

# Programsko inženjerstvo

Ak. god. 2021./2022.

## Digitalizacija

Dokumentacija, Rev. 2

Grupa: *PiratesOfCode*

Voditelj: *Ivan Futivić*

Datum predaje: *14. siječnja 2022.*

Nastavnik: *Igor Stančin*

# Sadržaj

<b>1</b>	<b>Dnevnik promjena dokumentacije</b>	<b>3</b>
<b>2</b>	<b>Opis projektnog zadatka</b>	<b>6</b>
2.1	Uvod . . . . .	6
2.2	Korisnici aplikacije . . . . .	6
2.3	OCR skener . . . . .	7
2.4	Ključni dijelovi sustava . . . . .	8
2.5	Slična rješenja . . . . .	8
<b>3</b>	<b>Specifikacija programske potpore</b>	<b>9</b>
3.1	Funkcionalni zahtjevi . . . . .	9
3.1.1	Obrasci uporabe . . . . .	11
3.1.2	Sekvencijski dijagrami . . . . .	23
3.2	Ostali zahtjevi . . . . .	28
<b>4</b>	<b>Arhitektura i dizajn sustava</b>	<b>29</b>
4.1	Baza podataka . . . . .	30
4.1.1	Opis tablica . . . . .	31
4.1.2	Dijagram baze podataka . . . . .	34
4.2	Dijagram razreda . . . . .	35
4.3	Dijagram stanja . . . . .	40
4.4	Dijagram aktivnosti . . . . .	41
4.5	Dijagram komponenti . . . . .	43
<b>5</b>	<b>Implementacija i korisničko sučelje</b>	<b>44</b>
5.1	Korištene tehnologije i alati . . . . .	44
5.2	Ispitivanje programskog rješenja . . . . .	46
5.2.1	Ispitivanje komponenti . . . . .	46
5.2.2	Ispitivanje sustava . . . . .	48
5.3	Dijagram razmještaja . . . . .	50
5.4	Upute za puštanje u pogon . . . . .	51

5.4.1	Kreiranje projekta i integracija SDK-a u Android . . . . .	51
5.4.2	Puštanje u pogon Cloud Funkcija . . . . .	54
5.4.3	Pokretanje Android aplikacije . . . . .	54
<b>6</b>	<b>Zaključak i budući rad</b>	<b>55</b>
	<b>Popis literature</b>	<b>57</b>
	<b>Indeks slika i dijagrama</b>	<b>58</b>
	<b>Dodatak: Prikaz aktivnosti grupe</b>	<b>59</b>

# 1. Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Dodani funkcionalni zahtjevi i obrasci uporabe	Blaž Solić	22.10.2021.
0.1.1	Dodani <i>Use Case</i> dijagrami	Blaž Solić	24.10.2021.
0.2	Dodan opis projektnog zadatka	Luka Hanžek	24.10.2021.
0.2.1	Uređen i nadograđen opis projektnog zadatka	Luka Hanžek	03.11.2021.
0.3	Dodani sekvencijalni dijagrami za registraciju i prijavu	Luka Hanžek	03.11.2021.
0.4	Dodan opis i dijagram baze podataka	Blaž Solić	10.11.2021.
0.4.1	Uređen dijagram baze podataka	Blaž Solić	14.11.2021.
0.5	Dodan sekvencijalni dijagram za skeniranje	Luka Hanžek	14.11.2021.
0.6	Dodan opis arhitekture	Blaž Solić	17.11.2021.
0.7	Dodani dijagrami razreda	Karlo Marković	17.11.2021.
0.8	Svi <i>Use Case</i> dijagrami potpuno uređeni, preuređenje svih sekvencijalnih dijagrama i dodavanje novih, dodani opisi sekvencijalnih dijagrama	Luka Hanžek	18.11.2021.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodatka	Autori	Datum
0.9	Dodana nova klasa dijagramu razreda	Karlo Marković	18.11.2021.
0.9.1	Dodani nefunkcionalni zahtjevi, opis projektnog zadatka potpuno uređen	Luka Hanžek	19.11.2021.
0.9.2	Preuređeni dijagrami razreda	Karlo Marković	19.11.2021.
1.0	Verzija samo s bitnim dijelovima za 1. ciklus	Blaž Solić, Ivan Futivić, Luka Hanžek, Karlo Marković	19.11.2021.
1.1	Dodan dijagram stanja	Blaž Solić	11.01.2022.
1.2	Izmijenjen dijagram baze	Blaž Solić	11.01.2022.
1.2.1	Popravljen dijagram baze	Blaž Solić	12.01.2022.
1.3	Dodane upute za puštanje u pogon	Ivan Futivić, Blaž Solić	12.01.2022.
1.4	Izmijenjen dijagram stanja i dodana slika dijagrama baze	Blaž Solić	12.01.2022.
1.5	Dodan dijagram razmještaja	Karlo Marković	12.01.2022.
1.6	Popravljen dijagram razreda	Karlo Marković	13.01.2022.
1.6.1	Dodane nove slike dijagrama razreda	Karlo Marković	13.01.2022.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodatka	Autori	Datum
1.7	Dodan dijagram aktivnosti i slika	Karlo Kada	14.01.2022.
1.8	Dodan dijagram komponenti	Luka Hanžek	14.01.2022.
1.9	Opisano testiranje aplikacije	Nina Petrušić	14.01.2022.
1.10	Opisane korištene tehnologije i alati	Ivan Futivić, Blaž Solić	14.01.2022.
1.11	Dodana literatura	Luka Hanžek	14.01.2022.
1.12	Dodana zaključak i daljnji rad	Rafael Boban, Luka Hanžek	14.01.2022.
2.0	Konačna verzija dokumentacije	Ivan Futivić, Luka Hanžek, Karlo Marković, Nina Petrušić, Rafael Boban, Karlo Kada, Blaž Solić	14.01.2022.

## 2. Opis projektnog zadatka

### 2.1 Uvod

Rezultat ovog projektnog zadatka je mobilna aplikacija "SpyGlass" i sustav namijenjen lakšoj internoj administraciji različitih tipova dokumenata unutar nekog poduzeća. Aplikacija će svim korisnicima unutar poslovnog subjekta pomoću ugrađene kamere na mobilnom uređaju omogućiti skeniranje dokumenata. Skenirani dokumenti će se pomoću optičkog prepoznavanje znakova pretvarati iz slikovnog formata u tekstualni format te će se u skladu s tipom dokumenta tekst na određeni način raščlaniti na dijelove. Dokumenti će se nakon skeniranja spremati u sustav i slati na daljnju obradu drugim korisnicima aplikacije zaduženim za određeni zadatak. Svaki korisnik će moći pristupiti svim već svojim skeniranim dokumentima, a ovisno o specijalizaciji korisnika aplikacija će korisniku nuditi i dodatne opcije. Korisnici aplikacije podijeljeni su u 4 kategorije koji tijekom rada sa sustavom međusobno razmjenjuju dokumente i tako administraciju dokumenata uvelike pojednostavljaju.

### 2.2 Korisnici aplikacije

Korisnici aplikacije svrstani su u četiri grupe od kojih svaka ima neke zajedničke i neke specifične mogućnosti. Korisnici aplikacije dijele se na sljedeće skupine: zaposlenik, revizor, računovođa i direktor. Korisnici se prije korištenja usluga moraju registrirati uz odabir određene specijalizacije. Prilikom registracije nužno je unijeti e-mail adresu, lozinku, specijalizaciju, ime i prezime. Ako korisnik odabire specijalizaciju računovođe, dodatno mora izabrati za koji tip dokumenta se opredjeljuje. Nakon uspješne registracije ili logiranja u sustav, korisnik ostaje zapamćen.

#### Zajedničke mogućnosti

Svi korisnici aplikacije imat će mogućnost skeniranja dokumenata pomoću ugrađene kamere na mobilnom uređaju te skenirani dokument potvrditi ili odbaciti, imati

pristup svojim skeniranim dokumentima unutar same aplikacije i moći se registrirati ili prijaviti u aplikaciju ovisno o tome je li korisnik već registriran u sustavu ili nije.

### **Zaposlenik**

Zadatak zaposlenika je skeniranje dokumenta te uz opciju skeniranja ima mogućnost preko gumba potvrditi ispravnost skeniranog dokumenta i tako poslati dokument na daljnju obradu revizoru ili ponovno skenirati isti dokument.

### **Revizor**

Zadatak revizora je pregledavanje svih poslanih dokumenata od strane zaposlenika. Za svaki dobiveni dokument dužan je odrediti tip dokumenta i preusmjeriti ga računovođi koji je za taj tip dokumenta zadužen. Ako revizor skenira dokument, aplikacija automatski prepoznaje tip dokumenta.

### **Računovođa**

Zadatak računovođe je arhiviranje dokumenata u bazu podataka. Dodatna opcija koju ima računovođa je slanje dokumenta na potpis direktoru prije arhiviranja. Nakon potpisa dokumenta računovođa dobiva obavijest da se dokument može arhivirati. Nakon otvaranja obavijesti, aplikacija automatski otvara dokument koji je potrebno potpisati i daje računovođi opciju za potpis.

### **Direktor**

Zadatak direktora je potpisivanje dokumenata koje mu pošalje računovođa. Potpisivanje se može provesti preko obavijesti koju dobije kada računovođa pošalje dokument na potpisivanje ili preko odabira dokumenta iz liste koja prikazuje povijest skeniranih dokumenata spremnih za potpisivanje. Također ima mogućnost generirati i pregledavati statistike za sve zaposlenike i pregledavati povijest skeniranih dokumenata svih zaposlenika.

## **2.3 OCR skener**

OCR (Optical Character Recognition) je glavna funkcionalnost aplikacije pomoću koje će se slike koje sadrže dokument skeniran pomoću kamere na mobilnom uređaju



pretvarati iz slikovnog u tekstualni format. OCR skener dopušta skeniranje samo onda kada su ispunjena 2 uvjeta. Jedan od uvjeta je da je unutar vidnog polja prisutan dokument, a drugi da mobilni uređaj miruje minimalno 0.5 sekundi što se postiže praćenjem senzora mobilnog uređaja.

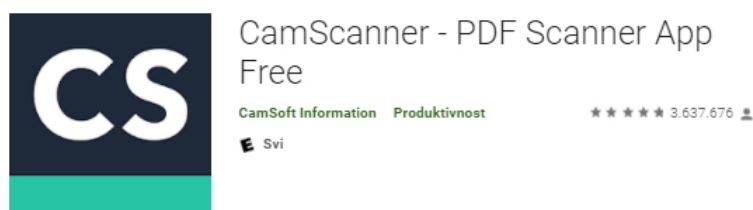
Aplikacija pomoću OCR skenera uz samo skeniranje prepoznaje određeni tip dokumenta i na temelju izvornog teksta dokumenta izlučuje određeni sadržaj. Tipovi dokumenata koje OCR skener prepoznaje su: računi, ponuda i interni dokument. Računi će u tekstu dokumenta imati oznaku koja se sastoji od velikog slova "R" te šest znamenaka, a sadržaj će se sastojati od imena klijenta, artikala s cijenama i ukupne cijene. Ponude će u tekstu dokumenta imati kao oznaku veliko slovo "P" i devet znamenaka, a sadržaj će se sastojati od artikala s cijenama i ukupne cijene. Interni dokument kao oznaku ima tri slova "INT" i četiri znamenke te nema određenu strukturu.

## 2.4 Ključni dijelovi sustava

Projekt će isporučiti javno dostupnu aplikaciju koja će se moći instalirati na većini suvremenih uređaja. Aplikacija zbog same prirode problematike koju rješava mora biti povezana na internet što uključuje bazu podataka koja mora podržavati veliki promet. Aplikacija zbog mogućnosti primanja obavijesti zahtjeva servis koji komunicira s bazom podataka u svrhu upravljanja obavijestima.

## 2.5 Slična rješenja

Sustav opisan ovdje trenutno na tržištu ne postoji, ali postoje brojne aplikacije koje imaju prevođenje slike dokumenata u tekstualni format i brojne druge usluge. Potražnja i vrijednost takvih aplikacija je velika i neupitna. Primjeri sličnih aplikacija dostupnih na tržištu su: CamScanner, OCR Text Scanner, Text Scanner i mnoge druge s preko 100 milijuna preuzimanja.



Slika 2.1: Primjer postojeće aplikacije

## 3. Specifikacija programske potpore

### 3.1 Funkcionalni zahtjevi

#### Dionici:

1. Zaposlenik
2. Revizor
3. Računovođa
4. Direktor

#### Aktori i njihovi funkcionalni zahtjevi:

1. Zaposlenik (inicijator) može:
  - (a) skenirati dokument OCR-om
  - (b) vidjeti vlastitu povijest skeniranja
  - (c) poslati skenirani dokument revizoru ako je točno skeniran
2. Revizor (inicijator) može:
  - (a) skenirati dokument OCR-om
  - (b) vidjeti vlastitu povijest skeniranja
  - (c) poslati skenirani dokument računovođama koji su za to namijenjeni
3. Računovođa (inicijator) može:
  - (a) skenirati dokument OCR-om
  - (b) vidjeti vlastitu povijest skeniranja
  - (c) arhivirati dokumente ili ih poslati na potpisivanje direktoru
  - (d) primiti direktorove obavijesti
4. Direktor (inicijator) može:
  - (a) skenirati dokument OCR-om
  - (b) vidjeti vlastitu povijest skeniranja
  - (c) vidjeti povijest svih dokumenata svih korisnika

- (d) vidjeti statistiku svih zaposlenika
- (e) primati računovodine obavijesti
- (f) potpisivati dokumente

### 3.1.1 Obrasci uporabe

#### Opis obrazaca uporabe

##### UC1 - Registracija

- **Glavni sudionik:** Zaposlenik, revizor, računovođa, direktor
- **Cilj:** Stvaranje stalnog računa te pristup onim funkcionalnostima aplikacije koje su korisniku potrebne
- **Sudionici:** - Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Korisnik upisuje potrebne identifikacijske podatke
  2. Korisnik odabire specijalizaciju iz ponuđene liste dostupnih specijalizacija
  3. Korisnik biva obaviješten o registraciji
- **Opis mogućih odstupanja:**
  - 1.a Korisnik upisuje podatke u neispravnom formatu
    1. Sustav iznad krivo upisanog teksta navodi grešku
    2. Korisnik prepravlja unos prema uputama
  - 3.a Korisnik je unio e-mail adresu koja već vezana za registriranog korisnika
    1. Sustav obavještava korisnika o greški i vraća ga na ponovnu registraciju
    2. Korisnik unosi drugu e-mail adresu ili se prijavljuje sa zadnje unešenom

##### UC1.1: Odabir specijalizacije revizora

- **Glavni sudionik:** Korisnik
- **Cilj:** Odabir vrste dokumenta za kojeg će korisnik biti zadužen
- **Sudionici:** - Baza podataka
- **Preduvjet:** - Korisnik je na zaslonu registracije odabrao specijalizaciju računovođe
- **Opis osnovnog tijeka:**
  1. Korisnik odabire za koji će tip dokumenta biti zadužen

##### UC2 - Prijava

- **Glavni sudionik:** Zaposlenik, revizor, računovođa, direktor
- **Cilj:** Prijava u sustav i početak korištenja usluga aplikacije

- **Sudionici:** - Baza podataka
- **Preduvjet:** - Korisnik je registriran
- **Opis osnovnog tijeka:**
  1. Korisnik upisuje potrebne identifikacijske podatke
  2. Korisnik je preusmjeren na početni zaslon aplikacije
- **Opis mogućih odstupanja:**
  - 1.a Korisnik upisuje neispravne podatke
    1. Sustav u obliku pop-up teksta ispisuje poruku o neispravno unesenim podacima
    2. Korisnik prepravlja lozinku ili e-mail adresu

