru opciju za potpis
 4. Direktor potpisuje dokument ili odbija potpisati dokument

UC15 - Promjena kategorije dokumenta

- **Glavni sudionik:** Računovođa, Direktor, Revizor
- **Cilj:** Promjena kategorije dokumenta kojeg su u sustav unijeli računovođa ili direktor
- **Sudionici:** Baza podataka
- **Preduvjet:** Prisutnost jednog ili više dokumenata u sustavu čija je ispravnost konverzije ispravna no automatska kategorizacija nije
- **Opis osnovnog tijeka:**
 1. Direktor, računovođa ili revizor odabiru opciju za potvrdu ispravnosti automatske kategorizacije dokumenta
 2. Aplikacija prikazuje fotografiju i dokument generiran iz fotografije te mu se nudi opcija za potvrdu točnosti ili promjenu kategorije dokumenta
 3. Baza podataka pohranjuje odabir korisnika

UC16 -Objava dokumenata na društvenoj mreži

- **Glavni sudionik:** Direktor
- **Cilj:** Objava dokumenata na društvenoj mreži
- **Sudionici:** Baza podataka
- **Preduvjet:** Prisutnost dokumenata u sustavu
- **Opis osnovnog tijeka:**

1. Direktor odabire opciju za objavu dokumenta na društvenoj mreži
2. Direktor odabire na kojoj od ponuđenih društvenih mreža želi objaviti dokument
3. Direktor unosi potrebne podatke za objavu dokumenta koje traži specifična društvena mreža
4. Dokument se objavljuje na društvenoj mreži

UC17 -Pregled povijesti svih dokumenata

- **Glavni sudionik:** Direktor
- **Cilj:** Pregled povijesti svih dokumenata
- **Sudionici:** Baza podataka
- **Preduvjet:** Prisutnost dokumenata u sustavu
- **Opis osnovnog tijeka:**
 1. Direktor odabire opciju za pregled povijesti dokumenata
 2. Aplikacija direktoru prikazuje listu svih dokumenata ikad unesenih u sustav
 3. Odabirom dokumenta sa liste prikazuje se fotografija i dokument generiran iz fotografije

UC18 -Brisanje dokumenta iz Arhiva

- **Glavni sudionik:** Direktor
- **Cilj:** Brisanje dokumenta iz Arhiva
- **Sudionici:** Baza podataka
- **Preduvjet:** Prisutnost jednog ili više arhiviranih dokumenata u sustavu
- **Opis osnovnog tijeka:**
 1. Direktor odabire opciju za brisanje dokumenta iz arhiva
 2. Aplikacija prikazuje listu svih arhiviranih dokumenata, te direktor odabire onog kojeg želi učitati
 3. Direktor unosi svoju lozinku kao potvrdu svog identiteta i odluke
 4. Baza briše dokument iz sustava

UC19 -Pregled statistike zaposlenika

- **Glavni sudionik:** Direktor
- **Cilj:** Pregled statistike zaposlenika
- **Sudionici:** Baza podataka, korisnici

- **Preduvjet:** Postojanje jednog ili više korisnika s razinom ovlasti manjom od direktor u sustavu
- **Opis osnovnog tijeka:**
 1. Direktor odabire opciju za pregled statistike zaposlenika
 2. Aplikacija nudi listu svih korisnika
 3. Direktor odabire zaposlenika čije statistike želi pogledati
 4. Aplikacija prikazuje tražene podatke

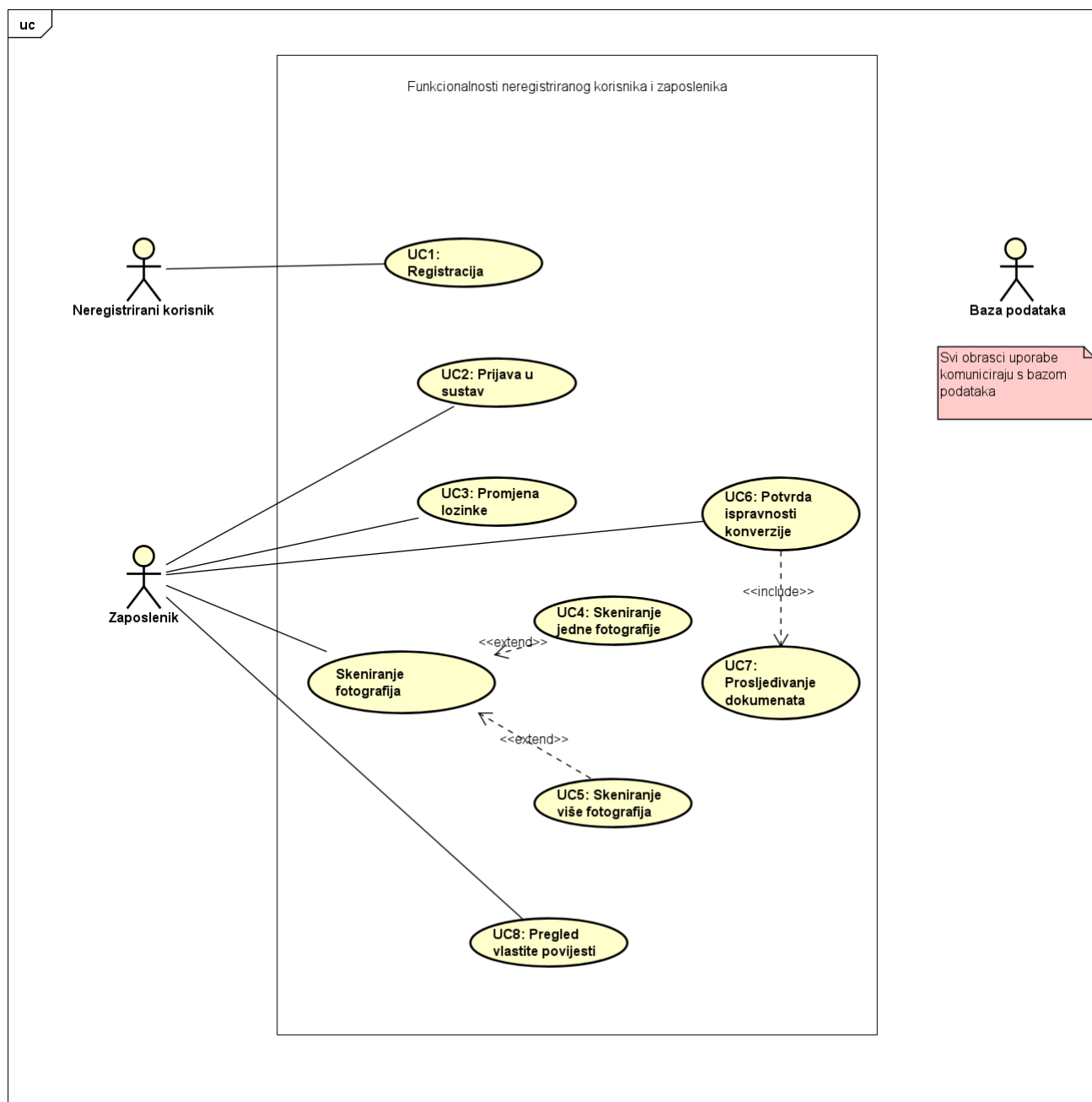
UC20 -Promjena razine ovlasti

- **Glavni sudionik:** Direktor
- **Cilj:** Promjena razine ovlasti korisnika
- **Sudionici:** Baza podataka, korisnici
- **Preduvjet:** Postojanje jednog ili više korisnika s razinom ovlasti manjom od direktor u sustavu
- **Opis osnovnog tijeka:**
 1. Direktor odabire opciju za promjenu razine ovlasti korisnika
 2. Sustav prikazuje direktoru listu svih korisnika sustava
 3. Direktor odabire zaposlenika te mu dodjeljuje novu razinu ovlasti
 4. Baza podataka sprema unesene promjene

