

Programsko inženjerstvo

Ak. god. 2021./2022.

Inventura

Dokumentacija, Rev. 1

Grupa: *Jarci&Magarci*

Voditelj: *Elena Wachtler*

Datum predaje: 19. 11. 2021.

Nastavnik: *Igor Stančin, mag. ing.*

Sadržaj

1	Dnevnik promjena dokumentacije	3
2	Opis projektnog zadatka	6
3	Specifikacija programske potpore	13
3.1	Funkcionalni zahtjevi	13
3.1.1	Obrasci uporabe	15
3.1.2	Sekvencijski dijagrami	23
3.2	Ostali zahtjevi	26
4	Arhitektura i dizajn sustava	28
4.1	Baza podataka	28
4.1.1	Opis tablica	29
4.1.2	Dijagram baze podataka	35
4.2	Dijagram razreda	36
4.3	Dijagram stanja	38
4.4	Dijagram aktivnosti	39
4.5	Dijagram komponenti	41
5	Implementacija i korisničko sučelje	42
5.1	Korištene tehnologije i alati	42
5.2	Ispitivanje programskog rješenja	43
5.2.1	Ispitivanje komponenti	43
5.2.2	Ispitivanje sustava	44
5.3	Dijagram razmještaja	46
5.4	Upute za puštanje u pogon	48
6	Zaključak i budući rad	49
	Popis literature	51
	Indeks slika i dijagrama	52

Dodatak: Prikaz aktivnosti grupe

53

1. Dnevnik promjena dokumentacije

Kontinuirano osvježavanje

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Napravljen predložak.	Filip Begović	26.10.2021.
0.2	Dodan opis projektnog zadatka i funkcionalni zahtjevi.	Nina Anić, Filip Begović, Marko Bagarić	26.10.2021.
0.3	Napravljeni obrasci uporabe.	Nina Anić, Filip Begović	2.11.2021.
0.4	Dijagrami obrazaca uporebe, sekvencijski dijagram i ostali zahtjevi	Filip Begović, Tin Jukić	10.11.2021. 13.1.2022.
0.4	Unos sličnih aplikacija	Marko Bagarić	27.10.2021.
0.5	Dodan <i>Use Case</i> dijagram i jedan sekvencijski dijagram, funkcionalni i nefunkcionalni zahtjevi i dodatak A	Filip Begović	15.11.2021.
0.6	Inicijalni dizajn ekrana aplikacije	Filip Begović	28.10.2021.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodatka	Autori	Datum
0.9	Opisi obrazaca uporabe	Filip Begović	15.11.2021.
0.10	ER dijagram i opis tablice baze podataka	Elena Wachtler	18.11.2021.
0.11	Sekvencijski dijagrami	Filip Begović	15.11.2021.
0.12.1	Dijagrami razreda	Tin Jukić	15.11.2021. 12.1.2022.
0.12.2	Dovršen dio dokumentacije Arhitektura i dizajn sustava	Tin Jukić	12.1.2022.
1.0	Verzija samo s bitnim dijelovima za 1. ciklus	Elena Wachtler, Tin Jukić, Filip Begović	18.11.2021.
1.1	Ostali dijagrami iz Arhitekture	Tin Jukić	12.1.2022.
1.2	Korištene tehnologije i alati	Marko Bagarić	13.1.2022.
1.5	Ispitivanje programskog rješenja	Marko Bagarić	13.1.2022.
1.5.1	Dijagram razmještaja	Marko Bagarić, Borna Rebić- Taučer	13.1.2022.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodatka	Autori	Datum
1.5.2	Upute za puštanje aplikacije u pogon	Borna Rebić- Taučer	13.1.2022.
2.0	Konačni tekst predloška dokumentacije	*	13.1.2022.

2. Opis projektnog zadatka

Cilj ovog projekta je izrada mobilne aplikacije „Inventura“ koja će korisniku olakšati posao inventure skladišta. Tvrtke čiji zaposlenici će koristiti ovu aplikaciju imaju nekoliko skladišta na različitim lokacijama te će pomoću ove aplikacije i mobilne kamare moći očitati bar kodove i QR kodove artikala čiju inventuru žele napraviti. Također, ovisno o tome koju ulogu u tvrtci korisnik ima, aplikacija mu nudi određene funkcionalnosti kao što su: prikaz informacija o artiklu, njegova lokacija, broj artikala, mogućnost povećanja i smanjenja broja artikala, spremanje promjena u bazu, slanje obavijesti nadležnima, kontrola inventure, preraspodjela artikala po grupama i mnoge druge.

Prilikom pokretanja aplikacije korisniku se prikazuje stranica za prijavu registriranih korisnika na kojoj se nalaze polja za unos email adrese i lozinke. Neregistrirani korisnik pri dnu stranice za prijavu treba odabrati opciju „Registriraj se“ koja ga vodi na stranicu za registraciju. Za kreiranje novog računa potrebno je navesti:

- *Ime*
- *Prezime*
- *Email*
- *Lozinku*
- *Ulogu u tvrtci*
- *Ovisno o ulozi u tvrtci, skladište u kojem zaposlenik radi*

Kada se korisnik prijavi u sustav uključuje mu se stražnja kamera mobilnog uređaja te se prikazuje početni zaslon koji je spreman za očitavanje bar koda i QR koda. Osim što može očitati kodove, korisnik ovisno o svojoj ulozi ima još dodatnih mogućnosti.

Korisnik može imati jednu od ove tri uloge:

- *Skladištar*
- *Šef skladišta*
- *Direktor*

Skladištar je osoba koja obavlja inventuru skladišta te po potrebi obavještava svoje nadležne ako određenog artikla nema na skladištu. Biranje artikla za skeniranje se vrši automatski, na način da prvi artikl koji skenira u toj inventuri je onaj odabrani. Zatim se uključi stražnja kamera mobilnog uređaja te se na ekranu aplikacije prikazuje početni zaslon koji je spreman za skeniranje. Kada se na proizvodu skenira te mu se detektira bar kod ili QR kod, skladištaru na zaslonu aplikacije iskače prozor (engl. *pop-up window*) koji se prikazuje 5 sekundi. Taj prozor sadrži informacije o skeniranom proizvodu u obliku kratkog opisa, lokaciju na kojoj je skeniran (dohvaćenu automatski pomoću GPS modula mobitela) i brojač takvih proizvoda u skladištu, a također i nudi opcije povećanja i smanjenja broja proizvoda, odbacivanje trenutnog očitavanja bez spremanja u bazu, te pohranu trenutnog očitavanja u bazu bez čekanja da istekne 5 sekundi. Promjenom broja očitanih artikala resetira se brojač 5 sekundi, a istekom vremena brojača zapis se automatski sprema u bazu. Na početnom zaslonu postoji i opcija *Završi*. Odabirom te opcije skladištar završava svoj rad nad tim proizvodom. Svaki skladištar može skenirati samo jednu vrstu proizvoda dok ne označi završetak skeniranja tog proizvoda. Osim opcije za skeniranje proizvoda, skladištaru se na početnoj stranici nudi još i mogućnost odjave iz aplikacije te povijest svih unosa proizvoda. Odabirom prikaza povijesti, sustav korisniku prikazuje pregled povijesti svih unosa korisnika. Prvo prema odrađenim inventurama, uz koje se prikazuje datum inventure. Pritiskom na izabranu inventuru prikazuju se svi proizvodi koje je prijavljeni skladištar unio u aplikaciju, dok pritiskom na još aktivnu inventuru, uza sve unesene proizvode skladištar može dojaviti šefu skladišta da nekog artikla nema na skladištu. Prilikom pregleda povijesti unosa, za svaki artikl navodi se njegov naziv, lokacija na kojoj je skeniran te broj koliko je takvih artikala prijavljeni skladištar skenirao.

Šef skladišta, za razliku od ostalih korisnika, ima mogućnost korištenja aplikacije na 1 od 2 načina: kao izvršitelj inventure ili kao kontrolor inventure. Prilikom ulaska u aplikaciju otvara mu se kamera za detekciju bar kodova i QR kodova. Ukoliko je skenirao proizvod koji u trenutnoj inventuri nije bio skeniran tada ima ulogu izvršitelja inventure. Za razliku od toga, ako je proizvod već skeniran u trenutnoj inventuri, tada ima ulogu kontrolora inventure.

Kao izvršitelj inventure, dobiva jednaku funkcionalnost kao i skladištar i uključuje mu se kamera mobilnog uređaja. Zatim se na ekranu aplikacije prikazuje početni zaslon koji je spreman za skeniranje. Kada se proizvodu skenira i detektira bar kod ili QR kod, šefu skladišta na zaslonu aplikacije iskače prozor (engl. *pop-up*

window) koji se prikazuje 5 sekundi. Taj prozor sadrži informacije o skeniranom proizvodu, lokaciju na kojoj je skeniran (automatsko dohvaćanje pomoću GPS modula mobitela) i brojač takvih proizvoda u skladištu te nudi opcije povećanja i smanjenja broja proizvoda, odbacivanje trenutnog očitavanja bez spremanja u bazu te pohranu trenutnog očitavanja u bazu bez čekanja da istekne 5 sekundi. Promjenom broja očitanih artikala resetira se brojač 5 sekundi, a istekom vremena brojača zapis se automatski sprema u bazu. Na početnom zaslonu postoji i opcija *Završi*. Odabirom te opcije označava se završetak skeniranja odabranog artikla. Osim opcije za skeniranje proizvoda, šefu skladišta se nude još četiri mogućnosti na početnom zaslonu: pregled svih artikala na skladištu, pregled svih izvršenih inventura nad tim skladištem, pregled obavijesti te izlazak (odjava) iz aplikacije. Odabirom prikaza povijest prikazuje se pregled povijesti svih unosa korisnika. Prvo prema odrađenim inventurama, uz koje se prikazuje datum inventure. Odabirom određene inventure, šefu skladišta se nudi opcija pregleda svih artikala na skladištu tako da se za svaki artikl ispisuje njegov naziv, lokacija te broj takvih artikala na skladištu. Odabirom prikaza pošte na početnom zaslonu, šefu skladišta se otvara stranica s obavijestima. Navedene obavijesti poslali su skladištari koji su uočili da određenih artikala nema na skladištu. Odabirom opcije prikaza svih proizvoda, šef skladišta može vidjeti sve artikle koji se trenutno nalaze na njegovom skladištu.

Kao kontrolor inventure šef skladišta ima iste funkcionalnosti kao i skladištari. Jedina razlika je što odabirom opcije *Završi*, ukoliko je broj skeniranih artikala od strane šefa skladišta različit od broja koje je skenirao skladištar, aplikacija automatski o tome obavještava direktora.

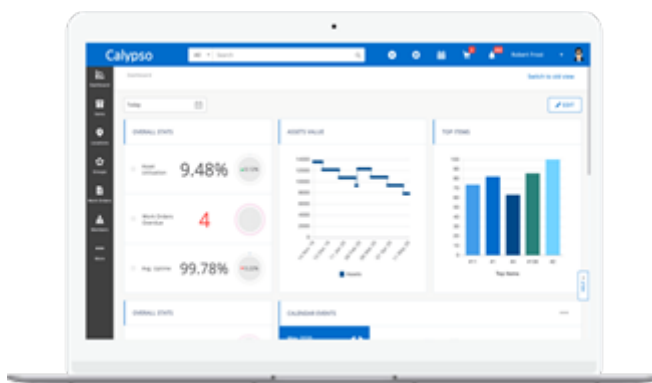
Direktor tvrtke je korisnik aplikacije kojem su omogućene najveće funkcionalnosti i koji ima pristup gotovo svim do sad navedenim mogućnostima. Naime, nakon prijave u sustav otvara mu se početni zaslon i uključuje stražnja kamera mobilnog uređaja kako bi mogao skenirati artikl. Kada se proizvodu skenira i detektira bar kod ili QR kod, direktoru na zaslonu aplikacije iskače prozor (engl. *pop-up window*) koji se prikazuje 5 sekundi. Taj prozor sadrži informacije o skeniranom proizvodu, lokaciju na kojoj je skeniran (automatsko dohvaćanje pomoću GPS modula mobitela) i brojač takvih proizvoda u skladištu te nudi opcije povećanja i smanjenja broja proizvoda, odbacivanje trenutnog očitavanja bez spremanja u bazu te pohranu trenutnog očitavanja u bazu bez čekanja da istekne 5 sekundi. Promjenom broja očitanih artikala resetira se brojač 5 sekundi, a istekom vremena brojača

zapis se automatski sprema u bazu. Na zaslonu postoji i opcija *Završi sken* čijim se odabirom završava rad nad tim artiklom, kao i opcija *Završi inventuru* čijim se odabirom završava trenutna inventura te automatski počinje nova. Osim opcije za skeniranje proizvoda, direktoru tvrtke nude se još tri mogućnosti na početnom zaslonu: pregled svih skladišta u kojima se obavljala inventura, pregled obavijesti te izlazak (odjava) iz aplikacije. Odabirom svih inventura direktoru se nudi opcija pregleda svih skladišta u kojima se obavljala inventura. Nakon što direktor odabere određeno skladište, otvara mu se nova stranica na kojoj može odabrati želi li pretraživati skladištare koji su obavljali inventure ili artikle koji su bili skenirani. Ako odabere opciju *Artikli* prikazuju mu se svi skenirani artikli u tom skladištu, te za svaki artikal njegovo ime i broj takvih artikala u odabranom skladištu. Osim što ima uvid u postojeće artikle, odabirom dodavanja novih kategorija direktoru se nudi opcija dodavanja grupa proizvoda i preraspodjela artikala po grupama. Za te funkcionalnosti otvara mu se nova stranica u kojoj može dodati novu grupu proizvoda, otići na već postojeću grupu ili preraspodijeliti artikle po grupama. Grupe proizvoda su organizirane u stablo s korijenom „proizvod“, a dubina i širina stabla nisu ograničeni. Osim opcije *Artikli* postoji i opcija *Skladištari* koja direktoru tvrtke prikazuje ime skladištara koji je radio inventuru u odabranom skladištu te njegovu statistiku pogreške. Odabirom popisa skladišta, prikazuje se ponovno popis svih skladišta, ali pritiskom na jedno od skladišta otvara se nova stranica s popisom svih inventura na tom skladištu. Odabirom određene inventure, direktor ima uvid u sve proizvode koji su se skenirali u toj inventuri. Odabirom opcije pregleda obavijesti, direktoru tvrtke otvara se stranica s obavijestima koje su poslali šefovi skladišta. Navedene obavijesti direktor može označiti odabirom odgovarajuće opcije kao riješene ili kao odbačene. Također, direktor ima pristup povijesti svih obavijesti.

Postojeća slična rješenja

1. EZOfficeInventory

Jedna od popularnih aplikacija za vođenje inventure je EZOfficeInventory, aplikacija koja na bazi QR koda i bar koda omogućava skeniranje i praćenje inventure. Uz samo skeniranje kodova pomoću aplikacije, omogućuje povezivanje i RFID čitača za olakšanu primjenu. Kako bi se lakše pratili proizvodi unutar skladišta omogućeno je i praćenje statistike podjeljive na više lokacija, timova itd., te podjelu ljudi po ulogama unutar same firme.