### UC3 - Odjava

- **Glavni sudionik:** Zaposlenik, revizor, računovođa, direktor
- **Cilj:** Odjava iz aplikacije u slučaju prijave drugog korisnika
- **Sudionici:** - Baza podataka
- **Preduvjet:** - Korisnik je prijavljen
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju "user info" na glavnom izborniku
  2. Korisnik se odjavljuje pritiskom na gumb za odjavu iz aplikacije
  3. Korisnik biva preusmjeren na zaslon aplikacije za prijavu

### UC4 - Skeniranje dokumenta

- **Glavni sudionik:** Zaposlenik, revizor, računovođa, direktor
- **Cilj:** Ispravno skenirati dokument te ga pohraniti u bazu podataka i u ovisnosti o specijalizaciji proslijediti ga drugom korisniku
- **Sudionici:** - Baza podataka
- **Preduvjet:** - Korisnik ima kameru na mobilnom uređaju
- **Opis osnovnog tijeka:**
  1. Korisnik na glavnom izborniku odabire opciju "Scan"
  2. Korisnik uređaj pozicionira u skladu s dokumentom
  3. Korisnik čeka izvjesno vrijeme dok se uređaj ne stabilizira
  4. Korisnik pritiska gumb za okidanje slike
  5. Korisnik nakon provedenog OCR algoritma potvrđuje ispravnost skeniranja
  6. Korisnik o ovisnosti o specijalizaciji šalje dokument drugom korisniku ili ga arhivira

7. Korisnik biva preusmjeren na zaslon spreman za ponovno skeniranje
- **Opis mogućih odstupanja:**
  - 1.a Korisnik nema kameru na mobilnom uređaju
    1. Na zaslonu se ispisuje poruka o nemogućnosti korištenja usluge skeniranja
  - 6.a Korisnik nije zadovoljan skeniranim dokumentom
    1. Korisnik odabire opciju odbacivanja skeniranog dokumenta
    2. Korisnik ponovno skenira dokument

### UC5 - Pregled dokumenata

- **Glavni sudionik:** Zaposlenik, revizor, računovođa, direktor
- **Cilj:** Pregled onih dokumenata za koje je određeni korisnik odgovoran
- **Sudionici:** - Baza podataka
- **Preduvjet:** - Korisnik je prijavljen u sustav
- **Opis osnovnog tijeka:**
  1. Korisnik na glavnom izborniku odabire opciju "History"
  2. Korisnik na temelju specijalizacije odabire opcije prikaza dokumenata
  3. Korisniku se prikazuju dokumenti
- **Opis mogućih odstupanja:**
  - 3.a Korisnik za odabranu opciju nema dostupnih dokumenata
    1. Na zaslonu se umjesto popisa dokumenata ispisuje poruka

#### UC5.1 - Pregled svih dokumenata

- **Glavni sudionik:** Direktor
- **Cilj:** Pregled svih skeniranih dokumenata svih korisnika
- **Sudionici:** - Baza podataka
- **Preduvjet:** - Korisnik je prijavljen u sustav i ima nivo autorizacije direktora
- **Opis osnovnog tijeka:**
  1. Korisnik na glavnom izborniku odabire opciju "History"
  2. Korisnik iz izbornika filtera odabire opciju prikaza svih dokumenata
  3. Korisniku se prikazuju svi dokumenti svih korisnika
- **Opis mogućih odstupanja:**
  - 3.a Korisnik za odabranu opciju nema dostupnih dokumenata
    1. Na zaslonu se umjesto popisa dokumenata ispisuje poruka

#### UC5.2 - Pregled dokumenata za potpisivanje

- **Glavni sudionik:** Direktor
- **Cilj:** Pregled i potpisivanje dokumenata koje je prije arhiviranja potrebno potpisati
- **Sudionici:** - Baza podataka
- **Preduvjet:** - Korisnik je prijavljen u sustav i ima nivo autorizacije direktora
- **Opis osnovnog tijeka:**
  1. Korisnik na glavnom izborniku odabire opciju "History"
  2. Korisnik iz izbornika filtera odabire opciju prikaza dokumenata za potpisivanje
  3. Korisniku se prikazuju svi dokumenti koji su spremni za potpisivanje
- **Opis mogućih odstupanja:**
  - 3.a Korisnik za odabranu opciju nema dostupnih dokumenata
    1. Na zaslonu se umjesto popisa dokumenata ispsuje poruka

### UC5.3 - Pregled dokumenata za arhiviranje

- **Glavni sudionik:** Računovođa
- **Cilj:** Pregled svih dokumenata spremnih za arhiviranje
- **Sudionici:** - Baza podataka
- **Preduvjet:** - Korisnik je prijavljen u sustav i ima nivo autorizacije računovođe
- **Opis osnovnog tijeka:**
  1. Korisnik na glavnom izborniku odabire opciju "History"
  2. Korisnik iz izbornika filtera odabire opciju prikaza dokumenata spremnih za arhiviranje
  3. Korisniku se prikazuju svi dokumenti spremni za arhiviranje
- **Opis mogućih odstupanja:**
  - 3.a Korisnik za odabranu opciju nema dostupnih dokumenata
    1. Na zaslonu se umjesto popisa dokumenata ispsuje poruka

### UC5.4 - Pregled dokumenata za reviziju

- **Glavni sudionik:** Revizor
- **Cilj:** Pregled dokumenata spremnih za reviziju
- **Sudionici:** - Baza podataka
- **Preduvjet:** - Korisnik je prijavljen u sustav i ima nivo autorizacije revizora
- **Opis osnovnog tijeka:**
  1. Korisnik na glavnom izborniku odabire opciju "History"

2. Korisnik iz izbornika filtera odabire opciju prikaza dokumenata sprem-  
nih za reviziju
3. Korisniku se prikazuju svi dokumenti svih korisnika
- **Opis mogućih odstupanja:**
  - 3.a Korisnik za odabranu opciju nema dostupnih dokumenata
    1. Na zaslonu se umjesto popisa dokumenata ispsuje poruka

#### UC5.5 - Pregled vlastito skeniranih dokumenata

- **Glavni sudionik:** Zaposlenik, revizor, računovođa, direktor
- **Cilj:** Pregled svih dokumenata koje je skenirao trenutno prijavljeni korisnik
- **Sudionici:** - Baza podataka
- **Preduvjet:** - Korisnik je prijavljen u sustav
- **Opis osnovnog tijeka:**
  1. Korisnik na glavnom izborniku odabire opciju "History"
  2. Korisnik iz izbornika filtera odabire opciju prikaza svih vlastito skenira-  
nih
  3. Korisniku se prikazuju svi dokumenti svih korisnika
- **Opis mogućih odstupanja:**
  - 3.a Korisnik za odabranu opciju nema dostupnih dokumenata
    1. Na zaslonu se umjesto popisa dokumenata ispsuje poruka

#### UC6 - Pregled korisničkih podataka

- **Glavni sudionik:** Zaposlenik, revizor, računovođa, direktor
- **Cilj:** Pregled korisničkih podataka vezanih za trenutno prijavljenog koris-  
nika
- **Sudionici:** - Baza podataka
- **Preduvjet:** - Korisnik je prijavljen u sustav
- **Opis osnovnog tijeka:**
  1. Korisnik na glavnom izborniku odabire opciju "Info"
  2. Korisniku se prikazuju korisnički podaci

#### UC7 - Slanje dokumenta revizoru

- **Glavni sudionik:** Zaposlenik
- **Cilj:** Slanje skeniranog dokumenta odgovarajućem korisniku u svrhu daljnje  
obrade
- **Sudionici:** - Baza podataka



- **Preduvjet:** - Korisnik je prijavljen u sustav i ima nivo autorizacije zaposlenika
- **Opis osnovnog tijeka:**
  1. Korisnik na glavnom izborniku odabire opciju "scan"
  2. Korisnik skenira dokument
  3. Korisniku potvrđuje kvalitetu skeniranog dokumenta
  4. Korisnik šalje dokument revizoru

#### **UC8 - Slanje dokumenta računovodi**

- **Glavni sudionik:** Revizor
- **Cilj:** Slanje dokumenta računovođama koji su zaduženi za određeni tip skeniranog dokumenta
- **Sudionici:** - Baza podataka
- **Preduvjet:** - Korisnik je prijavljen i ima nivo autorizacije revizora
- **Opis osnovnog tijeka:**
  1. Korisnik prilikom pregleda dokumenata odabire dokument spreman za reviziju
  2. Korisnik odabire tip dokumenta kako bi dokument bio prosljeđen ispravnim računovođama
- **Opis mogućih odstupanja:**
  - 1.a Korisnik kod pregleda dokumenata za odabranu opciju nema dostupnih dokumenata
    1. Na zaslonu se umjesto popisa dokumenata ispisuje poruka
- **Alternativni tijek:**
  - 1.a Korisnik skenira dokument
  - 2.a Ako je korisnik skenirao dokument, aplikacija sama prepoznaje tip dokumenta

#### **UC9 - Arhiviranje dokumenata**

- **Glavni sudionik:** Računovođa
- **Cilj:** Spremanje potpuno obrađenih dokumenata u bazu podataka
- **Sudionici:** - Baza podataka
- **Preduvjet:** - Korisnik je prijavljen u sustav i ima nivo autorizacije računovođe
- **Opis osnovnog tijeka:**
  1. Korisnik odabire pregled dokumenata
  2. Korisnik odabire opciju prikaza dokumenata dostupnih za arhiviranje
  3. Korisnik odabire dokument koji želi arhivirati

#### 4. Korisnik arhivira zadani dokument

- **Alternativni tijek:**

##### 1-3.a Korisnik dobiva obavijest

1. Korisnik pritiskom na obavijest dolazi do zaslona na kojem se prikazuje dokument

### UC10 - Slanje dokumenta na potpis

- **Glavni sudionik:** Računovođa
- **Cilj:** Pregled svih skeniranih dokumenata svih korisnika
- **Sudionici:** - Baza podataka
- **Preduvjet:** - Korisnik je prijavljen u sustav i ima nivo autorizacije računovođe
- **Opis osnovnog tijeka:**

1. Korisnik odabire opciju "History"
2. Korisnik iz izbornika filtera odabire opciju prikaza dokumenata za potpis
3. Korisnik odabire dokument koji želi poslati na potpis
4. korisnik šalje dokument na potpis

- **Alternativni tijek:**

##### 1-3.a Korisnik skenira dokument

1. Korisnik odabire opciju "Scan"
2. Korisnik skenira dokument
3. Korisnik potvrđuje kvalitetu skeniranog dokumenta
4. Korisnik odabire opciju slanja dokumenta na potpis

- **Opis mogućih odstupanja:**

4. Korisnik odabire opciju arhiviranja dokumenta
  1. Korisnik biva prebačen na zaslon skeniranja

### UC11 - Primanje obavijesti

- **Glavni sudionik:** Direktor, računovođa
- **Cilj:** Brzi pristup dokumentu kojega je potrebno obraditi
- **Sudionici:** - Baza podataka
- **Preduvjet:** - Korisniku je poslan dokument od strane drugog korisnika
- **Opis osnovnog tijeka:**
  1. Korisnik otvara obavijest
  2. Korisnik biva preusmjeren na zaslon koji sadrži podatke o dokumentu
  3. Korisnik obrađuje dokument na način na koji je za to zadužen

- **Opis mogućih odstupanja:**

- 1.a Korisnik briše obavijest

- 1. Korisnik nastavlja rad, a dokument se obrađuje klikom na određeni dokument na zaslonu pregleda dokumenata

### UC12 - Potpisivanje dokumenta

- **Glavni sudionik:** Direktor

- **Cilj:** Označavanje dokumenta potpisanim

- **Sudionici:** - Baza podataka

- **Preduvjet:** - Korisnik je prijavljen u sustav i ima nivo autorizacije direktora

- **Opis osnovnog tijeka:**

- 1. Korisnik na glavnom izborniku odabire opciju "History"
  - 2. Korisnik iz izbornika filtera odabire opciju prikaza dokumenata za potpisivanje
  - 3. Korisniku se prikazuju svi dokumenti za potpisivanje
  - 4. Korisnik odabire dokument koji želi potpisati
  - 5. Korisniku se prikazuju podaci odabranog dokumenta
  - 6. Korisnik potpisuje dokument

- **Opis mogućih odstupanja:**

- 4.a Dokument je već potpisan od strane drugog računovođe

- 1. Korisnik nastavlja rad

- **Alternativni koraci:**

- 1-4.a Korisnik je dobio obavijest

- 1. Korisnik pritišće na obavijest
    - 2. Korisniku se prikazuju podaci dokumenta referenciranog u obavijesti

### UC13 - Pregled statistike

- **Glavni sudionik:** Direktor

- **Cilj:** Uvid u statistiku zaposlenika

- **Sudionici:** - Baza podataka

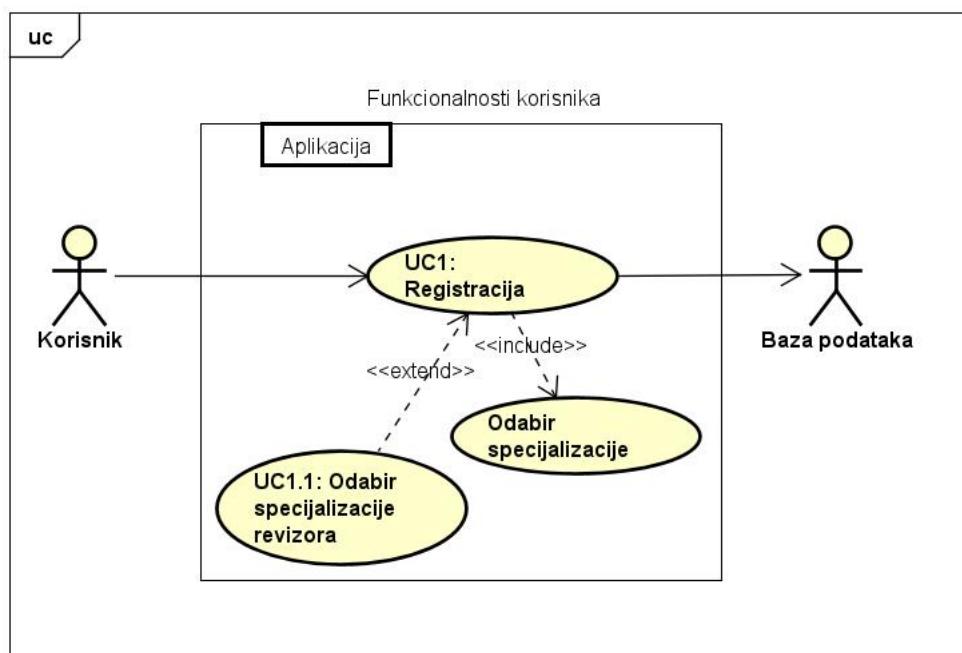
- **Preduvjet:** - Korisnik je prijavljen u sustav i ima nivo autorizacije direktora