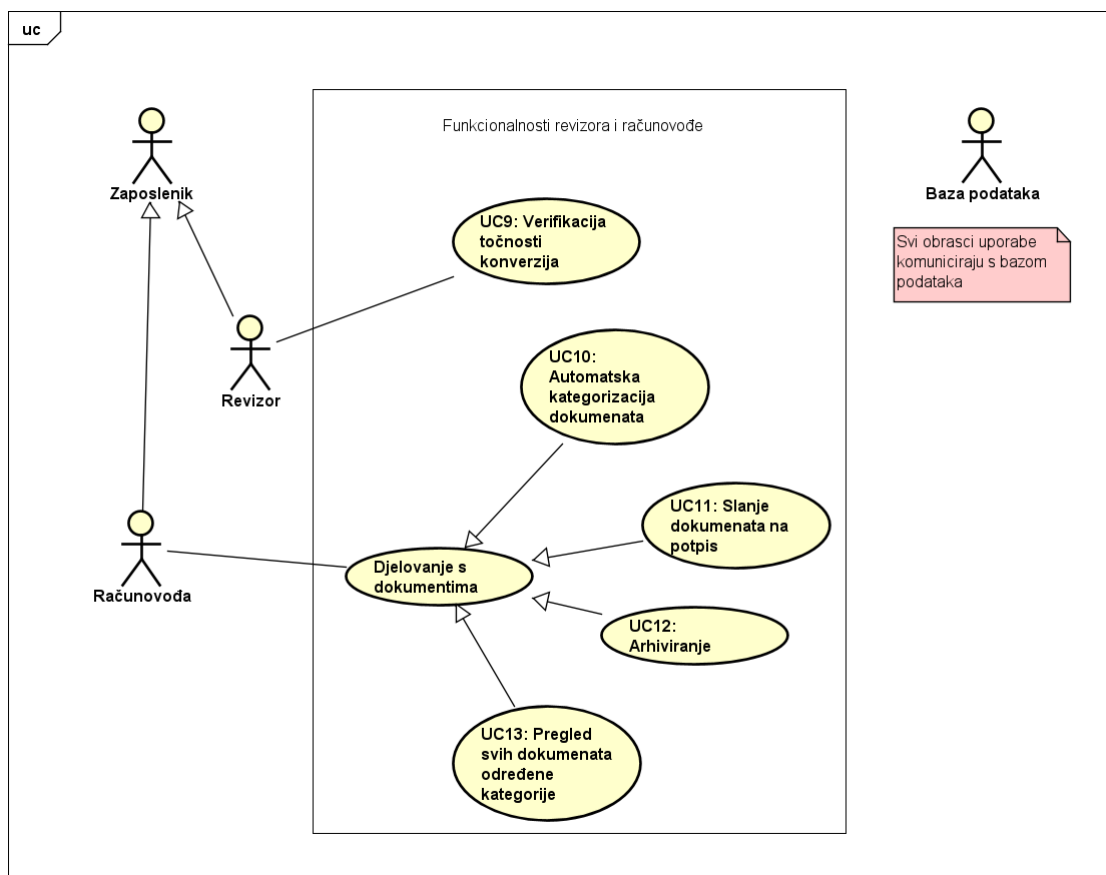
UC21 -Brisanje korisničkog računa

- **Glavni sudionik:** Direktor
- **Cilj:** Brisanje korisničkog računa
- **Sudionici:** Baza podataka
- **Preduvjet:** Registracija korisnika i prijava osobe s razinom ovlasti direktor u sustav
- **Opis osnovnog tijeka:**
 1. Direktor bira opciju za brisanje korisničkog računa
 2. Direktoru se prikazuje lista svih korisnika sustava (osim njega samog) te odabire jednog korisnika s liste
 3. Direktor unosi svoju korisničku lozinku kao potvrdu svog odabira i identiteta
 4. Baza podataka briše korisnički račun
 5. Direktor dobiva obavijest o uspješnom brisanju korisničkog računa