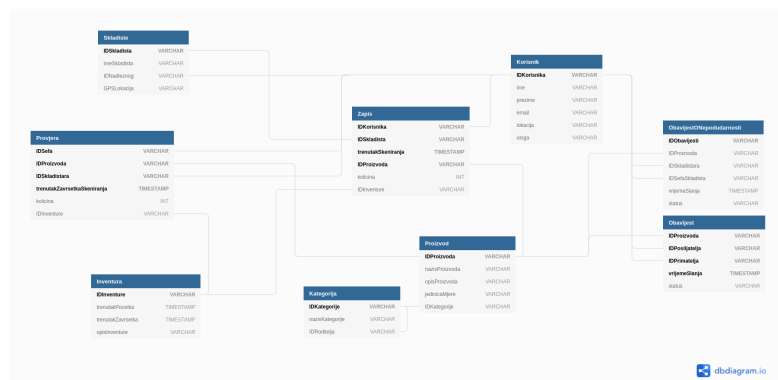
2. CountIt - Inventory application

CountIt omogućava povezivanje više mobilnih uređaja na više lokacija skladišta te sve promjene sprema na oblak. Praćenje broja uživo, skeniranje podataka i spremanje uživo na oblak. Uz to, aplikacija omogućuje i dodjelu poslova, statistike praćenja cijele inventure te prilično fleksibilan plan korištenja, jeftiniji od ostalih konkurenata te se može bilo kada otkazati (za veće grupe i više opcija skuplje).



3. Erply: Stocktake app

Erply, uz dosadašnje implementacije različitih funkcija, nudi i offline brojanje artikla (prilikom spremanja u bazu naravno ponovno mora ići online ili prebaciti podatke na računalo kablom), pomoću povijesti skeniranih kodova nudi brže prepoznavanje i samo skeniranje, notifikacije korisnicima ukoliko se broj proizvoda previše smanji.

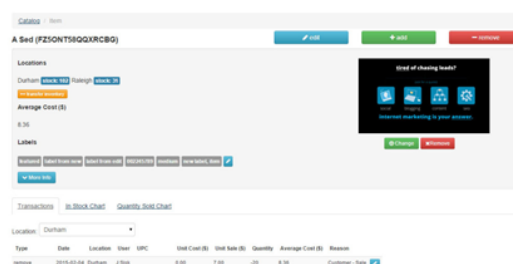


4. Stockpile by Canvas

Besplatna i jednostavnija verzija dosadašnjih aplikacija koja se fokusira na manje privatne firme kako bi im olakšali provođenje inventure. Aplikacija, za razliku od ostalih, ne nudi skeniranje kodova proizvoda, ali sadrži osnovne funkcionalnosti potrebne za provođenje inventure.



CANVUS
applications



Tržište i ciljana skupina

Ciljana skupina su tvrtke koje posjeduju veliki broj skladišta na brojnim i udaljenim lokacijama te im je tako proces inventure otežan. Tržište su sve kompanije koje posjeduju skladišta, posebice kompanije koje rade kontinuirane inventure.

Opseg projektnog zadatka

Aplikacija se sastoji od intuitivnog korisničkog sučelja, a osnovne su funkcionalnosti detekcija i skeniranje QR ili bar kodova artikala u skladištu prilikom

inventure. Svaki korisnik mora moći skenirati artikle, a ostale funkcionalnosti se dijele prema razini pristupa. Direktor ima pristup svim funkcionalnostima aplikacije, između ostaloga uvid u sva skladišta, artikle i statistiku pogrešaka pojedinog skladištara. Važna funkcionalnost je dodavanje artikala u stablo proizvoda. Pristup toj funkcionalnosti ima samo direktor tvrtke. Također, aplikacija mora omogućiti svakom skladištaru skeniranje samo jedne vrste artikala dok ne završi s prebrojavanjem te vrste artikla. Svaki korisnik može se prijaviti, registrirati ako nema napravljen račun, kao i odjaviti ako želi.

Moguće nadogradnje projektnog zadatka

Projektni zadatak mogao bi se nadograditi integracijom s nekom od popularnih aplikacija za upravljanje timom (npr. *Slack*). Tako bismo ubacili opciju međusobne komunikacije između članova tvrtke. Osim automatski generiranih poruka koje aplikacija nudi, svi članovi kompanije mogli bi razmjenjivati poruke međusobno. Uz to, jedna od mogućih nadogradnji svakako je višejezičnost. Aplikacija bi mogla evidentirati nabavne i prodajne naloge te automatski dodavati odnosno oduzimati broj artikala ovisno o njihovoj prodaji ili nabavi.

Za pomoć pogledati reference navedene u poglavlju „Popis literature“, a po potrebi konzultirati sadržaj na internetu koji nudi dobre smjernice u tom pogledu.

3. Specifikacija programske potpore

3.1 Funkcionalni zahtjevi

Dionici:

1. Skladištar
2. Šef skladišta
3. Direktor
4. Razvojni tim

Aktori i njihovi funkcionalni zahtjevi:

1. Skladištar (inicijator) može:
 - (a) Skenirati artikle za inventuru
 - (b) Pristupiti povijesti svih svojih unosa
 - (c) Dojaviti šefu skladišta da artikla nema na skladištu
 - (d) Odjaviti se
2. Šef skladišta (inicijator) može:
 - (a) Skenirati artikle za inventuru
 - i. Kao kontrolor
 - ii. Kao izvršitelj inventure
 - (b) Vidjeti sve artikle na skladištu u kojem je nadležan
 - (c) Pristupiti povijesti svih svojih unosa
 - (d) Primati obavijesti o nedostatku pojedinog artikla koje dojava skladištar
 - i. Proslijediti tu obavijest direktoru
 - (e) Odjaviti se
3. Direktor (inicijator) može:
 - (a) Skenirati artikle za inventuru
 - (b) Vidjeti povijest svih svojih unosa
 - (c) Vidjeti stanje artikala na svim skladištima

- (d) Vidjeti statistiku pogrešaka svakog skladištara
- (e) Izabrati skladišta koje želi provjeriti
 - i. Izabrati inventure na odabranom skladištu
 - ii. Pregledati sve skenirane artikle u toj inventuri
- (f) Dodati novo skladište
- (g) Urediti stablo artikala
 - i. Dodati artikl
 - ii. Ukloniti Artikl
 - iii. Dodati kategoriju
 - iv. Ukloniti kategoriju
- (h) Pristupiti svim obavijestima
 - i. Razriješiti obavijesti
- (i) Odjaviti se

4. Baza podataka (sudionik)

- (a) Pohranjivati sve podatke o korisnicima i njihovim ovlastima
- (b) Pohranjivati sve podatke o inventurama, skladištima, obavijestima i artiklima

3.1.1 Obrasci uporabe

Opis obrazaca uporabe

UC1 - Registracija

- **Glavni sudionik:** Korisnik
- **Cilj:** Stvoriti korisnički račun za pristup sustavu
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju za registraciju
 2. Korisnik unosi potrebne korisničke podatke te bira jednu od ponuđenih uloga
 3. Podaci se spremaju u bazu podataka
- **Opis mogućih odstupanja:**
 - 2.a Odabir email adrese koja se već koristi, unos podataka u nedozvoljenom formatu, ne popunjavanje svih potrebnih podataka
 1. Sustav obavještava korisnika o neuspjelom upisu i vraća ga na stranicu za prijavu
 2. Korisnik mijenja potrebne podatke te završava unos ili odustaje od registracije

UC2 - Prijava u sustav

- **Glavni sudionik:** Skladištar, šef skladišta, direktor
- **Cilj:** Prijaviti se u sustav i dobiti pristup korisničkom sučelju
- **Sudionici:** Baza podataka
- **Preduvjet:** Registracija
- **Opis osnovnog tijeka:**
 1. Korisnik unosi email adresu i lozinku
 2. Uspoređuje se korisnikov unos s podacima u bazi podataka
 3. Prikazuje se početni zaslon aplikacije
- **Opis mogućih odstupanja:**
 - 2.a Neispravna email adresa ili lozinka
 1. Sustav obavještava korisnika o neuspjelom upisu i vraća ga na stranicu za registraciju

UC3 - Odjava iz sustava

- **Glavni sudionik:** Skladištar, šef skladišta, direktor
- **Cilj:** Odjaviti se iz sustava
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju "Odjava" s ikonom mobilnog uređaja i strelice u gornjem desnom kutu
 2. Prikazuje se stranica za registraciju

UC4 - Skeniranje artikla

- **Glavni sudionik:** Skladištar, šef skladišta, direktor
- **Cilj:** Skenirati artikl
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav i odabrao je koji artikl želi skenirati
- **Opis osnovnog tijeka:**
 1. Korisnik postavlja mobilni uređaj iznad bar ili QR koda koji želi skenirati
 2. Korisnik pritiskom na ikonu QR koda (sredina, dolje) skenira kod artikla
 3. Prikazuje se pop-up window s informacijama o skeniranom artiklu
 4. Korisnik u pop-up window-u odabire mogućnost povećanja ili smanjenja broja takvih artikala na skladištu pritiskom na ikone "-" i "+", odbacivanja trenutno skeniranog proizvoda bez spremanja u bazu pritiskom na opciju "Odbaci", spremanje artikla u bazu pritiskom na opciju "Spremi" te izlaska iz pop-up window-a pritiskom na opciju "Završi"
- **Opis mogućih odstupanja:**
 - 2.a Skenirani artikl ne postoji u bazi podataka
 1. Sustav obavještava korisnika o neuspjelom pokušaju skeniranja uz poruku "Skenirani artikl ne postoji u bazi podataka" te ga vraća na početni zaslon za skeniranje artikala
 - 2.b Skenirani artikl nije onaj odabrani
 1. Sustav obavještava korisnika o neuspjelom pokušaju skeniranja uz poruku "Skenirani artikl nije onaj odabrani" te ga vraća na početni zaslon za skeniranje artikala

UC5 - Završetak skeniranja artikla

- **Glavni sudionik:** Skladištar, šef skladišta, direktor
- **Cilj:** Završiti sa skeniranjem odabranog artikla
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav i odabrao je artikl koji želi skenirati
- **Opis osnovnog tijeka:**
 1. Korisnik na početnom zaslonu pritišće gumb "Završi"
 2. Korisnik potvrđuje da želi završiti sa skeniranjem odabranog artikla

UC6 - Pregled povijesti inventura

- **Glavni sudionik:** Skladištar, šef skladišta
- **Cilj:** Pregled svih unosa artikala po inventurama
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav
- **Opis osnovnog tijeka:**
 1. Korisnik pritišće ikonu liste u donjem lijevom kutu početnog zaslona
 2. Prikazuje se popis inventura
 3. Korisnik bira inventuru
 4. Prikazuje se popis skeniranih artikala

UC7 - Pregled povijesti inventura

- **Glavni sudionik:** Direktor
- **Cilj:** Pregled svih unosa artikala po inventurama u svim skladištima
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav
- **Opis osnovnog tijeka:**
 1. Korisnik pritišće ikonu liste u donjem lijevom kutu početnog zaslona
 2. Prikazuje se popis svih skladišta
 3. Korisnik bira skladište
 4. Prikazuje se popis inventura
 5. Korisnik bira inventuru
 6. Prikazuje se popis skeniranih artikala

UC8 - Dojava da određenog artikla nema na skladištu

- **Glavni sudionik:** Skladištar, šef skladišta
- **Cilj:** Dojaviti nadređenima da određenog artikla nema na skladištu
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju "Nema artikla"
 2. Korisnik odabire naziv artikla
 3. Korisnik odabire opciju "Dojavi"

UC9 - Pregled obavijesti

- **Glavni sudionik:** Šef skladišta
- **Cilj:** Pregled obavijesti koje su poslali skladištari
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju s ikonom pošte
 2. Prikazuje se stranica s obavijestima o artiklima kojih nema na skladištu
 3. Korisnik odabire opciju "Šalji direktoru" ako to smatra opravdanim
 4. Odabrana obavijest se šalje direktoru

UC10 - Pregled obavijesti

- **Glavni sudionik:** Direktor
- **Cilj:** Pregled obavijesti koje su poslali šefovi skladišta
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju s ikonom pošte
 2. Prikazuje se stranica s obavijestima o artiklima kojih nema na skladištu i o pogreškama skladištara
 3. Korisnik odabire kvačicu ako želi određenu obavijest označiti kao riješenu, križić ako ju želi označiti kao odbačenu, ili opcije "Razriješi sve" i "Odbaci sve"

UC11 - Pregled svih artikala na skladištu

- **Glavni sudionik:** Šef skladišta
- **Cilj:** Pregled broj svih artikala na skladištu
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju s ikonom kuće
 2. Prikazuje se popis artikala u skladištu temeljem zadnje odrađene inventure i njihova količina

UC12 - Pregled svih skladišta kompanije

- **Glavni sudionik:** Direktor
- **Cilj:** Pregled svih skladišta kompanije te artikala i skladištara u njima
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju s ikonom kuće
 2. Prikazuje se popis skladišta
 3. Korisnik bira skladište
 4. Prikazuje se popis artikala i skladištara u odabranom skladištu
 5. Korisnik odabire opciju "Artikl"
 6. Prikazuje se popis artikala u odabranom skladištu
 7. Korisnik odabire opciju "Skladištar"
 8. Prikazuje se popis skladištara i njihova statistika pogreške u odabranom skladištu

UC13 - Dodavanje skladišta

- **Glavni sudionik:** Direktor
- **Cilj:** Dodati novo skladišta
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju "+" na stranici s popisom skladišta
 2. Prikazuje se stranica za unos novog skladišta
 3. Korisnik unosi novo skladište

4. Korisnik izabire drugog korisnika s oznakom "Šef skladišta" za odabranu lokaciju
5. Novo skladište se sprema u bazu podataka

UC14 - Brisanje skladišta

- **Glavni sudionik:** Direktor
- **Cilj:** Obrisati skladište
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav i skladište postoji
- **Opis osnovnog tijeka:**
 1. Korisnik odabire skladište koje želi obrisati
 2. Korisnik pritiskom na ikonu kante za smeće briše odabrano skladište

UC15 - Dodavanje artikla

- **Glavni sudionik:** Direktor
- **Cilj:** Dodati novi artikl ili kategoriju proizvoda
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju "+" na stranici s popisom artikala
 2. Prikazuje se stranica za unos novog artikla
 3. Korisnik unosi novi artikl ili kategoriju
 4. Promjene se spremaju u bazu podataka

UC16 - Preraspodjela artikla

- **Glavni sudionik:** Direktor
- **Cilj:** Preraspodijeliti postojeći artikl
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav i artikl postoji u sustavu
- **Opis osnovnog tijeka:**
 1. Korisnik odabire koji proizvod želi preraspodijeliti
 2. Otvara se pop-up window sa mogućnosti odabira nove kategorije
 3. Korisnik odabire u novu kategoriju
 4. Korisnik potvrđuje novu kategoriju
 5. Proizvod ili kategorija proizvoda se preraspodijeli u novu kategoriju