- **Opis osnovnog tijeka:**

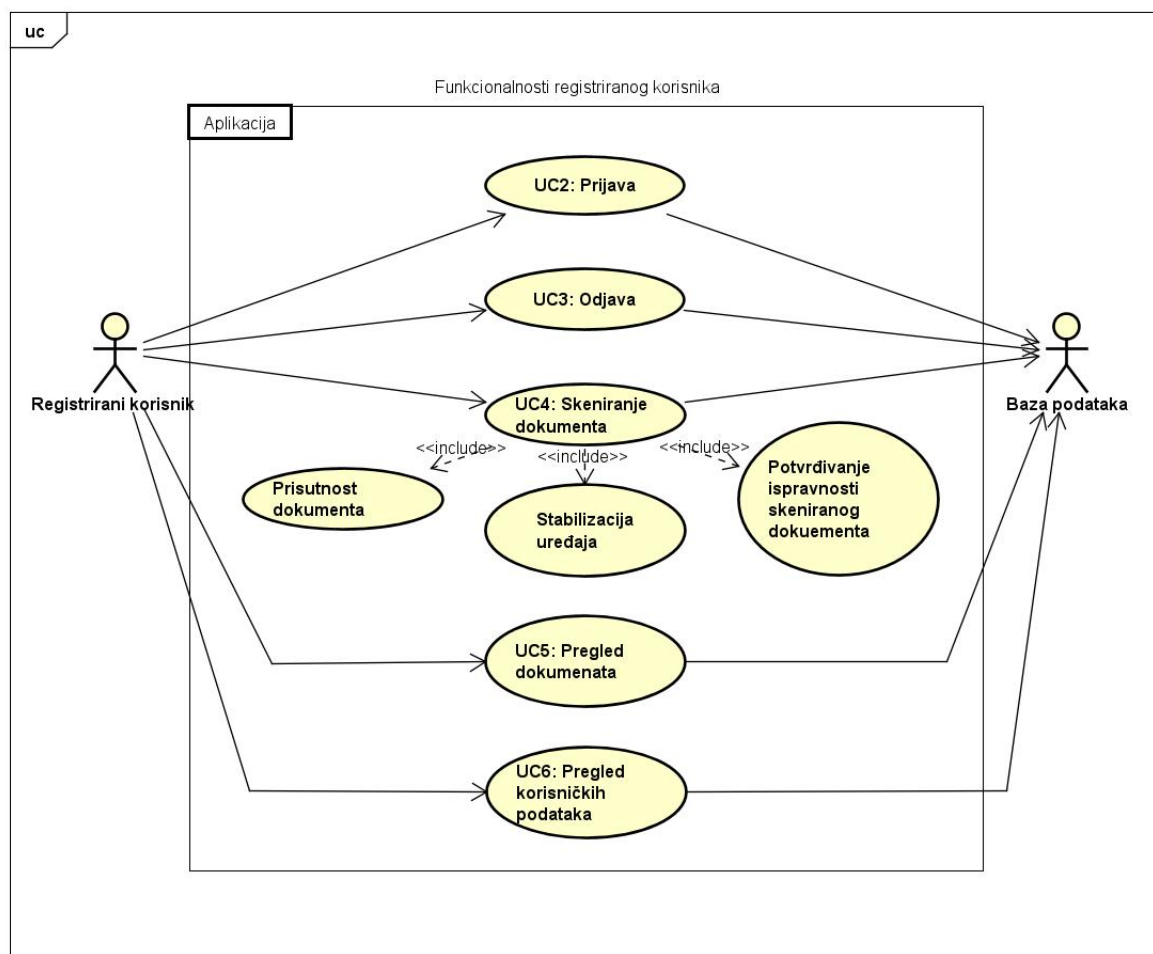
- 1. Korisnik na glavnom izborniku odabire opciju "Info"
  - 2. Korisnik odabire opciju "Statistics"
  - 3. Korisniku se prikazuje lista svih zaposlenika

4. Korisnik odabire zaposlenika za kojega želi prikazati statistiku
5. Korisniku se prikazuje zaslon s odgovarajućom statistikom

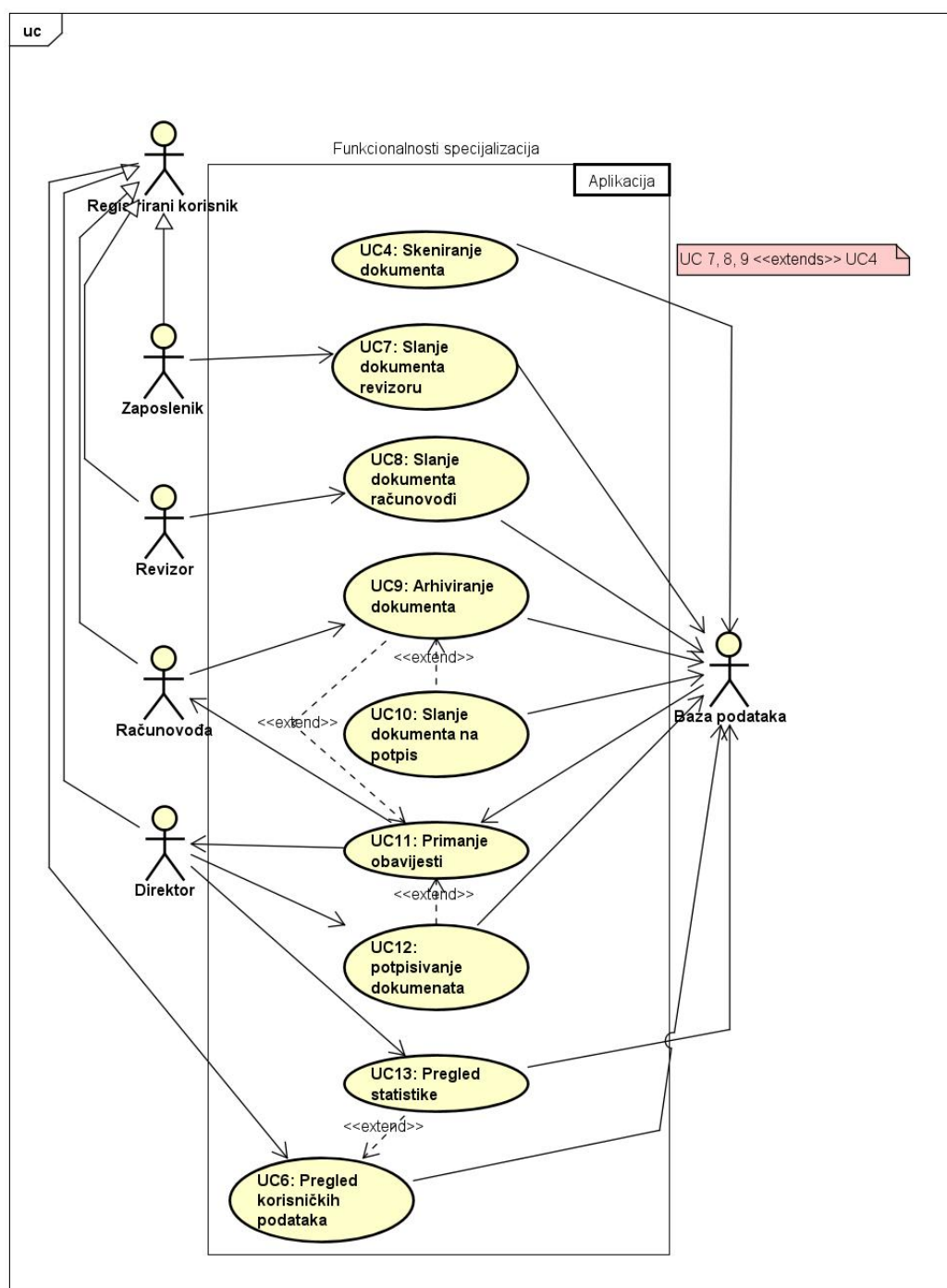
### Dijagrami obrazaca uporabe



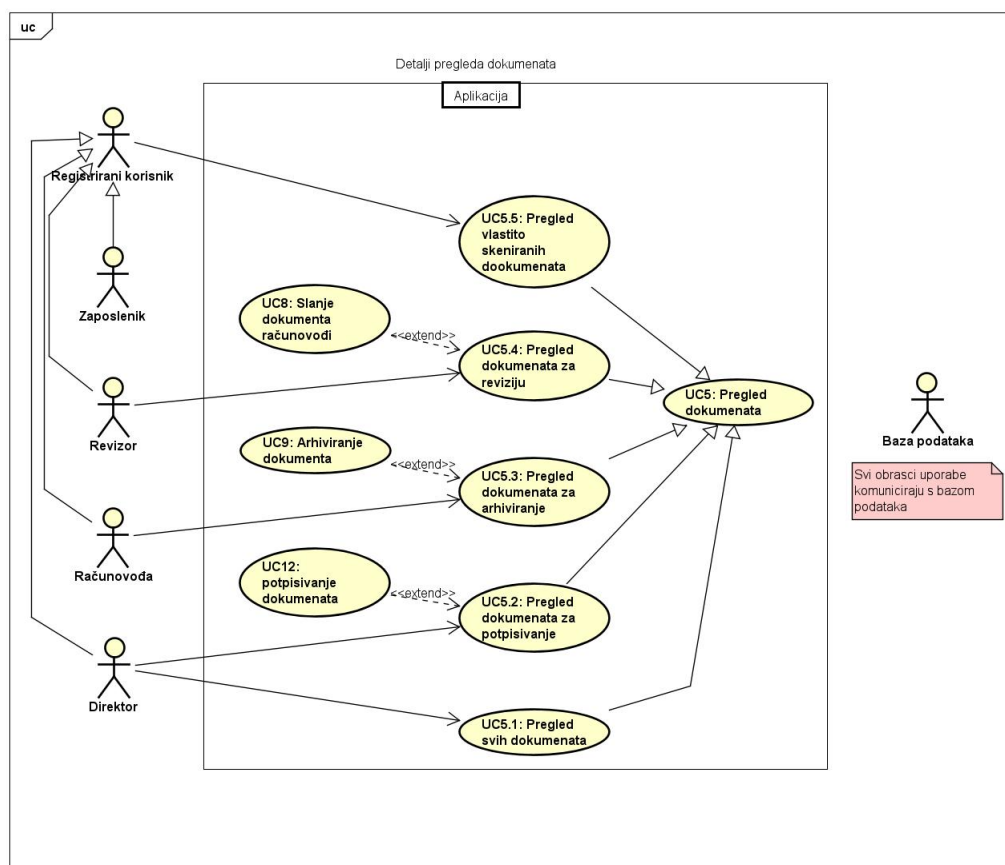
Slika 3.1: Dijagram obrasca uporabe, funkcionalnosti korisnika



Slika 3.2: Dijagram obrasca uporabe, funkcionalnosti registriranog korisnika



Slika 3.3: Dijagram obrasca uporabe, funkcionalnosti specijalizacija



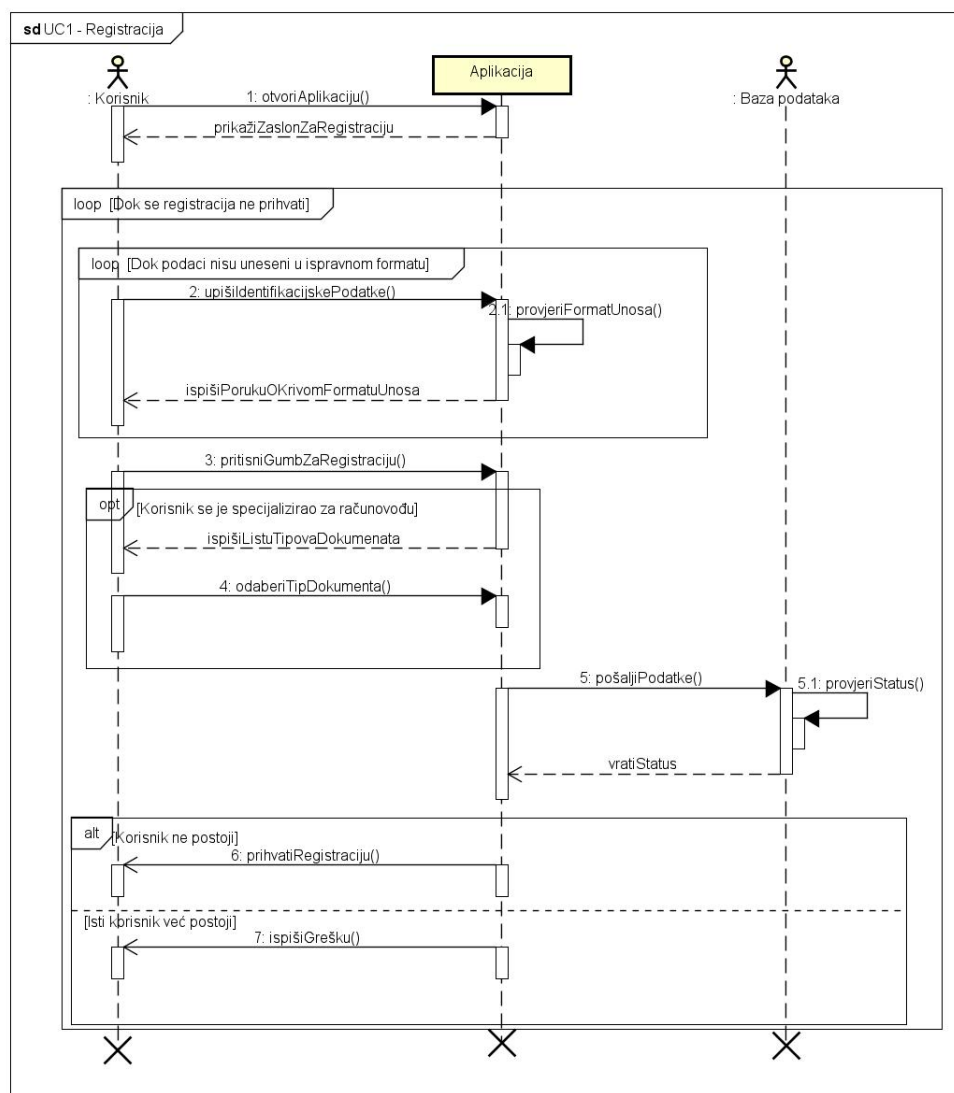
Slika 3.4: Dijagram obrasca uporabe, pregled dokumenata

### 3.1.2 Sekvencijski dijagrami

#### Obrazac uporabe UC1 - Registracija

Korisnik otvara aplikaciju te mu aplikacija u slučaju da nitko na uređaju nije prijavljen u sustav prikazuje zaslon za registraciju novog korisnika. Dok korisnik unosi identifikacijske podatke aplikacija konstantno provjerava ispravnost formata unesenih podataka kao što je dovoljno jaka lozinka ili ispravan format e-mail adrese te u slučaju detektiranja greške ispisuje odgovarajuću poruku iznad polja u kojem je greška. Nakon što korisnik unese podatke ispravnog formata, pritišće gumb za registraciju. Ako se je korisnik specijalizirao kao računovođa, aplikacija mu ispisuje izbornik sa svim tipovima dokumenata koji su dostupni. Korisnik odabire dokument za koji će biti zadužen te potvrđuje odabir. Aplikacija unesene podatke šalje u bazu podataka gdje se podaci provjeravaju. Ako baza podataka ustanovi da korisnik s istom e-mail adresom već postoji, registracija se odbacuje i korisniku se ispisuje odgovarajuća poruka. Ako je registracija uspjela, korisnik biva preusmjeren na početni zaslon aplikacije.