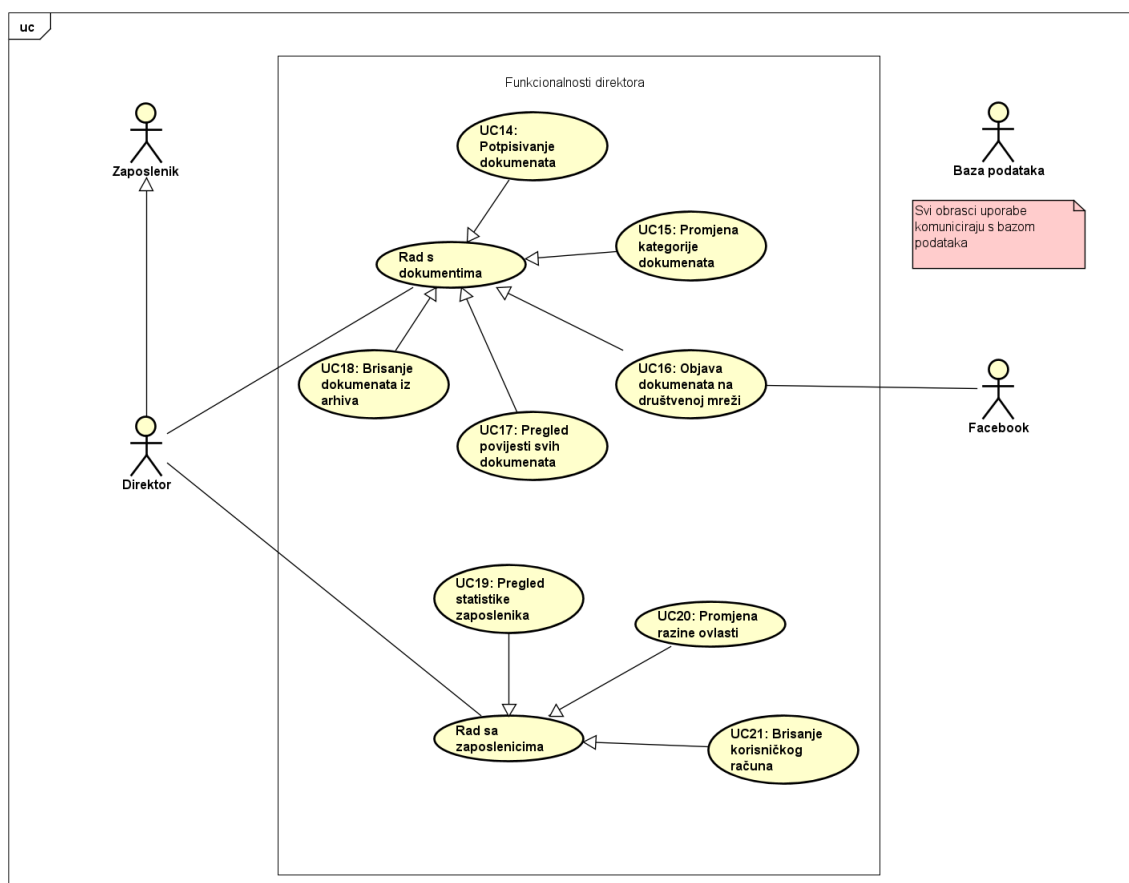
Dijagrami obrazaca uporabe



Slika 3.1: Use case dijagram 1

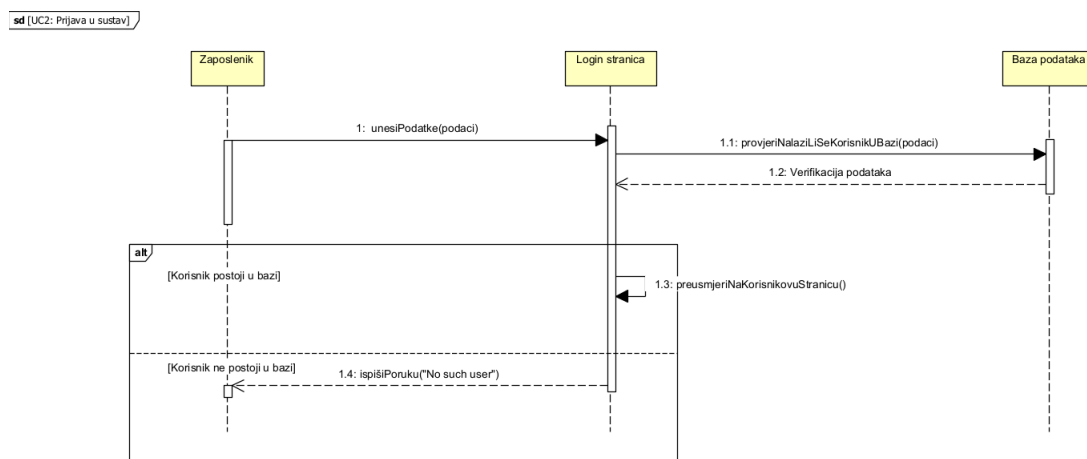


Slika 3.2: Use case dijagram 2

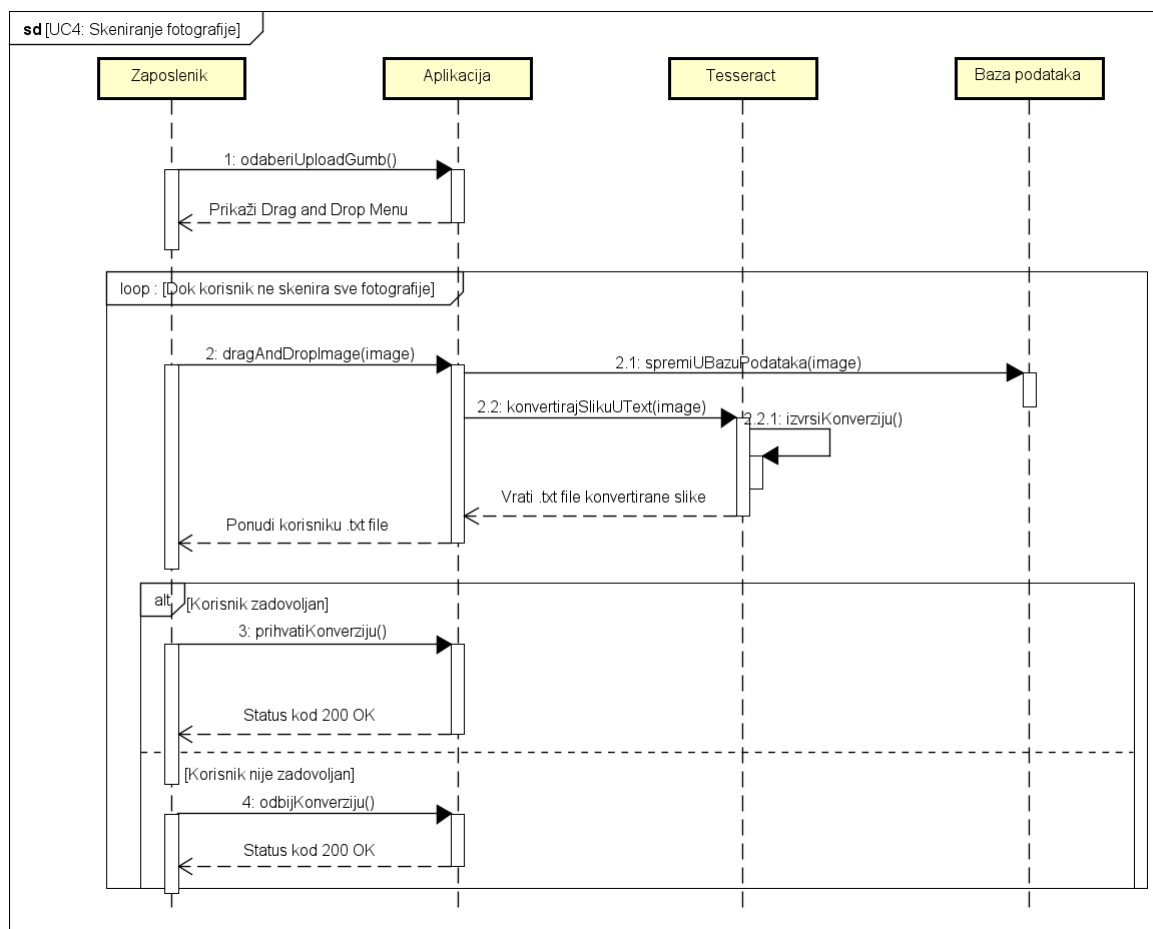


Slika 3.3: Use case dijagram 3

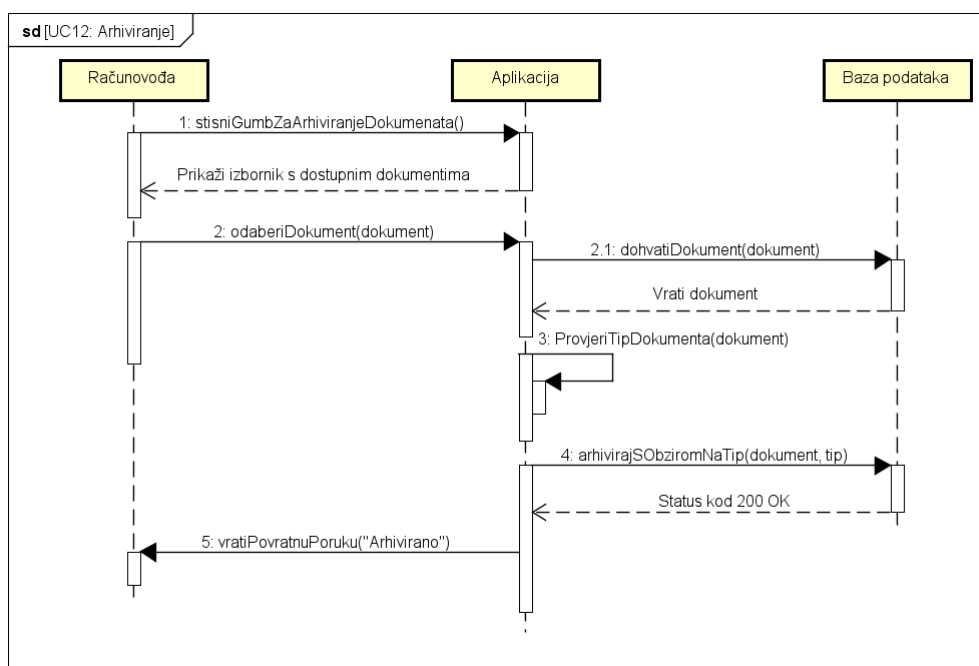
3.1.2 Sekvencijski dijagrami



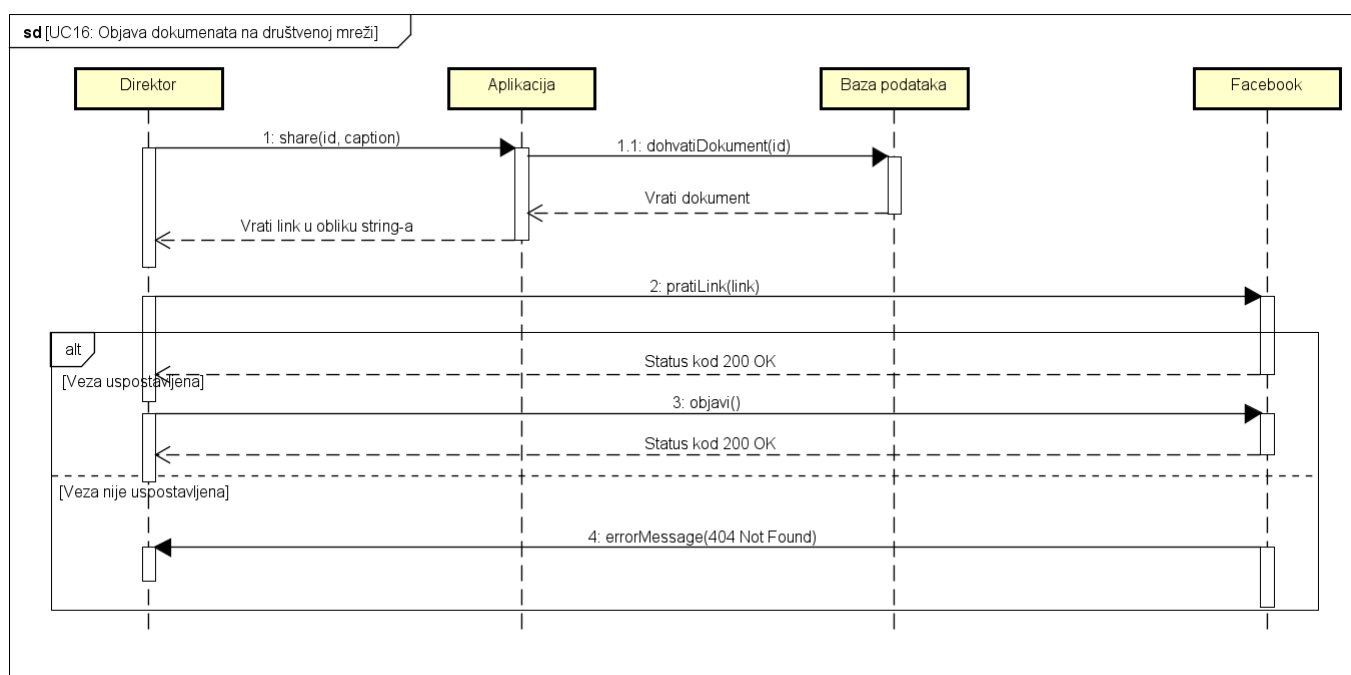
Slika 3.4: Sekvencijski dijagram 1 - Login



Slika 3.5: Sekvencijski dijagram 2 - Skeniranje fotografija



Slika 3.6: Sekvencijski dijagram 3 - Arhiviranje



Slika 3.7: Sekvencijski dijagram 4 - Objava dokumenata na društvenoj mreži

3.2 Ostali zahtjevi

- Korisničko sučelje mora podržavati čitavu hrvatsku abecedu
- Odgovor na HTTP zahtjev ne smije trajati duže od nekoliko sekundi
- Aplikacija mora podržati rad više korisnika istovremeno
- Korisničko sučelje mora biti jasno i jednostavno za upotrebu
- Sustav mora biti implementiran kao web-aplikacija
- Sustavu se mora putem HTTPS protokola moći pristupiti s javne mreže
- Nadogradnja sustava ne smije narušiti postojeće funkcionalnosti
- Mora postojati mogućnost brze i učinkovite nadogradnje sustava

4. Arhitektura i dizajn sustava

Arhitektura aplikacije može se podijeliti na 3 djela.

- web poslužitelj
- web preglednik
- baza podataka

Web preglednik je program koji korisniku omogućuje pristup web stranicama te multimedijском sadržaju na njima. Web preglednik omogućuje korisniku pristup web poslužitelju putem HTTPS (eng. *Hypertext Transfer Protokol Secure*) protokola te interpretira sadržaj kojeg je poslao Web poslužitelj u ljudski razumljiv oblik.

Web poslužitelj je program koji prima i odgovara na korisnikove HTTPS zahtjeve. Web poslužitelj pokreće aplikaciju te korisniku šalje potrebne podatke kako bi korisnikov web preglednik mogao generirati frontend aplikacije. Frontend predstavlja korisničko sučelje te je pisan u HTML-u, CSS-u i JavaScriptu uz pomoć React frameworka. ReactJS je biblioteka čiji su glavni element komponente. Komponente unutar sebe sadržavaju dijelove korisničkog sučelja te opisuju njihovo ponašanje. Glavna prednost korištenja komponenti je što se one mogu ponovno koristiti te se nove složenije komponente mogu spajati u složenije. Ovakav modularan pristup dizajniranju korisničkog sučelja čini nadogradnju i promjenu korisničkog sučelja vrlo jednostavnom. React dodatno sadrži virtualan DOM kojim optimizira rad korisničkog sučelja tako što minimizira broj manipulacija stvarnog DOM-a. Web poslužitelj je pisan u programskom jeziku Java i Spring Boot radnom okviru. Web poslužitelj komunicira sa bazom podataka te šalje podatke iz nje korisniku u prikladnom obliku. Spring Boot je radno sučelje za brz i učinkovit razvoj WEB aplikacija. Drži se principa konvencija iznad konfiguracije (engl. *convention over configuration*), to znači da Spring Boot oduzima mogućnost potpuno kontrole nad funkcioniranjem sustava od korisnika no zauzvrat pojednostavljuje razvoj web poslužitelja. Spring Boot ima višeslojnu arhitekturu gdje se podatci i zahtjevi šalju među raznim slojevima od kojih svaki ima svoju zadaću i logiku. Tako controller sloj prima i obrađuje HTTP zahtjeve, data access sloj komunicira s bazom podataka, exception handling layer je odgovoran za hvatanje i obrađivanje pogreški...

Baza podataka implementirana je putem PostgreSQL-a. Sadrži podatke potrebne za funkcioniranje aplikacije te njihove međusobne odnose.

4.1 Baza podataka

Baza podataka za web aplikaciju koristi relacijski model podataka. Kao RDBMS (Relational Database Management System) koristi se PostgreSQL verzija 15.0. Koristeći ERDPlus, stvaramo ER dijagram baze podataka u kojem navodimo sve entitete, njihove attribute te međusobne veze. Daljnje tipove podataka, primarne, strane i alternativne ključeve navodimo u relacijskoj shemi baze podataka.

4.1.1 Opis tablica

Employee Ovaj entitet sadržava sve važne informacije o zaposleniku koji koristi aplikaciju. Sadrži attribute: `id`, `first_name`, `last_name`, `email`, `password` i `role`. Ovaj entitet u vezi je One-to-Many s entitetom **Photos** preko zaposlenikovog `id`-a te je u vezi One-to-Many s entitetom **Document** preko zaposlenikovog `id`-a.