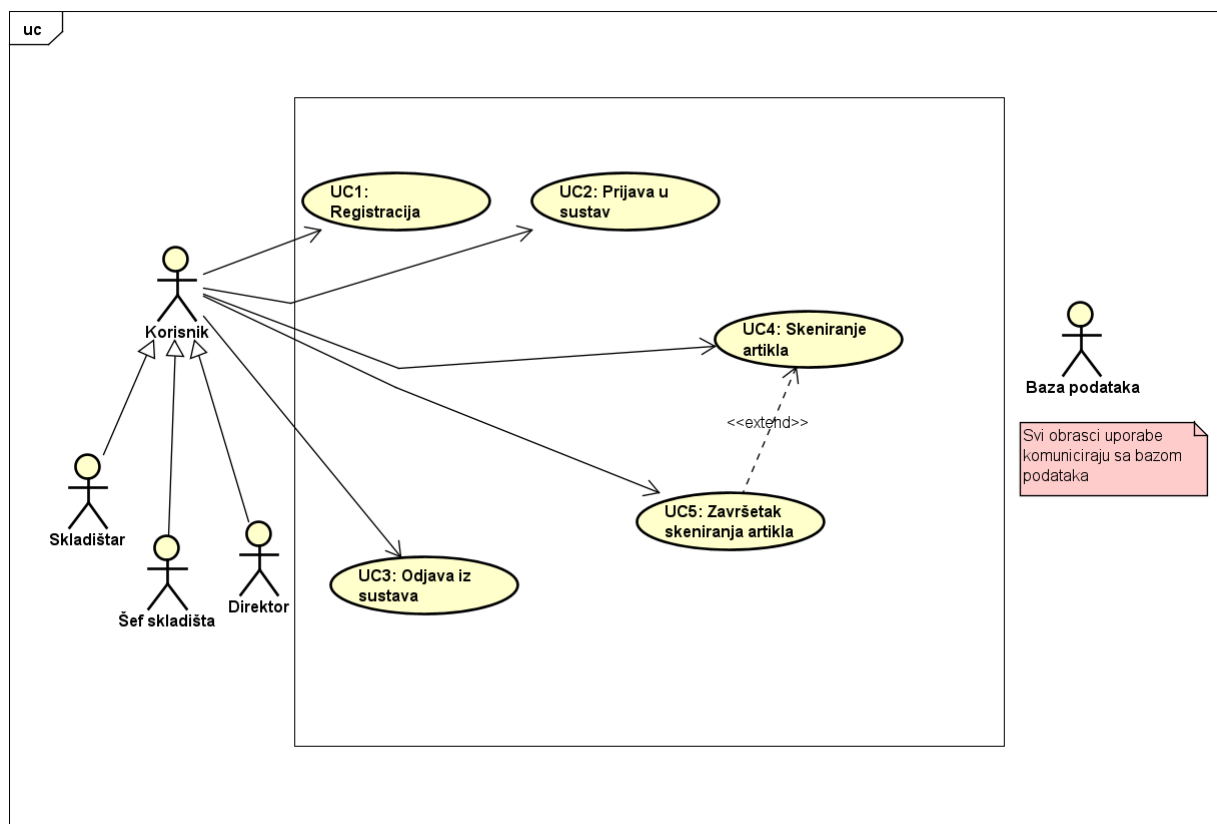
UC17 - Brisanje artikla

- **Glavni sudionik:** Direktor
- **Cilj:** Obrisati postojeći artikl ili kategoriju proizvoda
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav i artikl postoji u sustavu
- **Opis osnovnog tijeka:**
 1. Korisnik odabire koji proizvod ili kategoriju proizvoda želi obrisati
 2. Otvara se pop-up window sa upozorenjem
 3. Korisnik potvrđuje da želi obrisati proizvod
- **Opis mogućih odstupanja:**
 - 2.a Čvor kategorije proizvoda koji se želi obrisati ima djecu
 1. Sustav obavještava korisnika o pogrešci i ne dozvoljava brisanje kategorije proizvoda

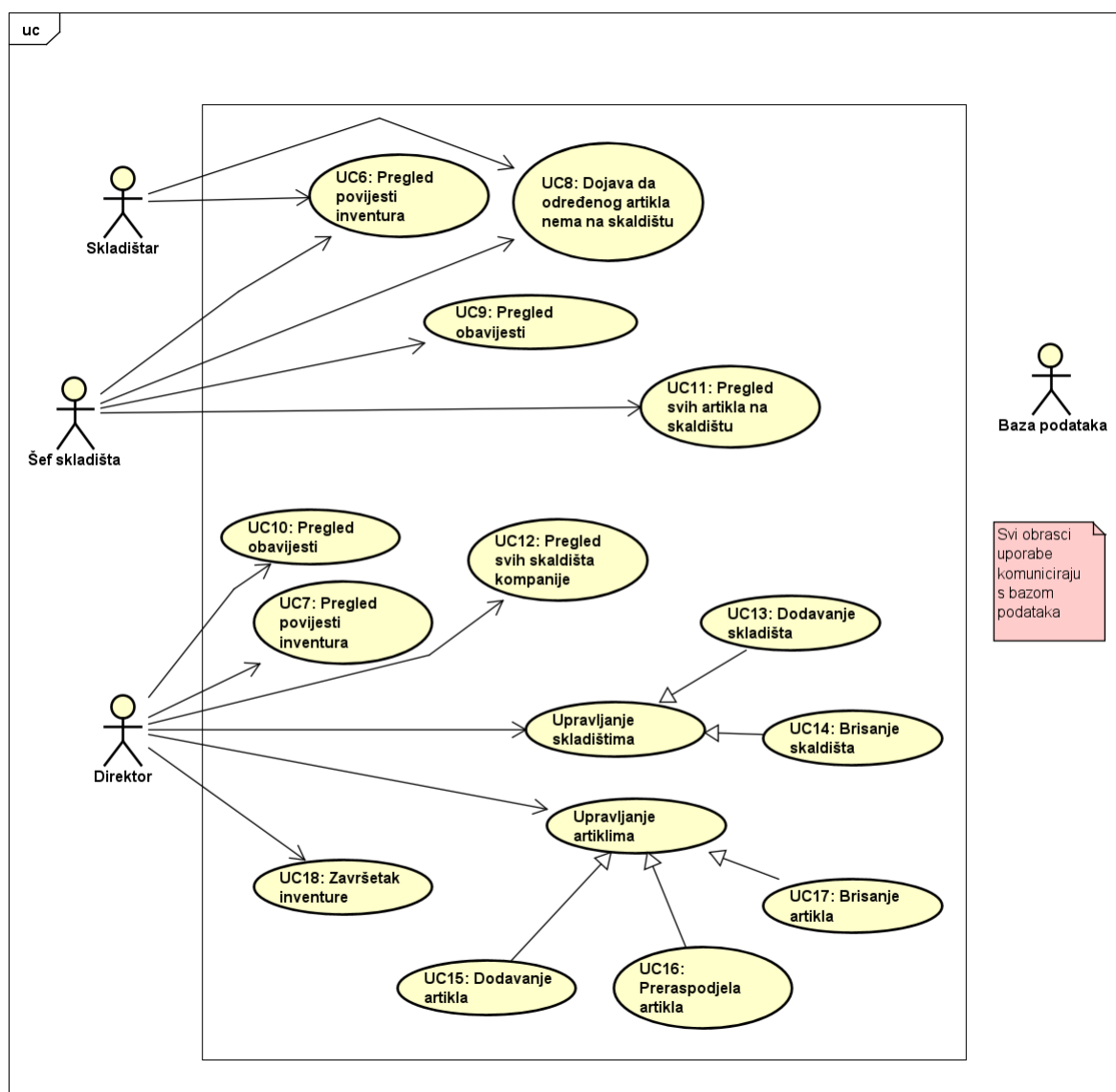
UC18 - Završetak inventure

- **Glavni sudionik:** Direktor
- **Cilj:** Završiti inventuru i započeti sljedeću
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav i inventura traje
- **Opis osnovnog tijeka:**
 1. Korisnik na početnom zaslonu odabire gumb "završi inventuru"
 2. Korisnik potvrđuje da želi završiti inventuru
 3. Trenutna inventura završava te se svi njezini unosi šalju u bazu podataka
 4. Automatski počinje nova inventura

Dijagrami obrazaca uporabe



Slika 3.1: Dijagram obrasca uporabe, zajedničke funkcionalnost svih korisnika



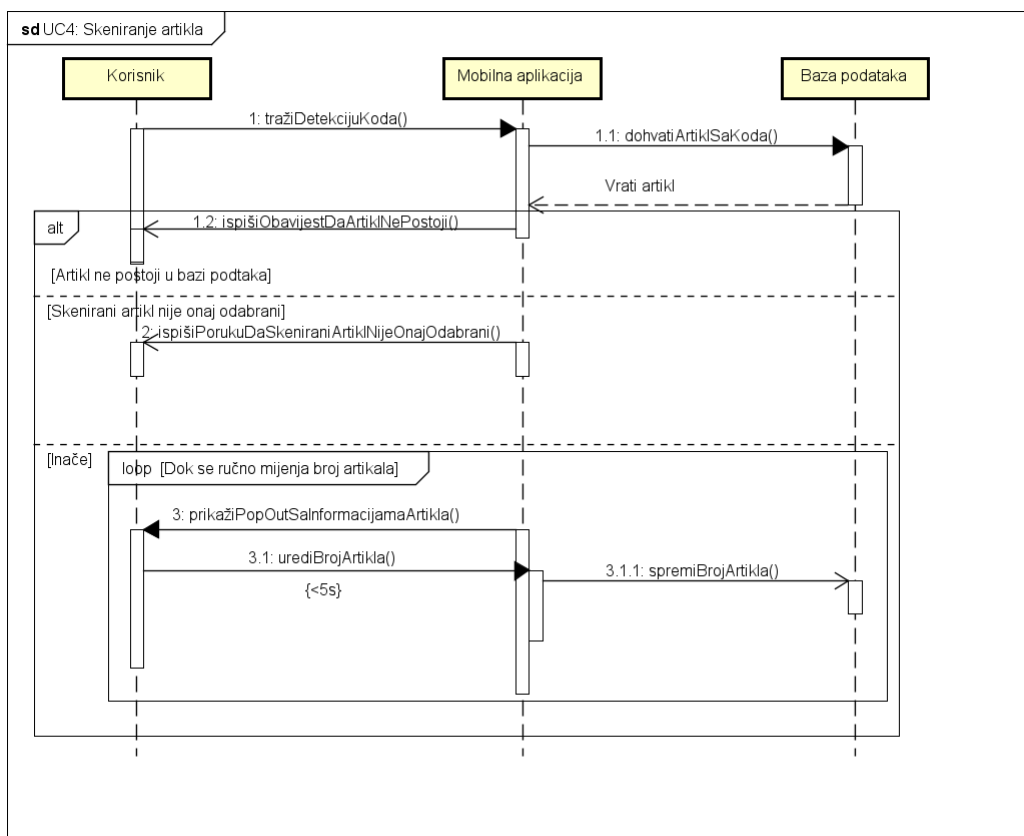
Slika 3.2: Dijagram obrasca uporabe, zasebne funkcionalnost skladištara, šefa skladišta i direktora

3.1.2 Sekvencijski dijagrami

Obrazac uporabe UC4 - Skeniranje artikla

Korisnik detekcijom koda šalje zahtjev mobilnoj aplikaciji za prikaz informacija o artiklu kako bi mogao uređivati broj tog artikla na skladištu. Ukoliko u bazi podataka ne postoji artikl sa odgovarajućim kodom mobilna aplikacija korisniku na zaslon ispiše odgovarajuću obavijest. Inače ako skenirani artikl nije onaj koji se trenutno skenira mobilna aplikacija također korisniku ispiše odgovarajuću obavijest. Ako u bazi podataka postoji artikl sa odgovarajućim kodom i ako je taj

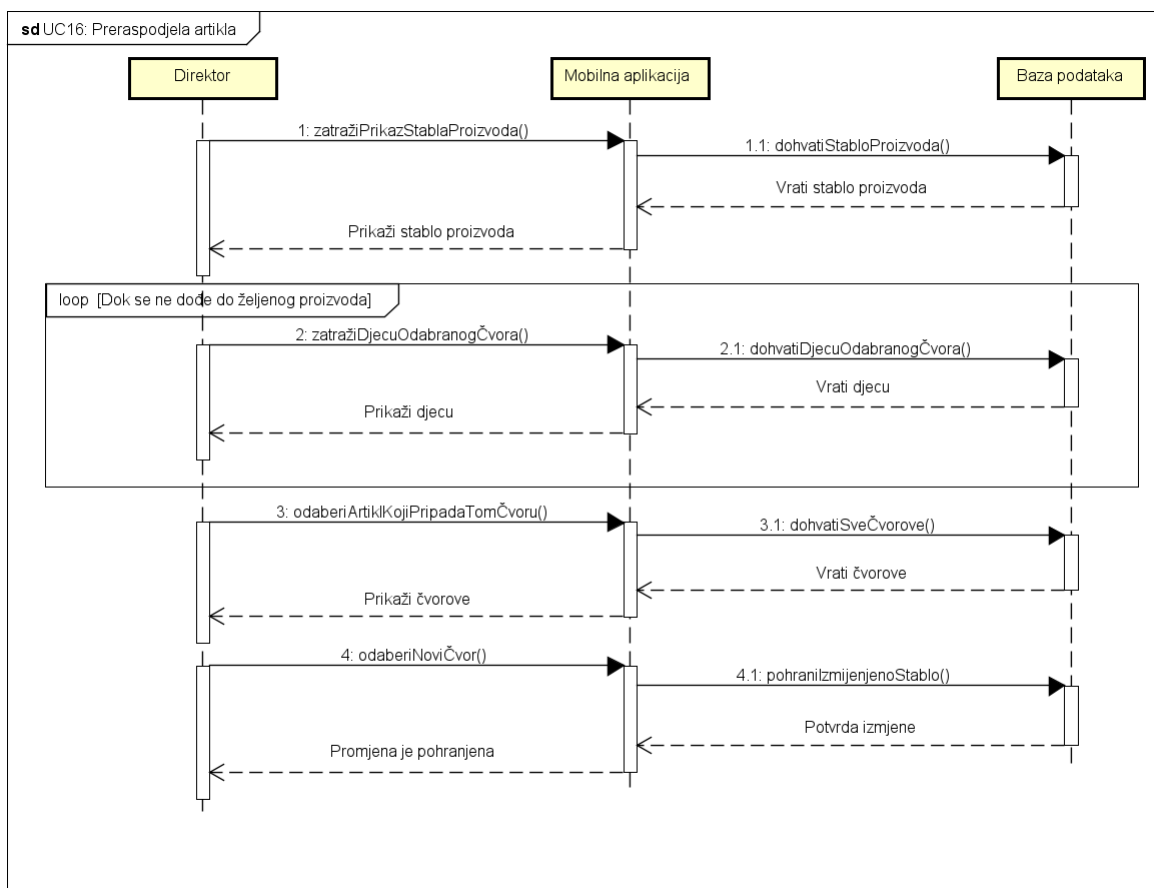
artikl onaj koji se trenutno skenira, mobilna aplikacija dohvaća podatke o artiklu i u obliku pop-up prozora prikazuje te informacije. Na zaslonu s informacijama o skeniranom proizvodu korisniku se nudi opcija da poveća ili smanji broj takvih artikala u skladištu, da odbaci trenutno očitavanje bez spremanja u bazu ili da trenutno očitavanje odmah pohrani u bazu (bez čekanja da istekne 5 sekundi). Promjenom broja očitanih artikala resetira se brojač 5 sekundi. Istekom vremena brojača zapis se automatski pohranjuje u bazu.



