

Opis projekta

Ovaj projekt je rezultat timskog rada u sklopu projektlog zadatka kolegija [Programsko inženjerstvo](#) na Fakultetu elektrotehnike i računarstva Sveučilišta u Zagrebu.

Ova aplikacija omogućuje korisnicima upravljanje stvarima u svojim ormarima, dijeljenje odjevnih predmeta s drugim korisnicima u blizini te pregled ponuda oglašivača. Korisnici koristeći mapu unutar aplikacije mogu vidjeti javne ormare drugih korisnika u stvarnom vremenu. Aplikacija je istovremeno korisna za organizaciju i za razmjenu stvari s drugim korisnicima. Projekt obuhvaća funkcionalne i nefunkcionalne zahtjeve koji omogućuju jednostavno korištenje, skalabilnost i proširivost aplikacije.

Aplikacija podržava tri rad tipa korisnika. Prvi tip je neregistrirani korisnik, kojemu je omogućeno da pri ulasku u aplikaciju može pregledavati i pretraživati artikle koji su označeni za dijeljenje u ormarima drugih korisnika, te artikle koji su oglašeni. Pretraživanje je omogućeno po svim karakteristikama artikla. Osim toga omogućeno je i pretraživanje artikla po lokaciji u odnosu na trenutnu geolokaciju. Odabirom nekog od artikala prikazuju se sve informacije o tom artiklu, uključujući i kontakt informacije korisnika koji je podijelio ili oglasio taj artikl. Drugi tip korisnika je registrirani korisnik koji ima sve mogućnosti neregistriranog korisnika. Registrirani korisnik ima mogućnost stvaranja svojih ormara, to uključuje definiranje strukture ormara (broj polica, ladica i šipki za odjeću). Također mu je omogućeno kasnije mijenjanje te strukture, brisanje ormara, stvaranje artikala, brisanje artikala. Svaki artikl ima sljedeće karakteristike: naziv, sliku, opću kategoriju, kategoriju godišnjeg doba nošenja, kategoriju otvorenosti (samo za obuću), kategoriju ležernosti, glavnu i sporednu boju artikla, opis stanja artikla te lokaciju u virtualnom ormaru. Omogućeno mu je i pretraživanje vlastitih ormara po svim karakteristikama artikala, rezultat ovog pretraživanja je lokacija artikla unutar njegovih ormara (npr. Ormar 3, Ladica 4). Također može i podijeliti svoje artikle s drugim korisnicima, time omogućuje drugim korisnicima pregled njegovih dijeljenih artikala. Dijeljenjem svojih artikala informacije o tom artiklu te korisniku postaju javno dostupne. Još jedna od mogućnosti registriranog korisnika je prijedlog odjeвне kombinacije na temelju unesenih kriterija. Odabirom ove mogućnosti korisnik unese svoje zadane kriterije i dobije pregled svih preporučenih artikala u svojem ormaru te njihove lokacije. Registrirani korisnik u svojem profilu mora minimalno unijeti podatke o svojem imenu i prezimenu ili alijasu, geolokaciji i adresi e-pošte.

Treća vrsta korisnika su oglašivači koji se također moraju registrirati prije uporabe aplikacije. Svaki oglašivač može stvoriti i mijenjati svoju galeriju artikala koje nudi. Oni pri dodavanju novog artikla moraju minimalno unijeti naziv, opću kategoriju, cijenu te sliku artikla. U svakom trenutku su u mogućnosti obrisati ili dodati novi artikl. Njihovi artikli su također vezani s geolokacijom. Oglašivači se profilu registriranih i neregistriranih korisnika oglašuju s klikabilnim logoom koji su sami definirali. Odabirom logoa nekog od oglašivača korisnike se vodi na galeriju artikala koje taj oglašivač ponuđuje, te se prikazuju kontakt informacije tog oglašivača.

Aplikacija treba biti izvedena kao mobilna aplikacija ili kao web aplikacija prilagođena za mobilne uređaje i implementirana u objektnom programskom jeziku. Sustav treba podržavati rad više korisnika u stvarnom vremenu. Administraciju registriranih korisnika može se obaviti kroz sučelje baze podataka.

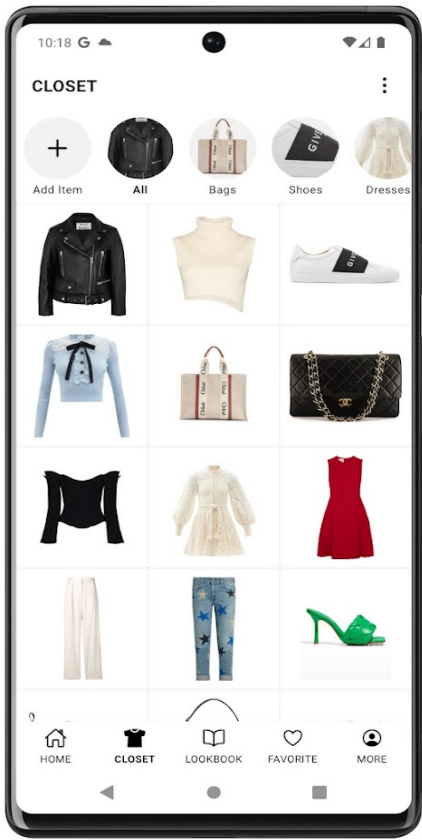
Potencijalna koristi ovog projekta

Korist ovog projekta je aplikacija koja omogućava korisnicima da lako organiziraju vlastiti ormar te pronađu artikle od drugih ljudi. Osim dovršene aplikacije svi članovi ovog tima će dobiti uvid u rješavanje složenog problema, rad u timu, organizaciju posla, podjelu odgovornosti te neka nova znanja.

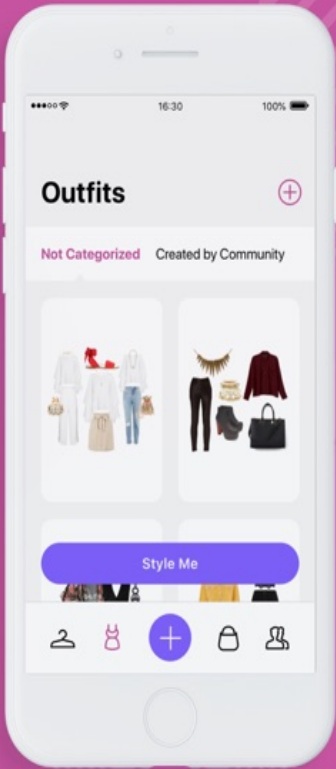
Postojeća slična rješenja

Closet+, Stylebook, Pureple, Smart Closet su neka od postojećih rješenja sličnih problema. Međutim ni jedna od ovih aplikacija ne sadržava sve mogućnosti kao i naše rješenje.

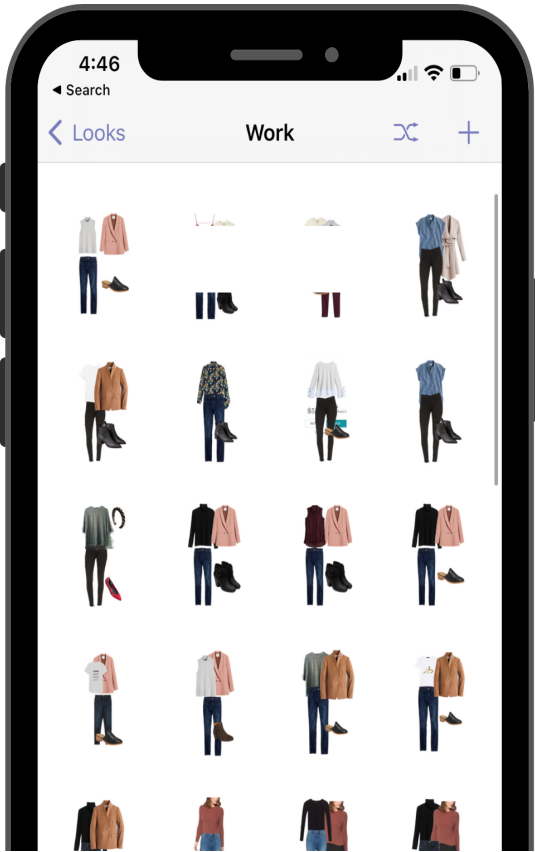
Organize Your Closet

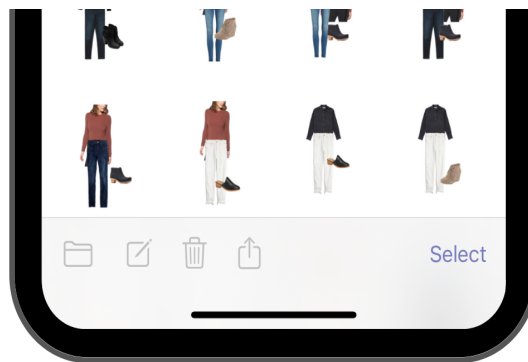
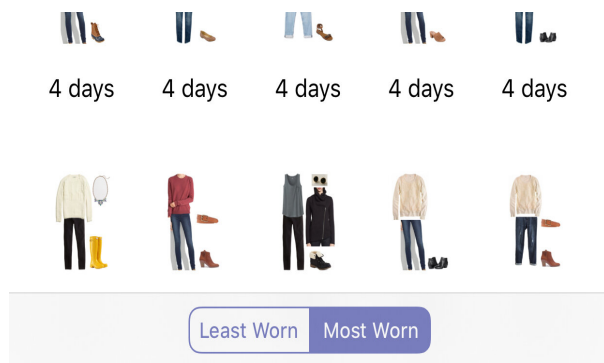


Plan your outfits



Style Stats Worn History Range





Skup korisnika koji bi mogao biti zainteresiran za ostvareno rješenje

Skup korisnika bi bio zainteresiran za stvaranje vlastitog virtualnog ormara i organizaciju u odnosu na stvarni ormar. Time bi si omogućili lagano pretraživanje svojeg stvarnog ormara uz pomoć virtualnog. Drugi skup bi bio zainteresiran u dijeljenje vlastite odjeće, te posuđivanje. Treći skup bi bile tvrtke koje bi svoje artikle oglasili na našoj aplikaciji i tako ostvarili profit. Skup potencijalnih korisnika za ovu aplikaciju je stvarno velik.

Mogućnost prilagodbe rješenja

Potencijalna prilagodba našeg rješenja bi bila u komercijalne svrhe.

Opseg projektnog zadatka

Uspješno korištenje znanja stečenog na ovom fakultetu, te njegovo proširenje i usavršavanje.

Moguće nadogradnje projektnog zadatka

Moguće su mnoge nadogradnje ovog zadatka, poput povezivanja aplikacije s vremenskom prognozom koja bi omogućavala bolji odabir odjevnih kombinacija. Još neko od potencijalnih nadogradnji je praćenje nošenja i održavanja artikala u ormarima. Moguće je i potencijalno povezivanje aplikacije s web trgovinama čime bi olakšali potražnju za nekim artiklima koji nam fale u ormaru.

Funkcionalni zahtjevi

ID zahtjeva	Opis	Prioritet	Izvor	Kriteriji prihvatanja
F-001	Neregistrirani korisnik može pregledavati i pretraživati artikle za dijeljenje.	Visok	Zahtjev dionika	Pretraživanje prema svim karakteristikama artikla i lokaciji; detaljan pregled artikla i kontakt podaci.
F-002	Registrirani korisnik može dodavati, uređivati i brisati vlastite virtualne ormare.	Visok	Zahtjev dionika	Omogućeno dodavanje više ormara, definiranje strukture (polica, ladica, šipka), brisanje ormara i sadržaja.