Employee		
<code>id</code>	BIGINT	Jedinstveni identifikator svakog zaposlenika
<code>first_name</code>	VARCHAR(255)	Ime zaposlenika
<code>last_name</code>	VARCHAR(255)	Prezime korisnika
<code>role</code>	VARCHAR(255)	Funkcija svakog zaposlenika (limitirano na Zaposlenik, Revizor, Računovođa i Direktor)
<code>password</code>	VARCHAR(255)	Zaposlenikova hashirana lozinka
<code>email</code>	VARCHAR(255)	E-mail korisnika

Photos Ovaj entitet sadržava sve važne informacije o fotografijama pohranjenima u bazu. Sadrži attribute: `photoid`, `url`, `image_name`, `upload_time` te strani ključ `upload_employee_id` preko korisnika. Ovaj entitet u vezi je Many-to-One s entitetom **Employee** preko zaposlenikovog `id`-a te je u One-to-One vezi s entitetom **Document** koji sadrži strani ključ `photo_id`.

Photos		
upload_employee_id	BIGINT	Jedinstveni identifikator zaposlenika koji je objavio fotografiju izveden iz tablice Employee
photo_id	BIGINT	Jedinstveni identifikator fotografije
url	VARCHAR(255)	URL fotografije
image_name	VARCHAR(255)	Ime prenesene fotografije
upload_time	TIMESTAMP	Datum i vrijeme prijenosa fotografije

Document Ovaj entitet sadržava sve važne informacije o dokumentima pohranjenima u bazi. Sadrži attribute: id, correct, type, to_be_signed, signed, verified, url, file_name te strane ključeve photo_id, scan_employee_id i verification_employee_id. Ovaj entitet u vezi je One-to-One s entitetom Photos preko photo_id, u Many-to-One vezi je s entitetom Employee te je u vezi One-to-One s entitetom Archive koji sadrži document_id kao strani ključ.

Document		
photo_id	BIGINT	Jedinstveni identifikator fotografije sadržane u dokumentu, izveden iz tablice Photos
scan_employee_id	BIGINT	Jedinstveni identifikator zaposlenika koji je skenirao dokument izveden iz tablice Employee
verification_employee_id	BIGINT	Jedinstveni identifikator zaposlenika zaduženog za verifikaciju ispravnosti konverzije, izveden iz tablice Employee
id	BIGINT	Jedinstveni identifikator prenesenog dokumenta

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Document		
correct	BOOLEAN	Potvrda ukoliko je rezultatni dokument ispravno skeniran
type	VARCHAR(255)	Tip dokumenta (limitirano na Račun, Ponuda i Interni dokument)
signed	BOOLEAN	Potvrda ukoliko je rezultatni dokument potpisan od strane direktora
verified	BOOLEAN	Potvrda ukoliko je rezultatni dokument potpisan od strane računovođe
to_be_signed	BOOLEAN	Potvrda ukoliko rezultatni dokument očekiva potpis od direktora
file_name	VARCHAR(255)	Ime pohranjenog dokumenta
url	VARCHAR(255)	URL zapis

Archive_entity Ovaj entitet sadržava sve važne informacije o arhiviranim dokumentima u bazi. Sadrži attribute: id te strani ključ document_id preko entiteta Dokument. Ovaj entitet u vezi je One-to-One s entitetom Dokument preko dokumentovog id-a.

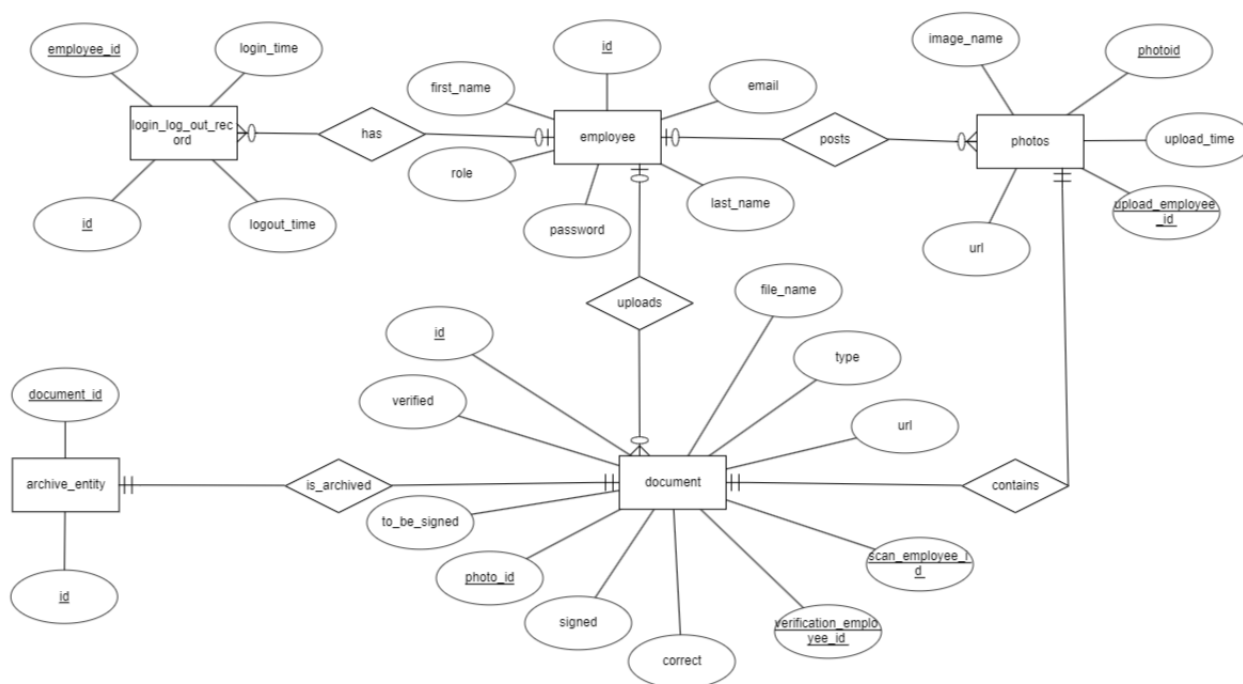
Archive		
document_id	BIGINT	Jedinstveni identifikator dokumenta koji se arhivira
id	BIGINT	Jedinstveni identifikator dokumenta koji se nalazi u arhivi

Login_log_out_record Ovaj entitet sadržava sve važne informacije o vremenu rada svakog pojedinačnog zaposlenika. Sadrži attribute: id, login_time, logout_time te strani ključ employee_id preko korisnikovog identifikatora. Ovaj entitet je u vezi Many-to-

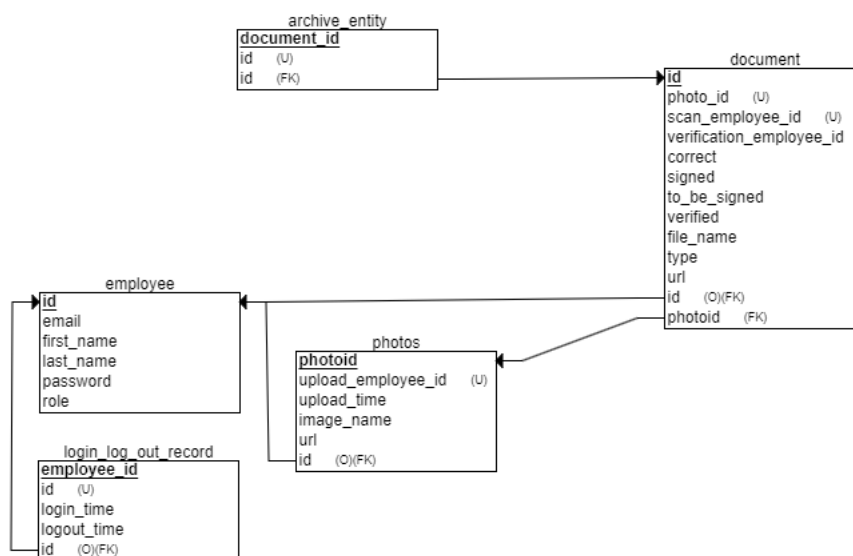
One s entitetom Employee preko zaposlenikovog id-a.

Login/Logout records		
employee_id	BIGINT	Jedinstveni identifikator zaposlenika čija se aktivnost na stranici sprema u statistiku
id	BIGINT	Jedinstveni identifikator session-a koji se gleda
login_time	TIMESTAMP	Datum i vrijeme login-a zaposlenika
logout_time	TIMESTAMP	Datum i vrijeme logout-a zaposlenika

