

# Programsko inženjerstvo ak.god 2024./2025

Sveučilište u Zagrebu

Fakultet elektrotehnike i računarstva

## Virtualni ormar

Tim:

Ime tima: Slagači ormara

Nastavnik: Vlado Sruk

## Web aplikacija

Front-end: <https://closetly-721y.onrender.com/>

Back-end: <https://ormarbackend.onrender.com/>

Inicijalno pokretanje stranice zahtjeva nekoliko minuta jer je servis deployan na besplatnu verziju Rendera, pa se nakon nekog vremena automatski izgasi i zahtjeva ponovno pokretanje.

### Članovi tima:

- **Lukas Marović** (voditelj/back-end/deployment)
- **Toni Bajan** (back-end)
- **Maja Miličević** (baza podataka i dokumentacija)
- **Lara Šimić** (baza podataka i dokumentacija)
- **Dražen Pešut** (front-end)
- **Filip Vrbnac** (front-end)
- **Patrik Kapetan** (front-end)

## 1. Opis projektnog zadatka

### Opis

Projekt „Virtualni ormar” ima za cilj pružiti rješenje za organizaciju odjeće i obuće putem intuitivne digitalne platforme. Ova aplikacija omogućuje korisnicima pregled, organizaciju i dijeljenje odjevnih artikala u virtualnom okruženju, što pojednostavljuje svakodnevno planiranje odjevnih kombinacija te olakšava pregledavanje i upravljanje garderobom. Posebno je koristan alat za osobe s velikim brojem odjevnih komada i obuće, koje često troše puno vremena u organizaciji ormara ili pretraživanju željenog artikla. Osim vlastite organizacije ormara, aplikacija će omogućiti dijeljenje određenih odjevnih komada s drugim korisnicima, što dodatno potiče društvenu interakciju, smanjuje potrebu za kupovinom i potiče održivo upravljanje odjećom.

#### Korist projekta

Kroz aplikaciju „Virtualni ormar” mogu poboljšati preglednost garderobe, automatizirati preporuke za odjevne kombinacije na temelju vremenskih uvjeta, preferencija i stila, dijeliti odjeću te uštedjeti vrijeme na organizaciji garderobe i osmišljavanju odjevnih kombinacija.

#### Postojeća slična rješenja

Trenutno na postoje aplikacije koje se bave organizacijom ormara i digitalnim popisivanjem odjeće, poput aplikacija „Cladwell”, „Stylebook” i „Smart Closet”. Svaka od njih ima specifične značajke, no postoje značajne razlike u odnosu na projekt „Virtualni ormar”:

Cladwell: omogućava planiranje odjevnih kombinacija, ali nije fokusiran na dijeljenje odjeće s drugim korisnicima

Stylebook: ima značajke za organizaciju ormara i kreiranje modnih kombinacija, ali nema komponentu dijeljenja

Smart Closet: pruža vizualnu organizaciju ormara i prijedloge odjevnih kombinacija, ali ne nudi prilagođavanje ormara prema fizičkoj strukturi

S obzirom na to, projekt „Virtualni ormar” donosi nekoliko inovacija:

- prilagodbu ormara stvarnim fizičkim lokacijama poput polica, ladica i šipki, čime se korisnicima omogućuje da digitalno preslikaju strukturu svojih ormara te se lakše snađu u organizaciji
- funkcionalnost dijeljenja artikala s drugim korisnicima u stvarnom vremenu
- geo-lokacijsko pretraživanje za brzo pronalaženje dostupnih artikala za posudbu ili razmjenu u blizini.

#### Skup zainteresiranih korisnika

- pojedinci s velikom garderobom koji traže sustav za lakše upravljanje i planiranje odjevnih kombinacija
- ekološki osviješteni korisnici zainteresirani za smanjenje tekstilnog otpada kroz dijeljenje odjeće
- stilisti i influenceri kojima aplikacija može pomoći u održavanju evidencije i predlaganju odjevnih kombinacija
- oglašivači u modnoj industriji koji kroz ovu platformu mogu lako doprijeti do zainteresiranih korisnika i prikazati svoje proizvode

#### Opseg projektnog zadatka

Osnovne funkcionalnosti projekta uključuju:

- organizaciju garderobe u virtualnom ormaru prema prilagodljivim fizičkim lokacijama (ladica, polica, šipka)
- dodavanje odjevnih artikala s detaljnim karakteristikama (boja, kategorija, sezona, stanje, lokacija)

- pretraživanje ormara i predlaganje odjevnih kombinacija prema zadanim kriterijima
- dijeljenje artikala s drugim korisnicima i pretraživanje dijeljenih artikala prema geolokaciji i karakteristikama
- upravljanje profilom i osnovnim informacijama korisnika te prilagodba ormara i artikala prema vlastitim potrebama