ID zahtjeva	Opis	Prioritet	Izvor	Kriteriji prihvatanja
F-003	Registrirani korisnik može dodati artikle u svoj ormar s pripadajućim karakteristikama.	Visok	Dokument zahtjeva	Artikli imaju naziv, sliku, kategorije (sezona, otvorenost, ležernost), boje, opis stanja, i lokaciju u ormaru.
F-004	Registrirani korisnik može dijeliti artikle s drugim korisnicima.	Srednji	Dokument zahtjeva	Podijeljeni artikli postaju javno vidljivi s korisničkim podacima (ime, lokacija, e-pošta).
F-005	Sustav omogućuje pretraživanje virtualnih ormara prema karakteristikama artikla.	Visok	Postojeći sustav	Rezultati pretraživanja prikazuju ormar, redni broj i vrstu lokacije (npr. 2. ormar, 3. lokacija – ladica).
F-006	Sustav predlaže odjevne kombinacije temeljem kriterija korisnika.	Srednji	Dokument zahtjeva	Generira popis preporučenih artikala i njihovih lokacija na temelju zadanih kriterija.
F-007	Neregistrirani korisnici se mogu registrirati kako bi dobili pristup dodatnim funkcijama.	Visok	Zahtjev dionika	Uspješna registracija s unosom minimalnih podataka (ime, geolokacija, e-mail).
F-008	Oglašivači mogu definirati galeriju artikala i oglašavati putem klikabilnog logotipa.	Nizak	Postojeći sustav	Galerija sadrži artikle, logotip vodi na galeriju i kontakt-informacije.
F-009	Sustav podržava prilagođavanje strukture virtualnog ormara.	Srednji	Zahtjev dionika	Korisnik može definirati broj i vrstu lokacija (polica, ladica, šipka).
F-010	Neregistrirani korisnik može dobiti osnovne informacije o aplikaciji bez registracije.	Nizak	Zahtjev dionika	Informacije uključuju opis aplikacije i njen osnovni cilj.
F-011	Aplikacija podržava rad više korisnika u stvarnom vremenu.	Visok	Dokument zahtjeva	Omogućeno simultano dodavanje, uređivanje i pretraživanje ormara od strane različitih korisnika.
F-012	Registrirani korisnik može uređivati svoj profil.	Srednji	Zahtjev dionika	Korisnik može mijenjati ime, geolokaciju i kontaktne podatke.
F-013	Administracija registriranih korisnika kroz bazu podataka.	Nizak	Postojeći sustav	Administrator može pregledavati i upravljati korisničkim računima izravno iz baze podataka.

ID zahtjeva	Opis	Prioritet	Izvor	Kriteriji prihvatanja
F-014	Oglašivači mogu dodavati i brisati artikle iz svoje galerije	Nizak	Postojeći sustav	Artikli sadrže sliku, naziv, kategoriju i cijenu

Ostali zahtjevi

Zahtjevi za održavanje

ID zahtjeva	Opis	Prioritet
NF-3.1	Sustav treba biti oblikovan tako da omogućuje jednostavno održavanje	Visok
NF-3.2	Sustav treba imati dovoljnu dokumentaciju	Visok
NF-3.3	Sustav treba omogućiti veći broj istovremenih korisnika	Visok
NF-3.4	Vrijeme odaziva sustava ne smije premašiti 2 sekunde u normalnim uvjetima	Srednji
NF-3.5	Sustav mora podržavati mobilnu uporabu	Visok
NF-3.6	Sustav mora osigurati sigurnost podataka korisnika	Visok
NF-3.7	Sustav treba biti dostupan 99.9% vremena u mjesecu	Visok
NF-3.7	Sustav mora biti jednostavan za korištenje, čak i za tehnički manje vješte korisnike	Srednji
NF-3.7	Aplikacija mora imati moderan i privlačan dizajn	Nizak

Dionici

1. **Korisnici sustava:**
 - Neregistrirani korisnici
 - Registrirani korisnici
 - Oglašivači
2. **Naručitelji projekta:** Određuju zahtjeve i smjer razvoja.
3. **Razvojni tim:** Implementira i održava sustav.

Aktori i njihovi funkcionalni zahtjevi

A-1: Neregistrirani korisnik (inicijator)

Neregistrirani korisnik može:

- **Funkcionalnost 1:** Pregledavati artikle za dijeljenje (F-001).
- **Funkcionalnost 2:** Pretraživati artikle prema karakteristikama i lokaciji (F-001).

- **Funkcionalnost 3:** Pregledavati detalje artikla i kontakt podatke vlasnika (F-001).
- **Funkcionalnost 4:** Registrirati se u sustav (F-007).
- **Funkcionalnost 5:** Saznati osnovne informacije o aplikaciji (F-010).

A-2: Registrirani korisnik (inicijator)

Registrirani korisnik može:

- **Funkcionalnost 1:** Dodavati i uređivati virtualne ormare (F-002).
- **Funkcionalnost 2:** Dodavati artikle u virtualne ormare s pripadajućim karakteristikama (F-003).
- **Funkcionalnost 3:** Dijeliti artikle s drugim korisnicima (F-004).
- **Funkcionalnost 4:** Pretraživati vlastite ormare prema zadanim kriterijima (F-005).
- **Funkcionalnost 5:** Primati preporuke odjevnih kombinacija temeljem kriterija (F-006).
- **Funkcionalnost 6:** Uređivati profil (ime, e-mail, geolokaciju) (F-012).

A-3: Oglašivač (sudionik)

Oglašivač može:

- **Funkcionalnost 1:** Dodavati artikle u vlastitu galeriju (F-014).
- **Funkcionalnost 2:** Upravlja galerijom (uređivati i brisati artikle) (F-014).
- **Funkcionalnost 3:** Oglašavati putem klikabilnog logotipa (F-008).
- **Funkcionalnost 4:** Stvoriti vlastitu galeriju artikala (F-008).


Obrasci uporabe

Dijagrami obrazaca uporabe

Visokorazinski dijagram obrazaca uporabe cijelog sustava

Generalni obrazac korištenja

Usecase v1.0

 Obrazac korištenja

Opis obrazaca uporabe

Obrazac uporabe UC1

Element	Opis
Naziv	UC1: registracija
Namjena	Registracija novih korisnika
Opis	Novi korisnik unosi podatke kako bi mogao ostati zapamćen u sustavu
Glavni aktor	Korisnik

Element	Opis
Preduvjeti	-
Opis osnovnog tijeka	1. Korisnik pošalje upit za registraciju web aplikaciji
	2. Web aplikacija pošalje upit korisniku o njegovim korisničkim podacima
	3. Korisnik upiše svoje korisničke podatke
	4. Web aplikacija registrira korisnika
Opis mogućih odstupanja	3.a) Ako je korisnik već registriran, aplikacija šalje upozorenje

Obrazac uporabe UC2

Element	Opis
Naziv	UC2: pregled artikala
Namjena	Pregled artikala
Opis	Korisnik pregledava i pretražuje artikle koji su označeni za dijeljenje u ormarima registriranih korisnika
Glavni aktor	Korisnik
Preduvjeti	Postoje artikli koje su registrirani korisnici podijelili
Opis osnovnog tijeka	1. Korisnik pretražuje artikle po nekim karakteristikama artikla
	2. Aplikacija mu prikaže sve dijeljene artikle koji odgovaraju unesenim karakteristikama
	3. Ako korisnik odabere artikl, aplikacija mu prikaže informacije o artiklu te kontakt informacije korisnika koji ga je podijelio
Opis mogućih odstupanja	2.a) Nema dijeljenih artikala koji odgovaraju unesenim karakteristikama pa aplikacija ne prikaže ni jedan artikl

Obrazac uporabe C3:

Element	Opis
Naziv	UC3: Klik na logo oglašivača
Namjena	Oglašavanje oglašivača
Opis	Oglašivači se na profilima korisnika oglašavaju klikabilnim logoom koji vodi na galeriju artikala i daljnjim kontakt-informacijama
Glavni aktor	Korisnik
Preduvjeti	Postoje oglasi oglašivača u aplikaciji

Element	Opis
Opis osnovnog tijeka	1. Korisnik klikne na logo oglašivača
	2. Aplikacija ga odvede na galeriju artikala oglašivača i prikaže mu daljnje kontakt informacije
Opis mogućih odstupanja	2.a) Oglašivač nema nikakvih artikala u svojoj galeriji pa ga o tome obavještava aplikacija

Obrazac uporabe C4:

Element	Opis
Naziv	UC4: Pretraživanje vlastitih ormara
Namjena	Pretraživanje vlastitih virtualnih ormara prema karakteristikama artikla
Opis	Registrirani korisnik može pretraživati svoje virtualne ormare prema unesenim karakteristikama
Glavni aktor	Registrirani korisnik
Preduvjeti	Korisnik mora imati barem jedan virtualni ormar
Opis osnovnog tijeka	1. Korisnik unese karakteristike prema kojima želi pretraživati svoje ormare
	2. Aplikacija mu prikaže artikle iz njegovih ormara koji odgovaraju unesenim karakteristikama
Opis mogućih odstupanja	2.a) Korisnik nema artikle u ormarima koji odgovaraju unesenim karakteristikama pa ga o tome obavještava aplikacija

Obrazac uporabe C5:

Element	Opis
Naziv	UC5: Prijedlog odjevne kombinacije
Namjena	Prijedlog odabira odjevne kombinacije na temelju unesenih kriterija
Opis	Registrirani korisnik može od aplikacije zatražiti prijedlog odjevne kombinacije na temelju unesenih kriterija
Glavni aktor	Registrirani korisnik
Preduvjeti	Korisnik mora imati artikle u svojim ormarima
Opis osnovnog tijeka	1. Korisnik zatraži prijedlog odjevne kombinacije zadavanjem kriterija
	2. Aplikacija mu prikaže popis preporučenih artikala i njihovih lokacija u ormarima

Element	Opis
Opis mogućih odstupanja	2.a) Korisnik nema artikle u ormarima koji odgovaraju unesenim karakteristikama pa ga o tome obavještava aplikacija

Obrazac uporabe C6:

Element	Opis
Naziv	UC6: Dodavanje novog virtualnog ormara
Namjena	Korisnik dodaje novi virtualni ormar na svom profilu
Opis	Registrirani korisnik može dodavati više virtualnih ormara na svom profilu
Glavni aktor	Registrirani korisnik
Preduvjeti	-
Opis osnovnog tijeka	1. Korisnik šalje upit za dodavanje novog virtualnog ormara
	2. Aplikacija ga zatraži da definira strukturu tog novog ormara
	3. Korisnik definira novi ormar
	4. Aplikacija ga sprema na njegovom profilu
Opis mogućih odstupanja	3.a) Korisnik nije dovršio definiranje strukture ormara o čemu ga obavještava aplikacija

Obrazac uporabe C7:

Element	Opis
Naziv	UC7: Upravljanje vlastitim virtualnim ormarima
Namjena	Promjena strukture i artikala u ormaru korisnika
Opis	Registrirani korisnik može definirati i mijenjati strukturu svog ormara te mu dodavati i brisati artikle
Glavni aktor	Registrirani korisnik
Preduvjeti	Korisnik mora imati barem jedan virtualni ormar
Opis osnovnog tijeka	1. Korisnik šalje upit za izmjenu strukture ormara, brisanje ili dodavanje artikala
	2. Ako se radi o dodavanju artikla, aplikacija ga zatraži dodavanje karakteristika tog artikla
	3. Ako se radi o dodavanju artikla, korisnik označi želi li označiti artikl za dijeljenje te doda potrebne karakteristike
	4. Aplikacija spremi promjene u korisničkom virtualnom ormaru

Element	Opis
Opis mogućih odstupanja	3.a) Korisnik nije unio sve potrebne karakteristike za artikl koji želi dodati pa ga aplikacija obavještava o tome

Obrazac uporabe C8:

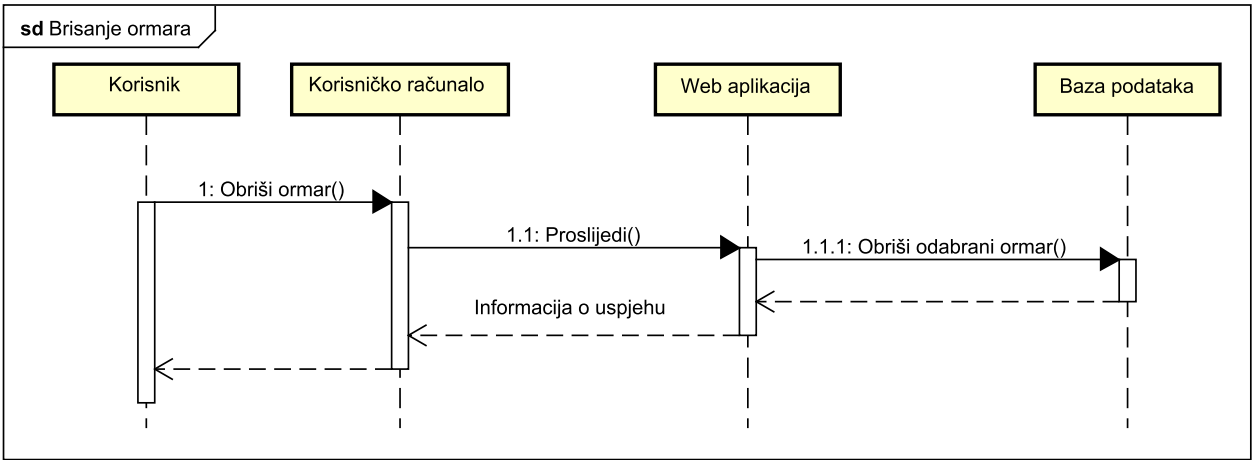
Element	Opis
Naziv	UC8: Stvaranje galerije artikala
Namjena	Oglašivač stvara svoju galeriju artikala
Opis	Oglašivač može na svom profilu definirati galeriju artikala koje nudi
Glavni aktor	Oglašivač
Preduvjeti	-
Opis osnovnog tijeka	1. Oglašivač šalje upit za stvaranje svoje galerije artikala
	2. Aplikacija ga zatraži da definira strukturu te galerije
	3. Oglašivač definira strukturu galerije
	4. Aplikacija ju sprema na njegovom profilu
Opis mogućih odstupanja	3.a) Oglašivač nije dovršio definiranje strukture galerije o čemu ga obavještava aplikacija

Obrazac uporabe C9:

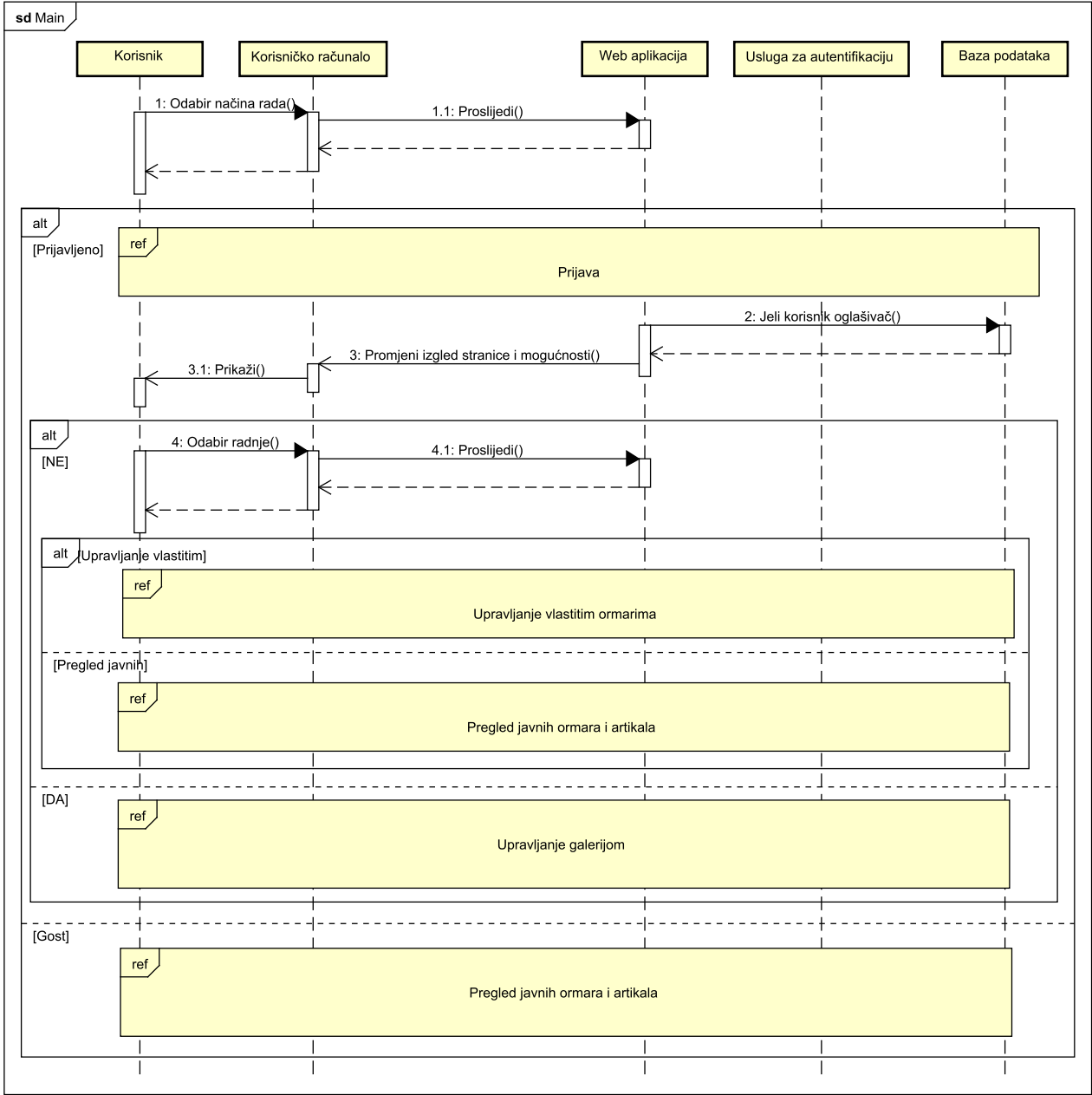
Element	Opis
Naziv	UC9: Upravljanje vlastitom galerijom artikala
Namjena	Promjena strukture i artikala u galeriji oglašivača
Opis	Oglašivač može definirati i mijenjati strukturu svoje galerije te joj dodavati i brisati artikle
Glavni aktor	Oglašivač
Preduvjeti	Oglašivač mora imati svoju galeriju artikala
Opis osnovnog tijeka	1. Oglašivač šalje upit za izmjenu strukture ormara, brisanje ili dodavanje artikala
	2. Ako se radi o dodavanju artikla, aplikacija ga zatraži dodavanje karakteristika tog artikla
	3. Ako se radi o dodavanju artikla, korisnik unese sve potrebne karakteristike
	4. Aplikacija spremi promjene u oglašivačevoj galeriji artikala
Opis mogućih odstupanja	3.a) Oglašivač nije unio sve potrebne karakteristike za artikl koji želi dodati pa ga aplikacija obavještava o tome