Slika 3.3: Sekvencijski dijagram za UC4

Obrazac uporabe UC16 - Preraspodjela artikla

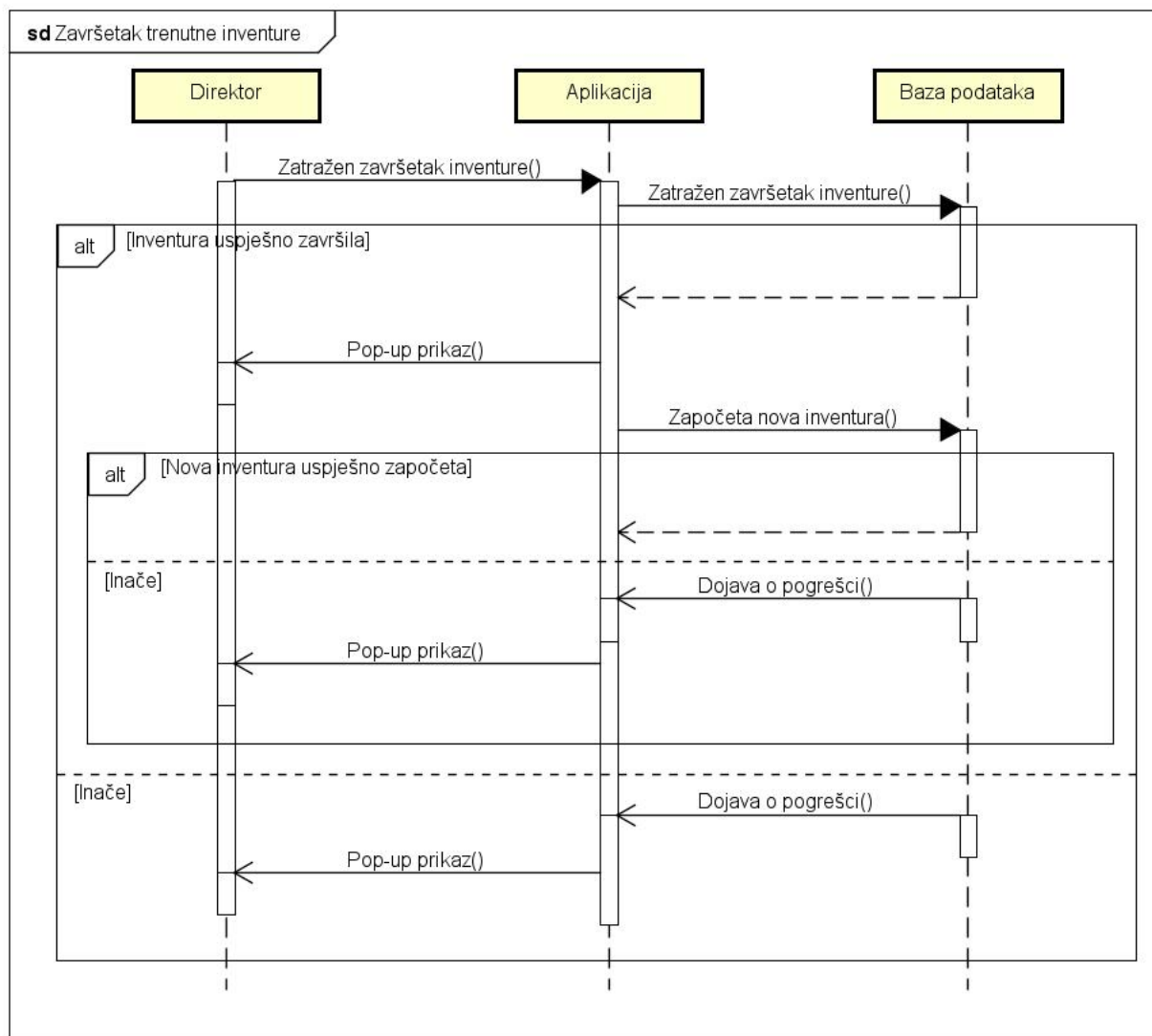
Korisnik šalje zahtjev mobilnoj aplikaciji za prikazom stabla proizvoda. Mobilna aplikacija dohvaća stablo proizvoda i prikazuje ga korisniku. Korisnik dok ne dođe do željenog proizvoda šalje zahtjeve za prikaz djece odabranog čvora. Kada korisnik odabere željeni list stabla mobilna aplikacija iz baze podataka dohvaća sve čvorove razine iznad izabranog lista. Mobilna aplikacija prikazuje sve čvorove razine iznad izabranog lista. Korisnik odabire novog roditelja. Nova struktura stabla se pohranjuje u bazu podataka.



Slika 3.4: Sekvencijski dijagram za UC16

Sekvencijski dijagram - završetak trenutne inventure

Korisnik od aplikacije traži završetak inventure. Aplikacija se povezuje s bazom podataka te od nje traži završetak trenutne inventure. U slučaju pogreške, to se dojavljuje korisniku pomoću pop-up prikaza. Ako je trenutna inventura uspješno završena, baza podataka to dojavljuje aplikacija, a ona dalje korisniku. Aplikacija nakon toga implicitno pokreće novu inventuru. Ukoliko se prilikom stvaranja nove inventure dogodi pogreška, status pogreške se dojavljuje korisniku.



Slika 3.5: Sekvencijski dijagram za završetak trenutne inventure

3.2 Ostali zahtjevi

- Sustav treba omogućiti rad više korisnika u stvarnom vremenu
- Korisničko sučelje i sustav moraju podržavati hrvatsku abecedu (dijakritičke znakove) pri unosu i prikazu tekstualnog sadržaja
- Sustav treba dohvaćati lokaciju korisnika preko GPS modula
- Izvršavanje dijela programa u kojem se pristupa bazi podataka ne smije trajati duže od nekoliko sekundi
- Sustav treba biti implementiran kao mobilna aplikacija koristeći objektno-orijentirane jezike

- Neispravno korištenje korisničkog sučelja ne smije narušiti funkcionalnost i rad sustava
- Sustav treba biti jednostavan za korištenje, korisnici se moraju znati koristiti sučeljem bez opširnih uputa
- Veza s bazom podataka mora biti kvalitetno zaštićena, brza i otporna na vanjske greške

4. Arhitektura i dizajn sustava

Aplikacija je napravljena u Flutteru koji je Googleov open-source framework za izradu aplikacija za više platformi iz jednog programskog koda. Programski jezik koji se koristi u Flutteru je Dart koji ima objektno orijentiranu prirodu, ali je uz to prilagođen za izradu UI-a. Sama arhitektura sustava slična je MVC modelu.

Postoji Model napravljen na temelju objektno orijentiranog programiranja, a poistovjećen je s dokumentima u odgovarajućim kolekcijama u bazi podataka.

Uz Model postoji i View čija je svrha prikazati svaki ekran koristeći podatke dobivene iz Modela. Ta dva dijela povezana su funkcijama tamo gdje su podatci statički, odnosno Streamovima gdje se podatci mijenjaju, bilo akcijom trenutnog korisnika ili nekog drugog korisnika čija akcija utječe na trenutno izvođenje aplikacije. Stream je također vrsta arhitekture koja se koristi u Flutteru, a povezuje View i Model na način da kad god dođe do promjene u Modelu, View se automatski ažurira.

Model je tako povezan sa streamom podataka iz baze te se svakom promjenom podataka automatski okine i promjena Modela, što uzrokuje automatsko osvježavanje korisnikovog ekrana s najnovijim podacima. Arhitekturu čine još i Provideri koji se također često koriste u Flutteru, a njihova je svrha da propagiraju podatke između raznih stranica aplikacije.

Na ovaj način Controller smo zamijenili Streamovima i Providerima koji nam omogućuju kvalitetan protok informacija kroz čitavu aplikaciju.

4.1 Baza podataka

Baza podataka odabrana za potrebe zadatka je Firebase Firestore.

Firebase je platforma razvijena od strane Googlea za izradu mobilnih i web aplikacija, a izrazito je dobro podržana za aplikacije pisane u Flutteru. Firestore je *document-oriented* NoSQL baza podataka, zbog čega se ER model prikazan u nastavku u određenoj mjeri razlikuje od realne implementacije u Firebaseu. Onome što bi u bazi podataka modeliranoj prema ER modelu bio jedan redak neke tablice, u Firebaseu je jedan primjerak *document*-a. On je reprezentiran jedinstvenim

ključem koji generira sam Firebase, a pripada nekoj sebi nadređenoj kolekciji (eng. *collection*), na neki način istovjetnoj ER entitetu. Svaki *document* može sadržavati proizvoljno mnogo informacija proizvoljnog tipa podataka, povezanih u *collection*, što bi se moglo poistovijetiti s atributima u ER modeliranoj bazi podataka.

Baza podataka ove mobilne aplikacije sastoji se od sljedećih entiteta:

- Korisnik
- Skladište
- Proizvod
- Kategorija
- Zapis
- Inventura
- Provjera
- Obavijest
- Obavijest o nepodudarnosti

4.1.1 Opis tablica

Napomena: Svi su ID atributi tipa VARCHAR zbog toga što Firebase pri dodavanju novih instanci stvara jedinstveni identifikator - ID, koji je kombinacija slova i znamenki.

Entitet **Korisnik** predstavlja svakog registriranog korisnika aplikacije. Sastoji se od atributa: IDKorisnika, ime, prezime, email, lokacija na kojoj je skeniran prvi proizvod pri odabiru proizvoda te uloga koju korisnik ima u skladištu. Ovaj entitet povezan je s entitetom Skladište preko atributa IDKorisnika koji je strani ključ u Skladištu kao IDNadležnog odgovarajućeg skladišta. Budući da jedno skladište može imati samo jednog šefa, a svaki šef može biti nadležan za samo jedno skladište, veza je *One-to-One*.

Korisnik		
IDKorisnika	VARCHAR	jedinstveni identifikator, kombijacija slova i znamenki
ime	VARCHAR	ime korisnika
prezime	VARCHAR	prezime korisnika
email	VARCHAR	email s kojim se korisnik prijavio u sustav
lokacija	VARCHAR	ime skladišta u kojem je osoba trenutno, određuje se prema GPS lokaciji koju dohvaća mobitel u trenutku skeniranja prvog artikla (odabira artikla), a za šefove skladište je fiksna prema registraciji
uloga	VARCHAR	pozicija koju korisnik ima u sustavu, može biti skladištar, šef skladišta ili direktor, odabire se pri registraciji

Entitet **Skladište**, kao što mu ime kaže, reprezentacija je skladišta tvrtke. U njemu su zapisani atributi: IDSkadišta, njegovo u praksi korišteno ime, ID šefa skladišta koji je ondje nadležan, kao i GPS lokacija na kojoj se skladište nalazi.

Skladište		
IDSkladišta	VARCHAR	jedinstveni identifikator skladišta
imeSkladišta	VARCHAR	intuitivni naziv skladišta
IDNadležnog	VARCHAR	jedinstveni identifikator šefa koji je nadležan za skladište
GPSLokacija	VARCHAR	GPS koordinate skladišta, zadaje ih direktor pri kreiranju novog skladišta

Proizvod je entitet koji sadržava sve važne informacije o proizvodu, kao što su: njegov ID, kratak naziv, malo detaljniji opis te jedinica mjere koja se koristi za proizvod. U vezi je s nekoliko entiteta. Jedan od njih je Zapis koji predstavlja "jedno skeniranje" određenog broja istovrsnih proizvoda, a veza je *One-to-Many*. Druga je *Many-to-One* veza s entitetom Provjera, gdje je IDProizvoda ujedno i dio kompo-

zitnog ključa. U *Many-to-One* vezi je i s entitetom Obavijest, gdje je IDProizvoda također dio kompozitnog ključa, dok je u *Many-to-One* vezi s entitetom Obavijest o nepodudarnosti samo strani ključ. Strani mu je ključ i IDKategorije iz entiteta Kategorija, povezanog *Many-to-One* vezom, a predstavlja kategoriju u koju se proizvod svrstava.

Proizvod		
IDProizvoda	VARCHAR	jedinstveni identifikator proizvoda, čita se iz QR ili bar koda
nazivProizvoda	VARCHAR	kratak naziv proizvoda, može imati minimalan opis kako bi se razlikovao od drugih (npr. CocaCola Zero 0.5l)
opisProizvoda	VARCHAR	malo detaljniji opis proizvoda, u njemu su sadržani prilagođeni podatci (npr. veličina i materijal za odjeću, rok trajanja za hranu i sl.)
jedinicaMjere	VARCHAR	jedinica u kojoj se proizvod prodaje, može biti komad, kilogram, metar i sl.
IDKategorije	VARCHAR	jedinstveni identifikator kategorije kojoj proizvod pripada

Kategorija je entitet koji služi za kategorizaciju proizvoda u određene grupe i podgrupe. Zbog toga ima attribute IDKategorije, nazivKategorije te IDRoditelja, i to na način da je IDRoditelja strani ključ u rekurzivnoj *One-to-many* vezi tako da svako dijete može imati samo jednog roditelja, a roditelj može imati N djece.

Kategorija		
IDKategorije	VARCHAR	jedinstveni identifikator kategorije
nazivKategorije	VARCHAR	kratak naziv kategorije (npr. hrana, mliječni proizvod)
IDRoditelja	VARCHAR	jedinstveni identifikator kategorije koja joj je nadređena

Zapis predstavlja jedno skeniranje jedne vrste proizvoda (svaki put kada koris-

nik odabere da želi ažurirati stanje proizvoda, dodaje se novi zapis). Ima attribute IDKorisnika, IDSklađišta, trenutakSkeniranja, ID skeniranog proizvoda, količina koja je skenirana te ID inventure tijekom koje se skeniranje dogodilo. Primarni mu je ključ kompozitni, posuđuje IDKorisnika iz Korisnika, IDSklađišta iz Sklađišta te IDProizvoda iz Proizvoda. Sadrži i atribut količina te IDInventure koji je posuđen iz entiteta Inventura. Veza je *Many-to-One*.

Zapis		
IDKorisnika	VARCHAR	jedinstveni identifikator korisnika koji je skenirao proizvod
IDSkladišta	VARCHAR	jedinstveni identifikator skladišta u kojem je proizvod skeniran
trenutak Skeniranja	TIMESTAMP	trenutak u kojem je očitana QR ili bar kod
IDProizvoda	VARCHAR	jedinstveni identifikator skeniranog proizvoda
količina	INT	broj proizvoda koji je skeniran
IDInventure	VARCHAR	jedinstveni identifikator inventure

Entitet **Inventura** sadrži osnovne informacije o inventuri. Njezini su atributi: IDInventure, trenutak početka i završetka inventure, te kratak opis.

Inventura		
IDInventure	VARCHAR	jedinstveni identifikator inventure
trenutak Početka	TIMESTAMP	trenutak početka inventure
trenutak Završetka	TIMESTAMP	trenutak završetka inventure
opisInventure	VARCHAR	kratak opis inventure (npr. božićna, polugodišnja i sl.)

U entitetu **Provjera** zapisane su informacije o situaciji kada šef skladišta skenira proizvod koji je već ranije skenirao skladištar. Ima posuđene ključeve ID šefa,

ID proizvoda koji je skeniran te ID skladištara kojeg šef provjerava, trenutak u kojem je šef završio skeniranje, količinu koju je izbrojio šef te ID inventure o kojoj se radi. Veze su *Many-to-One* s entitetom Korisnik preko ID-a šefa i skladištara, entitetom Proizvod preko ID-a proizvoda, a primarni ključ čini još i trenutak završetka skeniranja od strane šefa.

Provjera		
IDŠefa	VARCHAR	jedinstveni identifikator šefa skladišta koji provjerava broj skeniranih proizvoda
IDProizvoda	VARCHAR	jedinstveni identifikator proizvoda koji skenira šef, a skenirao ga je skladištar
IDSkladištara	VARCHAR	jedinstveni identifikator skladištara koji je skenirao proizvod koji je i šef skenirao
trenutak Završetka Skeniranja	TIMESTAMP	trenutak kada je šef skladišta prekinuo skeniranje odabranog proizvoda
količina	INT	broj proizvoda koje je skenirao šef skladišta
IDInventure	VARCHAR	jedinstveni identifikator inventure

Obavijest je entitet koji opisuje funkcionalnost slanja da određenog artikla nema na skladištu od strane skladištara sebi nadređenom šefu, a šef ju onda dalje može proslijediti direktoru. Kompozitni ključ čine posuđeni atributi ID proizvoda (veza *Many-to-One* s entitetom Proizvod), ID pošiljatelja i ID primatelja (veza *One-to-One* s entitetom Korisnik) i vrijeme slanja, a postoji i atribut status. On služi za označavanje akcija koje direktor može provesti nad obavijesti - može ju označiti kao odbačenu ili kao riješenu.