Moguće nadogradnje projektnog zadatka

- povezivanje s e-trgovinama – omogućilo bi kupnju predmeta direktno iz aplikacije kada korisnik otkrije potrebu za određenim komadom odjeće koji nema u garderobi
- podrška za razmjenu i prodaju odjeće – korisnicima bi omogućila postavljanje cijena za predmete koje žele prodati ili mijenjati s drugim korisnicima
- preporuke temeljene na umjetnoj inteligenciji – automatizirani sustav mogao bi nuditi personalizirane prijedloge za kombinacije i kupovinu novih komada na temelju korisnikovih navika i preferenca
- praćenje održivosti – kroz aplikaciju korisnici bi mogli pratiti održivost svojih modnih izbora, primjerice broj razmjena odjeće ili procjenu uštede tekstilnog otpada

2. Analiza zahtjeva

Funkcionalni zahtjevi

ID Zahtjeva	Opis	Prioritet	Izvor	Kriteriji prihvatanja
F-001	Mogućnost registracije korisnika	Visok	Neregistrirani korisnik	Korisnik pritišće gumb za registraciju, unosi osobne podatke (ime, prezime, email, lozinku), a sustav provjerava unikatnost email adrese i validira unesene podatke.
F-002	Mogućnost prijave korisnika	Visok	Registrirani korisnik i oglašivač	Korisnik pritišće gumb za prijavu, unosi podatke za prijavu (email, lozinka), a sustav provjerava i validira unošene podatke.
F-003	Podržava rad tri tipa korisnika - registriranih, neregistriranih te oglašivača	Visok	Sustav	Sustav podržava tri tipa korisnika, neregistrirane i registrirane korisnike te oglašivače. Svaki tip korisnika ima različite mogućnosti s obzirom na tip.
F-004	Mogućnost pretraživanja svih artikala označenih za dijeljenje	Visok	Neregistrirani i registrirani korisnik	Korisnik može pretraživati sve artikle drugih korisnika koji su postavljeni za dijeljenje od strane drugih korisnika.
F-005	Mogućnost pretraživanja svih artikala prema zadanim kriterijima	Visok	Neregistrirani i registrirani korisnik	Korisnik može unijeti kriterije za pretraživanje (npr. veličina, boja, vrsta artikla).Sustav prikazuje sve artikle koji odgovaraju zadanim kriterijima.
F-006	Dodavanje, pregled i uklanjanje ormara	Visok	Registrirani korisnik	Korisnik može dodati, ukloniti i pregledati svoje virtualne ormare.
F-007	U ormare se mogu dodavati i uklanjati artikli odjeće i obuće te dijeliti artikli	Visok	Registrirani korisnik	Korisnik može dodati i ukloniti artikle iz ormara te ih označi za dijeljenje drugim korisnicima.
F-008	Mogućnost pregledavanja artikala	Visok	Registrirani korisnik	Korisnik može odabrati postojeći artikl i pregledati njegove karakteristike.
F-009	Definiranje i mijenjanje strukture ormara	Visok	Registrirani korisnik	Korisnik može definirati i mijenjati raspodjelu prostora u ormaru.
F-010	Artikli imaju karakteristike prema kojima se može pretraživati	Visok	Registrirani korisnik	Korisnik može pretraživati artikle u svojem ormaru prema nekim karakteristikama.
F-011	Aplikacija predlaže odjevnu kombinaciju na temelju zadanih kriterija te vremenskih uvjeta kod geolokacije korisnika	Visok	Registrirani korisnik	Korisnik može zatražiti prijedlog odjevne kombinacije prema zadanim kriterijima i vremenskim uvjetima
F-012	Definiranje karakteristike artikla	Visok	Registrirani korisnik	Korisnik može dodati svoj artikl te definirati njegove karakteristike
F-013	Oglašavanje artikala te brisanje oglasa	Visok	Oglašivač	Oglašivač može dodati artikl u svoju galeriju oglašivača, mogu se oglašavati na profilima registriranih korisnika i obrisati svoj oglas.
F-014	Mogućnost pregleda galerija artikala	Visok	Oglašivač	Oglašivač može pregledavati galeriju postavljenih oglasa.

Ostali zahtjevi

ID Zahtjeva	Opis	Prioritet
NF-1.1	Podaci korisnika će biti pohranjeni u sigurnoj bazi podataka	Visok
NF-1.2	Aplikacija će imati respozivan dizajn	Visok
NF-1.3	Sustav će podržavati rad više korisnika u stvarnom vremenu	Visok
NF-1.4	Koristi OAuth 2.0 za autorizaciju	Srednji