Sekvencijski dijagrami

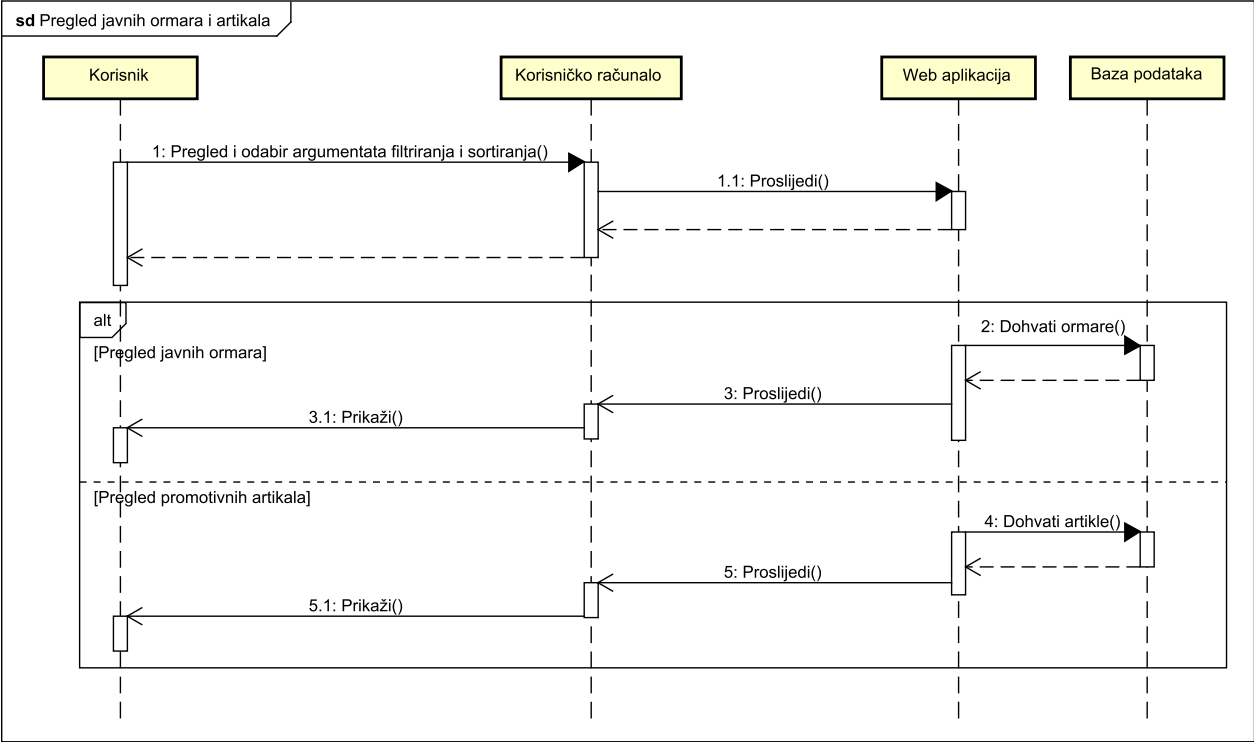
- Brisanje ormara



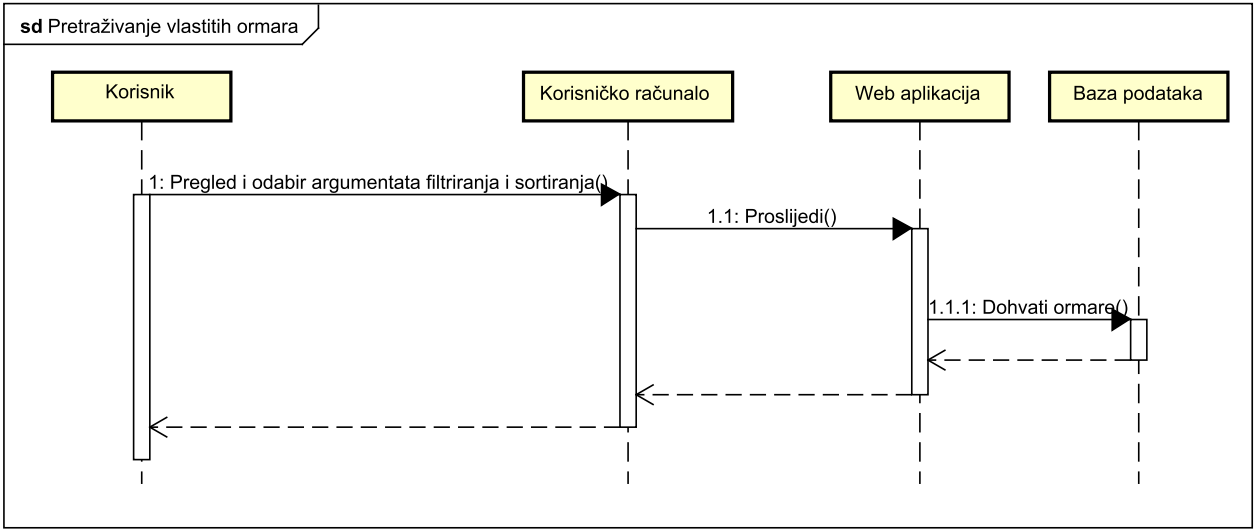
- Main



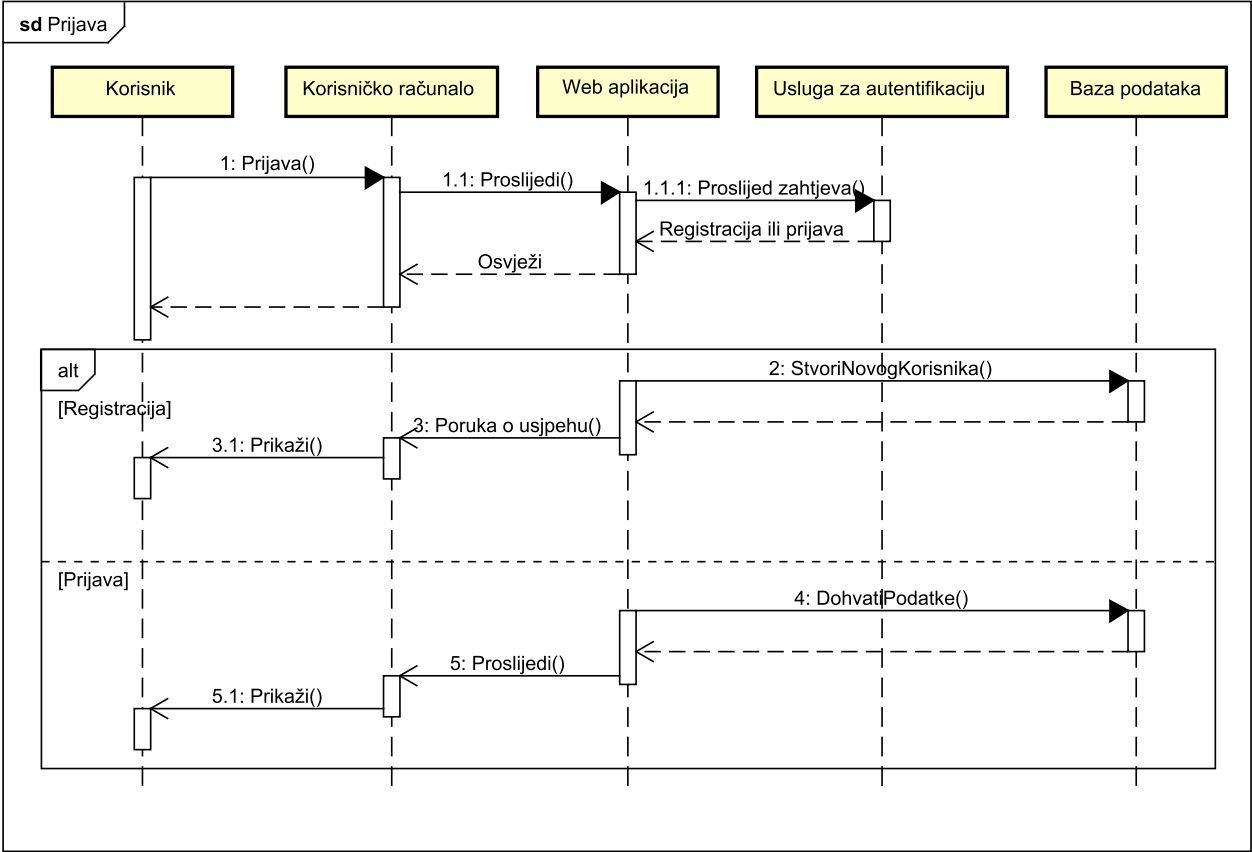
- Pregled javnih ormara i artikala



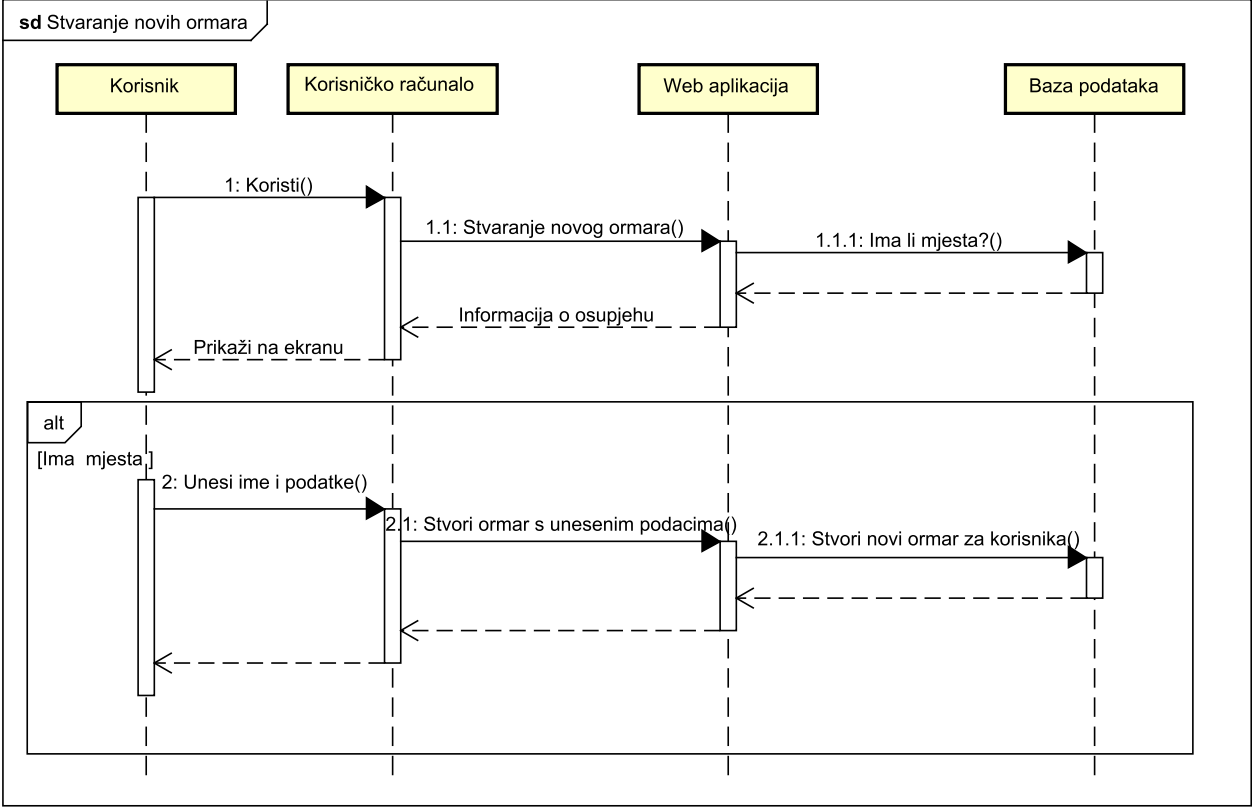
• Pretraživanje vlastitih ormara



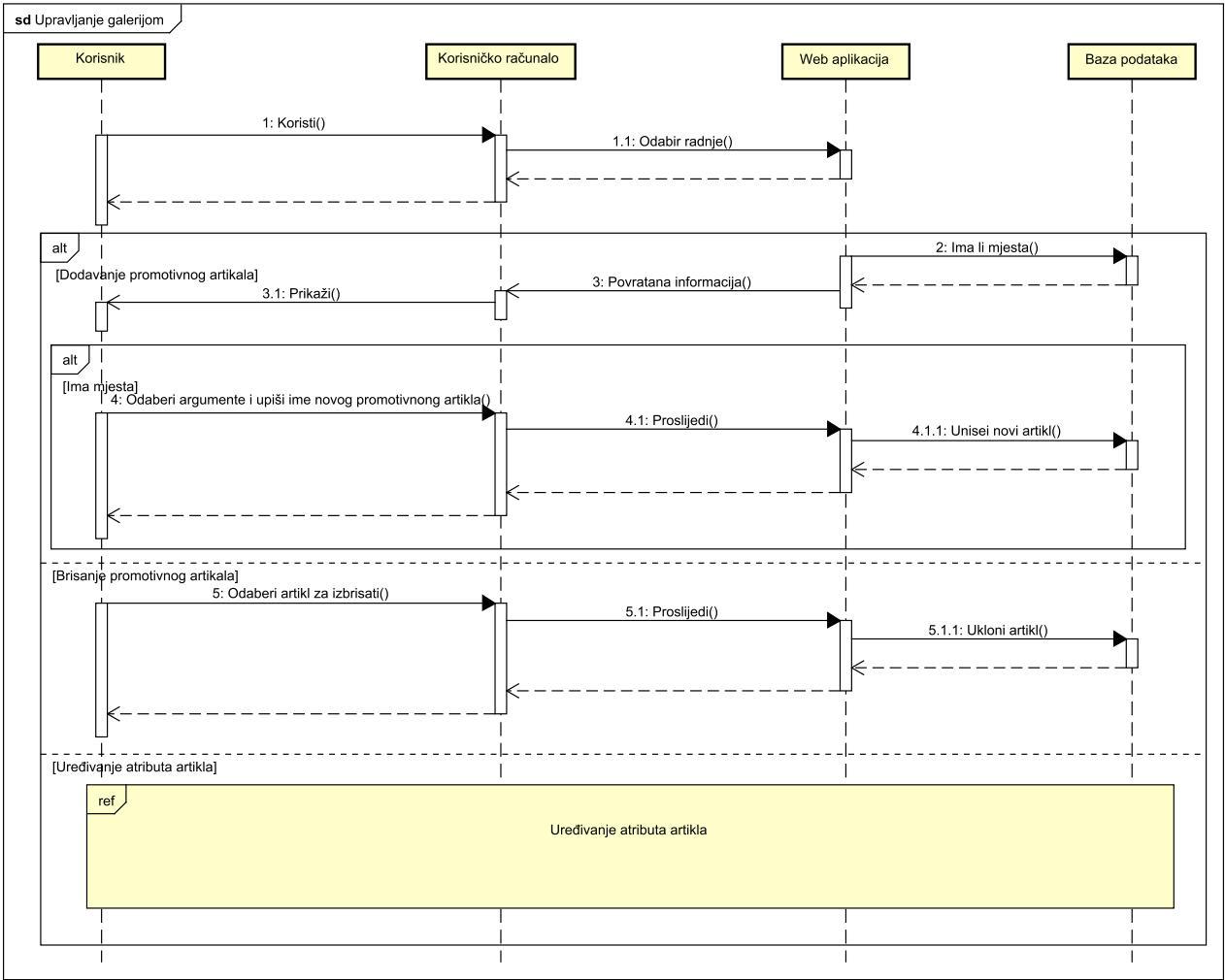
• Prijava



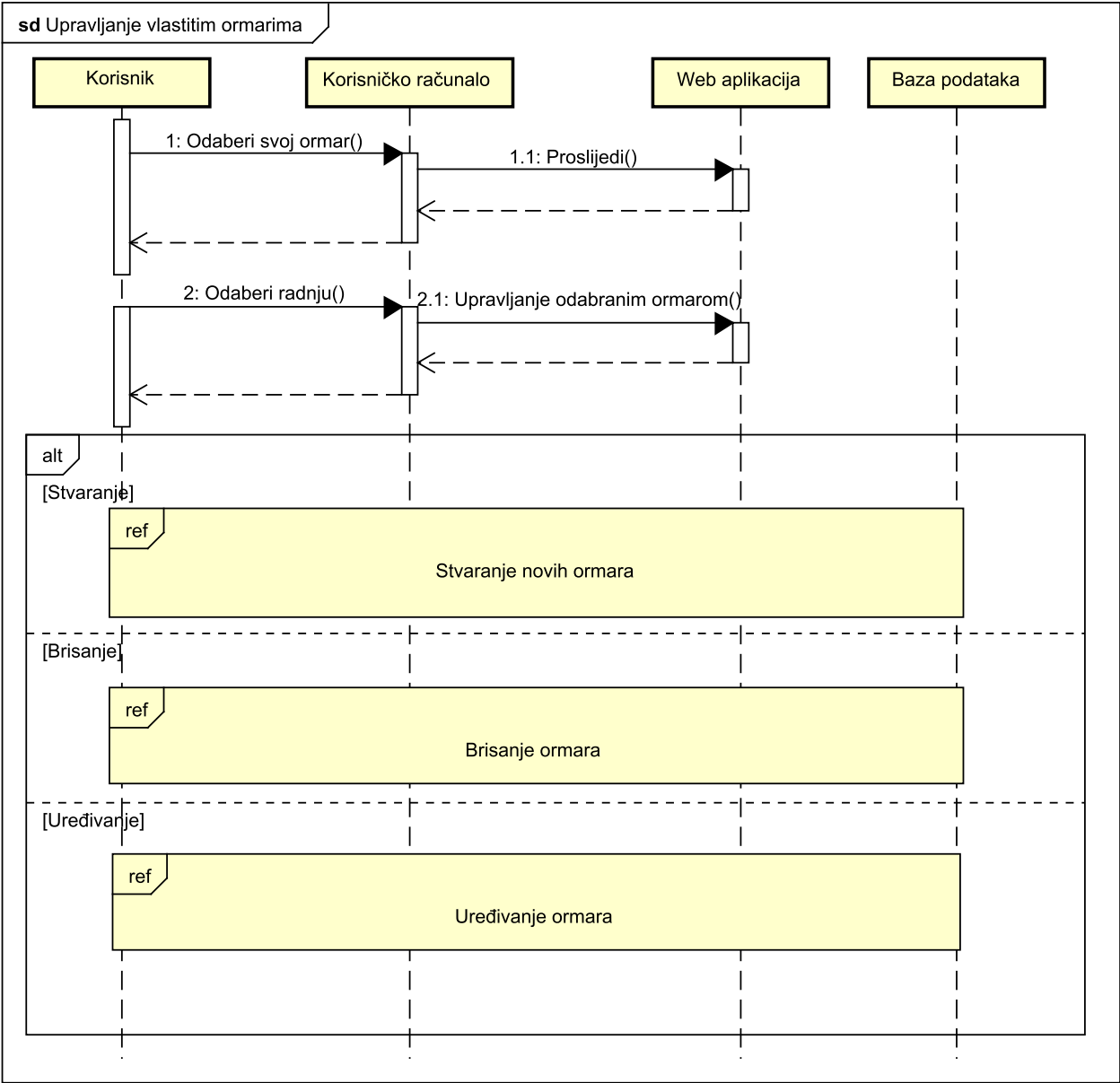
• Stvaranje novih ormara



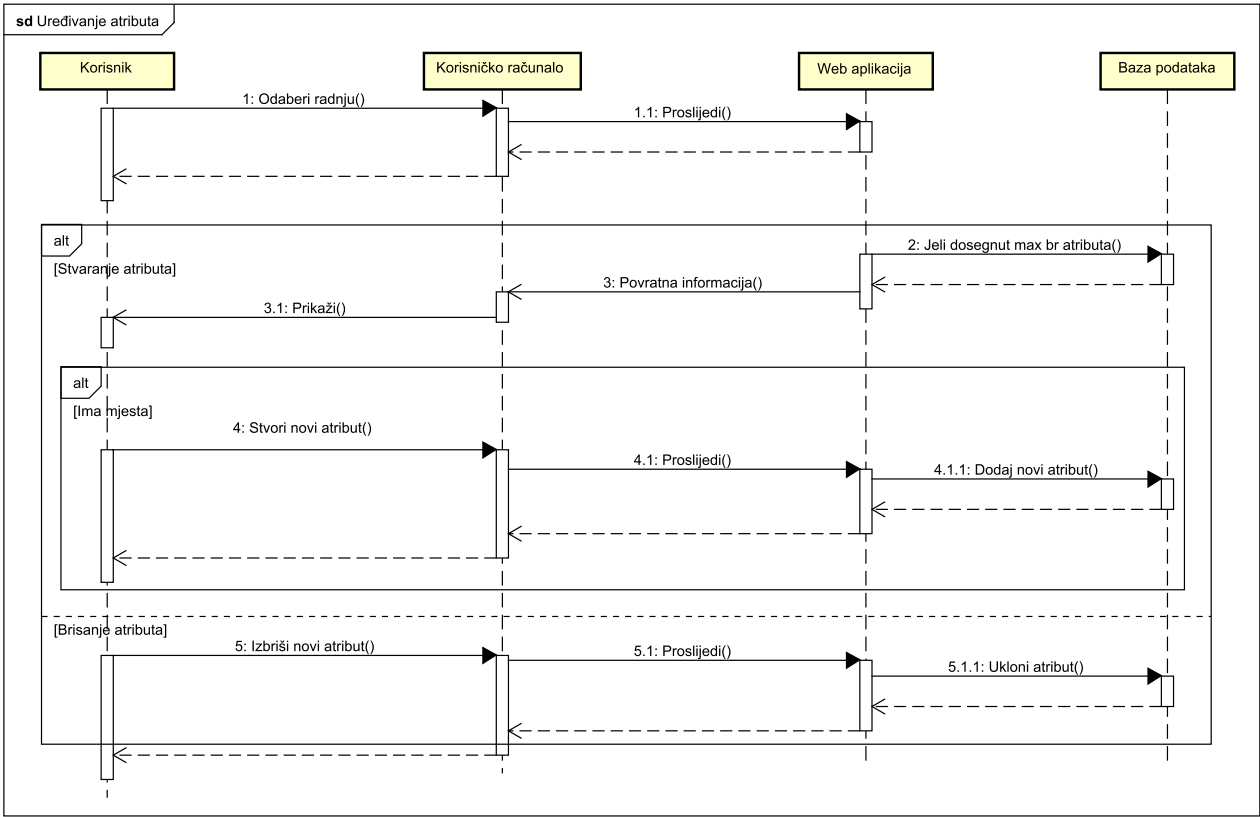
• Upravljanje galerijom



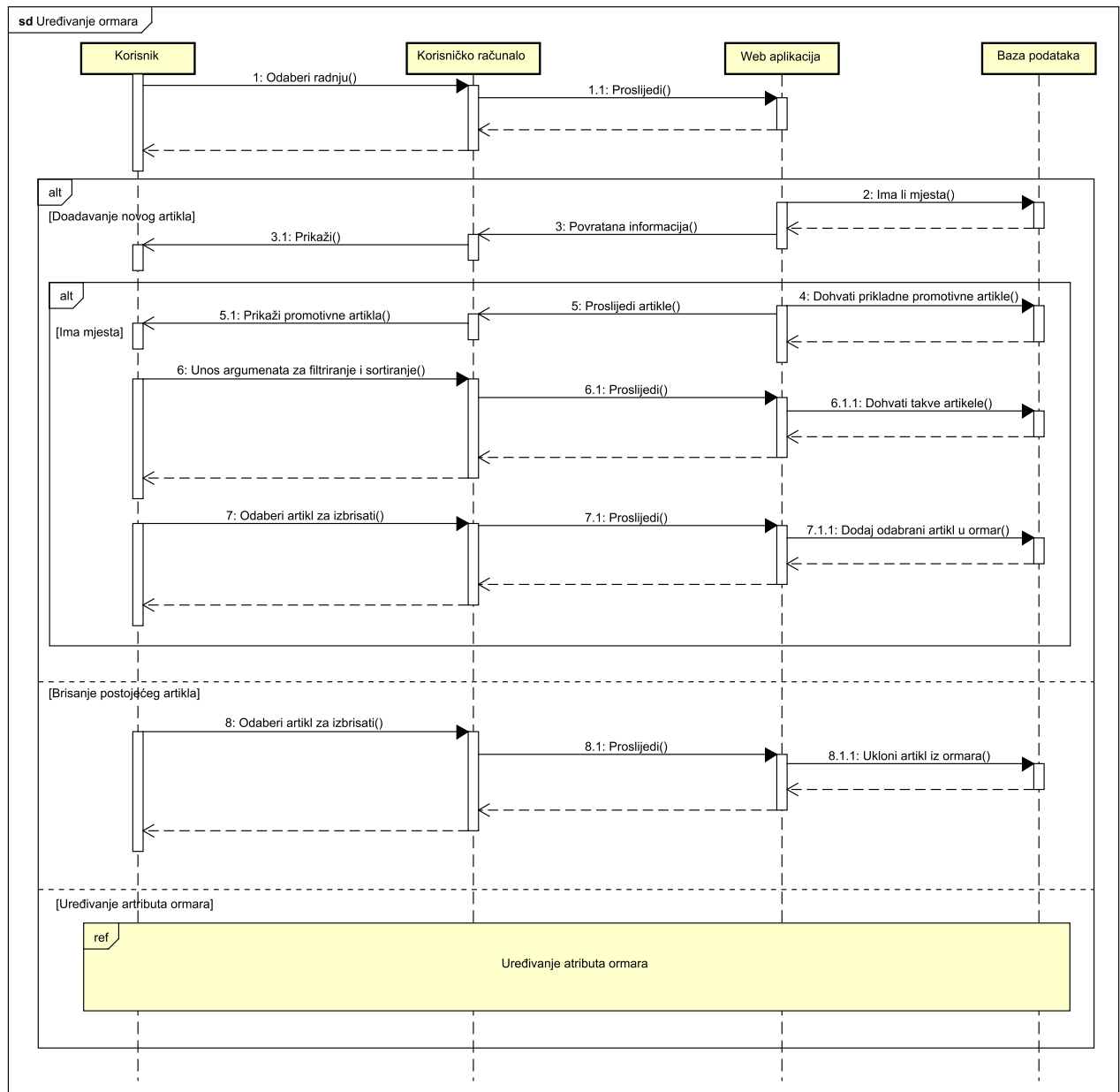
- Upravljanje vlastitim ormarima



- Uređivanje atributa



- Uređivanje ormara



Arhitektura sustava

 Dijagram arhitekture sustava

Opis arhitekture

Backend

- **Java Spring Boot:** Koristimo Spring Boot kao osnovu za backend aplikaciju zbog njegove podrške za izgradnju kompleksnih aplikacija na jednostavan način.
- **Spring Data JPA:** Spring Data JPA olakšava rad s bazom podataka te omogućava pristup SQL podacima na deklarativan način, a koristi se za upravljanje modelima unutar PostgreSQL baze podataka.
- **PostgreSQL Driver:** PostgreSQL baza podataka koristi se kao pouzdano skladište podataka, a Spring Boot aplikacija koristi PostgreSQL web driver za povezivanje i rad s bazom.
- **Spring Security i OAuth2:** Za autentifikaciju i autorizaciju koristimo Spring Security s podrškom za OAuth2, omogućavajući prijavu putem vanjskih providera kao što su Google, Facebook i GitHub.

- **JWT:** JSON Web Token (JWT) koristi se za autentifikaciju korisnika i dodjeljivanje korisničkih uloga (role) unutar sustava. JWT omogućava siguran i pouzdan prijenos identiteta korisnika između frontenda i backenda.

Frontend

- **React:** Frontend aplikacija razvijena je u Reactu kako bismo omogućili modernu, responzivnu i brzu korisničku interakciju.
- **Tailwind CSS i Material UI:** Kombinacija Tailwind CSS-a i Material UI-a koristi se za stilizaciju korisničkog sučelja. Tailwind CSS omogućava fleksibilnost i prilagodljivost dizajna, dok Material UI doprinosi elementima dizajna koji se lako integriraju i prilagođavaju korisničkom iskustvu.
- **Axios:** Axios se koristi za slanje HTTP zahtjeva s frontenda prema backendu, omogućujući komunikaciju i pristup podacima putem REST API-ja.

Verzioniranje i Deployment

- **Git i GitHub:** Git omogućava verzioniranje koda, dok se GitHub koristi za timsku suradnju i verzioniranje cjelokupnog razvojnog procesa.
- **Render:** Render platforma koristi se za deployment aplikacije. Na Renderu su hostani baza podataka, backend i frontend, omogućavajući stabilan i skalabilan pristup sustavu za sve korisnike.

Ostale komponente i alati

- **JWT i Cookie-based autentifikacija:** Korisnička autentifikacija održava se putem JWT-a, pohranjenog u kolačićima na frontend strani, kako bi se postigao siguran prijenos tokena između korisnika i servera.
- **Leaflet i OpenStreetMap:** Prikazivanje mape unutar aplikacije ostvaruje se koristeći Leaflet library za React, a mapa i podaci o mapi se dohvaćaju preko otvorenog OpenStreetMap servera.

Obrazloženje odabira arhitekture