Obavijest		
IDProizvoda	VARCHAR	jedinstveni identifikator proizvoda kojeg nema na skladištu

Nastavljeno na idućoj stranici

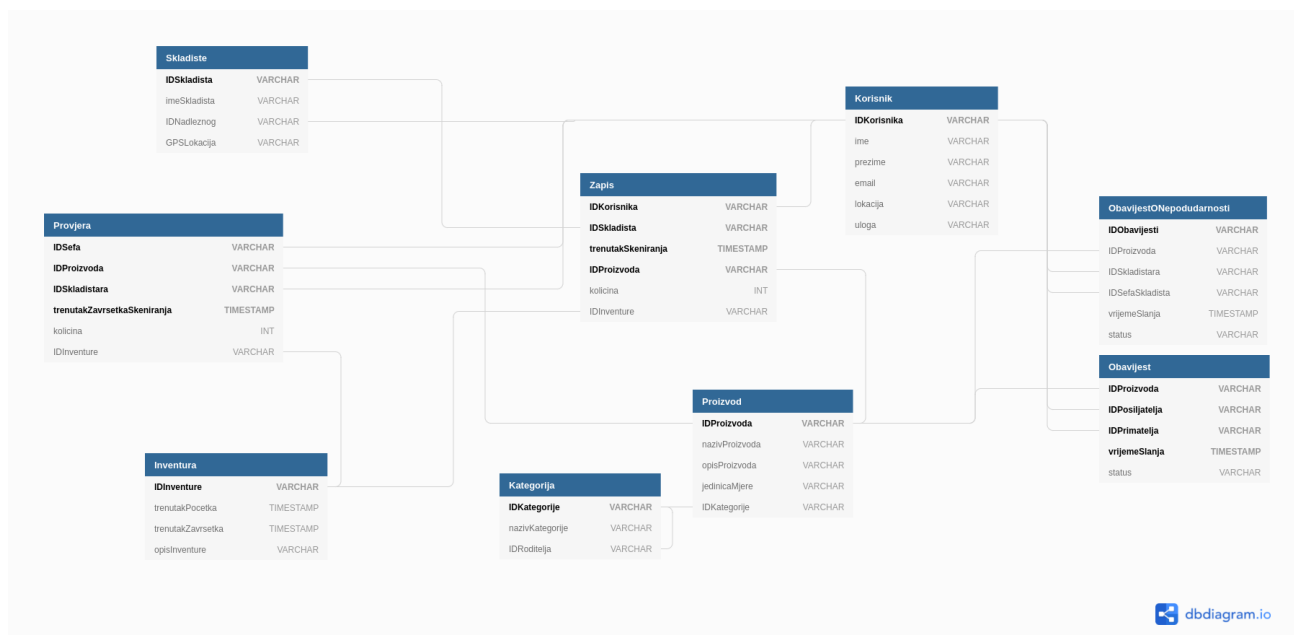
Nastavljeno od prethodne stranice

Obavijest		
IDPošiljatelja	VARCHAR	jedinstveni identifikator pošiljatelja obavijest, skladištar ili šef skladišta
IDPrimatelja	VARCHAR	jedinstveni identifikator primatelja obavijesti, može biti šef skladišta ili direktor
vrijemeSlanja	TIMESTAMP	trenutak slanja obavijesti
status	VARCHAR	stanje u kojem je poruka trenutno, odbačena od strane direktora ili riješena

Obavijest o nepodudarnosti entitet je koji nosi informacije o obavijesti koju šalje sustav kada dođe do toga da šef i skladištar nisu skenirali jednak broj proizvoda. Atributi su joj: ID obavijesti, ID proizvoda (strani ključ iz entiteta Proizvod, veza *Many-to-One*), ID skladištara i šefa skladišta (strani ključevi iz entiteta Korisnik povezanog vezom *Many-to-One*), vrijeme slanja i status (kao u entitetu Obavijest).

Obavijest o nepodudarnosti		
IDObavijesti	VARCHAR	jedinstveni identifikator obavijesti o nepodudarnosti
IDProizvoda	VARCHAR	jedinstveni identifikator proizvoda kojeg nema na skladištu
IDSkladištara	VARCHAR	jedinstveni identifikator skladištara čiji se broj skeniranih artikala ne podudara sa šefovim
IDŠefaSkladišta	VARCHAR	jedinstveni identifikator šefa skladišta čiji se broj skeniranih artikala ne podudara sa skladištarevim
vrijemeSlanja	TIMESTAMP	trenutak slanja obavijesti
status	VARCHAR	stanje u kojem je poruka trenutno, odbačena od strane direktora ili riješena

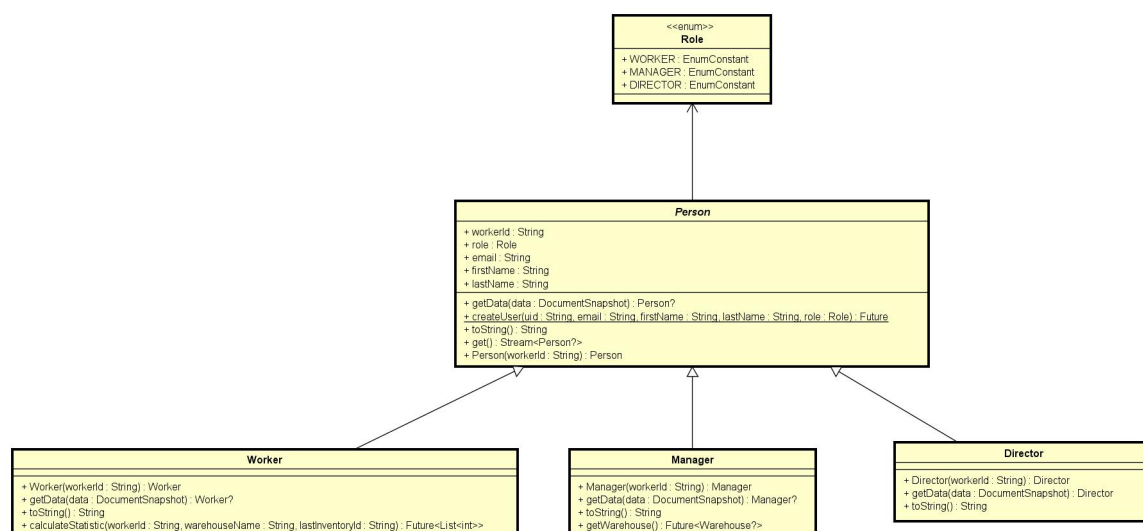
4.1.2 Dijagram baze podataka



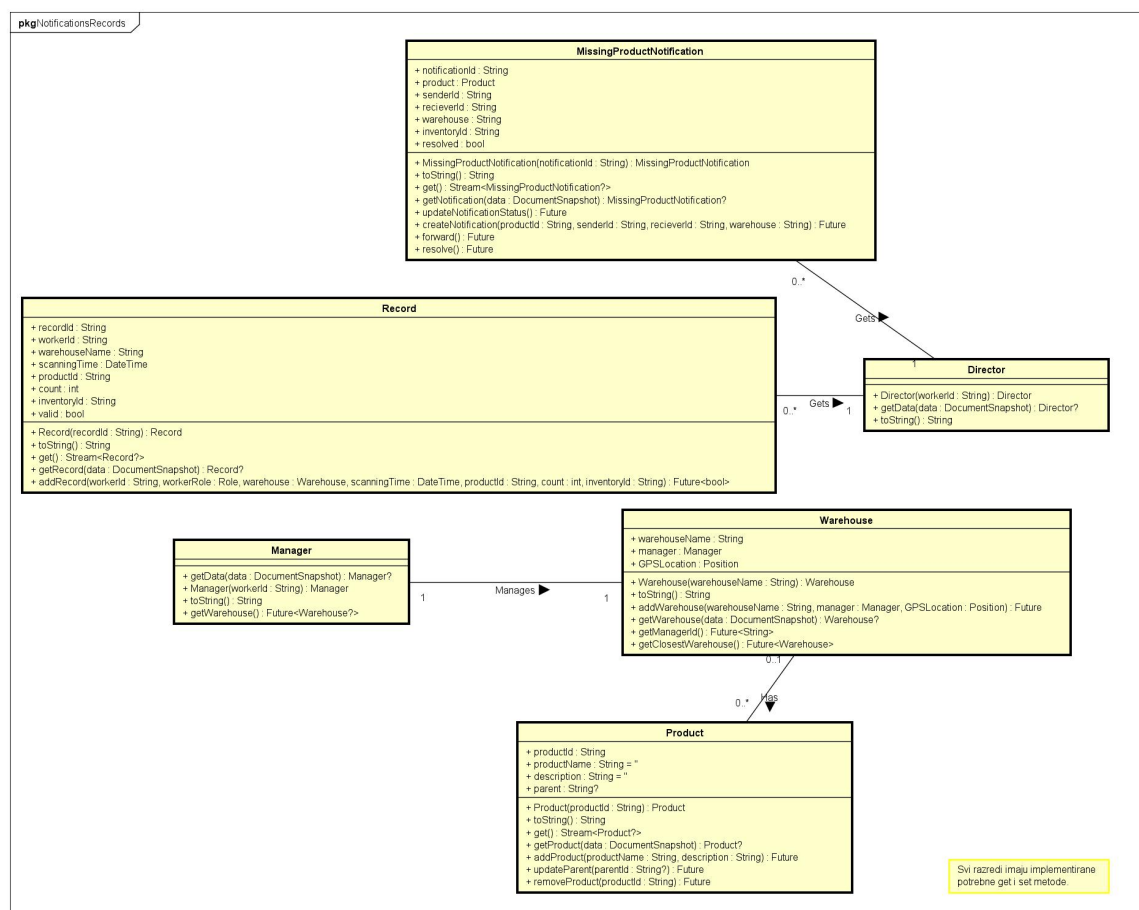
Slika 4.1: ER dijagram baze podataka

4.2 Dijagram razreda

Na slikama 4.2 i 4.3 prikazani su razredi koji pripadaju backend dijelu aplikacije. Razredi sa slike 4.2 nasljeđuju razred Person i implementiraju dodatne potrebne metode. Razred Person sadrži još dodatno atribut role, koji je po tipu enumeracija Role. Slika 4.3 prikazuje odnose razreda koji se međusobno ne nasljeđuju.



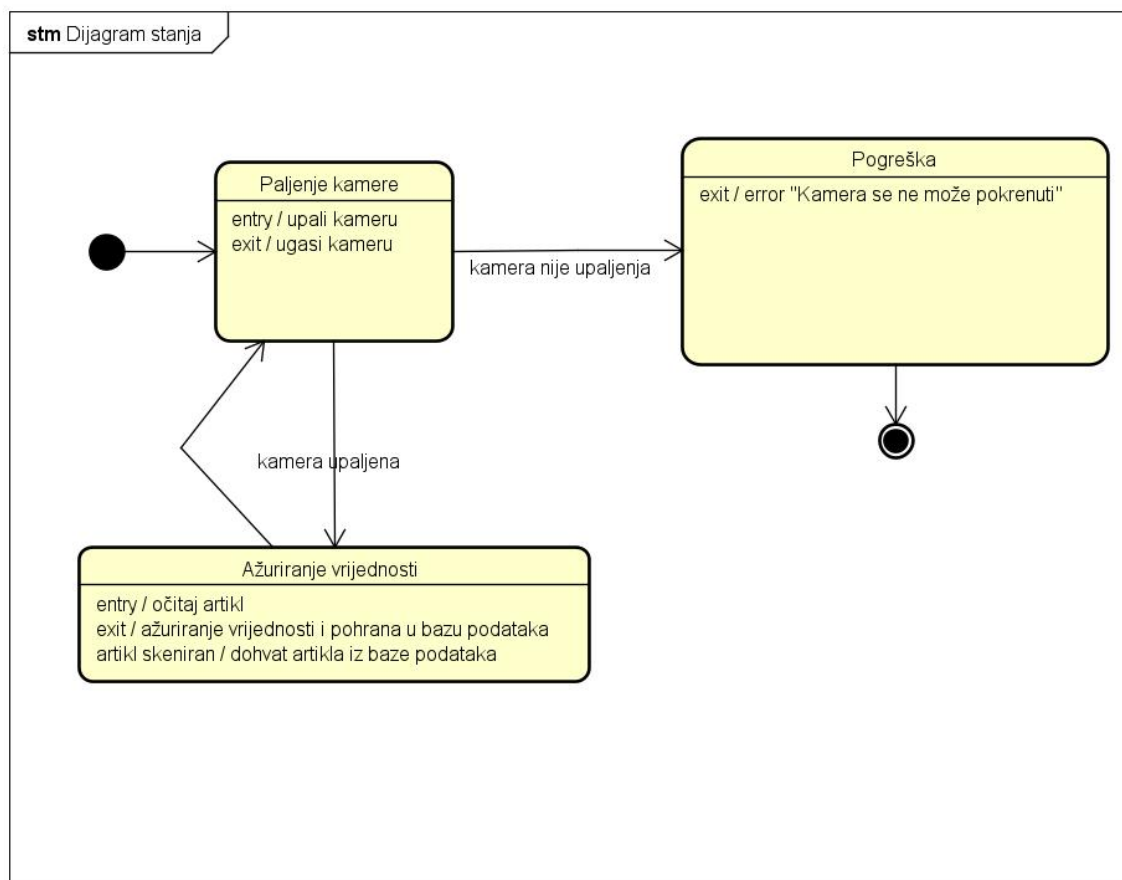
Slika 4.2: Dijagram razreda - razredi koji nasljeđuju razred Person



Slika 4.3: Dijagram razreda - odnosi među klasama

Svi razredi prikazani na slici 4.3 komuniciraju s bazom podataka kako bi dohvatili i spremili potrebne attribute. Razred Director ima najveće ovlasti. Nakon njega ide razred Manager, pa nakon njega Worker.

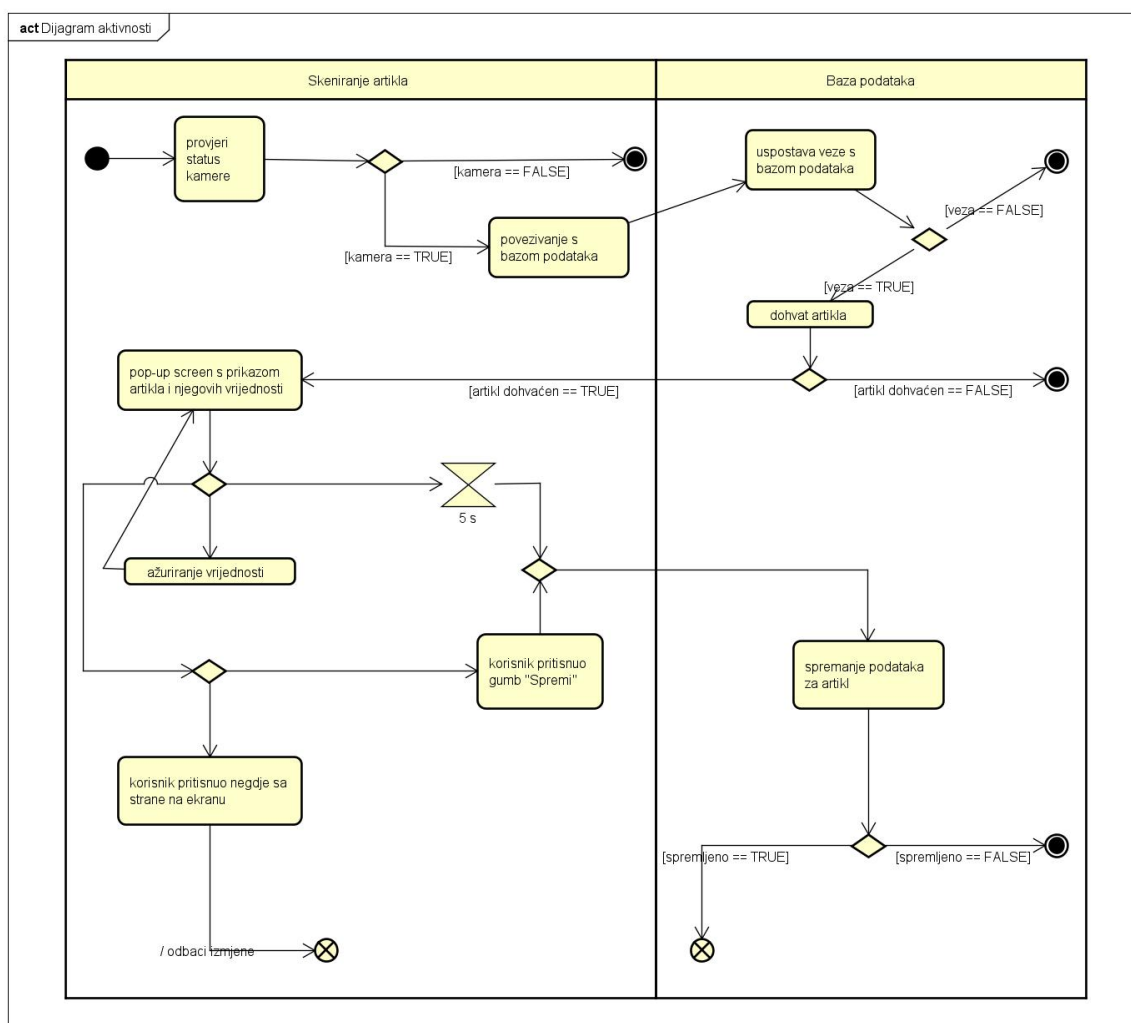
4.3 Dijagram stanja



Slika 4.4: Dijagram stanja - paljenje kamere i skeniranje artikala

Slika 4.4 prikazuje dijagram stanja aplikacije prilikom pokretanja kamere. Prvi zadatak aplikacije je pokrenuti kameru na uređaju korisnika. Ukoliko pokretanje kamere nije moguće, dojavljuje se pogreška i izlazi se iz prozora. Ako je kamera uspješno pokrenuta, korisnik započinje skeniranje artikla. Najprije se dohvaćaju podaci o artiklu iz baze podataka i prikazuju se korisniku. Zatim, korisnik ažurira vrijednosti artikala te se, prilikom dovršetka rada, promjene spremaju u bazu podataka i kamera se gasi.