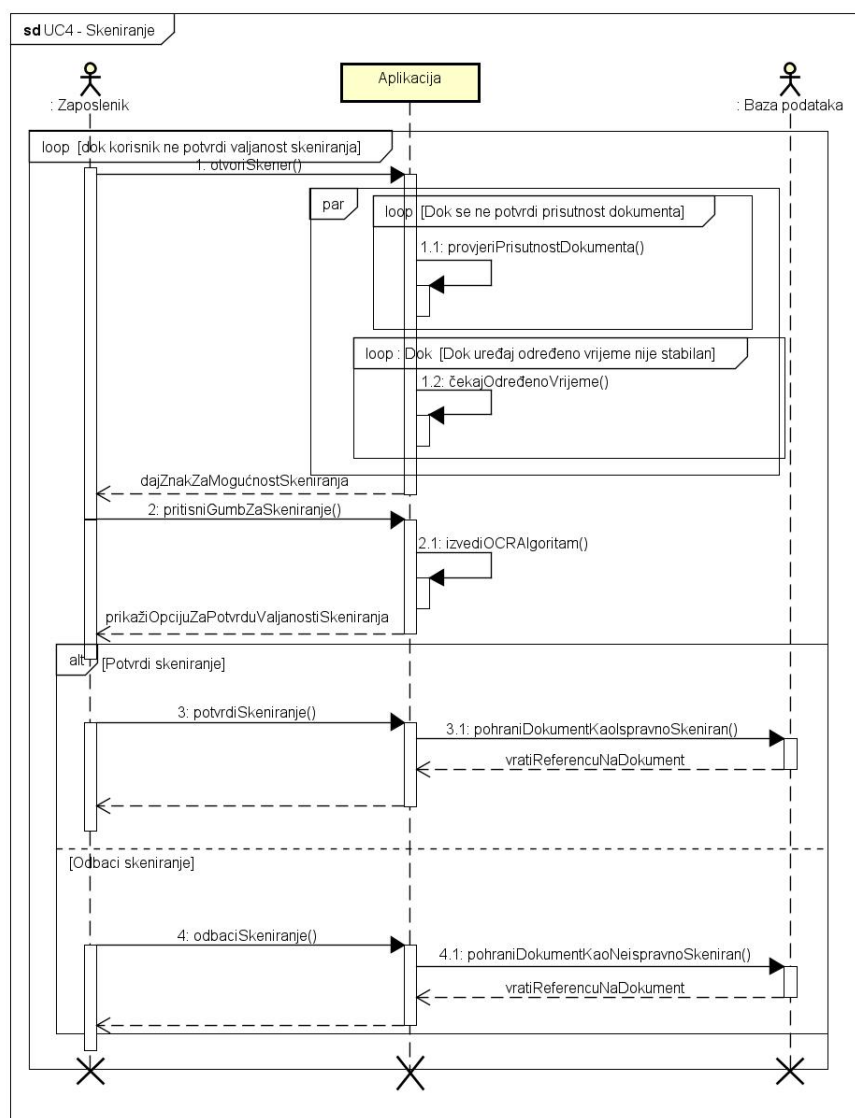


Slika 3.5: Sekvencijalni dijagram, UC1

### Obrazac uporabe UC4 - Skeniranje

Korisnik na glavnom izborniku odabire opciju skeniranja te mu aplikacija na zaslonu prikazuje pogled kamere. Aplikacija korisniku ne dozvoljava skeniranje prije nego što su zadovoljena dva uvjeta. Prvi uvjet je da se u pogledu kamere nalazi dokument. Drugi uvjet je ispunjen kada aplikacija u određenom periodu vremena ne izmjeri vibracije sa senzora na mobilnom uređaju. Kada je dokument prisutan i uređaj je stabilan, aplikacija korisniku omogućuje okidanje fotografije. Aplikacija nakon okidanja fotografije izvodi OCR algoritam. Nakon izvedenog algoritma aplikacija korisniku daje opciju prihvatanja ili odbacivanja skeniranog dokumenta na temelju kvalitete skeniranja. Ako korisnik odbaci skenirani dokument, aplika-

cija dokument pohranjuje u bazu podataka s oznakom da je neispravno skeniran. Ako korisnik prihvati skenirani dokument, aplikacija u bazu podataka pohranjuje dokument s oznakom da je ispravno skeniran.



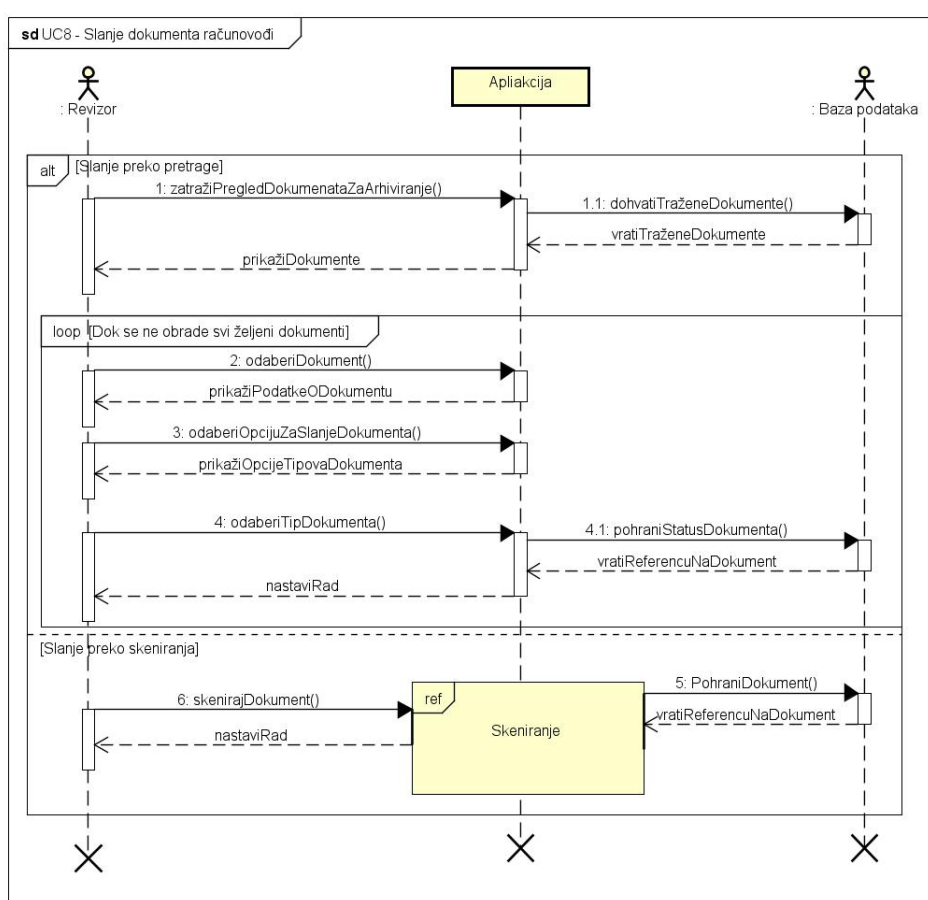
Slika 3.6: Sekvencijalni dijagram, UC4

### Obrazac uporabe UC8 - Slanje dokumenta računovođi

Revizor dokument računovođi na arhiviranje može poslati na dva načina. U slučaju da računovođi treba poslati dokument koji je već skeniran od strane zaposlenika, revizor u pregledu dokumenata odabire filter za prikaz dokumenata spremnih za slanje računovođi. Aplikacija iz baze podataka dohvaća tražene dokumente i prikazuje ih revizoru. Revizor na izborniku odabire željeni dokument. Aplikacija

otvara izabrani dokument i prikazuje ga revizoru. Revizor odabire opciju slanja dokumenta računovođi. Budući da je svaki računovođa zadužen za određeni tip dokumenta, aplikacija revizoru prikazuje izbornik koji sadrži popis svih tipova dokumenta. Revizor izabire onaj tip dokumenta koji odgovara skeniranom dokumentu. Aplikacija u bazi podataka mijenja status dokumenta što dokument čini vidljivim računovođi. Revizor ponavlja radnju sve dok ne pošalje sve željene dokumente.

U slučaju da revizor treba poslati dokument kojeg još nije skenirao zaposlenik, revizor treba skenirati dokument sam. Nakon skeniranja aplikacija dokumentu automatski odredi tip i pohrani ga u bazu podataka.



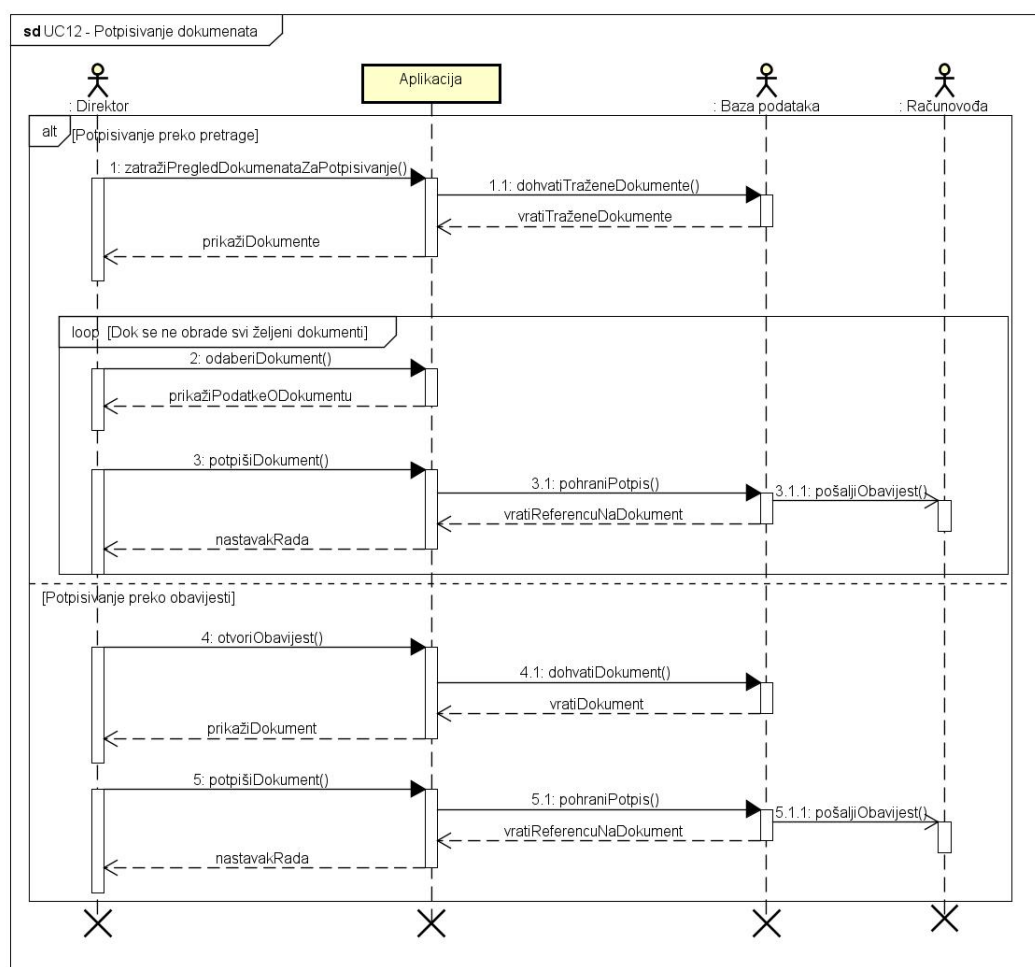
Slika 3.7: Sekvencijalni dijagram, UC8

### Obrazac uporabe UC12 - Potpisivanje dokumenata

Računovođa dokument može potpisati na dva načina. Jedan način je da direktor nakon što računovođa pošalje dokument na potpisivanje dobije obavijest. Aplikacija direktoru prikazuje obavijest na alatnoj traci mobilnog uređaja. Nakon što

direktor otvara obavijest, aplikacija dohvaća dokument referenciran u obavijesti iz baze podataka i prikazuje ga na zaslonu. Direktor odabire opciju potpisivanja dokumenta. Aplikacija u bazu podataka pohrani potpis gdje se aktivira slanje obavijesti računovođi koji je zadužen za arhiviranje istog dokumenta. Direktor nastavlja rad.

Drugi način je da direktor sam dolazi do dokumenta kojega treba potpisati. Direktor u pregledu povijesti dokumenata odabire filter kojim od aplikacije traži prikaz samo onih dokumenata spremnih za potpisivanje. Aplikacija prikazuje tražene dokumente. Direktor odabire željeni dokument te ga potpisuje. Aplikacija potpis sprema u bazu podataka što uzrokuje slanje obavijesti računovođi koji je zadužen za arhiviranje istog dokumenta. Direktor ponavlja radnje sve dok ne završi s potpisivanjem svih željenih dokumenata.



Slika 3.8: Sekvencijalni dijagram, UC12

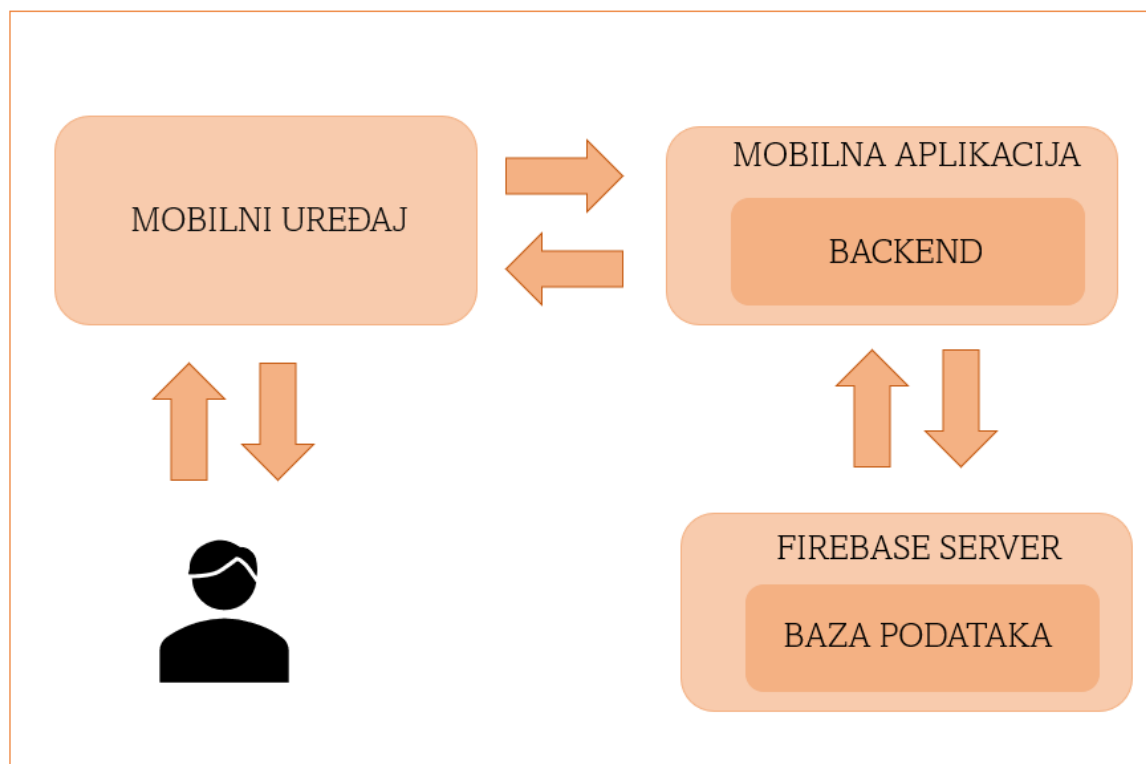
## 3.2 Ostali zahtjevi

- Aplikacija mora omogućiti primanje obavijesti i kada korisnik nema otvorenu instancu aplikacije
- Izvođenje OCR algoritma ne smije trajati duže od nekoliko sekundi
- Sustav mora podržavati rad od 50 do 100 korisnika u stvarnom vremenu
- Aplikacija za detektiranje prisutnosti dokumenta mora koristiti detekciju rubova
- Sustav za praćenje broja skeniranih dokumenata koristi pomoćnu strukturu podataka kako bi se smanjio broj upita u bazu podataka
- Sustav za prikaz velikog broja dokumenata mora koristiti paginaciju (*eng. pagination*)
- Aplikacija prilikom ponovnog otvaranja mora pamtit i zadnje prijavljenog korisnika
- Aplikacija prilikom pregleda dokumenata mora omogućiti naprednije filtriranje i sortiranje
- Aplikacija mora podržavati prijavu različitih korisnika na istom uređaju

## 4. Arhitektura i dizajn sustava

Arhitektura se može podijeliti na sljedeće podsustave:

- Mobilni uređaj
- Mobilna aplikacija
- Baza podataka



Slika 4.1: Arhitektura sustava

**Mobilni (Android) uređaj** je uređaj koji korisniku omogućuje pokretanje Android mobilne aplikacije. Aplikacija je napisana u kodu te nakon toga uređaj taj kod interpretira i prikazuje ga korisniku kao grafičko sučelje. Pomoću uređaja korisnik interaktira s aplikacijom i preko nje šalje upite bazi podataka.

**Mobilna aplikacija** je glavni dio arhitekture. Korisnik istu koristi kako bi obradio svoje zahtjeve. Ona se sastoji od *Frontend* i *Backend* dijela te pomoću *Backenda* komunicira s bazom podataka, čitajući ili pišući podatke u nju. Nakon toga se korisniku preko *Frontend* dijela prikazuju potrebni podaci.