4.1.2 Dijagram baze podataka



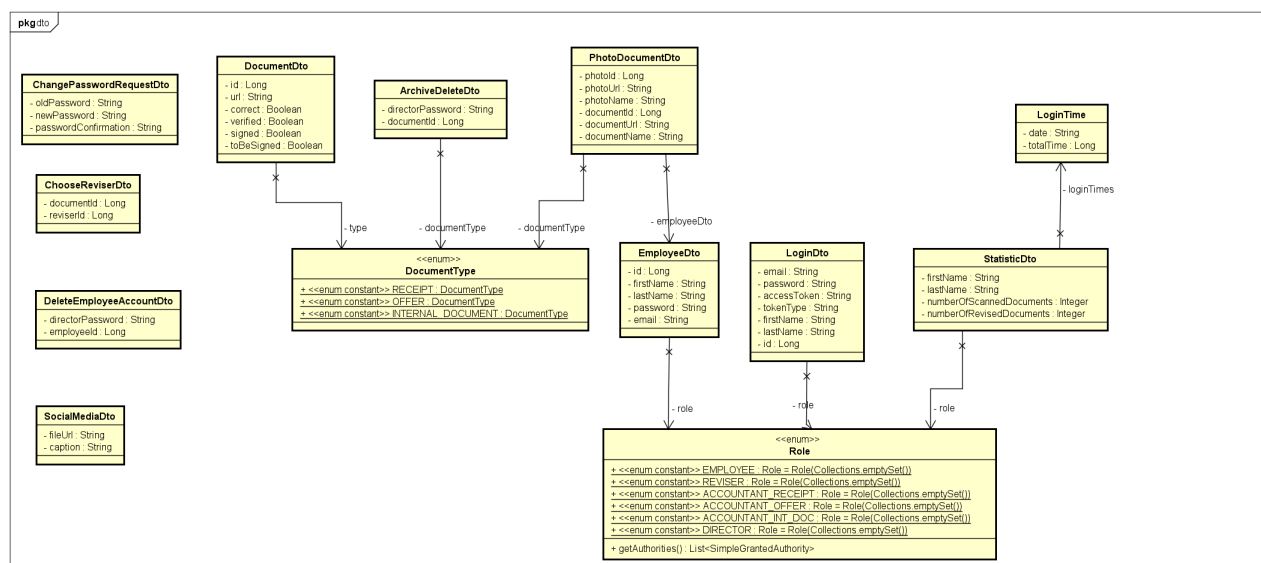
Slika 4.1: ER dijagram baze podataka



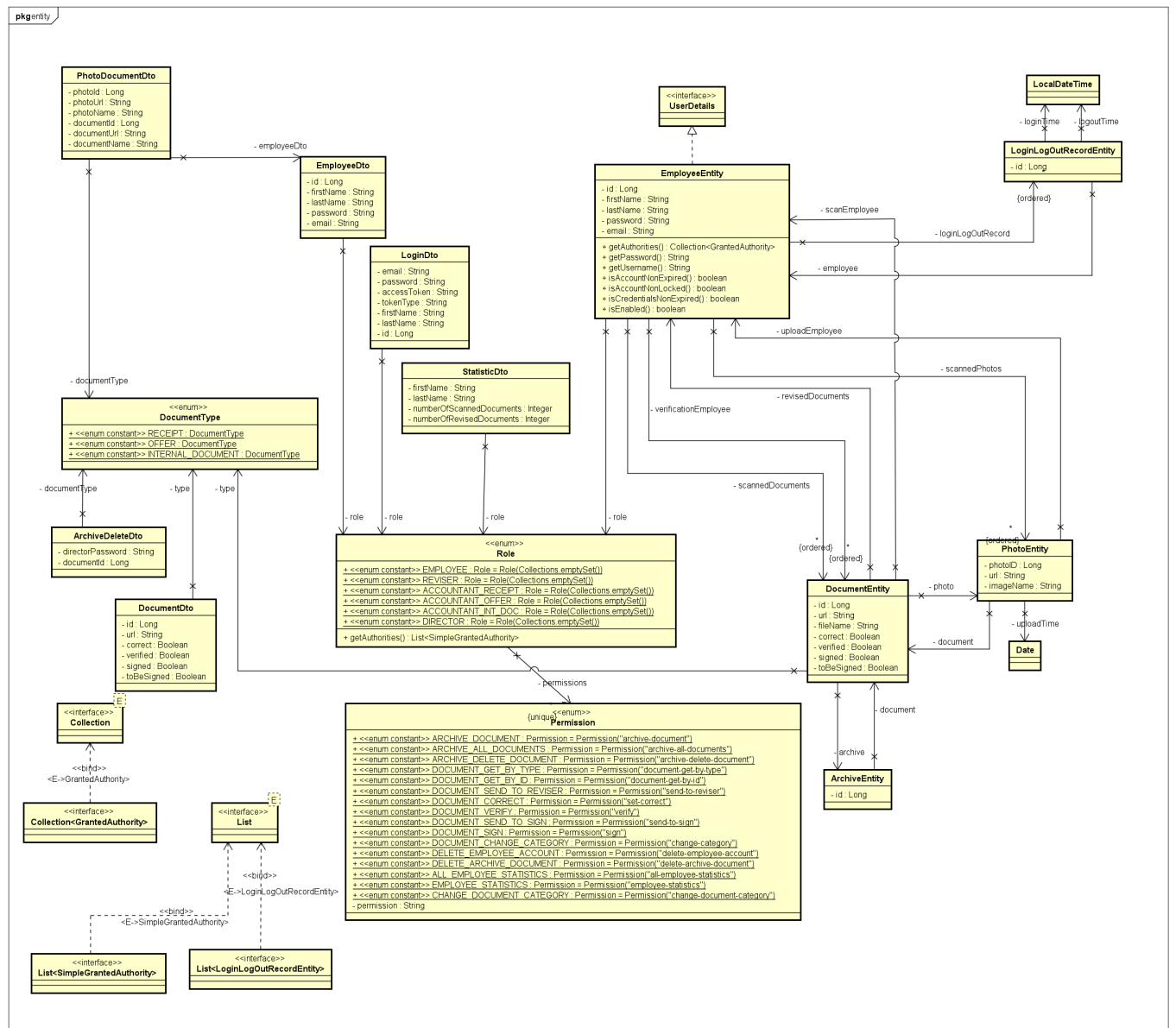
Slika 4.2: REL dijagram baze podataka

Na slikama su prikazani razredi koji pripadaju backend dijelu arhitekture. Na slikama su prikazani razredi koji nasljeđuju Controller razred, metode pomoću kojih oni manipuliraju s DTO (Data transfer objects), te razredi koji implementiraju razred Entity.





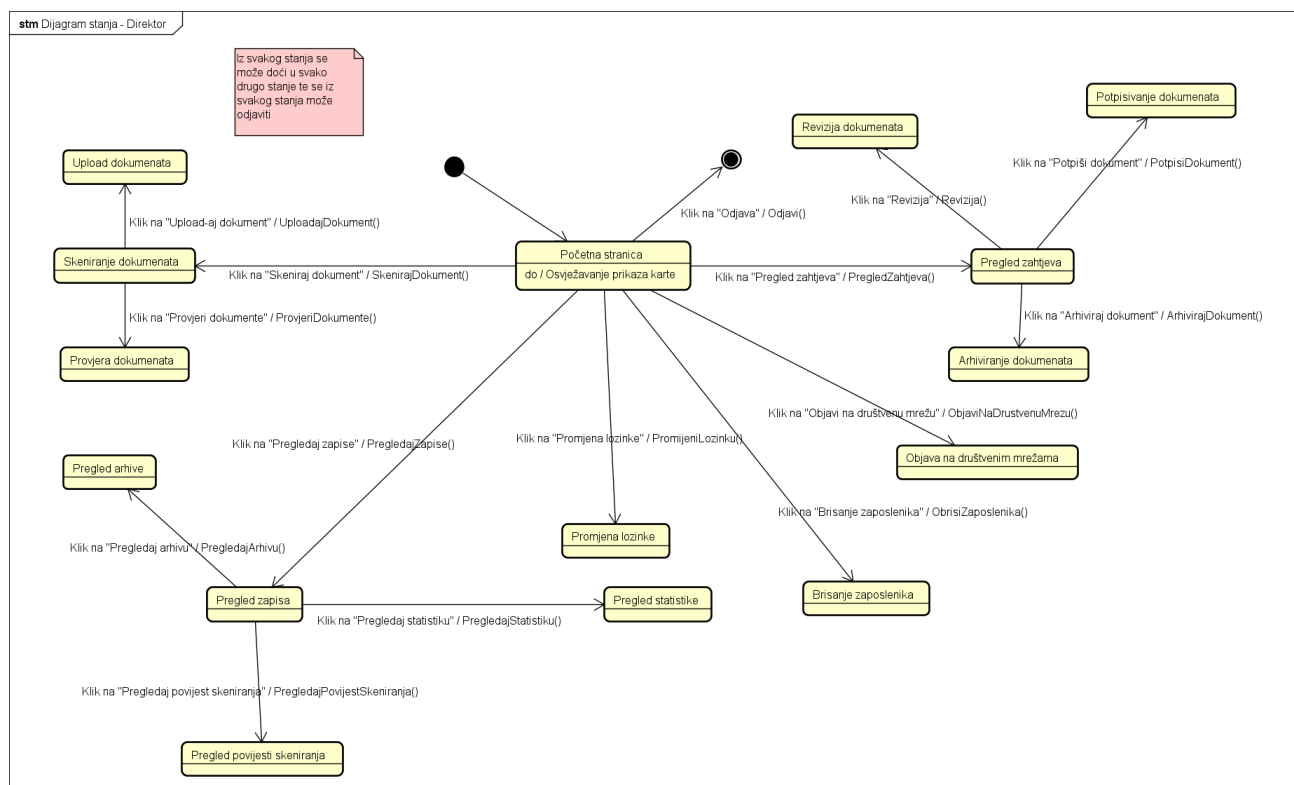
Slika 4.4: Dijagram razreda - dio Data transfer objects