4.4 Dijagram aktivnosti

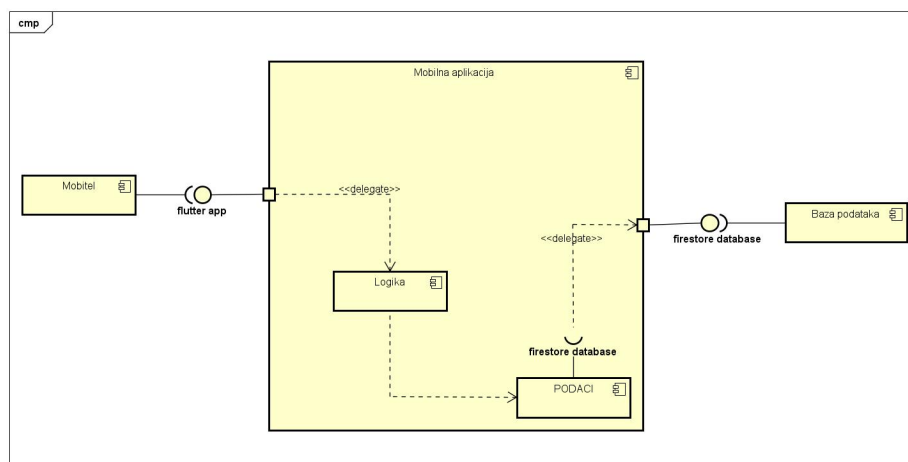


Slika 4.5: Dijagram aktivnosti - skeniranje artikla i komunikacija s bazom podataka

Slika 4.5 prikazuje paljenje kamere i ažuriranje podataka o skeniranom artiklu u bazu podataka. Prvo se provjerava status kamere. Ukoliko se kamera nije uspjela pokrenuti, prekida se proces skeniranja artikla. Ako se kamera pokrenula, uspostavlja se veza s bazom podataka, kako bi korisnik mogao dohvatiti podatke o skeniranom artiklu. Ukoliko se veza s bazom podataka nije uspjela uspostaviti, prekida se proces skeniranja artikla. U suprotnom, baza podataka vraća podatke o traženom artiklu te se ti podaci prikazuju korisniku u obliku pop-up screena. Korisnik tu unosi nove vrijednosti, koje se u bazu podataka spremaju na dva načina: nakon isteka vremena od 5 sekundi ili kada korisnik pritisne gumb "Spremi". Ko-

risnik također može prekinuti skeniranje artikla u bilo kojem trenutku, pritiskom bilo gdje na ekran uređaja, prilikom čega se odbacuju izmjene. Na kraju, ukoliko se podaci nisu uspjeli spremiti u bazu, gasi se skeniranje artikala. Ukoliko su podaci uspješno spremljeni, aplikacija završava normalno s radom i korisnik može započeti novo skeniranje artikla.

4.5 Dijagram komponenti



Slika 4.6: Dijagram komponenti - struktura aplikacije

Slika 4.6 prikazuje rad i komunikaciju između pojedinih komponenti aplikacije. Aplikacija je namijenjena za pokretanje na mobilnim uređajima, a pisana je u Flutteru. Podaci potrebni za rad aplikacije se dohvaćaju iz Firestore baze podataka.

5. Implementacija i korisničko sučelje

5.1 Korištene tehnologije i alati

Da bi postigli bolju komunikaciju te se bolje organizirali oko podjele zadataka koristili smo aplikacije Discord¹ i WhatsApp². Za izradu dokumentacije smo koristili TeXstudio³ - integrirano okruženje za pisanje LaTeX dokumenata, a za UML dijagrame smo koristili Astah Professional⁴. Udaljeni repozitorij projekta dostupan je na web platformi Gitlab⁵.

Kao razvojno okruženje koristili smo Visual Studio Code⁶ - uređivač izvornog koda koji je napravio Microsoft. VS Code ima ugrađeni sustav za upravljanje izvornim kodom Git⁷ koji smo koristili u ovom projektu.

Aplikacija je napravljena uz pomoć radnog okvira Flutter⁸ i programsog jezika Dart⁹. Flutter je održavan od strane Google-a, a koristi se za razvoj aplikacija za različite sustave - najčešće za razvoj mobilnih aplikacija za android i IOS. Baza podataka se nalazi na poslužitelju Firebase¹⁰ koji je također nastao od strane Google-a. Firebase je software koji nudi razne usluge za lakšu i bržu izradu Android, IOS i web aplikacija.

¹<https://discord.com/>

²<https://www.whatsapp.com/>

³<https://www.texstudio.org/>

⁴<http://astah.net/editions/professional>

⁵<https://gitlab.com/>

⁶<https://code.visualstudio.com/>

⁷<https://git-scm.com/>

⁸<https://flutter.dev/multi-platform/mobile>

⁹<https://dart.dev/>

¹⁰<https://firebase.google.com/>

5.2 Ispitivanje programskog rješenja

5.2.1 Ispitivanje komponenti

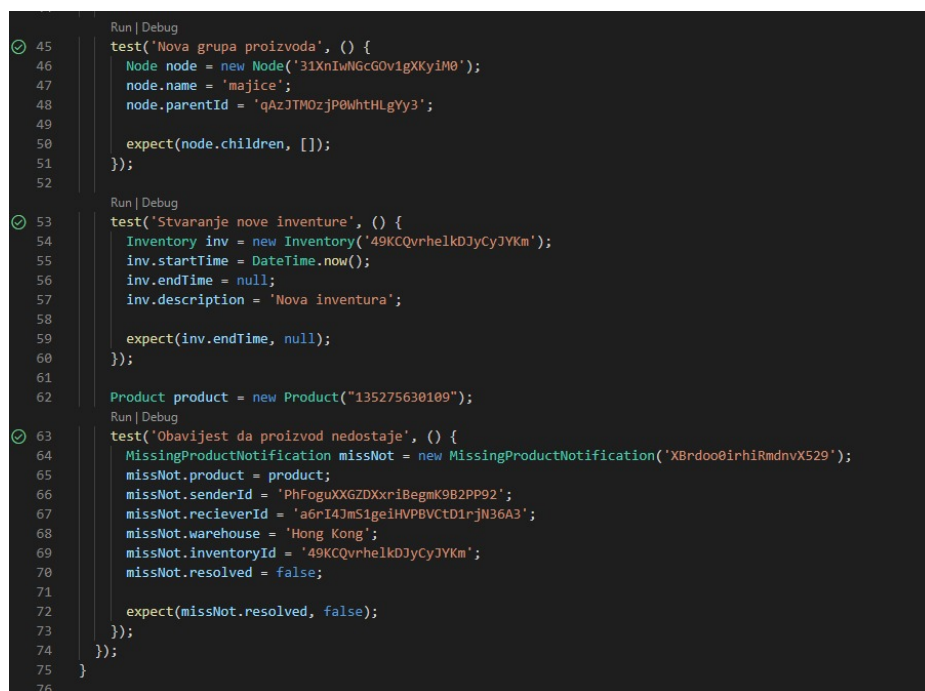
Budući da je aplikacija napisana za Android uređaje koristeći Flutter, za testove je korišten implementirani paket *flutter_test: sdk: flutter* pomoću kojeg su napravljeni unit testovi aplikacije.

Testiraju se razne funkcionalnosti dostupne na android uređaju, među kojima vrijedi izdvojiti testiranje stvaranja novog skladištara, stvaranje novog proizvoda, Unos novog zapisa skeniranja artikla, stvaranje nove grupe proizvoda, stvaranje nove inventure i obavijesti da proizvod ne postoji.

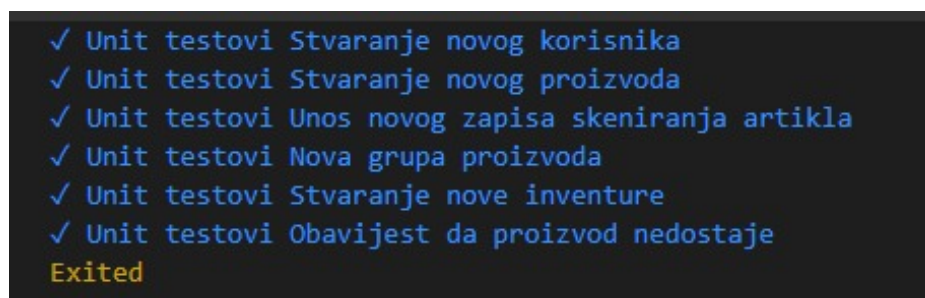
Primjer testiranja:

```
10 void main() {  
    Run | Debug  
11   group('Unit testovi', () {  
    Run | Debug  
12     test('Stvaranje novog korisnika', () {  
13       Worker worker = new Worker("A1b1hSr90XPXXk33JYz00VTSRm44");  
14       worker.email = 'ih@gmail.com';  
15       worker.firstName = 'Ivan';  
16       worker.lastName = 'Horvat';  
17       worker.firstName = 'Ivan';  
18       worker.role = Role.WORKER;  
19  
20       expect(worker.firstName, 'Ivan');  
21     });  
22  
    Run | Debug  
23     test('Stvaranje novog proizvoda', () {  
24       Product product = new Product("135275630109");  
25       product.productName = 'Nutella';  
26       product.parent = null;  
27       product.description = 'Slasni čokoladni namaz';  
28  
29       expect(product.parent, null);  
30     });  
31  
    Run | Debug  
32     test('Unos novog zapisa skeniranja artikla', () {  
33       Record record = new Record('1fpcFk3KL6nOPfBe3Ktk');  
34       record.workerId = 'A1b1hSr90XPXXk33JYz00VTSRm44';  
35       record.warehouseName = 'Maksimir';  
36       record.scanningTime = DateTime.now();  
37       record.productId = '135275630109';  
38       record.count = 10;  
39       record.inventoryId = '49KCQvrhelkDJyCyJYKm';  
40       record.valid = false;  
41  
42       expect(record.valid, false);  
43     });  
44   });  
}
```

Slika 5.1: Screenshot unit test koda (1)



Slika 5.2: Screenshot unit test koda (2)



Slika 5.3: Rezultati unit testiranja

5.2.2 Ispitivanje sustava

Ispitivanje sustava je provedeno koristeći funkcionalnosti iz flutterovog paketa `integration_test` kojeg je potrebno instalirati tako da se u `pubspec.yaml` file pod `dev_dependencies` dodaju sljedeće linije

```

integration_test:
  sdk: flutter

```

i u terminal unese naredba flutter pub get. Sav kod za ispitivanje sustava napisan je u programskom jeziku dart. Nad sustavom proveli smo sljedeća testiranja:

Otvaranje pop up prozora nakon pritiska gumba za skeniranje:

```
testWidgets('Skeniraj i otvori pop up', (WidgetTester tester) async {  
  app.main();  
  await tester.pumpAndSettle();  
  final Finder scan = find.byTooltip('scan');  
  await tester.tap(scan);  
  expect(find.byType(AlertDialog), findsOneWidget);  
});
```

Otvaranje pop up prozora prilikom dodavanja novoga skladišta:

```
testWidgets('Otvaranje dialoga prilikom dodavanja skladišta', (WidgetTester tester) async {  
  app.main();  
  await tester.pumpAndSettle();  
  final Finder addWarehouse = find.byTooltip('addWarehouse');  
  await tester.tap(addWarehouse);  
  expect(find.byType(AlertDialog), findsOneWidget);  
});
```

Upisivanje podataka za prijavu:

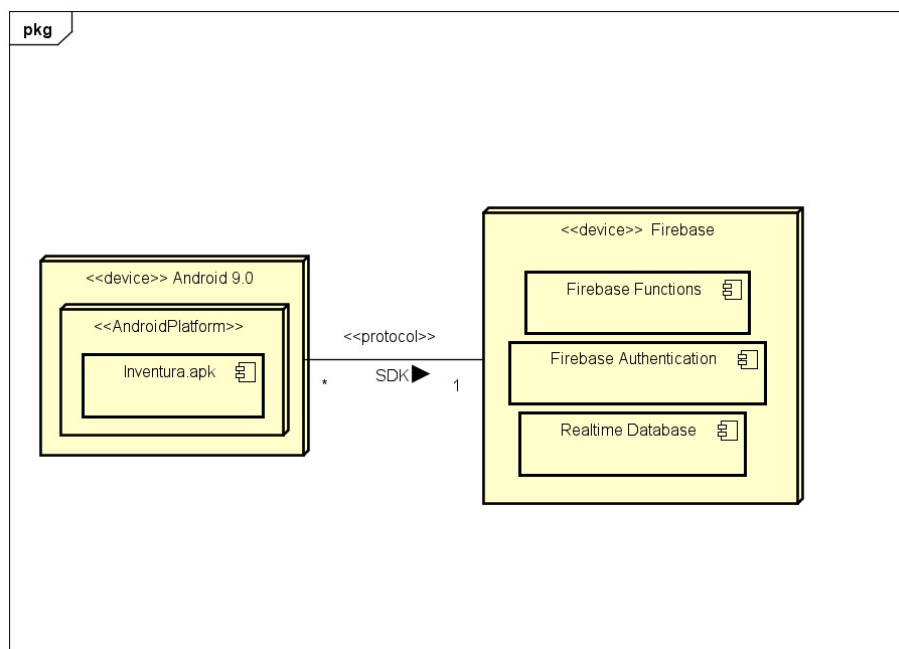
```
testWidgets('Upisivanje podataka za prijavu', (WidgetTester tester) async {  
  app.main();  
  await tester.pumpAndSettle();  
  final Finder inputEmail = find.byKey(Key('email'));  
  final Finder inputPass = find.byKey(Key('inputPass'));  
  await tester.enterText(inputEmail, 'fb@gmail.com');  
  await tester.enterText(inputPass, '123456');  
  expect(find.text('fb@gmail.com'), findsOneWidget);  
  expect(find.text('123456'), findsOneWidget);  
});
```

Upisivanje podataka za registraciju:

```
testWidgets('Upisivanje podataka za registraciju', (WidgetTester tester) async {  
  app.main();  
  await tester.pumpAndSettle();  
  final Finder inputName = find.byKey(Key('name'));  
  final Finder inputLastName = find.byKey(Key('lastName'));  
  final Finder inputEmail = find.byKey(Key('email'));  
  final Finder inputPass = find.byKey(Key('inputPass'));  
  await tester.enterText(inputName, 'Filip');  
  await tester.enterText(inputLastName, 'Begović');  
  await tester.enterText(inputEmail, 'fb@gmail.com');  
  await tester.enterText(inputPass, '123456');  
  expect(find.text('Filip'), findsOneWidget);  
  expect(find.text('Begović'), findsOneWidget);  
  expect(find.text('fb@gmail.com'), findsOneWidget);  
  expect(find.text('123456'), findsOneWidget);  
});
```

5.3 Dijagram razmještaja

Dijagram razmještaja opisuje topologiju sustava i usredotočen je na odnos sklopovskih i programskih dijelova. Na *Firebase* platformi nalaze se tri glavna artefakta kojima upravljamo bazom podataka. Klijent preko aplikacije pristupa bazi podataka. Sustav je baziran na obliku "poslužiteljske arhitekture". Komunikacija između mobilnog uređaja (skladištar, voditelj skladišta i direktora) i *Firebase* odvija se preko SDK veze.