Dionici

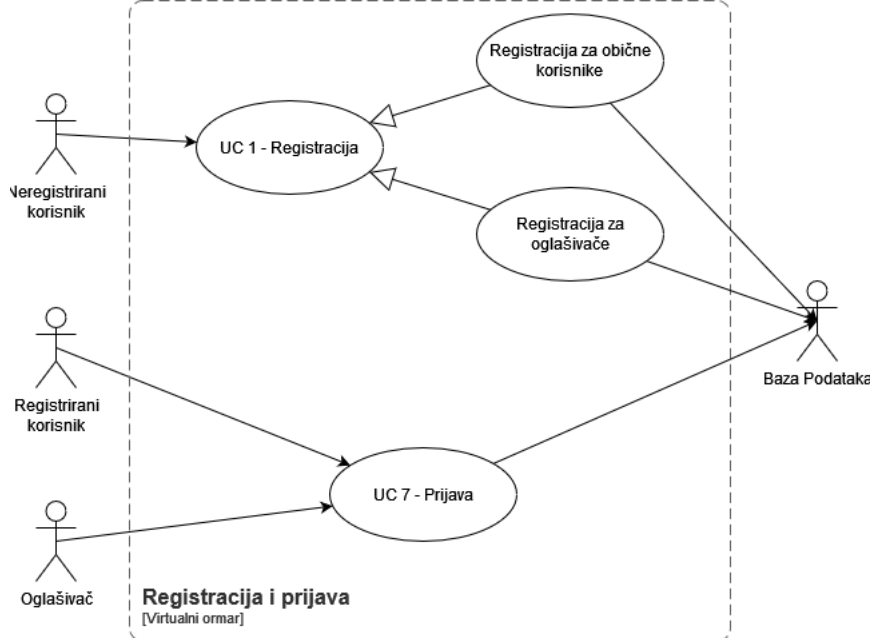
Popis dionika:

1. Neregistrirani korisnik
2. Registrirani korisnik
3. Oglašivač
4. Sustav

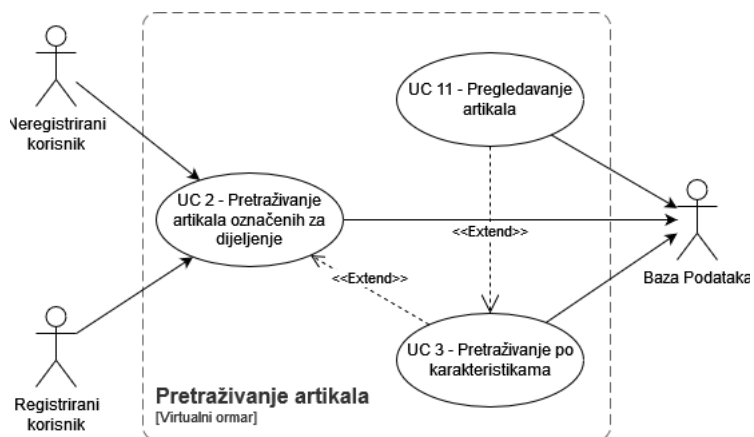
3. Specifikacija zahtjeva sustava

Dijagram obrazaca uporabe

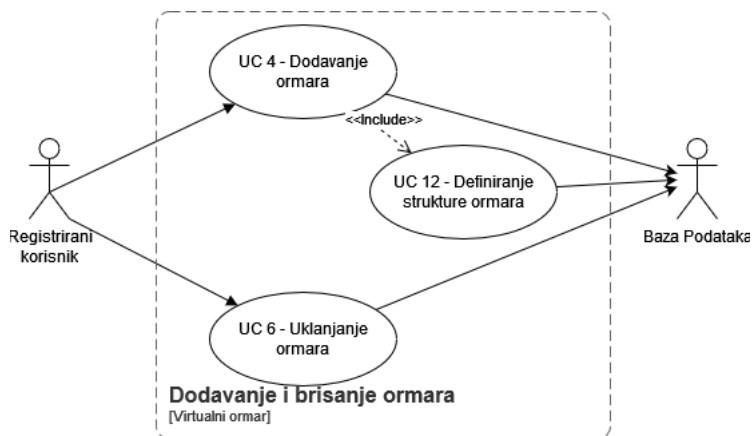
Registracija i prijava



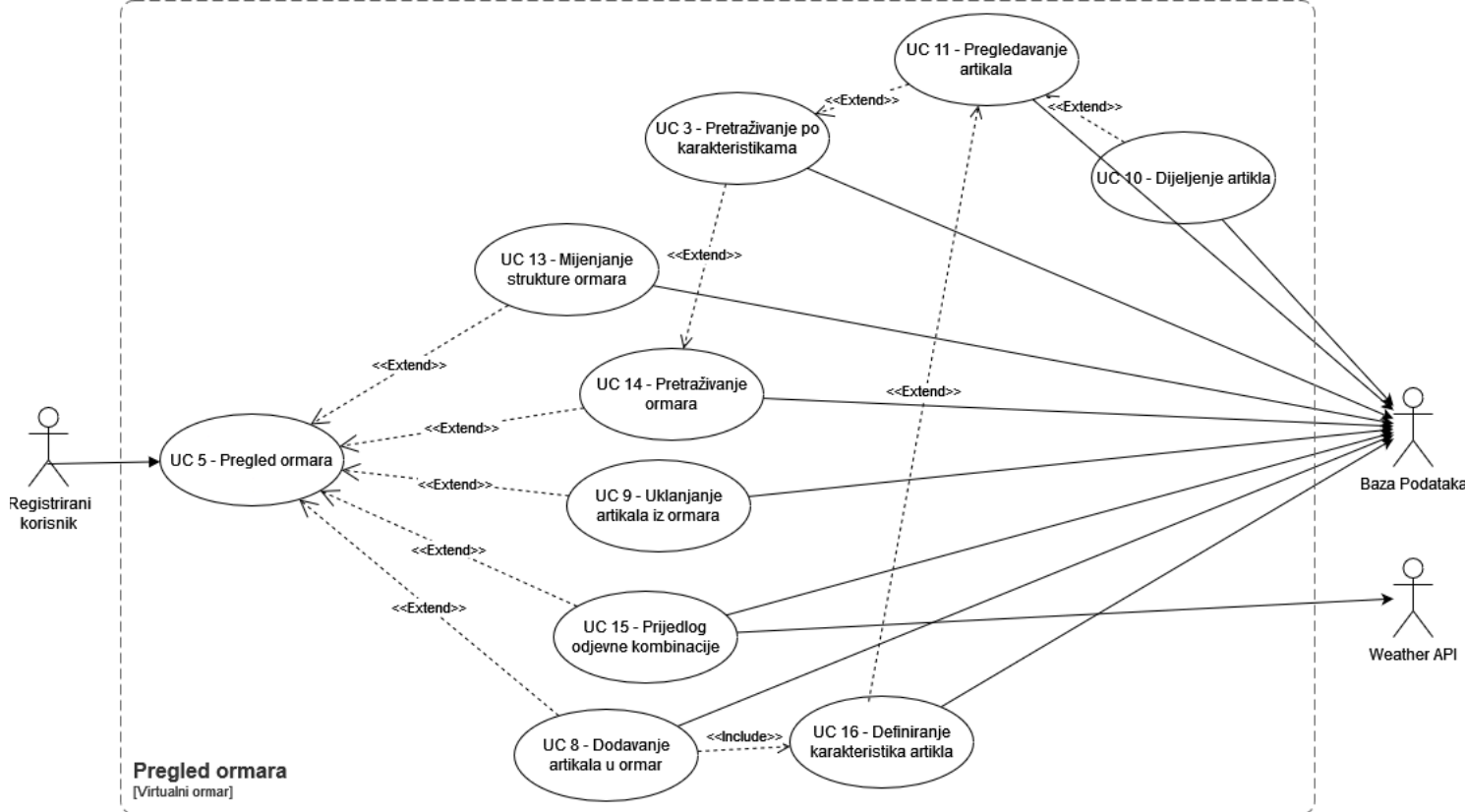
## Pretraživanje artikala



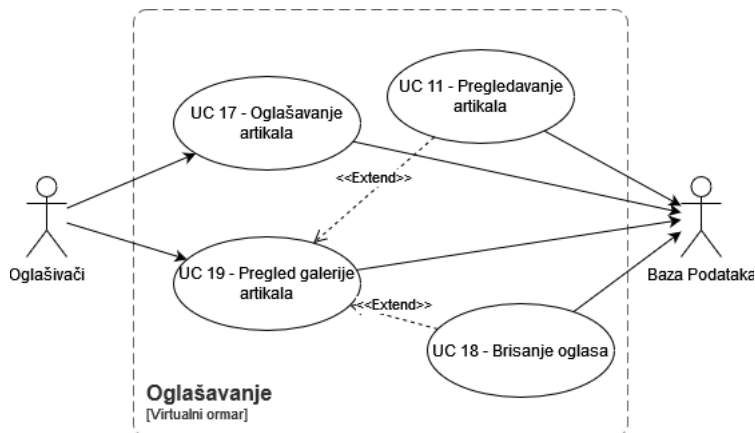
## Dodavanje i brisanje ormara



## Pregled ormara



## Oglašavanje



### UC 1 - Registracija

- Glavni sudionik: Neregistrirani korisnik
- Cilj: Registracija korisnika
- Sudionici: Neregistrirani korisnik
- Preduvjet: Korisnik nije već registriran
- Opis osnovnog tijeka:
  1. Korisnik pritisne gumb za registraciju
  2. Korisnik unosi osobne podatke
  3. Provjerava se jesu li podaci ispravni i unikatni
  4. Korisnik postaje registriran

### UC 2 - Pretraživanje artikala označenih za dijeljenje

- Glavni sudionik: Neregistrirani i registrirani korisnik
- Cilj: Pretraživanje svih artikala koji su postavljeni za dijeljenje od strane drugih korisnika
- Sudionici: Neregistrirani i registrirani korisnik
- Preduvjet:
- Opis osnovnog tijeka:
  1. Prikaz svih artikala na početnoj stranici