Slika 4.5: Dijagram razreda - dio Entity

4.3 Dijagram stanja

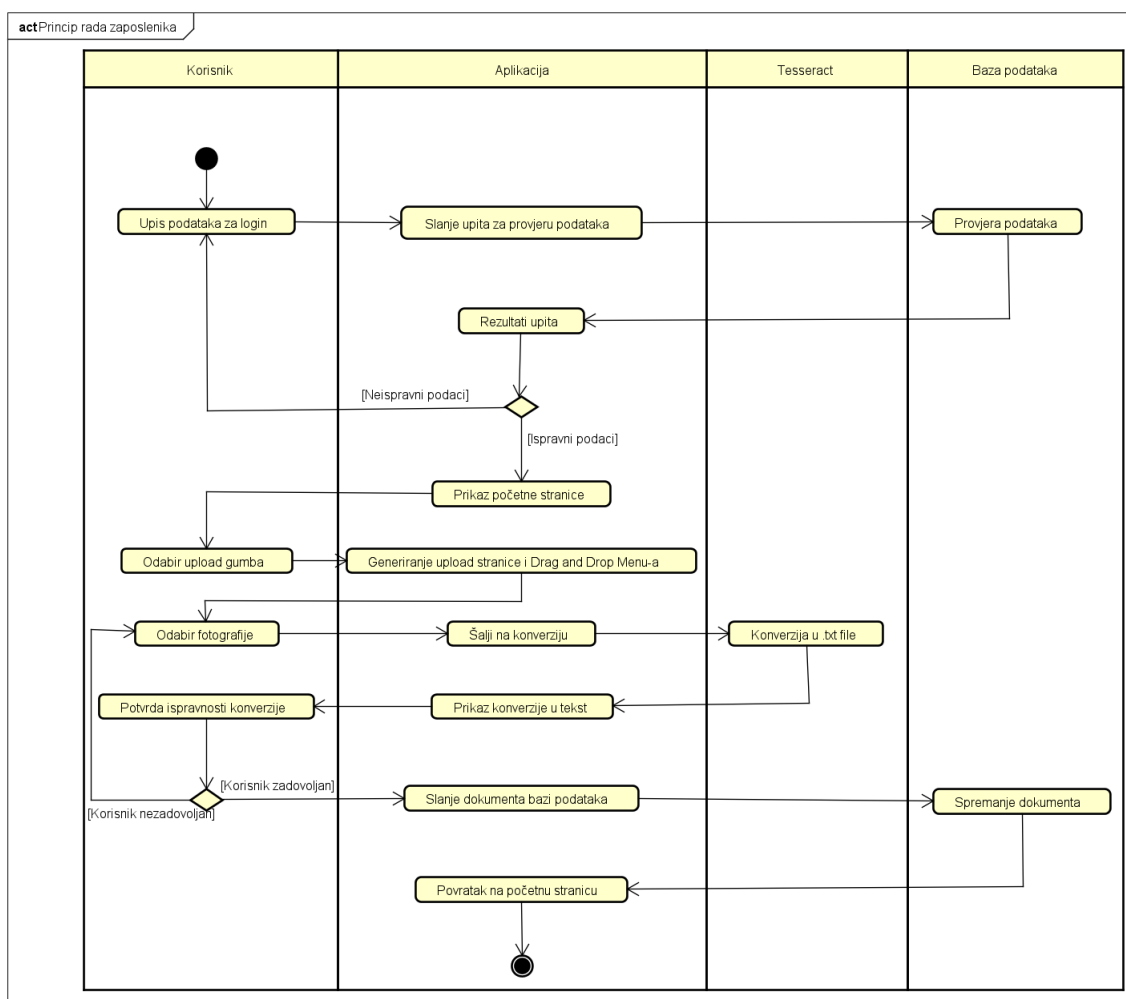
Dijagram stanja prikazuje stanja objekta te prijelaze iz jednog stanja u drugo temeljene na događajima. Na slici je prikazan dijagram stanja za direktora. Nakon prijave, direktoru se prikazuje početna stranica na kojoj može pristupiti svim svojim funkcionalnostima (skeniranju dokumenata, promjeni lozinke, pregledu zahtjeva itd.). Odabirom na skeniranje dokumenata direktoru se pruža mogućnost odabira upload-a dokumenta ili provjere već dostupnog dokumenta. Odabirom pregleda zahtjeva pružaju mu se opcije revizije, potpisivanja te arhiviranja dokumenta. Odabirom pregleda zapisa direktor može pregledati arhivu, povijest skeniranja te statistiku svih pojedinačnih zaposlenika. Direktoru se također pružaju opcije promjene lozinke, brisanja zaposlenika te objave dokumenta na društvenim mrežama (u ovom slučaju na društvenoj mreži "Facebook").



Slika 4.6: Dijagram stanja

4.4 Dijagram aktivnosti

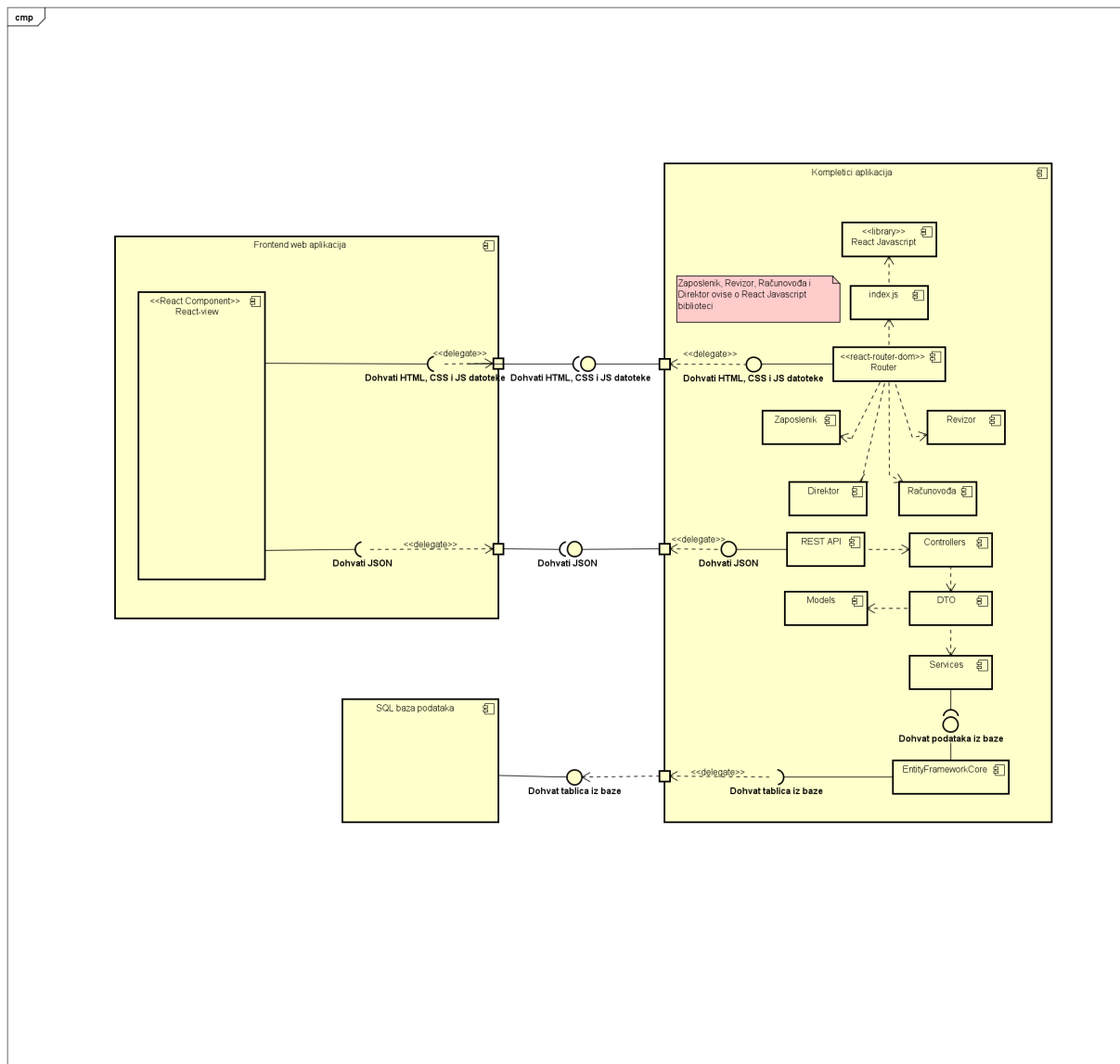
Dijagram aktivnosti primjenjuje se za opis modela toka upravljanja ili toka podataka. Ne upotrebljava se za modeliranje događajima poticanog ponašanja. U modeliranju toka upravljanja svaki novi korak poduzima se nakon završenog prethodnog, a naglasak je na jednostavnosti. Na dijagramu aktivnosti prikazan je princip rada zaposlenika. Zaposlenik nakon uspješne prijave u sustav upload-a fotografije koje se daju na skeniranje, te aplikacija iste fotografije konvertira koristeći Tesseract.



Slika 4.7: Dijagram aktivnosti

4.5 Dijagram komponenti

Dijagram komponenti prikazan na slici opisuje organizaciju i međuovisnost komponenti, interne strukture i odnose prema okolini. Sustavu se pristupa preko dva različita sučelja. Preko sučelja za dohvat HTML, CSS i JS datoteka poslužuju se datoteke koje pripadaju frontend dijelu aplikacije. Router je komponenta koja na upit s url određuje koja datoteka će se poslužiti na sučelje. Frontend dio se sastoji od niza JavaScript datoteka koje su raspoređene u logičke cjeline nazvane po tipovima aktera koji im pristupaju. Sve JavaScript datoteke ovise o React biblioteci iz koje dohvaćaju gotove komponente kao što su gumbi, forme i slično. Preko sučelja za dohvat JSON podataka pristupa se REST API komponenti. REST API poslužuje podatke koji pripadaju backend dijelu aplikacije. EntityFrameworkCore je zadužen za dohvaćanje tablica iz baze podataka pomoću SQL upita. Podaci koji su pristigli iz baze se šalju dalje MVC arhitekturi u obliku DTO preko Services. React-view komponenta preko dostupnih sučelja komunicira sa aplikacijom te ovisno o korisnikovim akcijama osvježava prikaz i dohvaća nove podatke ili datoteke.



Slika 4.8: Dijagram komponenti

5. Implementacija i korisničko sučelje

5.1 Korištene tehnologije i alati

Komunikacija u timu ostvarena je korištenjem aplikacije WhatsApp ¹. Kao sustav za upravljanje izvornim kodom korišten je Git ², a izvorni kod projekta dostupan je na Github ³ web platformi. Za izradu dokumentacije korištena je distribucija markup jezika LaTeX MiKTeX ⁴ u kombinaciji s TeXstudio ⁵ radnom okolinom.