Slika 5.4: Specifikacijski dijagram razmještaja

5.4 Upute za puštanje u pogon

Upute za puštanje u pogon napisane su za windows operacijski sustav. Za pokretanje aplikacije na vlastitom mobitelu potrebno je prvo napraviti *google profil*, te preuzeti aplikaciju s google diska (**Ovdje dolazi link aplikacije**). Kada se cijela aplikacija preuzela u željenu mapu na računalu potrebno je instalirati Flutter.

Prvo unesite link u vaš preglednik: <https://git-scm.com/download/win> te odaberite verziju koja odgovara vašem računalu (32-bit ili 64-bit). Nakon preuzimanja, pokrenite preuzeti *.exe file* te pratite upute do kraja instalacije.

Nakon toga unesite <https://docs.flutter.dev/get-started/install/windows> u vaš preglednik te provjerite je li vaše računalo zadovoljava potrebne specifikacije. Ispod podnaslova *Get the Flutter SDK* pritisnite plavi gumb: *flutter_windows_2.8.1-stable.zip*. Nakon preuzimanja *.zip* mape te izvucite datoteke u mapu *C:/src/*.

Potom je potrebno dodati PATH kako bi smo lakše koristili flutter kasnije. Pratite mape gdje ste instalirali flutter sve dok ne dođete do bit mape (najvjerojarnije putanja: *C:/src/FlutterSDK/flutter/bin*) te kopirajte tu putanju pošto će nam trebati poslije. U Windows tražilicu potrebno je upisati *Control panel* te odabrati *System*. S lijeve strane nalaziti će se popis opcija gdje je potrebno odabrati *Additional settings*. Odaberite *Edit enviroment variables for your account* te će vam se otvoriti prozor podijeljen na dva dijela. U gornjem prozoru nađite varijablu imenom *Path*, pritisnite na nju te ispod pritisnite *Edit*. S desne strane pritisnite *New* te zalijepite kopiranu putanju do mape *bin*. Potom u odjeljku ispod nađite ponovno varijablu *Path* te ponovite isti postupak kao i prethodno naveden. Pritisnite OK te izađite van.

Provjerom uspješnosti instalacije *fluttera* istovremeno ćete saznati je li određene potrebne datoteke nedostaju. U CMD-u napišite naredbu *flutter doctor*. Povratna informacija će izbaciti kako nedostaje Android SDK, koji nam je sada nepotreban.

Otvorite VS Code te u opciji *Open Folder* otvorite mapu u kojoj se nalazi aplikacija. Odaberite opciju *Extensions* te preuzmite i u VS Code *Flutter* i *Dart*.

Nakon što je sve potrebno instalirano i aplikacija otvorena u VS Code, u *Terminal* se može upisati naredba: *flutter build apk --build-name=1.0.1 --build-number=2* koja pokreće *build* aplikacije. Nakon što se proces završi u početnoj mapi projekta u *File Explorer* pratite putanju: *build/app/outputs/apk* te će datoteka pod nazivom *apk-release.apk* biti završna i primjenjiva aplikacija projekta. Za kraj je potrebno prebaciti apk datoteku na mobilni uređaj te pokrenuti instalaciju.

6. Zaključak i budući rad

Zadani zadatak naše grupe bio je razvoj rješenja problema vođenja inventure unutar više skladišta jedne kompanije, statistike greške skladištara koji vode inventuru i pomoću lokacije praćenje zaposlenika te kojim skladištima pripadaju. U razdoblju ovog semestra smo uspješno razvili aplikaciju te ostvarili zadani cilj. Provedba samog projekta odvijala se u dvije faze te smo unutar tima bili podijeljeni u dvije manje grupe zadužene za *frontend* i *backend* (podjela nije bila isključujuća te su članovi tima si međusobno kontinuirano pomagali radi bolje izvedbe i lakše komunikacije).

Prva faza razvoja aplikacije bila je okupljanje tima, dodjelu projektnog zadatka te praćenjem i dokumentiranjem zahtjeva određen obujam posla te inicijalna raspodjela poslova. Praćenjem daljnjih uputa te izradom obrazaca uporabe i dijagrama (sekvencijski dijagrami, dijagram razreda, model baze podataka (teže napravljen jer je baza podataka u Firebase-u, ali dovoljno da lakše shvatimo što je potrebno napraviti)) olakšana je raspodjela poslova među podtimovima i komunikacija među članovima jer smo se lakše mogli snaći u poslu koji je obavio drugi član. Izrada vizualnog dizajna aplikacije dodatno je olakšala razvoj u drugom ciklusu jer su podtimovi *frontend* i *backend* mogli brže raditi.

Druga faza razvoja, programiranje zadane aplikacije, bila je puno zahtjevnija individualno zbog manjka iskustva u programiranju u zadanom razvojnom okruženju. Prije samog programiranja svaki član samostalno je proučio potrebne alate kako bi otprilike znali kako i izvedivost zahtjeva. Podjela poslova, iako u početku raspodijeljena individualno, većinom je tražila rad više članova koji su si međusobno pomagali i zajedno nalazili potrebna rješenja. Uz sam razvoj aplikacije bilo je potrebno daljnje praćenje UML dijagrama i održavanja projektne dokumentacije kako bi budući korisnici, a i članovi tima lakše snalazili u određenim dijelovima projekta. Kontinuirano praćenje i razvoja kostura aplikacije, prilagodba već napravljenih rješenja problema i intenzivna komunikacija članova omogućila nam je razvoj s minimalnim brojem grešaka i uštedu vremena u daljnjem programiranju.

Za komunikaciju među članovima korištene su primarno dvije platforme: Whatsapp i Discord. Postojeće rješenje moguće je nastaviti razvijati i dalje koristeći

mobilnu aplikaciju te čak i proširiti funkcionalnost dodatnim razvojem web aplikacije.

Sudjelovanje na ovom projektu uvelike je koristilo svim članovima u razvoju vještina komuniciranja u većoj grupi, praćenje i pomaganje u radu više ljudi i učenjem u novim razvojnim okruženjima. Zadovoljni smo postignutim rješenjem koje je moguće nastaviti razvijati te primijeniti u stvarnom svijetu.

Popis literature

Kontinuirano osvježavanje

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

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

Indeks slika i dijagrama

3.1	Dijagram obrasca uporabe, zajedničke funkcionalnost svih korisnika	22
3.2	Dijagram obrasca uporabe, zasebne funkcionalnost skladištara, šefa skladišta i direktora	23
3.3	Sekvencijski dijagram za UC4	24
3.4	Sekvencijski dijagram za UC16	25
3.5	Sekvencijski dijagram za završetak trenutne inventure	26
4.1	ER dijagram baze podataka	35
4.2	Dijagram razreda - razredi koji nasljeđuju razred Person	36
4.3	Dijagram razreda - odnosi među klasama	37
4.4	Dijagram stanja - paljenje kamere i skeniranje artikala	38
4.5	Dijagram aktivnosti - skeniranje artikla i komunikacija s bazom podataka	39
4.6	Dijagram komponenti - struktura aplikacije	41
5.1	Screenshot unit test koda (1)	43
5.2	Screenshot unit test koda (2)	44
5.3	Rezultati unit testiranja	44
5.4	Specifikacijski dijagram razmještaja	47
6.1	Dijagram pregleda promjena 1	59
6.2	Dijagram pregleda promjena 2	60
6.3	Dijagram pregleda promjena 3	61
6.4	Dijagram pregleda promjena 4	62
6.5	Dijagram pregleda promjena 5	63

Dodatak: Prikaz aktivnosti grupe

Dnevnik sastajanja

Kontinuirano osvježavanje

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

1. sastanak

- Datum: 6. listopada 2021.
- Prisustvovali: K. Franić, E. Wachtler, T. Jukić, N. Anić, M. Bagarić, F. Begović
- Teme sastanka:
 - Prvi sastanak - razrada opće ideje za aplikaciju
 - Dogovor oko mogućih tehnologija za razvoj aplikacije, razmjena iskustava i opće upoznavanje

2. sastanak

- Datum: 13. listopada 2021.
- Prisustvovali: K. Franić, E. Wachtler, T. Jukić, N. Anić, M. Bagarić, F. Begović, B. Rebić-Taučer
- Teme sastanka:
 - Prvi sastanak s asistentom - opće upute

3. sastanak

- Datum: 20. listopada 2021.
- Prisustvovali: K. Franić, E. Wachtler, T. Jukić, N. Anić, M. Bagarić, F. Begović, B. Rebić-Taučer
- Teme sastanka:
 - Pokušaj instaliranja i pokretanja programa za razvoj aplikacije
 - Kratko upoznavanje s Flutterom, članovi s iskustvom (Franić, Begović) objasnili su glavni koncept ostatku tima
 - FrontEnd tim predstavio je ideju za dizajn screenova

4. sastanak

- Datum: 23. listopada 2021.
- Prisustvovali: E. Wachtler, T. Jukić, B. Rebić-Taučer
- Teme sastanka:
 - Razvijanje osnovne ideje za BackEnd dio aplikacije

5. sastanak

- Datum: 23. listopada 2021.
- Prisustvovali: F. Begović, M. Bagarić, N. Anić
- Teme sastanka:
 - Dogovor oko izgleda i rasporeda elemenata na ekranima

6. sastanak

- Datum: 28. listopada 2021.
- Prisustvovali: M. Bagarić, F. Begović
- Teme sastanka:
 - Razgovor s asistentom i rješavanje pitanja u vezi zadataka aplikacije
 - Konzultacije oko funkcionalnih zahtjeva

7. sastanak

- Datum: 29. listopada 2021.
- Prisustvovali: K. Franić, E. Wachtler, T. Jukić, N. Anić, M. Bagarić, F. Begović, B. Rebić-Taučer
- Teme sastanka:
 - Prezentacija dosad napravljenog
 - Raspravljanje o pitanjima koja su se pojavila prilikom razvoja

8. sastanak

- Datum: 04. studenoga 2021.
- Prisustvovali: K. Franić, E. Wachtler, T. Jukić, N. Anić, M. Bagarić, F. Begović, B. Rebić-Taučer
- Teme sastanka:
 - Prezentacija dosad napravljenoga
 - Raspravljanje o pitanjima koja su se pojavila prilikom razvoja

9. sastanak

- Datum: 07. studenoga 2021.
- Prisustvovali: K. Franić, N. Anić
- Teme sastanka:
 - Ispravci pogrešaka za signup screen
 - Poboljšanje strukture koda

10. sastanak

- Datum: 10. studenoga 2021.
- Prisustvovali: K. Franić, E. Wachtler, T. Jukić, B. Rebić-Taučer, N. Anić, F. Begović
- Teme sastanka:
 - Rješavanje pitanja s asistentom koja su se pojavila prilikom razvoja aplikacije

11. sastanak

- Datum: 12. studenoga 2021.
- Prisustvovali: K. Franić, E. Wachtler, T. Jukić, B. Rebić-Taučer
- Teme sastanka:
 - Rješavanje pitanja u vezi komunikacije s bazom podataka

12. sastanak

- Datum: 13. studenoga 2021.
- Prisustvovali: K. Franić, E. Wachtler
- Teme sastanka:
 - Ispravljanje pogreške koja se javljala na nekim uređajima

13. sastanak

- Datum: 14. studenoga 2021.
- Prisustvovali: K. Franić, E. Wachtler, T. Jukić, N. Anić, M. Bagarić, F. Begović, B. Rebić-Taučer
- Teme sastanka:
 - Presentacija dosad napravljenoga
 - Raspravljanje o pitanjima koja su se pojavila prilikom razvoja
 - Razrada ideje oko skeniranja artikala

14. sastanak

- Datum: 18. studenoga 2021.
- Prisustvovali: K. Franić, E. Wachtler, T. Jukić, N. Anić, M. Bagarić, F. Begović, B. Rebić-Taučer
- Teme sastanka:
 - Presentacija dosad napravljenoga
 - Rješavanje pogrešaka za prvu predaju
 - Presentacija dosad napravljenog asistentu

15. sastanak

- Datum: 18. prosinca 2021.