### UC 3 - Pretraživanje po karakteristikama

- Glavni sudionik: Neregistrirani i registrirani korisnik
- Cilj: Pretraživanje svih artikala po zadanim kriterijima
- Sudionici: Neregistrirani i registrirani korisnik
- Preduvjet:
- Opis osnovnog tijeka:

1. Unos kriterija za pretraživanje
2. Prikaz svih artikala koji zadovoljavaju kriterije

#### **UC 4 - Dodavanje ormara**

- Glavni sudionik: Registrirani korisnik
- Cilj: Dodavanje novog virtualnog ormara na profil korisnika
- Sudionici: Registrirani korisnik
- Preduvjet: Korisnik je prijavljen
- Opis osnovnog tijeka:

1. Korisnik stisne gumb za dodaju novog ormara
2. Korisnik definira izgled ormara
3. Ormar se prikazuje korisniku

#### **UC 5 - Pregled ormara**

- Glavni sudionik: Registrirani korisnik
- Cilj: Pregled ormara iz profila korisnika
- Sudionici: Registrirani korisnik
- Preduvjet: Korisnik je prijavljen, korisnik ima barem jedan ormar
- Opis osnovnog tijeka:

1. Korisnik stisne na željeni ormar
2. Korisniku se prikazuje ormar s svim njegovim artiklima

#### **UC 6 - Uklanjanje ormara**

- Glavni sudionik: Registrirani korisnik
- Cilj: Brisanje ormara s profila korisnika
- Sudionici: Registrirani korisnik
- Preduvjet: Korisnik je prijavljen, korisnik ima barem jedan ormar
- Opis osnovnog tijeka:

1. Korisnik odabere ormar
2. Korisnik stisne gumb za brisanje ormara
3. Korisnik potvrđuje
4. Ormar nestaje s profila korisnika

#### **UC 7 - Prijava**

- Glavni sudionik: Registrirani korisnik, oglašivač
- Cilj: Prijavljivanje korisnika na svoj profil
- Sudionici: Registrirani korisnik, oglašivač
- Preduvjet: Korisnik posjeduje profil
- Opis osnovnog tijeka:

1. Korisnik stišće na gumb prijave
2. Korisnik unosi podatke
3. Podaci se provjeravaju u bazi podataka
4. Ako ispravni, korisnik postaje prijavljen

#### **UC 8 - Dodavanje artikala u ormar**

- Glavni sudionik: Registrirani korisnik
- Cilj: Dodavanje artikala u ormar korisnika
- Sudionici: Registrirani korisnik
- Preduvjet: Korisnik je prijavljen, korisnik ima barem jedan ormar
- Opis osnovnog tijeka:

1. Korisnik stišće gumb za dodavanje artikla
2. Korisnik unosi sve karakteristike artikla
3. Artikl dodan u ormar

#### **UC 9 - Uklanjanje artikla iz ormara**

- Glavni sudionik: Registrirani korisnik
- Cilj: Uklanjanje artikla iz ormara korisnika
- Sudionici: Registrirani korisnik
- Preduvjet: Korisnik je prijavljen, korisnik ima barem jedan ormar i barem jedan artikl u njemu
- Opis osnovnog tijeka:

1. Korisnik stišće gumb za brisanje artikla
2. Korisnik potvrđuje da želi pobrisati artikl
3. Artikl nestaje iz ormara

#### **UC 10 - Dijeljenje artikla**

- Glavni sudionik: Registrirani korisnik
- Cilj: Postavljanje korisnikovog artikla da bude vidljiv svim korisnicima
- Sudionici: Registrirani korisnik
- Preduvjet: Korisnik je prijavljen, korisnik ima barem jedan ormar i barem jedan artikl u njemu
- Opis osnovnog tijeka:

1. Korisnik stišće na artikl
2. Odabire opciju za dijeljenje artikla
3. Korisnik potvrđuje

4. Artikl postaje javan za gledanje

UC 11 - Pregledavanje artikala

- Glavni sudionik: Neregistrirani korisnik, registrirani korisnik, oglašivač
- Cilj: Pregledati karakteristike artikla
- Sudionici: Neregistrirani korisnik, registrirani korisnik, oglašivač, baza podataka
- Preduvjet: /
- Opis osnovnog tijeka:

1. Korisnik stišće na artikl
2. Prikazuju se podaci o artiklu

UC 12 - Definiranje strukture ormara

- Glavni sudionik: Registrirani korisnik
- Cilj: Definiranje raspodjele prostora u ormaru
- Sudionici: Registrirani korisnik, baza podataka
- Preduvjet: Korisnik je prijavljen, korisnik dodaje novi ormar
- Opis osnovnog tijeka:

1. Korisnik odabire lokacije koje će biti u ormaru
2. Kada je zadovoljan, korisnik pritišće Done gumb

UC 13 - Mijenjanje strukture ormara

- Glavni sudionik: Registrirani korisnik
- Cilj: Mijenjaje raspodjele prostora u ormaru
- Sudionici: Registrirani korisnik, baza podataka
- Preduvjet: Korisnik je prijavljen, korisnik posjeduje barem jedan ormar
- Opis osnovnog tijeka:

1. Korisnik odabire gumb za izmjenu ormara
2. Korisnik odabire lokacije koje će biti u ormaru
3. Kada je zadovoljan, korisnik pritišće Done gumb

UC 14 - Pretraživanje ormara

- Glavni sudionik: Registrirani korisnik
- Cilj: Pretraživanje ormara korisnika za artikle
- Sudionici: Registrirani korisnik, baza podataka
- Preduvjet: Korisnik je prijavljen, korisnik posjeduje barem jedan ormar
- Opis osnovnog tijeka:

1. Korisnik odabire ormar
2. Korisnik upisuje riječi koje opisuju artikl kojeg traži
3. Traženi artikl se pokazuje korisniku

UC 15 - Prijedlog odjevne kombinacije

- Glavni sudionik: Registrirani korisnik
- Cilj: Aplikacija korisniku predlaže odjevnu kombinaciju na temelju odabranog ormara te vremenskih uvjeta
- Sudionici: Registrirani korisnik, baza podataka, weather API
- Preduvjet: Korisnik je prijavljen, korisnik posjeduje barem jedan ormar
- Opis osnovnog tijeka:

1. Korisnik odabire ormar
2. Korisnik pritišće gumb za prijedlog odjevne kombinacije
3. Aplikacija na temelju trenutnih vremenskih uvjeta korisniku pokazuje prikladnu odjevnu kombinaciju

UC 16 - Definiranje karakteristika artikla

- Glavni sudionik: Registrirani korisnik, oglašivač
- Cilj: Definiranje karakteristika artikla
- Sudionici: Registrirani korisnik, oglašivač, baza podataka
- Preduvjet: Korisnik je prijavljen, korisnik posjeduje barem jedan artikl
- Opis osnovnog tijeka:

1. Korisnik definira karakteristike odabranog artikla
2. Jednom kada je gotov, korisnik pritišće Done gumb

UC 17 - Oglašavanje artikala

- Glavni sudionik: Oglašivač
- Cilj: Staviti artikl za oglašivanje u galeriju oglašivača
- Sudionici: Oglašivač, baza podataka
- Preduvjet: Korisnik je prijavljen kao oglašivač
- Opis osnovnog tijeka:

1. Korisnik pritišće gumb za oglašavanje artikla
2. Korisnik definira sve potrebne karakteristike artikla
3. Korisnik označava da je završio definiranje artikla
4. Oglas je postavljen

UC 18 - Brisanje oglasa

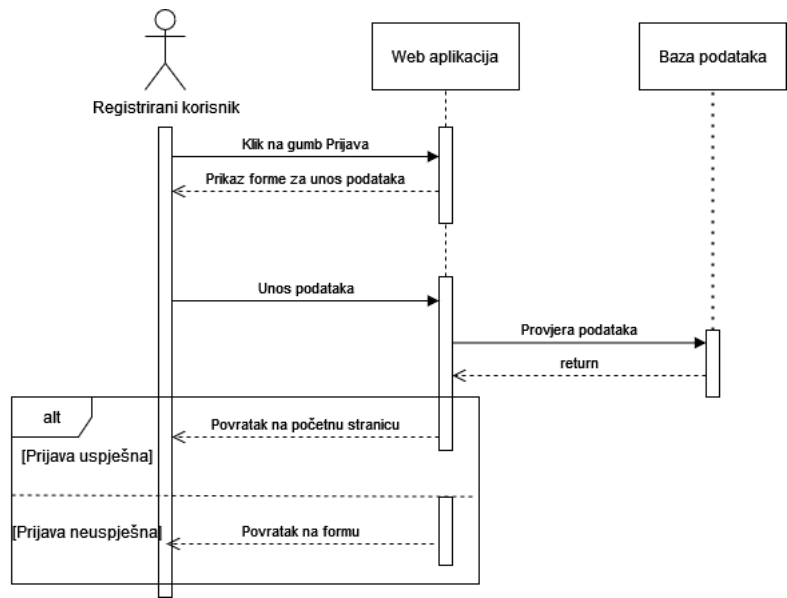
- Glavni sudionik: Oglašivač
- Cilj: Pobrisati oglas iz galerije oglašivača
- Sudionici: Oglašivač, baza podataka
- Preduvjet: Korisnik je prijavljen kao oglašivač, korisnik ima barem jedan oglas postavljen
- Opis osnovnog tijeka:
  - Korisnik odabire oglas
  - Korisnik pritišće gumb za brisanje oglasa
  - Oglas je pobrisan