Ova struktura omogućava nam stabilan razvoj, sigurnu autentifikaciju i jednostavno održavanje aplikacije, pružajući korisnicima optimalno iskustvo unutar digitalnog ormara.

Organizacija sustava na visokoj razini

- **Baza podataka:** Baza podataka nam služi za spremanje informacija o korisnicima, ormarima i artiklima
- **Grafičko sučelje:** Korisnik ima pristup samo grafičkom sučelju, to grafičko sučelje je dalje povezano sa ostatkom aplikacije

Organizacija aplikacije

Aplikacija je organizirana u slojevit arhitekturu koja osigurava modularnost, skalabilnost i održivost sustava. Ključni slojevi su frontend i backend, a arhitektura koristi MVC (Model-View-Controller) obrazac za jasno odvajanje odgovornosti.

Frontend

Frontend sloj odgovoran je za korisničko sučelje i interakciju korisnika s aplikacijom. Implementiran je koristeći moderne web tehnologije kako bi aplikacija bila responzivna i prilagođena različitim veličinama ekrana.

Komponente:

- View komponenta: Renderira podatke korisniku i omogućuje interaktivnost (navigacija, pretraživanje, registracija).
- API klijent: Komunicira s backendom putem HTTP zahtjeva koristeći REST API.

Backend

Backend sloj pruža poslovnu logiku, upravljanje podacima i komunikaciju s bazom podataka. Implementiran je u objektno-orijentiranom jeziku.

Komponente:

- Controllers: Prima zahtjeve od frontend sloja, obrađuje ih i vraća odgovore.

MVC arhitektura

Aplikacija koristi MVC (Model-View-Controller) obrazac za odvajanje različitih dijelova sustava:

1. Model (Backend - Data Layer):

- Predstavlja podatke sustava, njihove karakteristike i odnose.
- Koristi ORM za pristup i manipulaciju podacima u bazi (npr. definiranje strukture ormara i artikala).
- Osigurava integritet podataka kroz validaciju i pravila.

2. View (Frontend):

- Prikazuje podatke korisnicima na vizualno privlačan i responzivan način.
- Povezuje se s kontrolerima putem API poziva.
- Omogućuje korisničke interakcije, poput dodavanja ili pretraživanja artikala.

3. Controller (Backend - Logic Layer):

- Posreduje između modela i pogleda.
- Prima zahtjeve od korisnika (npr. preko frontend API poziva), obrađuje ih i dohvaća ili mijenja podatke u modelu.
- Omogućuje poslovnu logiku, npr. preporuku odjevnih kombinacija na temelju kriterija.

Baza podataka

Koristimo PostgreSQL bazu podataka. Zbog bolje strukture i lakšeg korištenja odlučili smo se za SQL bazu podataka ispred NoSQL baze. Baza se poslužuje ne zasebnom poslužitelju tako da ju je moguće ponovno iskoristiti ukoliko dođe do značajnije promjene arhitekture backenda.

Opis tablica

User

Atribut	Tip podatka	Opis varijable
---------	-------------	----------------

Atribut	Tip podatka	Opis varijable
id	BIGINT	Jedinstveni identifikator korisnika
username	VARCHAR	Korisničko ime korisnika
password	VARCHAR	Šifra korisnika
email	VARCHAR	Email adresa korisnika
isAdvertiser	BOOLEAN	Oznaka je li korisnik oglašivač
displayName	VARCHAR	Prikazno ime korisnika

user_roles

Atribut	Tip podatka	Opis varijable
user_id	BIGINT	ID korisnika, vanjski ključ prema tablici User
role_id	BIGINT	ID uloge, vanjski ključ prema tablici Role

Item

Atribut	Tip podatka	Opis varijable