Za izradu aplikacije koristila su se dva razvojna okruženja, Visual Studio Code⁶ za frontend i JetBrains IntelliJ IDEA ⁷ za backend. Visual Studio Code je radna okolina koju razvija i održava Microsoft, vrlo je fleksibilna te omogućava razvoj u širokom spektru jezika i tehnologija. JetBrains IntelliJ IDEA je razvojno okruženje specifično dizajnirano za rad s programskim jezikom Java, održava je tvrtka JetBrains koja je poznata po proizvodnji razvojnih okolina.

Za izradu backenda korišten je programski jezik Java⁸ i radni okvir Spring ⁹. Spring je radni okvir koji nudi gotova rješenja za mnoge često potrebne funkcionalnosti programskih sustava, time programerima omogućuje jednostavniji, brži i sigurniji razvoj aplikacija.

Za izradu frontenda korišten je programski jezik JavaScript¹⁰ i biblioteka React¹¹. React je biblioteka za razvoj korisničkih sučelja, u složenijim sustavima koristi se s drugim bibliotekama gdje služi kao temeljni sustav sučelja. React održava Facebook.

Za izradu baze podataka korištena je implementacija SQL-a zvana PostgreSQL¹².

Za skeniranje dokumenata i njihovo pretvaranje u txt datoteke korišten je OCR

¹<https://www.whatsapp.com/>

²<https://git-scm.com/>

³<https://github.com/>

⁴<https://miktex.org/>

⁵<https://www.texstudio.org/>

⁶<https://code.visualstudio.com/>

⁷<https://www.jetbrains.com/idea/>

⁸<https://www.java.com/en/>

⁹<https://spring.io/>

¹⁰<https://developer.mozilla.org/en-US/docs/Web/javascript>

¹¹<https://react.dev/>

¹²<https://www.postgresql.org/>

Tesseract.¹³ Tesseract je projekt otvorenog izvornog koda kojeg održava zajednica volontera a omogućuje pretvorbu slika teksta u txt datoteke preko API-a.

Za pohranu slika korištena je Googleova usluga Firebase. Firebase je web platforma za razvoj video igara i aplikacija koja nudi gotova rješenja i infrastrukture. U sklopu ovog projekta korištena je za pohranu slika jer druge usluge nisu dopuštale dovoljno prostora.

Za objavu dokumenata na internetskim mrežama korišten je API društvene mreže Facebook.

Kao platformu za puštanje u pogon odabrane je Render ¹⁴ Render je WEB platforma specifično dizajnirana za puštanje aplikacija u pogon. Render pruža jednostavnu i učinkovitu infrastrukturu u oblaku zajedno s ograničenom memorijom za bazu podataka. Render održava istoimena tvrtka. Kako bi se zadovoljio format u kojem Render očekuje aplikaciju za puštanje u pogon dodatno se koristio alat Docker ¹⁵. Docker je alat za pakiranje aplikacije sa svim potrebnim sredstvima za pokretanje aplikacije, time se postiže mogućnost pokretanja aplikacije na širokom spektru arhitektura. Docker također održava istoimena tvrtka.

5.2 Ispitivanje programskog rješenja

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

Ispitivanje sustava nije provedeno

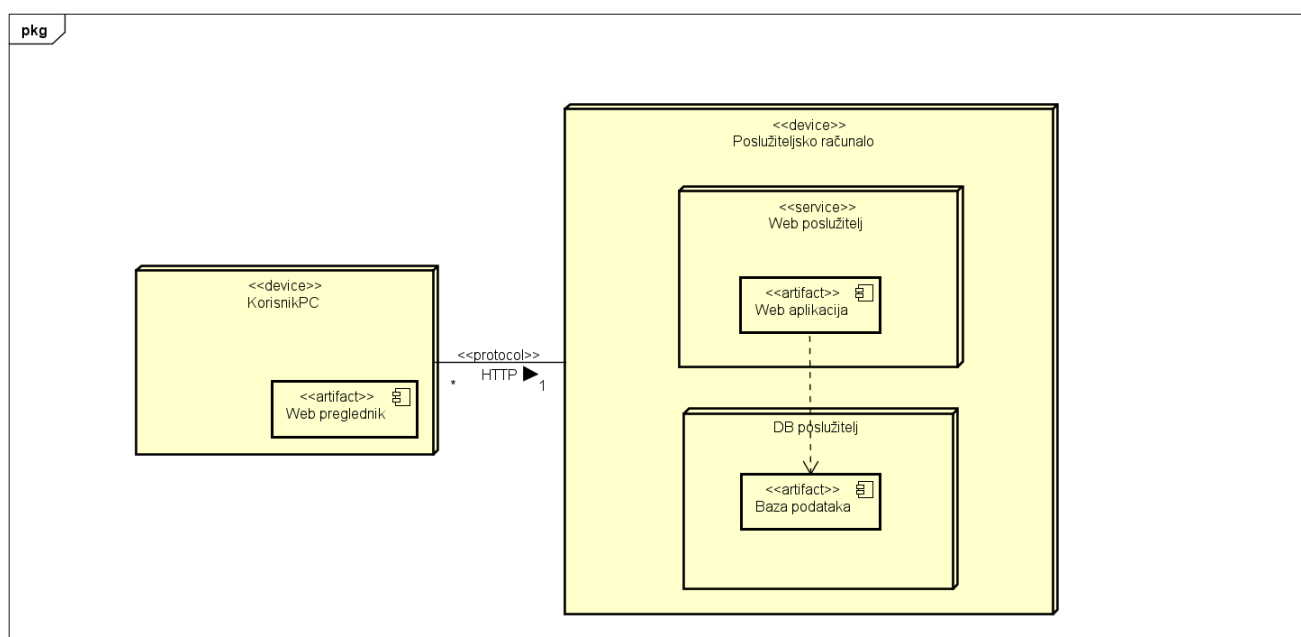
¹³<https://tesseract-ocr.github.io/tessdoc/Home.html>

¹⁴<https://render.com/>

¹⁵<https://www.docker.com/>

5.3 Dijagram razmještaja

Dijagrami razmještaja opisuju topologiju sklopovlja i programsku potporu koja se koristi u implementaciji sustava u njegovom radnom okruženju. 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 (zaposlenik, revizor, računovođa, direktor) i poslužitelja odvija se preko HTTP veze.



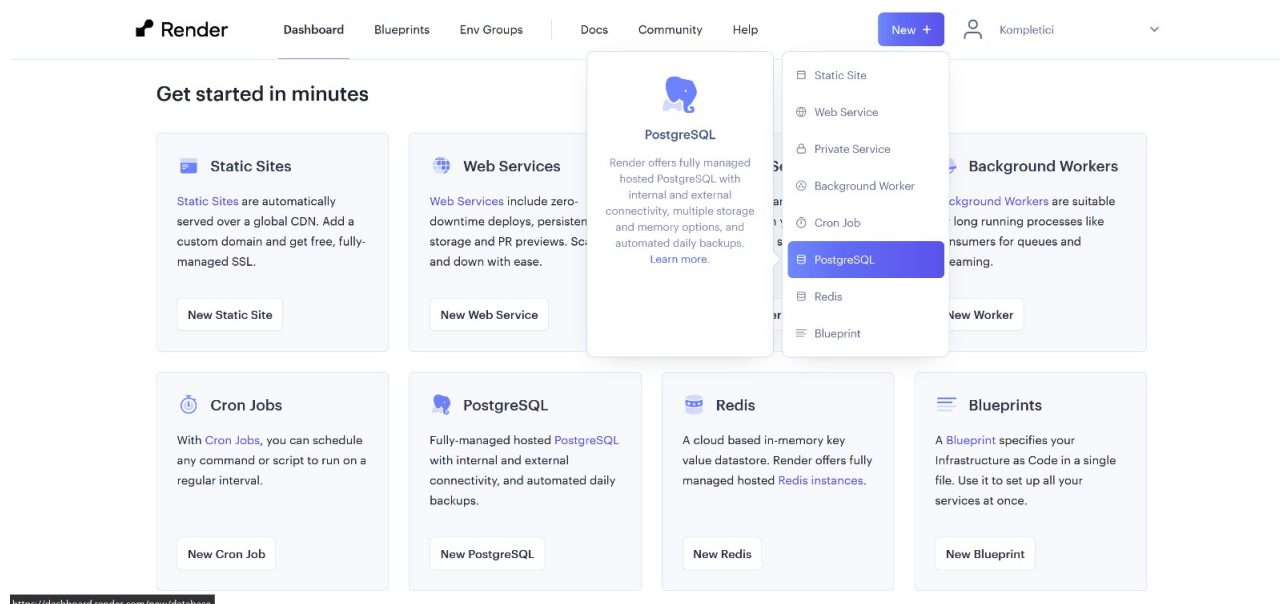
Slika 5.1: Dijagram razmještaja

5.4 Upute za puštanje u pogon

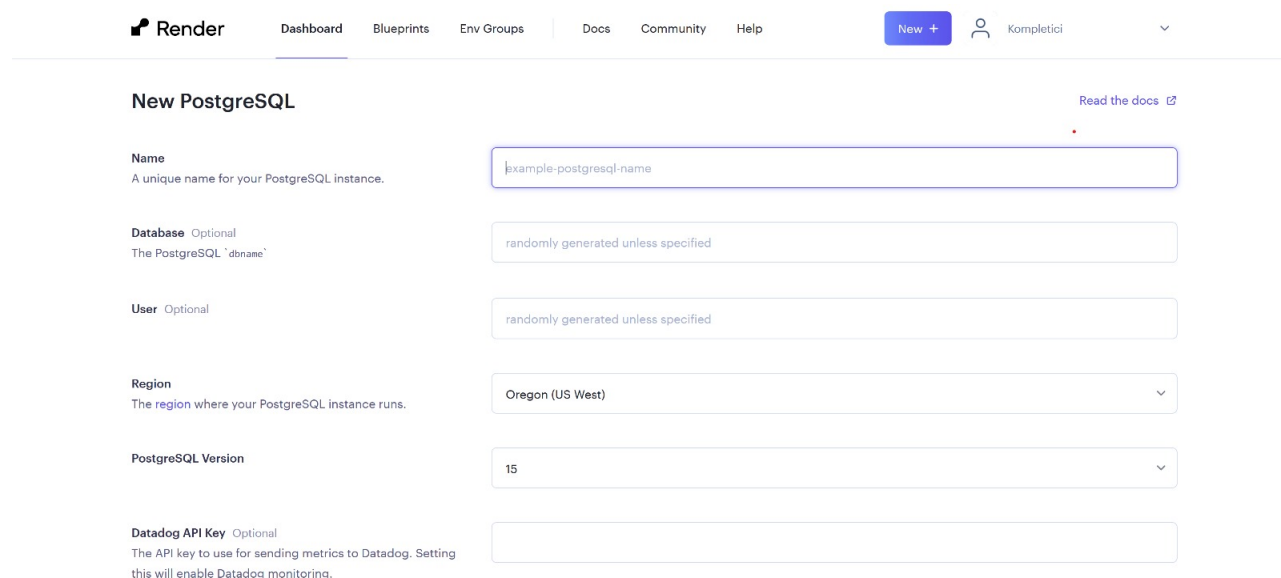
Za puštanje u pogon korištena je WEB usluga Render istoimene tvrtke, te se puštanje u pogon treba obaviti prema zahtjevima Render platforme.