Programski jezik koji smo koristili za izradu naše mobilne je Kotlin, koji je Google 2019. proglasio kao preferirani jezik izrade Android aplikacija. Razvojno okruženje s najboljom podrškom za izradu takvih aplikacija je Android Studio, koji smo i mi upotrebljavali. Jedan dio *Backend* programskog koda napisan je u JavaScript programskom jeziku, u Node.js okruženju. Arhitektura sustava se temelji na MVVM (Model-View-Viewmodel) konceptu, koji je dio svake Android aplikacije.

Odlika MVVM-a je što pomaže organizirati kod i podijeliti program u manje dijelove kako bi razvoj, ažuriranje i ponovna upotreba koda bila jednostavnija i brža. MVVM se sastoji od:

- **Model** - odnosi se ili na model domene, koji predstavlja sadržaj stvarnog stanja (objektno orijentirani pristup), ili na sloj pristupa podacima, koji predstavlja sadržaj.
- **View** - struktura, raspored i izgled onoga što korisnik vidi na zaslonu. Prikazuje model i prima interakciju korisnika s prikazom (klikovi mišem, unos tipkovnice, dodirivanja zaslona, itd.).
- **Viewmodel** - nalazi se između Viewa i Modela. Ovdje su smještene kontrole za interakciju s Viewom, dok se *binding* koristi za povezivanje elemenata korisničkog sučelja u Viewu s kontrolama u ViewModelu.

## 4.1 Baza podataka

Za potrebe našeg sustava koristit ćemo nerelacijsku bazu podataka (NoSQL) ugrađenu u Firebase alat koji olakšava rad s podacima. Firebase nudi nekoliko vrsta baza, a mi smo izabrali Firestore verziju. Baza se sastoji od nekoliko kolekcija koja se definiše imenom i skupom dokumenata, a svaki dokument ima svoje atribute. Atributi se sastoje od imena, tipa podataka i samog podatka. Temelje se na modelu ključ-vrijednost. Kako svaki dokument unutar kolekcije može imati različite atribute, važno je održati konzistentnost podataka, baza mora biti dosljedna i to se postiže kreiranjem modela unutar aplikacije. Na taj način će svi dokumenti unutar kolekcije imati iste atribute. Baza se sastoji od sljedećih kolekcija:

- documentArchive
- documentStatus
- documentByUser
- documents
- users
- tokens

#### 4.1.1 Opis tablica

**Napomena:** Iz razloga što se u službenoj Firebase dokumentaciji koriste izrazi *collection* i *document* te zato što se naša aplikacija bavi skeniranjem dokumenata, može doći do zabune kad je koji izraz za dokument upotrijebljen (Firebaseov ili iz aplikacije). Stoga će se u daljnjem tekstu za izraz koji se odnosi na fizičke (skenirane) dokumente koristiti riječ **dokument**, a za Firebaseov član kolekcije riječ **document**.

**documentArchive** Ova kolekcija sadrži informaciju o arhiviranju dokumenata. Dokumentu koji je arhiviran, tj. spremljen u bazu, se automatski generira identifikator te *documenti* u *documentArchive* kolekciji dobivaju iste te identifikatore, kako bi se znalo za koji skenirani dokument sadrži podatke. Kolekcija se sastoji od atributa *archiveId* i *archivedOn*, koji govori vrijeme kreiranja arhive. Svakom *documentu* iz kolekcije *documents* pripada jedan *document* iz kolekcije *documentArchive*.

documentArchive		
documentId	string	Jedinstveni identifikator arhive. Jednak id-u dokumenta koji je arhiviran
archiveId	string	
archivedOn	timestamp	datum i vrijeme arhiviranja dokumenta

**documentStatus** Ova se kolekcija sastoji od nekoliko atributa vezanih za status dokumenta. Sadrži attribute *archived*, *revised*, *scannedProperly*, *toBeSigned*, *signed*. Primjerice, nakon što se dokument uspješno skenira, vrijednosti *archived*, *scannedProperly* i *toBeSigned* se postavljaju na *true*. U slučaju da u aplikaciji treba filtrirati dokumente koji zadovoljavaju jedan ili više statusa, ova kolekcija dolazi u uporabu.



Za svaki *document* iz kolekcije *documents* postoji jedan *document* iz ove kolekcije, s istim identifikatorom.

documentStatus		
documentId	string	Jedinstveni identifikator. Jednak id-u dokumenta koji je skeniran
archived	boolean	oznaka je li dokument arhiviran
revised	boolean	oznaka je li dokument reviziran
scannedProperly	boolean	oznaka je li dokument pravilno skeniran
signed	boolean	oznaka je li dokument potpisan
toBeSigned	boolean	oznaka je li dokument poslan na slanje

**documentByUser** Kolekcija koja informira o detaljnijim podacima dokumenta, tj. prikazuje koji je *user* napravio koji korak u protoku dokumenta kroz aplikaciju. Sadrži attribute *archivedBy*, *revisedBy*, *sentToSignBy*, *signedBy*. U polju svakog od atributa nalazi se id nekog *usera*. Za svaki *document* iz kolekcije *documents* postoji jedan *document* iz ove kolekcije, s istim identifikatorom.

documentByUser		
documentId	string	Jedinstveni identifikator. Jednak id-u dokumenta koji je skeniran
archivedBy	string	id usera koji je arhivirao dokument
revisedBy	boolean	id usera koji je revizirao dokument
toBeSignBy	boolean	id usera koji je dokument poslao na potpisivanje
signedBy	boolean	id usera koji je potpisao dokument

**documents** Kolekcija koja sadrži informacije o skeniranim dokumentima. Postoji nekoliko specijalizacija dokumenta, a to su **Račun**, **Ponuda** i **Interni dokument**. Dokumenti sadrže zajednički skup atributa, *createdOn*, *type*, *user*, *label* te svaka od specijalizacija ima svoje zasebne attribute. Svakom dokumentu se može pridružiti jedan *document* iz kolekcija *documentArchive*, *documentStatus* i *users*. Identifikator

pojednog *usera* se u ovoj kolekciji sprema pod atribut *user*.

documents		
documentId	string	Jedinstveni identifikator skeniranog dokumenta
createdOn	timestamp	datum i vrijeme kreiranja dokumenta
label	string	oznaka s dokumenta
type	string	tip dokumenta: Račun, Ponuda, Interni dokument
user	string	ID korisnika koji je obavio skeniranje
clientName	string	ime i prezime klijenta s Računa
items	map of numbers	artikli s Računa ili Ponude
totalPrice	number	ukupna cijena artikala s Računa ili Ponude
content	string	sadržaj Internog dokumenta

**users** Ova kolekcija sadrži informacije o korisnicima aplikacije koji su registrirani. Pri prvoj uporabi aplikacije svaki korisnik mora proći registraciju. U kolekciji se pohranjuju korisnikov email, ime, prezime i uloga. Korisnik može biti Direktor, Računovođa, Revizor ili Zaposlenik. Svoju ulogu korisnik odabire pri registraciji. Svaki korisnik ima svoj ID i on se sprema u kolekciju *documents*, kako bi se znalo koji korisnik je skenirao koji dokument.

users		
documentId	string	Jedinstveni identifikator registriranog korisnika
email	string	email adresa korisnika
firstName	string	ime korisnika
lastName	string	prezime korisnika

Nastavljeno na idućoj stranici

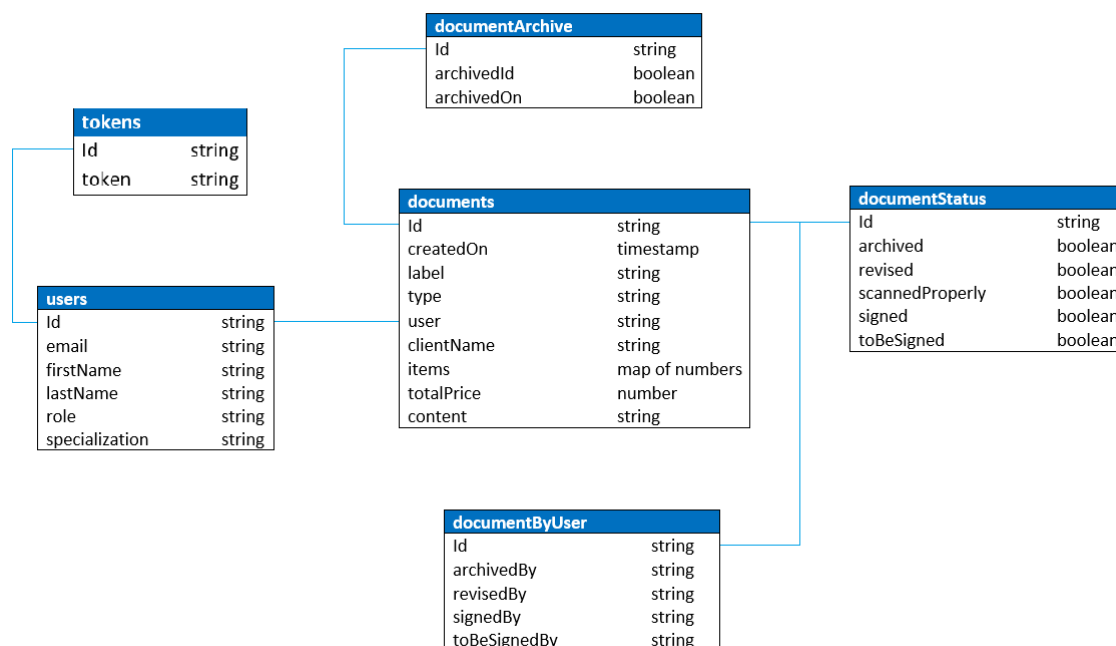
Nastavljeno od prethodne stranice

users		
role	string	uloga korisnika
specialization	string	opcionalna specijalizacija Računovođe

**tokens** Kolekcija s *Firebase Cloud Messaging* tokenima. Ona omogućava slanje notifikacijama onim tokenima koji su u tom trenutku aktivni ili zadovoljavaju određene uvjete. Sastoji se od samo jednog atributa, *token*, a jedinstveni identifikator tokena je jednak identifikatoru jednom *documentu* iz kolekcije *users*.

tokens		
documentId	string	Jedinstveni identifikator tokena. Jednak id-u usera kojem pripada.
token	string	Identifikator dodijeljen od <i>Firebase Cloud Messaginga</i>

#### 4.1.2 Dijagram baze podataka

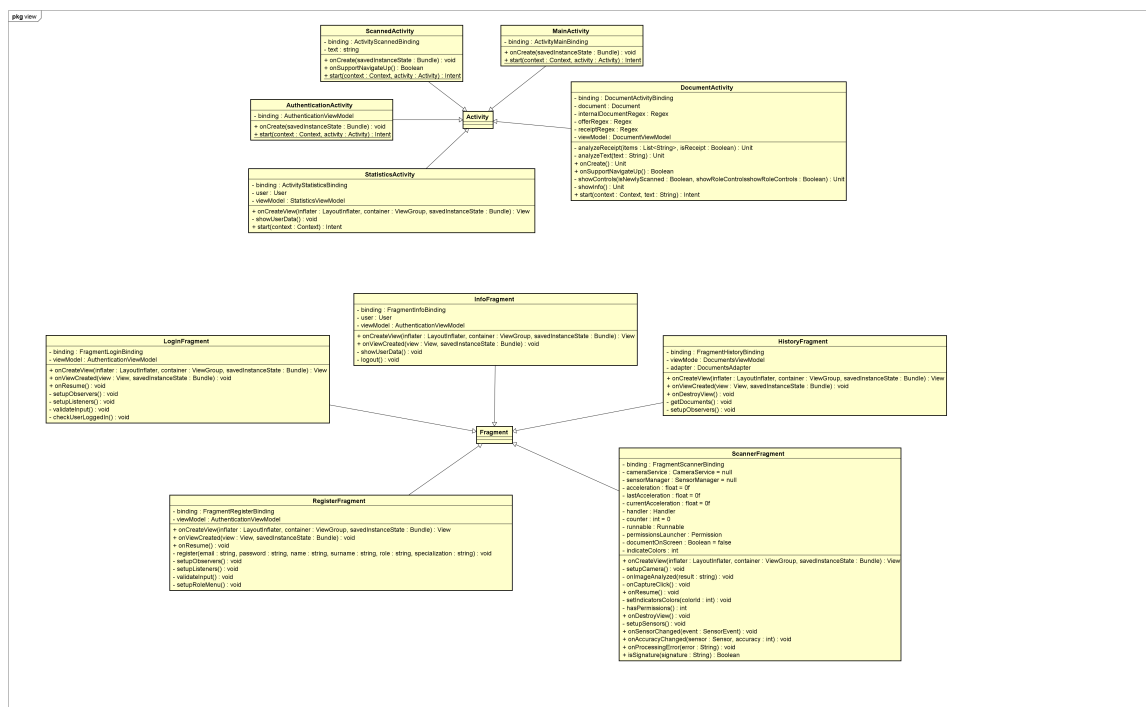


Slika 4.2: Dijagram baze podataka sustava

## 4.2 Dijagram razreda

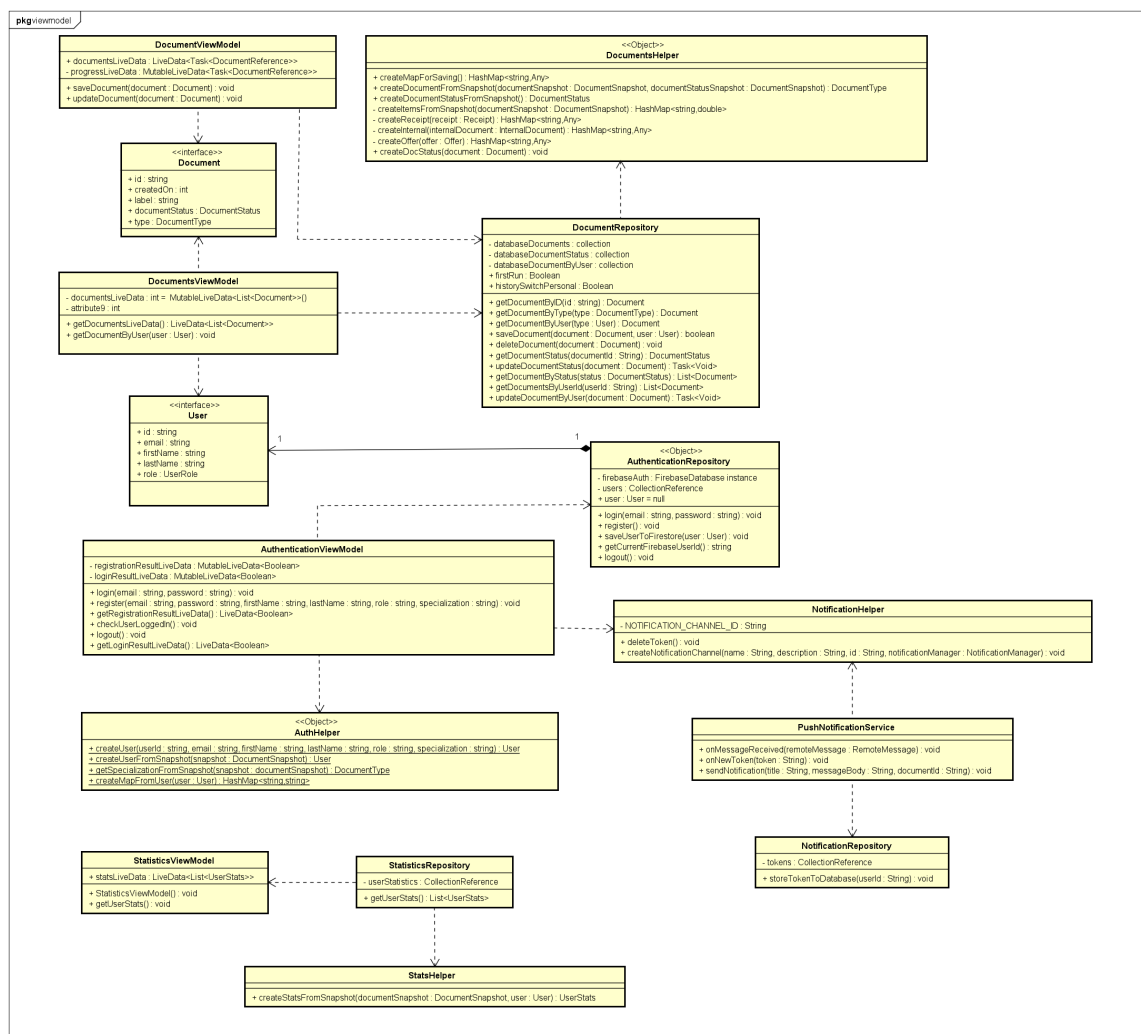
Na slikama 4.3., 4.4., 4.5., 4.6. prikazani su razredi koji pripadaju MVVM arhitekturi. Zbog lakše organizacije, razredi su podijeljeni prema arhitekturi, kako bi se smanjila prenapučenost unutar dijagrama. Iz naziva i tipova podataka može se zaključiti vrsta ovisnosti između razreda.