itemId	BIGINT	Jedinstveni identifikator predmeta
itemName	VARCHAR	Naziv predmeta
description	TEXT	Opis predmeta
category	VARCHAR	Kategorija predmeta
section_id	BIGINT	ID sekcije, vanjski ključ prema tablici Section

Role

Atribut	Tip podatka	Opis varijable
id	BIGINT	Jedinstveni identifikator uloge
name	VARCHAR	Ime uloge

Section

Atribut	Tip podatka	Opis varijable
sectionId	BIGINT	Jedinstveni identifikator sekcije
sectionName	VARCHAR	Naziv sekcije
sectionType	VARCHAR	Tip sekcije
itemCapacity	INT	Kapacitet predmeta u sekciji

Atribut	Tip podatka	Opis varijable
wardrobe_id	BIGINT	ID ormara, vanjski ključ prema tablici Wardrobe

Wardrobe

Atribut	Tip podatka	Opis varijable
wardrobeld	BIGINT	Jedinstveni identifikator ormara
wardrobeName	VARCHAR	Naziv ormara
isPublic	BOOLEAN	Oznaka je li ormar javan
user_id	BIGINT	ID korisnika, vanjski ključ prema tablici User

AdvertiserProfile

Atribut	Tip podatka	Opis varijable
id	BIGINT	Jedinstveni identifikator profila oglašivača
user_id	BIGINT	ID korisnika, vanjski ključ prema tablici User
companyAddress	VARCHAR	Adresa tvrtke oglašivača
companyEmail	VARCHAR	Email adresa tvrtke oglašivača

Sve tablice biti će dodane tijekom razvoja aplikacije kad im se atributi budu mogli točno definirati po potrebama aplikacije

Dijagram baze podataka

ER Model

ER v1.1

 Database ER Model

Dijagram razreda

 Dijagram razreda

dio 2. revizije

Prilikom druge predaje projekta dijagram razreda i opisi moraju odgovarati stvarnom stanju implementacije.

Dijagram stanja

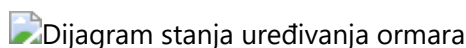
Glavni dijagram stanja



Opis dijagrama stanja

- Početno stanje:
Dijagram započinje na početnoj stranici (Homepage), koja ima uvjete za prijavljene i neprijavljene korisnike.
Ovisno o statusu korisnika, početna stranica nudi različite mogućnosti, poput pregleda javnih ormara ili artikala.
- Homepage funkcionalnosti:
Neprijavljeni korisnici imaju pristup osnovnim pregledima, dok prijavljeni korisnici i oglašivači mogu uređivati profile i upravljati sadržajem (npr. vlastitim ormarima i artiklima).
Pritisak na određene gumbе vodi korisnike do specifičnih akcija poput "Uređivanja profila" ili "Pregleda javnih ormara".
- Registracija i prijava:
Prijava i registracija razdvojene su u zasebne procese.
Korisnici biraju između dvije vrste računa: osobni korisnik ili oglašivač.
Svaka registracija vodi u uspješnu prijavu ili povratak na početnu stranicu.
- Uređivanje profila:
Prikazuje funkcionalnosti poput promjene podataka o korisniku, koje su pohranjene u bazu podataka.
Svaka promjena potvrđena je povratkom na početnu stranicu s ažuriranim podacima.
- Pregled javnih ormara i artikala:
Korisnicima omogućuje pregled dostupnih ormara i njihovih sadržaja.
Implementirana je funkcionalnost prikaza ormara na karti.
- Upravljanje sadržajem (ormari i artikli):
Omogućeno je dodavanje, ažuriranje i brisanje ormara i artikala.
Dodavanje artikala vodi kroz proces unosa podataka, slanja zahtjeva prema bazi podataka i potvrđivanja.
- Navigacija između stanja:
Korisnici se mogu vraćati na prethodna stanja ili navigirati kroz ključne funkcionalnosti sustava.