### UC 19 - Pregled galerije artikala

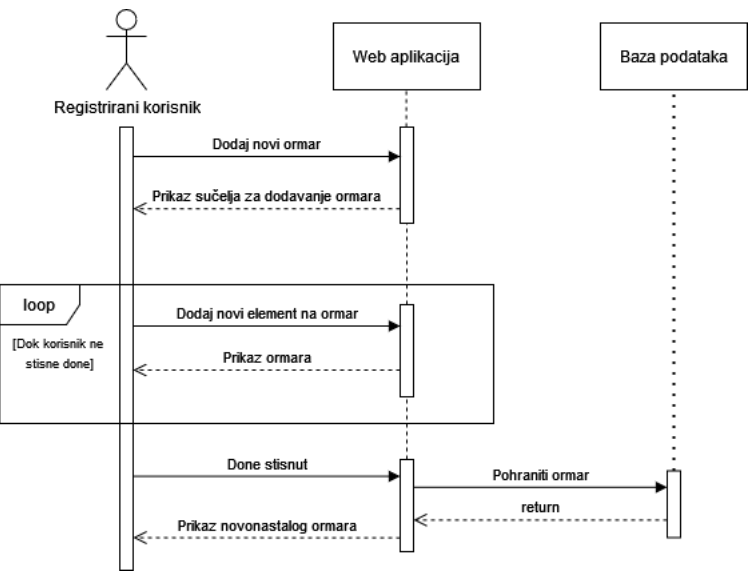
- Glavni sudionik: Oglašivač
- Cilj: Pregled galerije postavljenih oglasa
- Sudionici: Oglašivač, baza podataka
- Preduvjet: Korisnik je prijavljen kao oglašivač
- Opis osnovnog tijeka:
  - Korisnik odabire galeriju na svom profilu
  - Galerija artikala se prikazuje korisniku

## Sekvencijski dijagrami

### Prijava registriranog korisnika



### Dodavanje ormara



## Uključenost ključnih funkcionalnosti u obrasce uporabe

Broj obrasca uporabe	Opis obrasca uporabe	ID zahtjeva	Opis zahtjeva
UC1	Registracija	F-001	Mogućnost unosa podataka za registraciju novog korisnika.

UC2	Pretraživanje artikala označenih za dijeljenje	F-004	Korisnik može pretraživati sve artikle drugih korisnika koji su postavljeni za dijeljenje od strane drugih korisnika.
UC3	Pretraživanje po karakteristikama	F-005	Korisnik može unijeti kriterije za pretraživanje (npr. veličina, boja, vrsta artikla)
UC4	Dodavanje ormara	F-006	Mogućnost dodavanja i stvaranja virtualnog ormara.
UC5	Pregled ormara	F-006	Mogućnost pregleda virtualnog ormara.
UC6	Uklanjanje ormara	F-006	Mogućnost uklanjanja virtualnog ormara.
UC7	Prijava	F-002	Mogućnost unosa podataka za prijavu.
UC8	Dodavanje artikala u ormar	F-007	Mogućnost dodavanja vlastitih artikala u virtualni ormar.
UC9	Uklanjanje artikla iz ormara	F-007	Mogućnost uklanjanja artikala iz virtualnog ormara.
UC10	Dijeljenje artikla	F-007	Mogućnost dijeljenja artikal s drugim korisnicima.
UC11	Pregledavanje artikala	F-008	Korisnik može odabrati postojeći artikal i pregledati njegove karakteristike.
UC12	Definiranje strukture ormara	F-009	Korisnik može definirati strukturu ormara.
UC13	Mijenjanje strukture ormara	F-009	Korisnik može mijenjati strukturu ormara.
UC14	Pretraživanje ormara	F-006, F-010	Mogućnost pretraživanja vlastitih ormara i artikala u njemu.
UC15	Prijedlog odjevne kombinacije	F-011	Korisnik može zatražiti prijedlog odjevne kombinacije prema zadanim kriterijima i vremenskim uvjetima.
UC16	Definiranje karakteristika artikla	F-012	Korisnik može dodati svoj artikal te definirati njegove karakteristike.
UC17	Oglašavanje artikala	F-013	Oglašivač može dodati artikal u svoju galeriju oglašivača, mogu se oglašavati na profilima registriranih korisnika.
UC18	Brisanje oglasa	F-013	Oglašivač može obrisati svoj oglas.
UC19	Pregled galerije artikala	F-014	Oglašivač može pregledavati galeriju postavljenih oglasa.

- Arhitektura i dizajn sustava

## Arhitektura sustava

### Opis arhitekture

#### Stil arhitekture

Projekt se temelji na klijent-poslužiteljskoj arhitekturi koja odvaja korisnički dio (frontend) od poslovne logike i pristupa podacima (backend). Frontend aplikacija razvijena je u Reactu, s podrškom za CSS, Bootstrap i React-Bootstrap kako bi osigurala intuitivno korisničko sučelje. Backend, izrađen u Javi uz pomoć Spring frameworka, upravlja poslovnom logikom, sigurnošću i komunikacijom s bazom podataka. Ova arhitektura omogućava visok stupanj skalabilnosti kako bi frontend i backend mogu evoluirati odvojeno, čineći sustav prilagodljivijim za buduće optimizacije.