Razredi prikazani na slici 4.3. nasljeđuju Activity i Fragment razred. Metode implementirane u tim razredima manipuliraju objektima dobivenim iz razreda sa slike 4.4. (ViewModel) te omogućuju prikaz podataka.



Slika 4.3: Dijagram razreda - View.

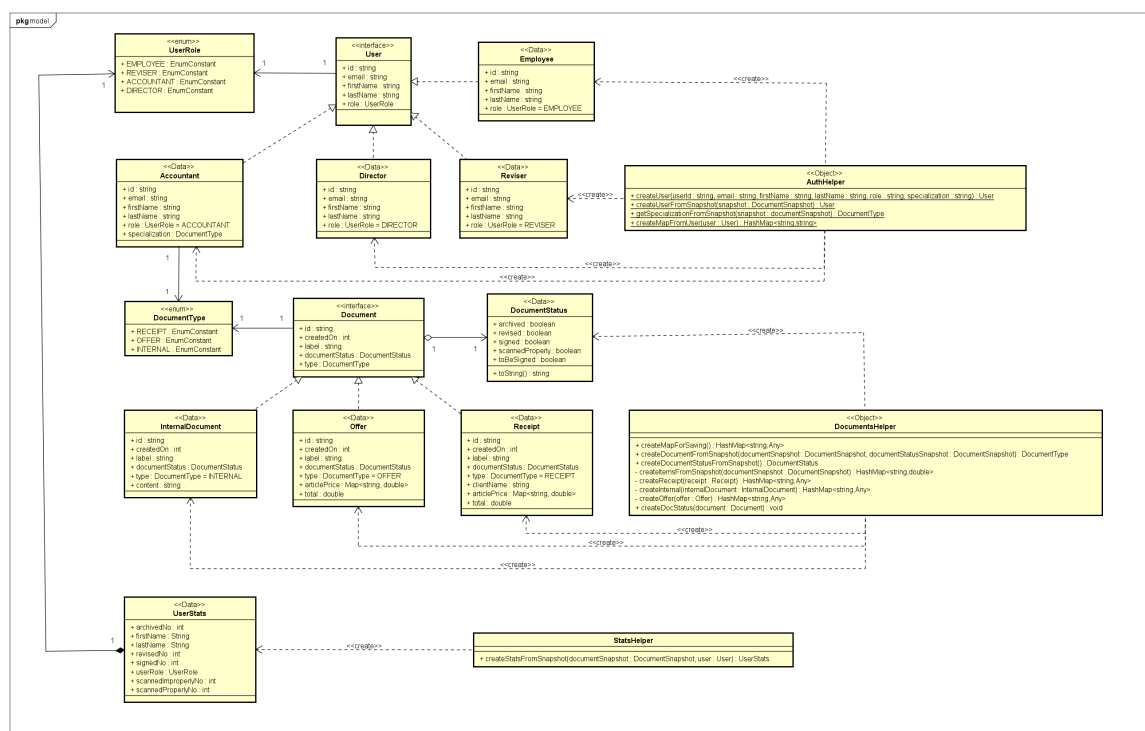
Razredi `DocumentsViewModel` i `AuthenticationViewModel` uzimaju željene podatke preko razreda `DocumentRepository` i `AuthenticationRepository`, koji su direktno povezani s bazom podataka. `AuthenticationRepository` razred koristi se razredom `AuthHelper` kako bi prilikom registracije stvorio novog korisnika, a `DocumentRepository` koristi se razredom `DocumentsHelper` kako bi omogućio stvaranje dokumenata. Način na koji su povezani razredi iz view i viewModel dijela arhitekture prikazani su na slici 4.5.



Slika 4.4: Dijagram razreda - ViewModel.



14. siječnja 2022.

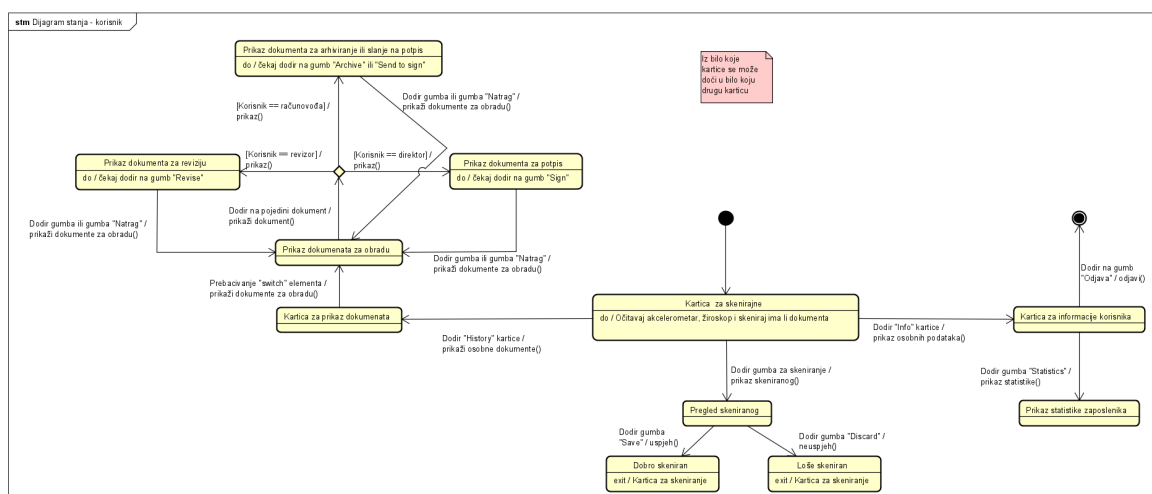


Slika 4.6: Dijagram razreda - Model



### 4.3 Dijagram stanja

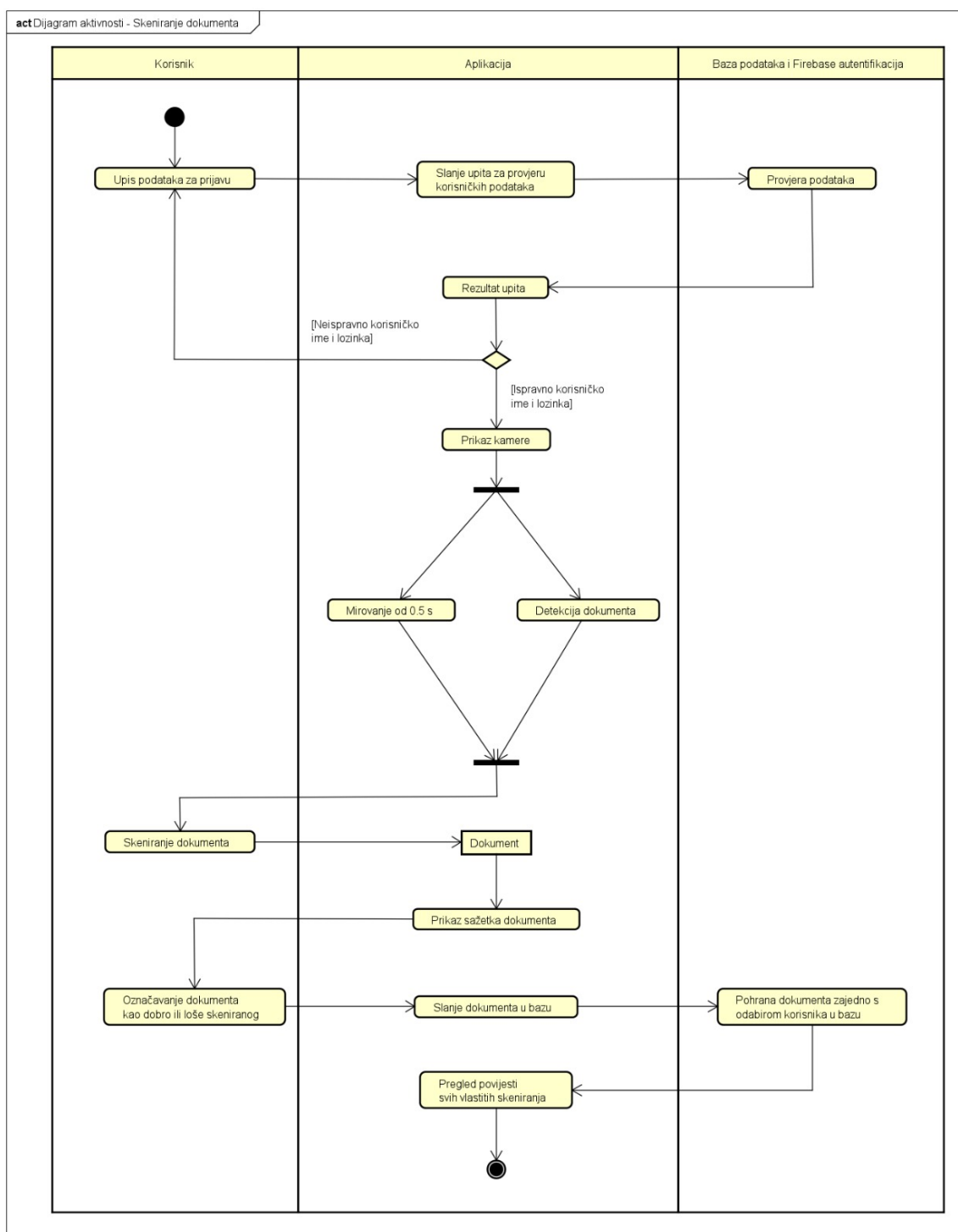
Dijagram stanja prikazuje stanja objekta te prijelaze iz jednog stanja u drugo temeljene na događajima. Na slici 4.7. prikazan je dijagram stanja za registriranog korisnika, koji može biti direktor, računovođa, revizor ili zaposlenik. Nakon prijave korisnik može skenirati dokument (nakon što akcelerometar i žiroskop to dopuste). Dodirom na gumb za skeniranje otvara se pregled skeniranog dokumenta. Ako korisnik nije zadovoljan skeniranim, ima opciju dodira na gumb *Discard*, a u suprotnom gumb *Save*. Bez obzira koji gumb je dodirnut, aplikacija vraća korisnika na karticu za skeniranje. Dodirom kartice *History* korisniku se prikazuju osobni skenirani dokumenti. Postoji *switch* koji kad se dodirne mijenja prikaz u dokumente koji su potrebni za obrađivanje (revizija, arhiviranje, potpisivanje – ovisno o tipu korisnika). U oba slučaja *switcha* korisnik može dodirom na pojedini dokument vidjeti detalje dokumenta i obaviti akciju. Odlaskom na karticu *Info* prijavljeni korisnik pregledava osobne podatke te ima mogućnost dodirnuti gumb za odjavu ili za prikaz statistike.



Slika 4.7: Dijagram stanja - Registrirani korisnik

## 4.4 Dijagram aktivnosti

Dijagram aktivnosti primjenjuje se za opis modela toka upravljanja ili toka podataka. Na dijagramu aktivnosti prikazan je proces skeniranja dokumenta. Pri modeliranju toka upravljanja svaki novi korak poduzima se nakon završenog prethodnog, osim koraka *Mirovanje od 5 s* i *Detekcija dokumenta*, koji se odvijaju istovremeno. Korisnik se s korisničkim podacima prijavljuje u sustav, skenira željeni dokument, nakon čega mu se prikazuje sažetak dokumenta. Korisnik zatim označava dokument kao dobro ili loše skeniran, dokument se šalje u bazu podataka, te se zajedno s odabirom korisnika pohranjuje u bazu.



Slika 4.8: Dijagram aktivnosti - skeniranje dokumenta

## 4.5 Dijagram komponenti

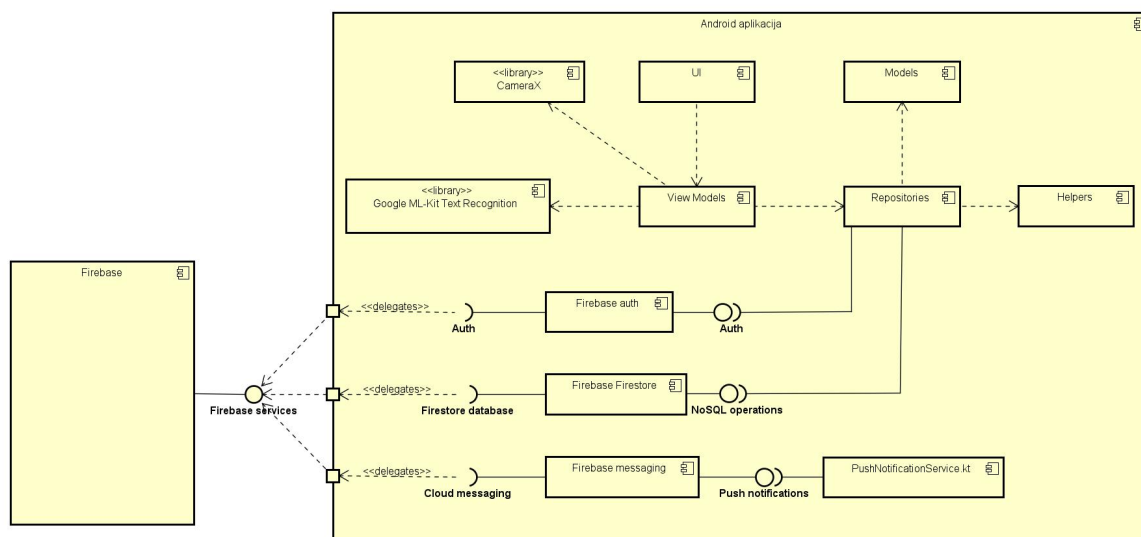
Dijagram komponenti na slici 4.9. prikazuje unutarnju strukturu aplikacije i komunikaciju s Firebase uslugom koja aplikaciji pruža sve potrebne usluge kao što su NoSQL baza podataka, push obavijesti i autentifikacija korisnika. Aplikacija je modelirana po uzoru na MVVM arhitekturu stoga svako korisničko sučelje ima pripadni model koji njime upravlja. Češće korištene funkcije enkapsulirane su u helper komponente.

Prijava i registracija korisnika ostvarena je preko sučelja Auth pomoću Firebase biblioteke za autentifikaciju.

Model korisničkog sučelja koristi repozitorij kako bi aplikacija dohvatila ili spremila određene podatke koji se modeliraju pomoću skupa modela za dokumente i korisnike preko sučelja za NoSQL operacije.

Skeniranje dokumenata ostvareno je bibliotekom CameraX, a procesiranje skeniranih dokumenata izvodi se interno preko pripadnog sučelja pomoću Google ML Text Recognition biblioteke.

Komponenta zadužena za push obavijesti konstantno sluša Firebase dolazne poruke preko sučelja za poruke te ih šalje drugim komponentama aplikacije na prikaz.



Slika 4.9: Dijagram komponenti

## 5. Implementacija i korisničko sučelje

### 5.1 Korištene tehnologije i alati

Komunikacija u timu ostvarena je korištenjem aplikacija Whatsapp<sup>1</sup> i Discord<sup>2</sup>. Za upravljanje izvornim kodom korišten je sustav Git<sup>3</sup>, a udaljeni repozitorij dostupan je na web platformi GitLab<sup>4</sup>. Za izradu UML dijagrama iskoristili smo alat Astah<sup>5</sup>.