Dijagram stanja uređivanja ormara



Opis dijagrama stanja

- Početno stanje:
Dijagram započinje izborom opcije za uređivanje ormara.
Korisnici mogu pregledavati i uređivati postojeće ormare ili stvarati nove sekcije i artikle.
- Uređivanje ormara:

- Središnji dio dijagrama fokusira se na tri glavne funkcionalnosti:
 1. Brisanje ormara: Korisnik bira određeni ormar, potvrđuje odluku o brisanju, što rezultira ažuriranjem baze podataka i povratkom na pregled svih ormara.
 2. Dodavanje nove sekcije: Korisnik definira naziv sekcije i unosi podatke. Ako je broj sekcija dostigao maksimalni dozvoljeni broj, proces je zaustavljen.
 3. Brisanje sekcije: Omogućeno je brisanje određene sekcije unutar ormara uz prethodnu potvrdu korisnika.
- Uređivanje sekcije:
 - Nakon odabira sekcije, korisniku je omogućeno:
 1. Dodavanje novih artikala: Korisnik unosi podatke o artiklu, nakon čega se podaci pohranjuju u bazu.
 2. Brisanje artikala: Prikazuje se mogućnost odabira artikala za brisanje, uz potvrdu odluke.
 3. Pretraživanje artikala: Korisnik može pretraživati artikle prema lokaciji (na karti) ili unesenim kriterijima.
- Navigacija između stanja:

Korisnik ima mogućnost vraćanja na prethodne korake ili nastavka prema sljedećim funkcionalnostima. Svaka akcija ažurira bazu podataka i prikazuje odgovarajuće promjene na korisničkom sučelju.
- Završetak uređivanja:

Nakon obrade svih promjena (brisanje, dodavanje ili ažuriranje), korisnik se vraća na početni pregled ormara.

Dijagram aktivnosti

Glavni dijagram aktivnosti

 Glavni dijagram aktivnosti


Dijagram aktivnosti obavijesti

 Dijagram aktivnosti obavijesti

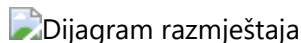
Dijagram aktivnosti predmeta

 Dijagram aktivnosti predmeta

Dijagram komponenti

 Dijagram komponenti

Dijagram razmještaja



Arhitektura sustava

Arhitektura sustava predstavlja temeljni okvir za razumijevanje i implementaciju svih njegovih funkcionalnosti. U kontekstu razvojne dokumentacije aplikacija, **dijagrami komponenata** i **dijagrami razmještaja** odlučujući su za prikaz povezanosti i rasporeda različitih komponenata sustava. Ovi dijagrami omogućuju sudionicima projekta razumijevanje i vizualizaciju fizičkog i logičkog dizajna sustava, uključujući interakcije između dijelova aplikacije, što je odlučujuće za efikasnu implementaciju i dugoročnu održivost sustava.

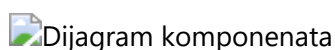
Arhitektura sustava, u kontekstu dijagrama komponenata i razmještaja, pruža uvid u strukturu i raspored ključnih dijelova aplikacije. Ovi dijagrami nisu korisni samo tijekom faza oblikovanja i implementacije, već služe i kao alati za održavanje i optimizaciju sustava u budućnosti.

Izrada dijagrama

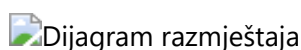
Kao dio razvoja aplikacije, važno je osmisliti i dokumentirati arhitekturu sustava s naglaskom na dijagrame komponenata i razmještaja:

- **Dijagram komponenata:** Prikaz ključnih funkcionalnih komponenata aplikacije, njihove međusobne povezanosti i sučelja za komunikaciju.
 - **Dijagram razmještaja:** Prikaz fizičkog i virtualnog rasporeda komponenata unutar infrastrukture sustava, uključujući fizičke i virtualne resurse poput poslužitelja ili uređaja krajnjih korisnika.
-

Dijagram komponenata



Dijagram razmještaja



Ovo poglavlje treba opisati provedena ispitivanja implementiranih funkcionalnosti na razini komponenti i sustava. Fokus je na odabiru i izvedbi ispitnih slučajeva koji obuhvaćaju redovne, rubne uvjete i testiranje grešaka, kao i upotrebu odgovarajućih alata za provedbu testiranja.

Komponentno Ispitivanje

Cilj ispitivanja komponenti je provjera osnovnih funkcionalnosti implementiranih u razredima sustava. Ovdje se testira svaka komponenta izolirano kako bi se osigurala ispravnost i reakcija na različite scenarije.

Zadaci

Razvijeno je minimalno 6 ispitnih slučajeva koji obuhvaćaju:

- **Redovne slučajeve:** Testiranje uobičajenog ponašanja funkcionalnosti.

- **Rubne uvjete:** Provjera ulaznih podataka na granici valjanosti.
- **Izazivanje pogreške (exception throwing):** Testiranje reakcije na iznimke.
- **Nepostojeće funkcionalnosti:** Provjera reakcije na poziv neimplementirane funkcionalnosti.

Struktura ispitivanja

1. Registracija korisnika

Funkcionalnost: Registracija novog korisnika.

- **Ulazni podaci:**
 - Korisničko ime: "newUser"
 - Lozinka: "password123"
 - Email: "newuser@test.com"
- **Očekivani rezultat:** Uspješna registracija.
- **Dobiveni rezultat:** Prolaz.
- **Postupak:**
 1. Pošalji zahtjev na `/api/register` s podacima korisnika.
 2. Verificiraj HTTP status 201 i poruku "User registered successfully".

2. Prijava korisnika

Funkcionalnost: Prijava postojećeg korisnika.

- **Ulazni podaci:**
 - Korisničko ime: "testUser"
 - Lozinka: "password123"
- **Očekivani rezultat:** Uspješna prijava.
- **Dobiveni rezultat:** Prolaz.
- **Postupak:**
 1. Pošalji zahtjev na `/api/login` s podacima za prijavu.
 2. Verificiraj HTTP status 200 i generirani JWT token.

3. Neuspjela prijava s pogrešnom lozinkom

Funkcionalnost: Obrada neuspjele prijave.

- **Ulazni podaci:**
 - Korisničko ime: "testUser"
 - Lozinka: "wrongPassword"
- **Očekivani rezultat:** Pogreška prijave.
- **Dobiveni rezultat:** Prolaz.
- **Postupak:**
 1. Pošalji zahtjev na `/api/login` s pogrešnom lozinkom.
 2. Verificiraj HTTP status 400.

4. Registracija s već postojećim emailom

Funkcionalnost: Obrada registracije s duplikatom emaila.

- **Ulazni podaci:**
 - Email: "existing@test.com"
 - Lozinka: "password123"
- **Očekivani rezultat:** Pogreška registracije.
- **Dobiveni rezultat:** Prolaz.
- **Postupak:**
 1. Pošalji zahtjev na `/api/register` s dupliciranim emailom.
 2. Verificiraj HTTP status 400 i poruku "Email is already taken".

5. Dohvat ormara korisnika

Funkcionalnost: Dohvat liste ormara za korisnika.

- **Ulazni podaci:**
 - Korisničko ime: "testUser"
- **Očekivani rezultat:** Lista ormara.
- **Dobiveni rezultat:** Prolaz.
- **Postupak:**
 1. Pošalji GET zahtjev na `/api/wardrobes/`.
 2. Verificiraj HTTP status 200 i sadržaj odgovora.

6. Brisanje ormara drugog korisnika

Funkcionalnost: Obrada neuspjelog brisanja.

- **Ulazni podaci:**
 - Korisničko ime: "testUser"
 - ID ormara: 2
- **Očekivani rezultat:** Pogreška autorizacije.
- **Dobiveni rezultat:** Prolaz.
- **Postupak:**
 1. Pošalji DELETE zahtjev na `/api/wardrobes/delete/2`.
 2. Verificiraj HTTP status 405 i poruku "You cannot edit outhers user's wardrobes".

Rezultati:

 Rezultati unit testova  Rezultati unit testova

Sustavno Ispitivanje

Cilj ispitivanja sustava je testiranje ponašanja cijelog sustava u uvjetima stvarnog korištenja, uz provjeru međusobne povezanosti komponenti.

Zadaci

Razvijeno je minimalno 4 ispitna slučaja:

- **Redovni slučajevi:** Očekivano ponašanje sustava.

- **Rubni uvjeti:** Reakcija sustava na granične ulaze.
- **Poziv nepostojećih funkcionalnosti:** Reakcija sustava na neispravne funkcije.

Struktura ispitnih slučajeva

1. Prijava s nevažećom lozinkom (Selenium)

- **Ulazi:**
 - Korisničko ime: "seleniumTestAgent"
 - Lozinka: "badPassword123"
- **Koraci:**
 1. Otvori aplikaciju u pregledniku.
 2. Unesi korisničko ime i lozinku.
 3. Klikni na "Prijava".
 4. Verificiraj poruku greške.
- **Očekivani izlaz:** "Login failed. Please check your credentials."
- **Dobiveni izlaz:** Prolaz.

2. Registracija bez korisničkog imena (Selenium)

- **Ulazi:**
 - Email: "seleniumTestAgent2@styletrack.com"
 - Lozinka: "seleniumTestPassw0rd"
 - Potvrda lozinke: "seleniumTestPassw0rd"
- **Koraci:**
 1. Otvori aplikaciju i idi na stranicu za registraciju.
 2. Unesi email, lozinku i potvrdu lozinke bez korisničkog imena.
 3. Klikni na "Registracija".
 4. Verificiraj poruku greške.
- **Očekivani izlaz:** "Username cannot be empty".
- **Dobiveni izlaz:** Prolaz.

3. Uspješna prijava (Selenium)

- **Ulazi:**
 - Korisničko ime: "seleniumTestAgent"
 - Lozinka: "seleniumTestPassw0rd"
- **Koraci:**
 1. Otvori aplikaciju u pregledniku.
 2. Unesi korisničko ime i lozinku.
 3. Klikni na "Prijava".
 4. Verificiraj da je korisnik preusmjeren na "Feed" stranicu.
- **Očekivani izlaz:** Prikazan "My Wardrobes" gumb.
- **Dobiveni izlaz:** Prolaz.

4. Kreiranje ormara (Selenium)

- **Ulazi:**

- Naziv ormara: "New Test Wardrobe"
- Opis ormara: "This is a new test wardrobe"
- **Koraci:**
 1. Prijavi se u aplikaciju.
 2. Otvori stranicu za kreiranje ormara.
 3. Unesi naziv i opis ormara.
 4. Klikni na "Spremi".
 5. Verificiraj da je ormar uspješno kreiran.
- **Očekivani izlaz:** Prikazan "Create" gumb.
- **Dobiveni izlaz:** Prolaz.

Rezultati:

 Rezultati selenium testova

Korištene tehnologije i alati

1. Programski jezici: Java (Spring Boot na backendu), Javascript (React na frontendu)
2. Radni okviri i biblioteke: Spring Boot, React, Material UI, JSONWebToken, Project Lombok (anotacije za lakše i više čitljivo pisanje klasa na backendu)
3. Baza podataka: Postgre SQL (Poslužuje se pute platfromer Neon [<https://neon.tech>])
4. Razvojni alati: VS Code, IntelliJ Idea Ultimate, GitHub Desktop, git bash
5. Alati za ispitivanje: Java + Selenium, Java Unit testovi
6. Cloud platforma: Render.com za posluživanje backenda i frontenda, Open Weather API za dohvaćanje podataka o vremenskoj prognozi, imagekit.io za pohranu fotografija

Ovaj odjeljak dokumentacije treba dati detaljne smjernice za instalaciju, konfiguraciju, pokretanje i administraciju aplikacije. Cilj je olakšati postavljanje aplikacije na razvojnom, ispitnom i produkcijskom okruženju.