- Prisustvovali: K. Franić, E. Wachtler, T. Jukić, N. Anić, M. Bagarić, F. Begović, B. Rebić-Taučer
- Teme sastanka:
 - Izrada aplikacije za prvu demonstraciju
 - Dodavanje potrebnih funkcionalnosti za komunikaciju s bazom podataka

16. sastanak

- Datum: 19. prosinca 2021.
- Prisustvovali: K. Franić, E. Wachtler, T. Jukić, N. Anić, M. Bagarić, F. Begović, B. Rebić-Taučer
- Teme sastanka:
 - Izrada aplikacije za prvu demonstraciju
 - Implementacija dohvaćanja lokacije
 - Implementacija skeniranja artikala

17. sastanak

- Datum: 9. siječnja 2022.
- Prisustvovali: K. Franić, E. Wachtler, T. Jukić, F. Begović
- Teme sastanka:
 - Izrada statistike za skladištara
 - Popravak rada aplikacije prilikom skeniranja
 - Popravak dohvata lokacije

18. sastanak

- Datum: 11. siječnja 2022.
- Prisustvovali: K. Franić, E. Wachtler, T. Jukić, F. Begović
- Teme sastanka:
 - Popravak statistike za skladištara
 - Ispravci pogrešaka u radu aplikacije
 - Raspodjela daljnjih poslova

Tablica aktivnosti

Kontinuirano osvježavanje

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

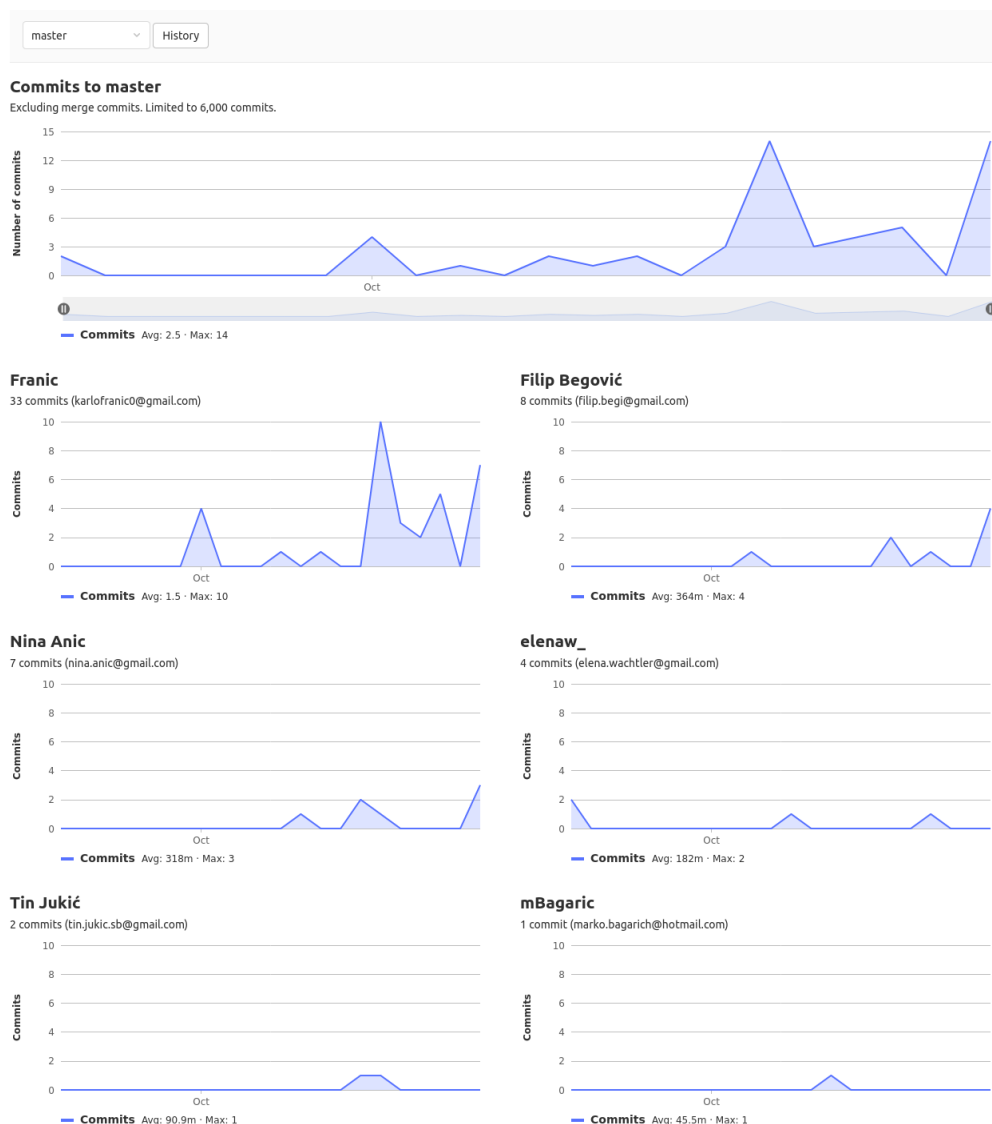
	Elena Wachtler	Karlo Franić	Filip Begović	Nina Anić	Marko Bagarić	Borna Rebić-Taučer	Tin Jukić	
Upravljanje projektom	20	10	3					
Opis projektnog zadatka	1		12	6	6	3		
Funkcionalni zahtjevi		2	5	13	13	7		
Opis pojedinih obrazaca	2		8	15	3			
Dijagram obrazaca			6			2		
Sekvencijski dijagrami			5				3	
Opis ostalih zahtjeva			1	5	1			
Arhitektura i dizajn sustava	8	8	9	8	8	5	5	
Baza podataka	10	10				5	5	
Dijagram razreda							10	
Dijagram stanja							2	
Dijagram aktivnosti							3	
Dijagram komponenti							2	
Korištene tehnologije i alati	4	4	3	3	5	5	5	
Ispitivanje programskog rješenja	2	2	25	15	27	10	7	
Dijagram razmještaja						2		

Nastavljeno na idućoj stranici

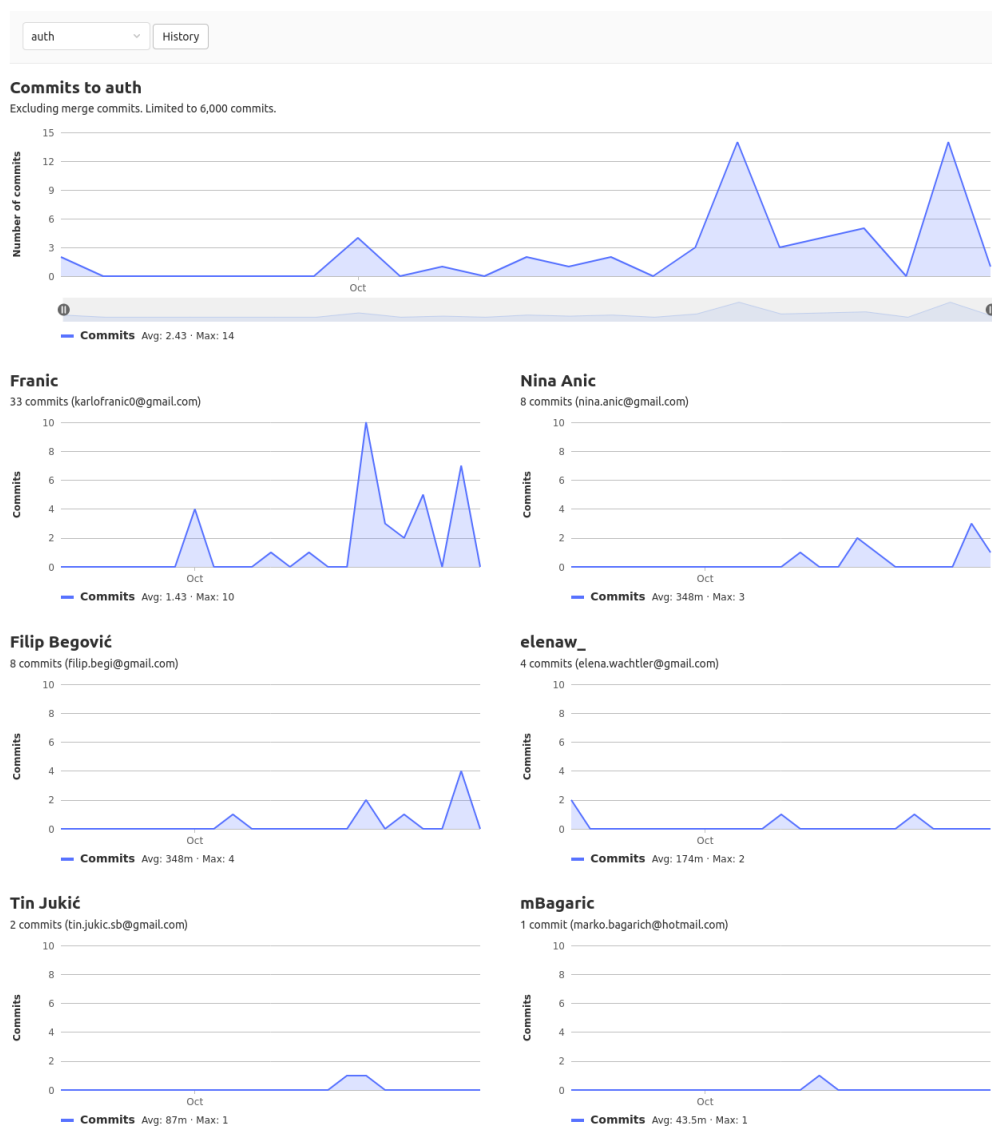
Nastavljeno od prethodne stranice

	Elena Wachtler	Karlo Franić	Filip Begović	Nina Anić	Marko Bagarić	Borna Rebić-Taučer	Tin Jukić	
Upute za puštanje u pogon					5	2		
Dnevnik sastajanja							7	
Zaključak i budući rad					6			
Popis literature						2		
<i>front end</i>		12	47	45	25	5		
<i>izrada baze podataka</i>	5	19				5	5	
<i>spajanje s bazom podataka</i>	20	10	7			5	5	
<i>back end</i>	49	55				40	40	
ukupno sati	121	132	131	110	99	98	99	

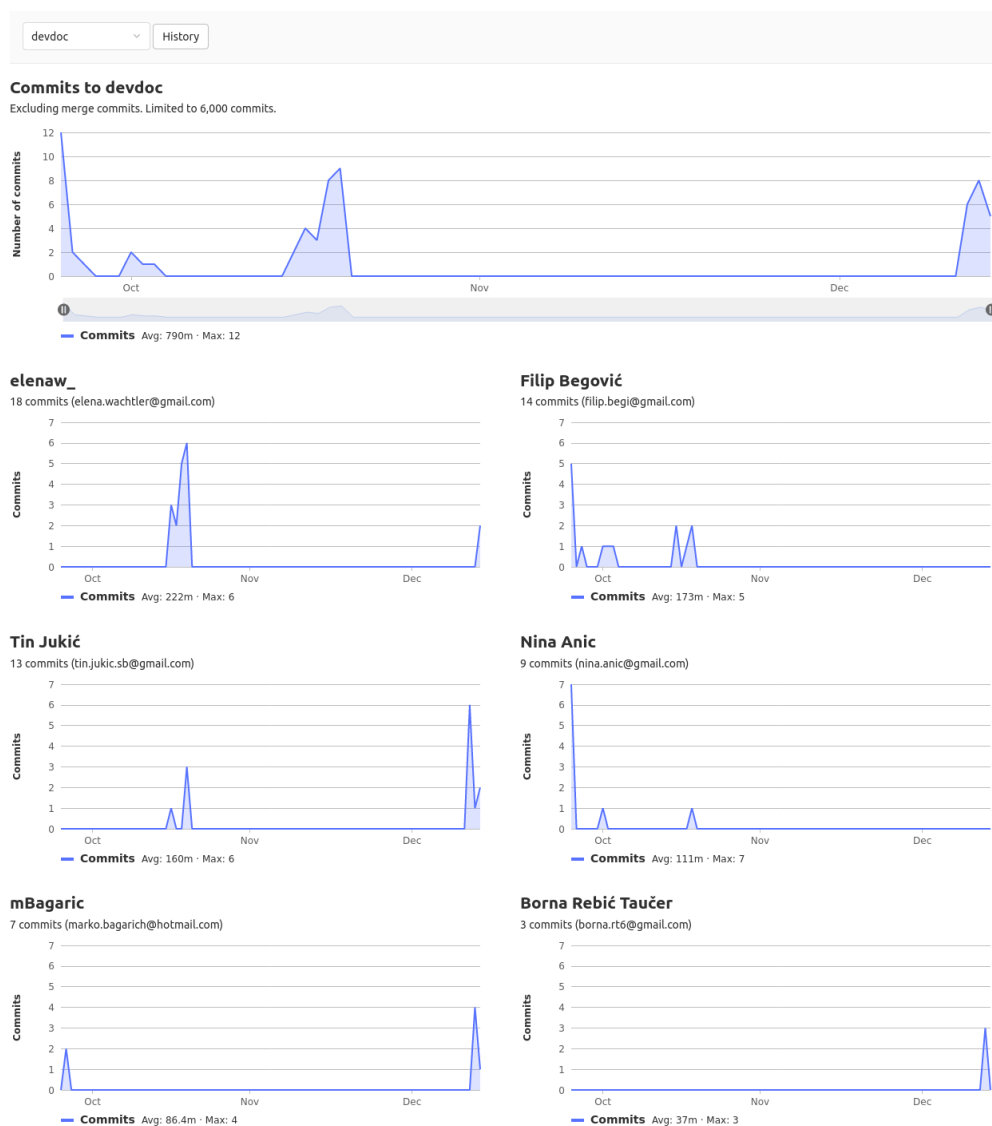
Dijagrami pregleda promjena



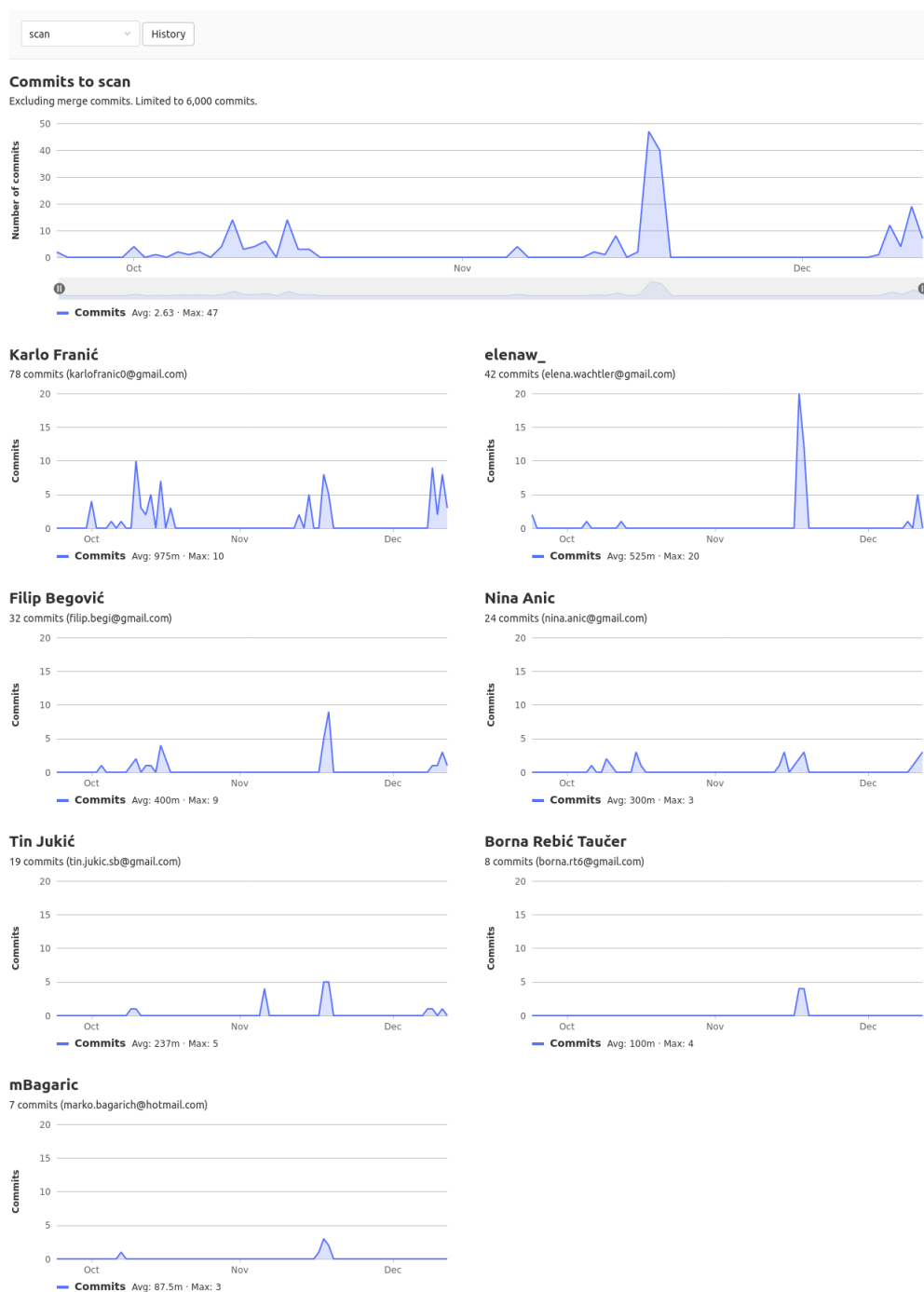
Slika 6.1: Dijagram pregleda promjena 1



Slika 6.2: Dijagram pregleda promjena 2



Slika 6.3: Dijagram pregleda promjena 3



Slika 6.4: Dijagram pregleda promjena 4



Slika 6.5: Dijagram pregleda promjena 5