Razvojno okruženje korišteno u Android dijelu aplikacije je Android Studio<sup>6</sup> - integrirano razvojno okruženje (IDE) kreirano na temeljima drugog JetBrainsovog IDE-a, IntelliJ. Također službeni je Googleov IDE za razvoj Android aplikacija, a rad u njemu je moguć u sva tri glavna operacijska sustava (Windows, Linux, MacOS).

Za backend dio programskog koda korišten je Microsoftov uređivač teksta Visual Studio Code<sup>7</sup>. Iako je samo uređivač teksta, VSC, zahvaljujući mnoštvu plugina razvijenih od programerske zajednice, podsjeća na razvojno okruženje. U njemu je integriran i sustav Git što nam je olakšalo rad.

Aplikacija je napisana za Android operacijski sustav s podrškom za verzije 10, 11, 12 koristeći programski jezik Kotlin<sup>8</sup>, a cloud funkcije za bazu podataka Firebase pisane su u Node.js<sup>9</sup>, Javascript<sup>10</sup> frameworku za izradu backenda. Node.js vrti se na V8 engineu te izvršava JavaScript kod izvan preglednika. Primarno služi za razvoj serverske strane web stranica za prikazivanje dinamičkog sadržaja, ali može poslužiti u bilo kojem dijelu serverskog programiranja.

Baza podataka se nalazi na Googleovom poslužitelju i razvijana je pomoću Firebase<sup>11</sup>

---

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

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

<sup>3</sup><https://git-scm.com>

<sup>4</sup><https://about.gitlab.com>

<sup>5</sup><https://astah.net>

<sup>6</sup><https://developer.android.com/studio>

<sup>7</sup><https://code.visualstudio.com>

<sup>8</sup><https://kotlinlang.org>

<sup>9</sup><https://nodejs.org>

<sup>10</sup><https://www.javascript.com>

<sup>11</sup><https://firebase.google.com>

Google platforme. Firebase se koristi za praćenje analitike, prijavljivanje i popravljivanje rušenja aplikacija, a ima ugrađene baze podataka (Realtime i Firestore).

Unit testiranje smo vršili pomoću Mockito<sup>12</sup> *frameworka* koji je razvijan pod MIT licencom. Programski jezik korišten za testiranje je Kotlin.

---

<sup>12</sup><https://site.mockito.org>

## 5.2 Ispitivanje programskog rješenja

### 5.2.1 Ispitivanje komponenti

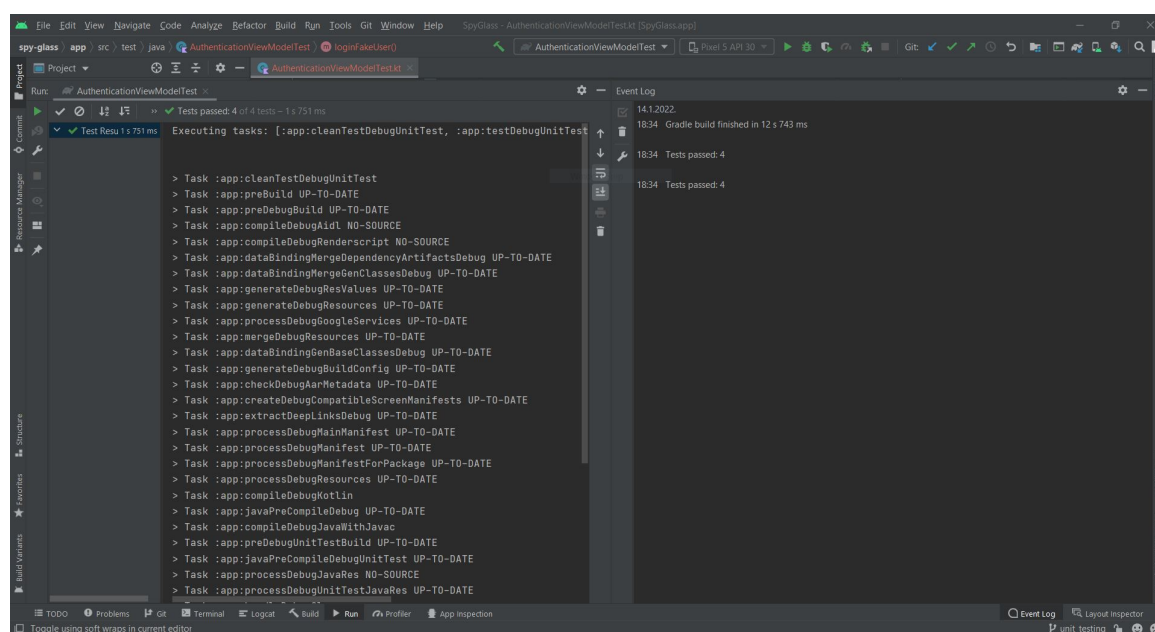
Napravljena su 4 testa za AuthenticationViewModel:

**loginFakeUser()** provjerava error handling za pokušaj prijave nepostojećeg korisnika

**loginEmptyEmail()** provjerava error handling za pokušaj prijave s praznim email-om

**registerEmptyEmail()** provjerava error handling za pokušaj registracije s praznim email-om

**logout()** provjerava error handling za pokušaj odjave bez prijavljenog korisnika



Slika 5.1: Rezultati testova - AuthenticationViewModelTests

```
30 //error handling for viewModel functions
31 @Test
32 fun loginFakeUser() {
33     val email = "unittest@test.com"
34     val password = "123_456"
35
36     result = authenticationViewModel.login(email, password)
37     assertEquals(result, kotlin.Unit)
38 }
39
40 @Test
41 fun loginEmptyEmail() {
42     val email = ""
43     val password = "123_456"
44
45     result = authenticationViewModel.login(email, password)
46     assertEquals(result, kotlin.Unit)
47 }
```

Slika 5.2: Isječak koda - AuthenticationViewModelTests

```
49 @Test
50 fun registerEmptyEmail() {
51     val email = ""
52     val password = "123_456"
53
54     result = authenticationViewModel.register(email, password, firstName: "ime", lastName: "prezime", role: "accountant", specialization: "offer")
55     assertEquals(result, kotlin.Unit)
56 }
57
58 @Test
59 fun logout() {
60     result = authenticationViewModel.logout()
61     assertEquals(result, kotlin.Unit)
62 }
63
64 }
```

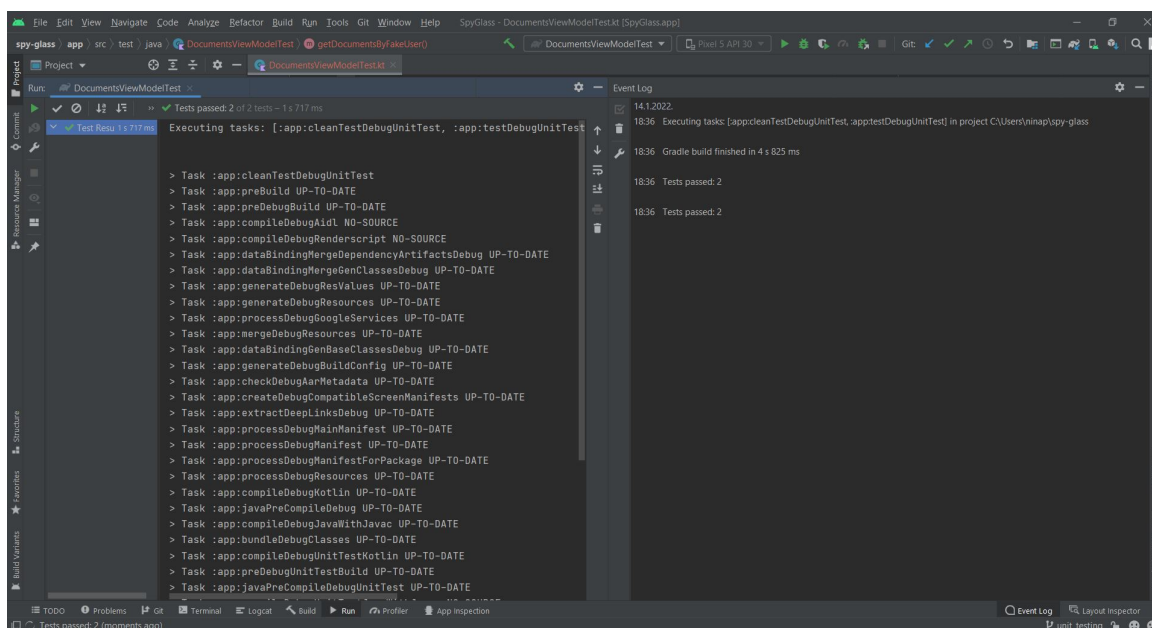
Slika 5.3: Isječak koda - AuthenticationViewModelTests

Napravljena su 2 testa za DocumentsViewModel:

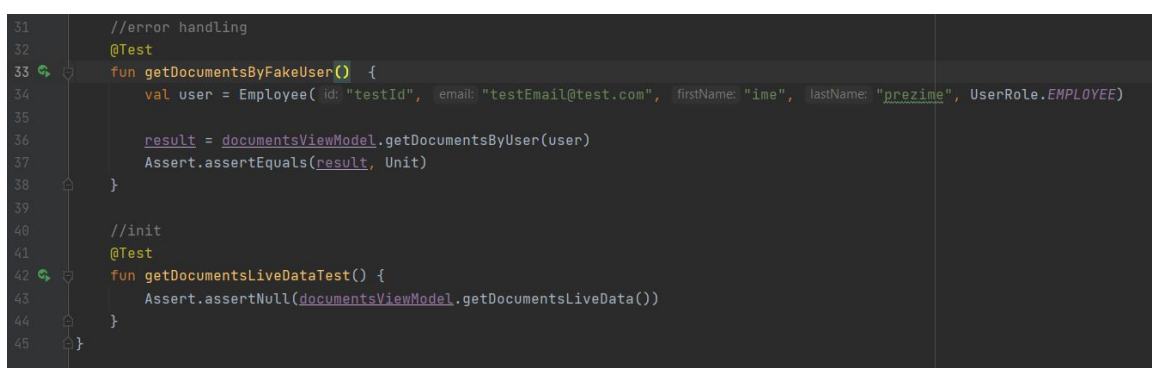
**getDocumentsByFakeUser()** provjerava error handling za pokušaj dohvata dokumenata nepostojećeg korisnika

**getDocumentsLiveDataTest()** provjerava inicijalizaciju varijable documentsLiveData: MutableLiveData<List<Document>>





Slika 5.4: Rezultati testova - DocumentsViewModelTests



Slika 5.5: Isječak koda - DocumentsViewModelTests

## 5.2.2 Ispitivanje sustava

Svaki dio sustava ispitan je ručno po obrascima uporabe kako bi se pronašla neočekivana ponašanja aplikacije. Prikazan je dio ispitivanja sustava zbog jednostavnosti (UC1 - Registracija).

### Ispitni slučaj 1: Ispitivanje registracije direktora

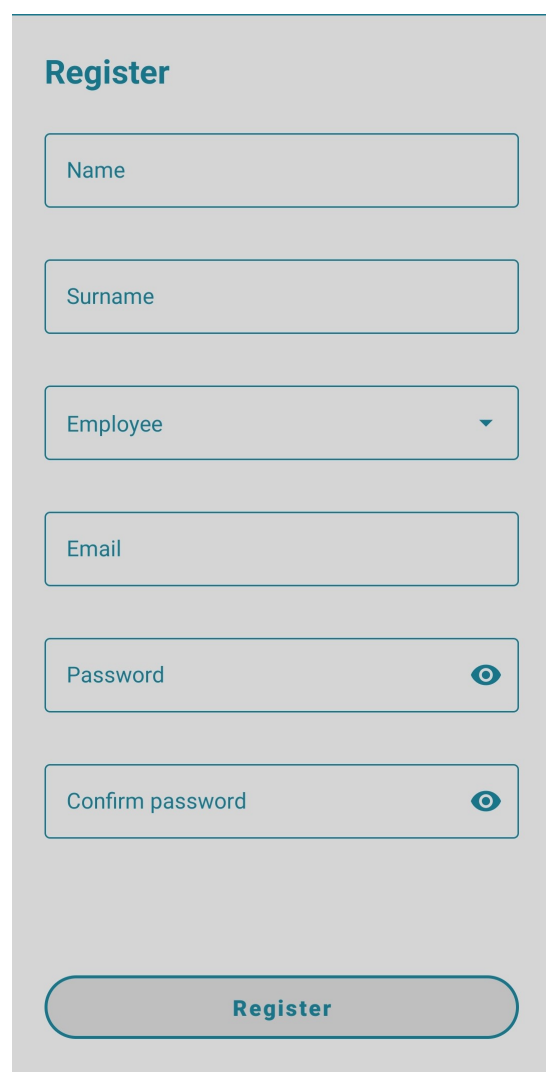
Ulaz:

1. Otvaranje aplikacije
2. Upisivanje identifikacijskih podataka
3. Pritiskanje gumba za registraciju

**Očekivani rezultat:**

1. Prikazuje se zaslon za registraciju
2. Provjera formata unosa
3. Pošalji podatke i prihvati registraciju

**Rezultat:** Očekivanja su zadovoljena. Aplikacija je prošla test.

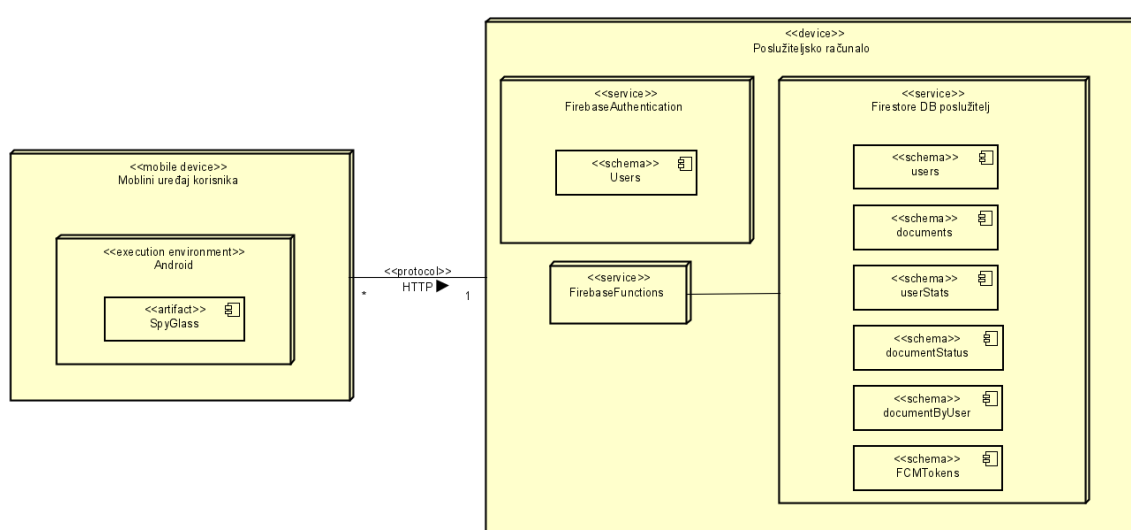


The image shows a registration form with the title "Register" in a bold, dark blue font. Below the title are six input fields, each with a light blue border and a light gray background. The first four fields are labeled "Name", "Surname", "Employee", and "Email". The "Employee" field has a small downward arrow on its right side. The fifth and sixth fields are labeled "Password" and "Confirm password", each with a small eye icon on its right side. At the bottom of the form is a rounded rectangular button with a dark blue border and a light gray background, containing the word "Register" in a bold, dark blue font.