Konfiguracija poslužitelja baze podataka

Unutar WEB platforme Render potrebno je konfigurirati poslužitelj baze podataka. Na radnoj traci odabiremo opciju new, nakon čega iz padajućeg izbornika treba odabiremo opciju PostgreSQL.



Slika 5.2: Render korisničko sučelje i radna traka



The screenshot shows the 'New PostgreSQL' configuration page on the Render platform. The page has a header with the Render logo and navigation links: Dashboard, Blueprints, Env Groups, Docs, Community, and Help. There is a 'New +' button and a user profile icon labeled 'Kompletići'. The main content area is titled 'New PostgreSQL' with a 'Read the docs' link. The configuration fields are as follows:

- Name:** A unique name for your PostgreSQL instance. The input field contains 'example-postgresql-name'.
- Database:** Optional. The PostgreSQL 'dbname'. The input field contains 'randomly generated unless specified'.
- User:** Optional. The input field contains 'randomly generated unless specified'.
- Region:** The region where your PostgreSQL instance runs. The dropdown menu is set to 'Oregon (US West)'.
- PostgreSQL Version:** The dropdown menu is set to '15'.
- Datadog API Key:** Optional. The API key to use for sending metrics to Datadog. Setting this will enable Datadog monitoring. The input field is empty.

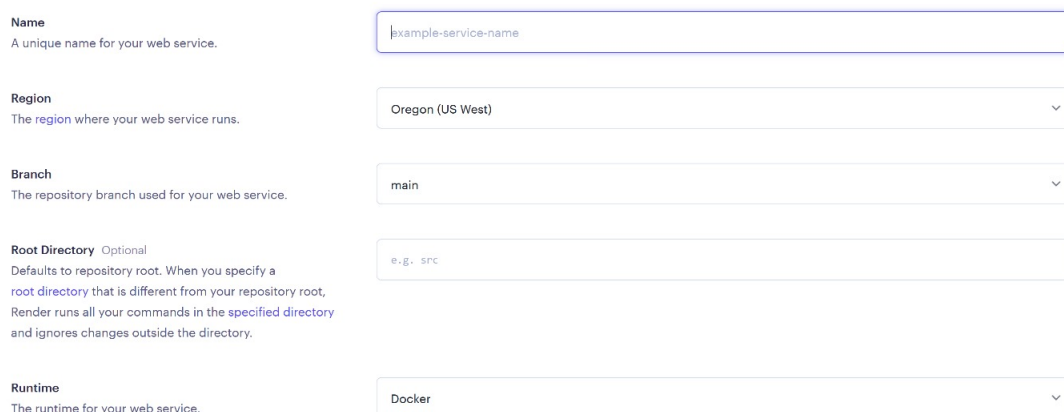
Slika 5.3: Opcije konfiguracije za bazu podataka

Otvoriti će se korisničko sučelje za konfiguraciju baze podataka. Na korisničkom sučelju potrebno je odabrati regiju poslužitelja s kojeg će Render posluživati korisnike. Render će zatim izgenerirati URL poslužitelja baze podataka i lozinku baze podataka, spremamo te podatke jer će nam trebati u daljnjim koracima

Konfiguracija backend poslužitelja

Na Render-ovoj radnoj traci odaberemo opciju new WEB service, te odaberemo opciju "Build and deploy from git repository."

Nakon toga sljedeći proces povezivanja GitHub korisničkog računa i repozitorija s Render korisničkim računom. Potom trebamo odabrati ime za WEB servis te ponovno odabrati regiju poslužitelja s koje će Render posluživati korisnike.



The image shows a configuration form for a new web service on the Render platform. It includes five sections: 'Name' with a text input field containing 'example-service-name'; 'Region' with a dropdown menu set to 'Oregon (US West)'; 'Branch' with a dropdown menu set to 'main'; 'Root Directory' with an optional text input field containing 'e.g., src'; and 'Runtime' with a dropdown menu set to 'Docker'.

Name A unique name for your web service.	<input type="text" value="example-service-name"/>
Region The region where your web service runs.	<div>Oregon (US West) ▾</div>
Branch The repository branch used for your web service.	<div>main ▾</div>
Root Directory <small>Optional</small> Defaults to repository root. When you specify a root directory that is different from your repository root, Render runs all your commands in the specified directory and ignores changes outside the directory.	<input type="text" value="e.g., src"/>
Runtime The runtime for your web service.	<div>Docker ▾</div>

Slika 5.4: Opcije konfiguracije za WEB servise

Nakon toga trebamo odabrati granu GitHub repozitorija s koje će Render povlačiti kod za automatsko puštanje u pogon i odabrati root direktorij backend aplikacije. Za runtime okolinu odabiremo Docker i proširujemo napredne postavke. Dodajemo potrebne varijable okoline poput imena baze, šifre baze i URL baze. Posebnu pozornost trebamo obratiti na format URL-a baze podataka, naime postoji mogućnost da ga treba preoblikovati. On mora biti u sljedećem formatu `jdbc:postgresql://<hostname>:<port>/<database>`. Konačno moramo dodati putanju Docker datoteke.

Konfiguracija frontend poslužitelja

Na Renderovoj radnoj traci odabiremo opciju new WEB service, te odabiremo opciju "Build and deploy from git repository." Nakon toga ponovno povezujemo GitHub korisnički račun s Renderom te odabiremo granu koju će Render puštati u pogon. Dodatno odabiremo regiju poslužitelja s kojeg će Render posluživati korisnike. Build komand postavljamo na yarn-build a start komandu na start-prod. Konačno proširujemo napredne postavke te kao varijablu okoline dodajemo adresu deployanog web servisa.

6. Zaključak i budući rad

Zadatak projektnog tima bio je razvoj aplikacije za skeniranje i distribuciju dokumenata unutar organizacije. Nakon dva i pol mjeseca rada ostvarili smo zadani cilj.

Projekt smo proveli unutar dvije faze razvoja. Prva faza uključivala je pisanje početne dokumentacije, odabir tehnologija te podjelu zadataka unutar projektnog tima. Prva faza je također uključivala implementaciju sustava registracije i prijave korisnika te kostur aplikacije. Druga faza sastojala se od implementacije ključnih djelova sustava kao što su skeniranje dokumenata, njihova distribucija i arhiviranje. Dodatno druga faza sadržavala je pisanje popratne dokumentacije.

Općenito govoreći rad na projektu bio je zanimljivo i korisno iskustvo za sve članove tima. Suočili smo se sa mnogim izazovima od kojih smo mnoge riješili. Glavne prepreke su bile manjak iskustva te ne optimalna koordinacija članova tima. Manjak iskustva stvarao je poteškoće jer su članovi tima morali samostalno učiti tehnologije koje prije nisu koristili te nisu znali najbolje prakse u mnogim situacijama. Ne optimalna koordinacija članova tima značila je da količina posla nije bila u potpunosti ravnopravna te svi članovi tima nisu u potpunosti mogli pridonijeti čak i kad je postojala volja. Poboljšanje radnog procesa u budućim nadogradnjama aplikacije ili radu na novim projektima uključivala bi korištenje poznatih alata i bolju koordinaciju projektnog tima.

Implementirali smo sve ključne značajke zadatka. Nedostaje implementacija jednog obrasca uporabe što je mogućnost da direktor mijenja razinu ovlasti drugih korisnika. Neka od poboljšanja koja bi se mogla implementirati su razvoj mobilne aplikacije, poboljšanje robusnosti OCR-a i razvoj boljeg korisničkog sučelja. Iako aplikacija ima mnoge nedostatke zadovoljni smo s odrađenim poslom.

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	Use case dijagram 1	19
3.2	Use case dijagram 2	20
3.3	Use case dijagram 3	21
3.4	Sekvencijski dijagram 1 - Login	21
3.5	Sekvencijski dijagram 2 - Skeniranje fotografija	22
3.6	Sekvencijski dijagram 3 - Arhiviranje	23
3.7	Sekvencijski dijagram 4 - Objava dokumenata na društvenoj mreži	23
4.1	ER dijagram baze podataka	29
4.2	REL dijagram baze podataka	29
4.3	Dijagram razreda - dio Controllers	30
4.4	Dijagram razreda - dio Data transfer objects	31
4.5	Dijagram razreda - dio Entity	32
4.6	Dijagram stanja	33
4.7	Dijagram aktivnosti	34
4.8	Dijagram komponenti	36
5.1	Dijagram razmještaja	39
5.2	Render korisničko sučelje i radna traka	40
5.3	Opcije konfiguracije za bazu podataka	41
5.4	Opcije konfiguracije za WEB servise	42