1. Instalacija
2. Postavke
3. Pokretanje aplikacije
4. Upute za administratore
5. Primjer za Render platformu (Cloud Deploy)

Opis pristupa aplikaciji na javnom poslužitelju

- <https://styletrack.onrender.com/>
- Na ovoj poveznici se nalazi naša web stranica koja je prilično intuitivna

Upute za pokretanje i puštanje u pogon

Kako bi se aplikacija pokrenula potrebno je preuzeti ovaj projekt s GitHuba te ga otvoriti u željenima alatima za uređivanje teksta/koda (mi smo koristili VSCode i IntelliJ). Otovrimo frontend projekt te u njemu pozovemo `npm install` kako bi se instalirale sve biblioteke potrebne za pokretanje aplikacije. Nakon što se aplikacija instalira potrebno je stvoriti `.env` file koji sadrži vrijednosti:

```
VITE_BACKEND_URL=http://localhost:8080/api VITE_WEATHER_API_KEY=
```

Ključ za weather API dohvatite prijavom na <https://www.weatherapi.com/>

Također na nekoliko mjesta u kodu potrebno je sve stringove "`https://styletrack-backend.onrender.com`" zamjeniti s "`http://localhost:8080`". Te nakon toga pozvati `npm run dev` kako bi se aplikacija uspješno pokrenula na portu 5173.

Za pokretanje backenda potrebno ga je otvoriti u IntelliJ ili nekom drugom alatu te prije samog pokretanja u konfiguracijskim datotekama postaviti vrijednosti za frontend URL. Tj. u datoteci `OAuth2AuthenticationSuccessHandler.java` potrebno je zamjeniti

```
String redirectUrl = "http://styletrack.onrender.com/oauth2/redirect?token=" + jwtToken.get() +  
"&username=" + username.get();
```

sa

```
String redirectUrl = "http://localhost:5173/oauth2/redirect?token=" + jwtToken.get() + "&username=" +  
username.get();
```

Te u datoteci `SecurityConfig.java`:

```
configuration.setAllowedOrigins(List.of("https://styletrack.onrender.com"));
```

sa:

```
configuration.setAllowedOrigins(List.of("http://localhost:5173")); // Make sure to change this for production
```

Još je potrebno dodati i sve `env` vrijednosti u konfiguraciju pokretanja:

```
DATABASE_PASSWORD= DATABASE_URL= DATABASE_USERNAME= GITHUB_CLIENT_ID=  
GITHUB_CLIENT_SECRET= GOOGLE_CLIENT_ID= GOOGLE_CLIENT_SECRET= IMAGEKIT_KEY= PORT=8080
```

Nakon toga pokrenite Spring Boot aplikaciju i backend će biti online. Ako to pak pokrećete putem rendera kako bi sve bilo online dostupno. Potrebno je postaviti varijable okoline tj. `env` varijable u samom render sučelju te se pobrinuti da se `Dockerfile` datoteka nalazi u repozitoriju kako bi render mogao izgraditi sliku za backend. Za frontend potrebno je isto postaviti varijable okruženja, ali nije potrebno imati `docker` sliku za taj web service jer je on podržan od strane rendera. U renderu je još potrebno podesiti postavku za ispravan rad React Routera tako da se u sekciji `Redirect/Rewrite` doda novo pravilo koje ima vrijednosti: `Source=/*` `Destination=/index.html` te `Action=Rewrite`

Projektni zadatak naše grupe bio je osmisлити i napraviti aplikaciju za uređivanje i dijeljenje virtualnih ormara. Tijekom cijelog zimskog semestra smo imali kontinuiran rad na projektu uz kraće pauze zbog blagdana i ispita. Cjelokupan projekt završen je sa svim našim inicijalnim očekivanjima unutar zadanoga roka od 17 tjedana.

Iznimno bitan dio ovog projekta je bilo inicijalno upoznavanje članova, odabir voditelja te podjela poslova. Uz kvalitetnu podjelu i komunikaciju uspjeli smo nadopuniti međusobne rupe u znanju koje smo imali. Uz mnogo

samostalnog rada na implementaciji rješenja uspjeli smo riješiti zadani problem. Zbog velikog neiskustva u rješavanju ovakvog problema bilo je potrebno uložiti puno vremena u učenje i upoznavanje programskih jezika i odabranih alata. Osim implementacije rješenja trebali smo napraviti i mnoge UML dijagrame te dokumentaciju s kojima se prvi put srećemo ove godine. Uz dobro odrađenu početnu fazu projekta podjele i osmišljavanja projekta kasnija implementacija je bila nešto lakša. Uz redovitu komunikaciju grupe svi su uvijek bili upućeni u razvitak aplikacije te dokumentacije.

Potencijalni budući rad na ovom projektu je implementacija još nekih korisnih opcija i mogućnosti naše web aplikacije, te povećanje kapaciteta podataka. Iako bi to bile neke naprednije opcije, vjerujemo da bi ih uspjeli implementirati puno brže uz sada već stečeno znanje i organizaciju.

Sudjelovanje na ovakvom projektu nam je svima bilo jako korisno jer smo po prvi put iskusili kako je to raditi u većem timu uz veliki zadatak kroz dulji period vremena. Također smo naučili da je dobra komunikacija i dobro raspoređen posao jako važan. Zadovoljni smo sa isporučenom aplikacijom iako ima mjesta za napredak.

Rev.	Opis promjene/dodatka	Autori	Datum
1	Napravljen predložak	Ivan Došlić	19.10.2024
2	Opisani inicijalni funkcijski zahtjevi	Ivan Došlić	19.10.2024
3	Napravljeni sekvencijski dijagrami	Benjamin Teskera	26.10.2024
4	Napravljen dijagram baze podataka	Ivan Došlić	28.10.2024
5	Napravljen UseCase dijagram	Ivan Došlić	28.10.2024
6	Popravljeni sekvencijski dijagrami	Benjamin Teskera	2.11.2024
7	Popravljen UseCase dijagram	Lucija Bralić	3.11.2024
8	Napravljen dijagram razreda	Ivan Došlić	11.11.2024
9	Popravljen predložak i popunjena sva dokumentacija za prvu reviziju	Dominik Kos	10.11.2024 - 15.11.2024
10	Popravljen dijagram razreda	Marko Lujo	15.11.2024
11	Napravljen dijagram arhitekture sustava	Ivan Došlić	15.11.2024
12	Napravljen dijagram stanja	Benjamin Teskera	19.1.2025
13	Napravljen dodatan dijagram stanja	Benjamin Teskera	20.1.2025
14	Popunjena sva dokumentacija za drugu reviziju	Dominik Kos	20.1.2025 - 24.1.2025
15	Napravljeni dijagrami aktivnosti	Marko Lujo	23.1.2025
16	Napravljeni dijagram komponenata i dijagram razmještaja	Lucija Bralić	24.1.2025

Rev.	Opis promjene/dodatka	Autori	Datum
17	Dodani rezultati testiranja i upute za pokretanje aplikacije lokalno	Ivan Došlić	24.1.2025

- ovo treba biti vidljivo u samoj dinamici promjena repozitorija

Evidencija promjena sadrži popis promjena izvršenih tijekom cijelo životnog trajanja predmeta evidencije (dokumentacije, projekta i sl.). Osnova svrha je praćenja napretka svake promjene na temelju njezina preispitivanja, odobrenja (ili odbijanja), provedbe, kao i zaključenja. Štoviše, dobar dnevnik promjena sadrži i datum promjene i njegov utjecaj na projekt u smislu rizika, vremena i troškova. Sve te promjene prenose se dionicima. Štoviše, odbačene promjene također su uključene u povijest promjena.

Literatura

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.Langanieri, "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/](http://www.ece.rutgers.edu/~marsic/books/SE) books/SE
5. The Unified Modeling Language, <https://www.uml-diagrams.org/>
6. Astah Community, <http://astah.net/editions/uml-new>
7. Markdown Guide, <https://www.markdownguide.org/basic-syntax/>
8. Figma Tutorial, <https://www.youtube.com/watch?v=ezldKx-jPag&t=1625s>

Prikaz grupnih aktivnosti

1. Prvi sastanak

- Datum: 16. listopada 2024.
- Prisustvovali: Lucija Bralić, Marko Lujo, Benjamin Teskera, Ivan Džankić, Dominik Kos, Domagoj Radman, Ivan Došlić
- Teme sastanka:
 - Uspostavljena Whatsapp grupa svih članova
 - Inicijalni dogovor oko raspodjela odgovornosti
 - Odabir voditelja grupe
 - Analiza zadatka
 - Odabir alata i tehnologija

2. Drugi sastanak

- Datum: 11. studenog 2024.
- Prisustvovali: Marko Lujo, Benjamin Teskera, Ivan Džankić, Dominik Kos, Domagoj Radman, Ivan Došlić
- Teme sastanka:
 - Dogovor za popravljivanje stavki u dokumentaciji
 - Daljnja podjela rada

3. Daljnji dogovor

- Daljnji dogovor bio je putem whatsapp grupe na tjednoj bazi, bez većih dogovorenih sastanaka

Plan rada


- Tablični/Gantt/Kanban prikaz
 - Prikaz vremenskog plana rada ključnih aktivnosti (tjedna granulacija)
 - Uključuje akronime angažiranih članova tima TODO: primjer

Tablica aktivnosti

Stavke	Lucija Bralić	Marko Lujo	Benjamin Teskera	Ivan Džankić	Dominik Kos	Domagoj Radman	Ivan Došlić
Upravljanje projektom							30
Opis projekt-nog zadatka					2		1
Funkcionalni zahtjevi					5		1
Opis pojedinih obrazaca	3				3		
Dijagram obrazaca	5						
Sekvencijski dijagrami		1	7				
Opis ostalih zahtjeva					3		
Arhitektura i dizajn sustava	7	15	25				10
Baza podataka							3
Dijagram razreda		4					
Dijagram stanja		1	7				
Dijagram aktivnosti		5					
Dijagram komponenti	4						
Korištene tehnologije i alati					1		2
Ispitivanje programskog rješenja			1				
Dijagram razmještaja	3						
Upute za puštanje u pogon							1
Dnevnik sastajanja					1		

Stavke	Lucija Bralić	Marko Lujo	Benjamin Teskera	Ivan Džankić	Dominik Kos	Domagoj Radman	Ivan Došlić
Zaključak i budući rad					1		
Popis literature					1		
Dodatne stavke kako ste podijelili	3				15		
Izrada početne stranice							10
Izrada baze podataka		1					2
Spajanje s bazom podataka		3					1
Izrada prezentacije					2		

Dijagram pregleda promjena

 Additions graph

 Commits graph

 Deletions graph

Ključni izazovi i rješenja

- Glavni izazov bio je nedostatak iskustva i znanja
- Izazov je bio i pravovremena organizacija vremena
- Rješenja ovih problema su sudjelovanje u više ovakvih projekata da se skupi iskustvo i znanje