Slika 5.6: Zaslon za registraciju

## 5.3 Dijagram razmještaja

Dijagram razmještaja je strukturni statički UML dijagram koji opisuje topologiju sustava i usredotočen je na odnos sklopovskih i programskih dijelova. Na slici 5.7 prikazan je specifikacijski dijagram razmještaja. Specifikacijski dijagram prikazuje pregled implementacije artefakata bez upućivanja na specifične slučajeve artefakata ili čvorova. Na poslužiteljskom računalu se nalazi poslužitelj baze podataka. Klijent koristi mobilnu aplikaciju namijenjenu za operacijski sustav Android. Komunikacija između klijenta i baze podataka odvija se preko protokola HTTP.



Slika 5.7: Dijagram razmještaja

## 5.4 Upute za puštanje u pogon

### 5.4.1 Kreiranje projekta i integracija SDK-a u Android

1. Kreirati Firebase projekt
  - (a) Imenovati projekt
  - (b) Omogućiti **Google Analytics**
  - (c) Kliknuti **Create project**
2. Registrirati aplikaciju
  - (a) Kliknuti opciju **Add app**
  - (b) Unijeti Android package name koji odgovara imenu od aplikacije
  - (c) Kliknuti **Register app**
3. Dodati Firebase configuration datoteku u aplikaciju
  - (a) Preuzeti **google-services.json** iz Firebase konzole
  - (b) Staviti datoteku u ispravni direktorij (module ili app-level direktorij)
  - (c) Dodati u **build.gradle** datoteku koja se nalazi u root-level ili project-level direktoriju navedeni tekst:

```
buildscript {  
  
    repositories {  
        // Check that you have the following  
        // line (if not, add it):  
        google() // Google's Maven  
        repository  
    }  
  
    dependencies {  
        // ...  
  
        // Add the following line:  
        classpath 'com.google.gms:google-services:4.3.10' // Google  
        Services plugin  
    }  
}
```

```
}

    allprojects {
        // ...

        repositories {
            // Check that you have the
            // following line (if not,
            // add it):
            google() // Google's Maven
            repository
            // ...
        }
    }
}
```

- (d) Dodati u **build.gradle** datoteku koja se nalazi u module ili app-level direktoriju navedeni tekst:

```
apply plugin:
    'com.android.application'
// Add the following line:
apply plugin: 'com.google.gms.google-services' //
    Google Services plugin

android {
    // ...
}
```

#### 4. Dodati Firebase SDK u aplikaciju

- (a) Dodati u **build.gradle** datoteku koja se nalazi u module ili app-level direktoriju navedeni tekst:

```
dependencies {
    // ...

    // Import the Firebase BoM
    implementation platform('com.google.firebase:
        firebase-bom:29.0.3')
```

```
// When using the BoM, you don't specify versions in
    Firebase library dependencies

// Declare the dependency for the Firebase SDK for
    Google Analytics
implementation 'com.google.firebase:firebase-
    analytics-ktx'

// Declare the dependencies for any other desired
    Firebase products
// For example, declare the dependencies for
    Firebase Authentication and Cloud Firestore
implementation 'com.google.firebase:firebase-auth-
    ktx'
implementation 'com.google.firebase:firebase-
    firestore-ktx'
```

- (b) Kliknuti gumb **Sync** u programu Android Studio kako bi svi **dependencies** bili na najnovijoj verziji

5. U Firebase konzoli omogućiti autentifikaciju preko email adrese i lozinke

- (a) Otvoriti **Auth** sekciju unutar konzole
- (b) Pod **Sign in method** odabrati opciju **Email/password** i kliknuti gumb **Save**

6. Dodati **dependencies** za Firebase autentifikaciju

- (a) Dodati u **build.gradle** datoteku koja se nalazi u module ili app-level direktoriju navedeni tekst:

```
dependencies {
    // Import the BoM for the Firebase platform
    implementation platform('com.google.firebase
        :firebase-bom:29.0.3')

    // Declare the dependency for the Firebase
        Authentication library
    // When using the BoM, you don't specify
        versions in Firebase library dependencies
```

```
        implementation 'com.google.firebase:firebase  
            -auth-ktx '  
    }
```

### 5.4.2 Puštanje u pogon Cloud Funkcija

**Cloud Functions** služe za automatsko pokretanje pozadinskog koda kao odgovor na događaje koje pokreću značajke Firebasea, primjerice triggeri kolekcija u bazi podataka. Nakon kreiranja Firebase projekta, opisanog u prethodnom dijelu teksta potrebni su sljedeći koraci:

1. Instalacija node.js i Firebase CLI-a. Potrebno je instalirati i npm, te pokrenuti sljedeću naredbu:

```
npm install -g firebase-tools
```

2. Povezati projekt lokalno s cloudom:

```
firebase login  
firebase init firestore  
firebase init functions
```

3. Napisati Cloud funkciju te pokrenuti

```
firebase deploy --only functions
```

### 5.4.3 Pokretanje Android aplikacije

Potrebno je preuzeti spyglass.apk datoteku s Gitlab repozitorija projekta pod direktorijem *apk* na mobilni uređaj. Zatim ju je potrebno pokrenuti čime će se pokrenuti i instalacija Android aplikacije nakon čega klikom na ikonu aplikacije pokrećemo aplikaciju.

## 6. Zaključak i budući rad

Zadatak zadan ovim projektom dovoljno je složen kako bi se članovi tima s nikakvim do malim iskustvom upoznali s mnogim problemima organizacije posla i vremena, programiranja, dizajna sustava i komunikacije unutar tima te tako dobili dojam o pravim problemima koje je potrebno riješiti kod razvoja programske podrške za stvarne klijente. Unatoč ambicioznom početku, većina članova tima su inicijalno slabo bili upoznati s razvojem programske podrške za sustav Android što je dodalo dodatnu razinu kompleksnosti na zadatak. Zajedničkim radom i suradnjom, projekt je u predefiniranom roku uspješno završen i tim je zadovoljan rezultatom.

Nakon sastavljanja tima, članovi tima krenuli su se upoznavati sa Android sustavom i alatima za razvoj programske podrške kao što je Git, Android studio i programski jezik Kotlin i sl. Nakon shvaćenog opsega projekta, članovi tima podijelili su se na zadatke koji najbolje odgovaraju sposobnostima pojedinog člana. Voditelj tima pratio je napredak svakog člana te uskakao u pomoć ukoliko je to bilo potrebno. Iskusniji članovi tima izradili su kostur aplikacije dok su se ostali članovi posvetili proučavanju zadatka u detalje i izradi tehničke dokumentacije kako bi za početak implementacije funkcionalnosti tim imao dobru podlogu.

Prva funkcionalnost koja je razvijena je sustav za registraciju i prijavu korisnika. Budući da je korištena dobro poznata i dokumentirana usluga Firebase, ova faza implementacije prošla je bez većih poteškoća.

Sljedeća funkcionalnost koja je implementirana je pospremanje tokena za push obavijesti u bazu podataka koja nije u cijelosti dokumentirana od strane Firebase usluge. Posljedica je funkcionalni sustav koji nije ostvaren najefikasnije moguće i podložan problemima kod proširenja aplikacije.

Naš najveći izazov bio je implementirati dinamičku detekciju dokumenta. Uspjeli smo to izvesti korištenjem Google-ovog ML Kita s kojim unutar implementacije CameraX API-ja provjeravamo je li na ekranu zapravo dokument. Osim toga nismo imali značajnijih izazova te su sve tražene funkcionalnosti implementirane.

Projekt bi bio završen ranije da smo imali više iskustva s radom u timu te iskustva s radom u Androidu. No s vremenom je tim postao sve bolji te smo brže i



kvalitetnije implementirali tražene značajke.

Rad na projektu pokazao se dovoljno izazovnim da svaki od članova tima steče nova znanja o tehnologijama, timskom radu i svojim sposobnostima, a u isto vrijeme dovoljno jednostavan da se tim rastane sa zadovoljstvom i iskustvom koje će im u budućoj karijeri olakšati prilagodbu ozbiljnom zadatku.

# Popis literature

1. M. Moskala, I. Wojda, "Android Development with Kotlin", Packt Publishing Ltd., 2017
2. UML Use Case Diagram Tutorial, <https://www.lucidchart.com/pages/uml-use-case-diagram>
3. Learn Git Branching, <https://learngitbranching.js.org/>
4. Guide to app architecture, <https://developer.android.com/jetpack/guide>
5. Model-View-ViewModel (MVVM) Explained, <https://www.wintellelect.com/model-view-viewmodel-mvvm-explained/>
6. Handling Lifecycles with Lifecycle-Aware Components, <https://developer.android.com/topic/libraries/architecture/lifecycle>
7. Saving UI States, <https://developer.android.com/topic/libraries/architecture/saving-states>
8. Handle configuration changes, <https://developer.android.com/guide/topics/resources/runtime-changes>
9. Firebase Documentation, <https://firebase.google.com/docs>
10. CameraX overview, <https://developer.android.com/training/camerax>
11. Recognize text in images with ML Kit on Android, <https://developers.google.com/ml-kit/vision/text-recognition/android>
12. Kotlin coroutines on Android, <https://developer.android.com/kotlin/coroutines>

# Indeks slika i dijagrama

2.1	Primjer postojeće aplikacije . . . . .	8
3.1	Dijagram obrasca uporabe, funkcionalnosti korisnika . . . . .	19
3.2	Dijagram obrasca uporabe, funkcionalnosti registriranog korisnika .	20
3.3	Dijagram obrasca uporabe, funkcionalnosti specijalizacija . . . . .	21
3.4	Dijagram obrasca uporabe, pregled dokumenata . . . . .	22
3.5	Sekvencijalni dijagram, UC1 . . . . .	24
3.6	Sekvencijalni dijagram, UC4 . . . . .	25
3.7	Sekvencijalni dijagram, UC8 . . . . .	26
3.8	Sekvencijalni dijagram, UC12 . . . . .	27
4.1	Arhitektura sustava . . . . .	29
4.2	Dijagram baze podataka sustava . . . . .	35
4.3	Dijagram razreda - View. . . . .	36
4.4	Dijagram razreda - ViewModel. . . . .	37
4.5	Dijagram razreda - View s ViewModelom. . . . .	38
4.6	Dijagram razreda - Model . . . . .	39
4.7	Dijagram stanja - Registrirani korisnik . . . . .	40
4.8	Dijagram aktivnosti - skeniranje dokumenta . . . . .	42
4.9	Dijagram komponenti . . . . .	43
5.1	Rezultati testova - AuthenticationViewModelTests . . . . .	46
5.2	Isječak koda - AuthenticationViewModelTests . . . . .	47
5.3	Isječak koda - AuthenticationViewModelTests . . . . .	47
5.4	Rezultati testova - DocumentsViewModelTests . . . . .	48
5.5	Isječak koda - DocumentsViewModelTests . . . . .	48
5.6	Zaslona za registraciju . . . . .	49
5.7	Dijagram razmjesta . . . . .	50

# Dodatak: Prikaz aktivnosti grupe

## Dnevnik sastajanja

### 1. sastanak

- Datum: 13. listopada 2021.
- Prisustvovali: Ivan Futivić, Karlo Marković, Rafael Boban, Blaž Solić, Nina Petrušić, Luka Hanžek
- Teme sastanka:
  - prvi sastanak s asistentom i timom
  - diskusija o projektu i zadanom zadatku s visokog pogleda
  - diskusija o tehnologijama prikladnim za projektni zadatak
  - međusobno upoznavanje

### 2. sastanak

- Datum: 19. listopada 2021.
- Prisustvovali: Ivan Futivić, Karlo Marković, Rafael Boban, Karlo Kada, Nina Petrušić, Luka Hanžek
- Teme sastanka:
  - specifikacija programske potpore
  - detaljna razrada zadatka
  - izgled baze podataka
  - diskutiranje o potencijalnim rješenjima
  - upoznavanje s NoSQL bazom podataka
  - diskutiranje o sustavu za slanje obavijesti

### 3. sastanak

- Datum: 2. studenog 2021.
- Prisustvovali: Ivan Futivić, Rafael Boban, Blaž Solić, Karlo Kada, Luka Hanžek
- Teme sastanka:
  - specifikacija programske potpore
  - struktura baze podataka
  - rješavanje nedoumica u dizajnu i zahtjevima

- podjela poslova
- arhitektura sustava

#### 4. sastanak

- Datum: 9. studenog 2021.
- Prisustvovali: Ivan Futivić, Karlo Marković, Blaž Solić, Karlo Kada, Nina Petrušić, Luka Hanžek
- Teme sastanka:
  - diskutiranje o dovršavanju dokumentiranja prve revizije projekta
  - rješavanje nedoumica u dizajnu i zahtjevima
  - prilagođavanje prvobitnog plana podjele posla

#### 5. sastanak

- Datum: 6. prosinca 2021.
- Prisustvovali: Ivan Futivić, Karlo Marković, Blaž Solić, Karlo Kada, Nina Petrušić, Luka Hanžek
- Teme sastanka:
  - rasprava o načinu ponašanja obavijesti
  - priprema za prvo kolokviranje
  - podjela ostatka posla vezano za aplikaciju

#### 6. sastanak

- Datum: 20. prosinca 2021.
- Prisustvovali: Ivan Futivić, Karlo Marković, Blaž Solić, Karlo Kada, Rafael Boban
- Teme sastanka:
  - podjela posla za završne dijelove dokumentacije
  - priprema za odmor tijekom blagdana

#### 7. sastanak

- Datum: 10. siječnja 2022.
- Prisustvovali: Ivan Futivić, Karlo Marković, Blaž Solić, Karlo Kada, Rafael Boban, Nina Petrušić, Luka Hanžek
- Teme sastanka:
  - dovršavanje dokumentacije
  - popravljivanje bugova u dokumentaciji
  - finaliziranje koda
  - razgovor o izradi prezentacije
  - testiranje koda

## Tablica aktivnosti

	Ivan Futivić	Rafael Boban	Karlo Marković	Blaž Solić	Karlo Kada	Nina Petrušić	Luka Hanžek
Upravljanje projektom	4	0	0	0	0	0	0
Opis projektnog zadatka	0	0	0	0	0	0	3
Funkcionalni zahtjevi	0	0	0	1	0	0	0,2
Opis pojedinih obrazaca	0	0	0	1	0	0	4
Dijagram obrazaca	0	0	0	1	0	0	3
Sekvencijski dijagrami	0	0	0	0	0	0	3
Opis ostalih zahtjeva	0,25	0	0	0	0	0	0,5
Arhitektura i dizajn sustava	5	5	0	1	0	0	0
Baza podataka	6	0	0	4	0	0	0
Dijagram razreda	0,2	0	8	0	0	0	0
Dijagram stanja	0	0	0	1	0	0	0
Dijagram aktivnosti	0	0	0	0	2	0	0
Dijagram komponenti	0	0	0	0	0	0	3
Korištene tehnologije i alati	0,5	0	0	0,5	0	0	0
Ispitivanje programskog rješenja	0	0	0	0	0	10	0
Dijagram razmještaja	0	0	2	0	0	0	0
Upute za puštanje u pogon	1	0	0	1	0	0	0
Dnevnik sastajanja	2	0	0	0	0	0	0,5
Zaključak i budući rad	0	1	0	0	0	0	0
Popis literature	0	0	0	0	0	0	1

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

	Ivan Futivić	Rafael Boban	Karlo Marković	Blaž Solić	Karlo Kada	Nina Petrušić	Luka Hanžek
<i>izrada predloška koda</i>	0	2	0	0	0	0	0
<i>izrada login/register sustava</i>	10	0	0	0	0	0	0
<i>spajanje s bazom podataka</i>	1	0	0	5	0	0	0
<i>izrada sustava za obavijesti</i>	0,2	0	0	0	0	0	6
<i>backend</i>	2	0	0	2	0	2	0
<i>izgled aplikacije</i>	2	8	0	0	0	0	0
<i>izrada pretrage povijesti dokumenata</i>	5	14	0	0	5	0	0
<i>izrada pregleda osobnih podataka</i>	0	0	8	0	0	0	0
<i>izrada OCR skenera</i>	0	15	0	0	0	0	0
<i>izrada prikaza statistike</i>	1	2	4	0	0	0	0
<i>izrada modela</i>	1	0	0	1	0	4	0