#### Podsustavi

Sustav je podijeljen na tri ključna podsustava, od kojih svaki ima svoju specifičnu ulogu u osiguravanju rada aplikacije:

- Frontend:** ovaj dio sustava razvijen je koristeći React, CSS, Bootstrap i React-Bootstrap kako bi pružio responzivno korisničko sučelje. Frontend je odgovoran za prikaz svih vizualnih elemenata i omogućuje korisnicima interakciju s aplikacijom putem preglednika. Korištenje Reacta omogućava brzi render i dinamičko ažuriranje korisničkog sučelja bez potrebe za ponovnim učitavanjem stranica, čime se ostvaruje interaktivno iskustvo.
- Backend:** odgovoran je za poslovnu logiku sustava, obradu zahtjeva i interakciju s bazom podataka. Izgrađen uz Java Spring framework, backend koristi Spring Data JPA za upravljanje podacima u bazi. Backend obrađuje zahtjeve od frontenda, izvršava potrebne operacije i vraća odgovore, čime omogućuje pravilnu funkcionalnost sustava.
- Baza podataka:** za pohranu podataka koristi se PostgreSQL. PostgreSQL je odabran zbog svoje sposobnosti upravljanja složenim strukturama podataka, visoke sigurnosti i podrške za ACID transakcije, što osigurava integritet podataka. Osim toga, omogućava efikasno izvođenje složenih SQL upita, podržava JSON podatke te se može skalirati kako bi podržao rastuće potrebe sustava.

#### Spremišta podataka

Za pohranu podataka koristi se baza podataka PostgreSQL. Odabrana je zbog svoje pouzdanosti i značajki upravljanja podacima. Baza podataka koristi standardizirane SQL upite za upravljanje podacima, što omogućava fleksibilnost u obavljanju složenih upita i operacija. Za administraciju i upravljanje bazom podataka koristi se PgAdmin jer omogućuje jednostavno upravljanje strukturama podataka, nadzor rada baze te izvođenje različitih administrativnih zadataka.

#### Mrežni protokoli

Za komunikaciju između klijentske i serverske strane koristit će se HTTP protokol koji omogućuje jednostavan prijenos podataka i osigurava stabilnost pri rukovanju velikim brojem zahtjeva.

#### Globalni upravljački tok

- Početna interakcija:** korisnik započinje interakciju putem frontend sučelja, pokrećući HTTP zahtjeve za komunikaciju s backendom kroz REST API endpointove, npr. ArticleController koji omogućuje manipulaciju artikulima putem različitih ruta.
- Obrada podataka i povratna informacija:** backend upravlja cijelim procesom obrade, pohranjuje podatke u bazu i vraća informacije frontend aplikaciji. Modeli podataka, poput entiteta Ormar, koriste JPA anotacije za povezivanje s PostgreSQL bazom podataka. ArticleRepository omogućuje pristup podacima, dok ArticleService obrađuje poslovnu logiku za unose, brisanje i dohvaćanje podataka. Nakon što backend obradi zahtjev, generira odgovor koji frontend koristi za ažuriranje korisničkog sučelja, npr. prikaz artikala.
- Konačni prikaz korisniku:** frontend koristi primljene podatke za ažuriranje korisničkog prikaza ili prikazuje odgovarajuće poruke o statusu zahtjeva.

#### Sklopovskoprogramski zahtjevi

Aplikacija je dizajnirana za rad na svim modernim operacijskim sustavima (Windows, macOS, Linux) i zahtjeva konfiguracije koje su sukladne osnovnim potrebama operacijskog sustava kojeg pojedino računalo koristi. Za pregled aplikacije potreban je web preglednik (Google Chrome, Mozilla Firefox, Safari, Microsoft Edge ili drugi). Također, potrebna je stabilna internetska veza za nesmetano korištenje aplikacije, posebno za učitavanje slika i pretraživanje podataka u stvarnom vremenu.

### Obrazloženje odabira arhitekture

Kod definiranja arhitekture sustava, u obzir smo uzeli nekoliko ključnih načela:

- Modularna struktura sustava:** omogućuje jednostavno proširenje i prilagodbu novim zahtjevima, omogućavajući dodavanje novih značajki bez utjecaja na postojeće komponente.



- **Visoka kohezija:** svaki dio sustava ima točno određenu ulogu — frontend je zadužen za korisničko sučelje i interakcije, backend obrađuje poslovnu logiku, dok se baza podataka koristi za sigurnu pohranu i pristup podacima.
- **Dostupnost i otpornost:** arhitektura sustava dizajnirana je za podršku više korisnika istovremeno, minimizirajući mogućnost zastoja i osiguravajući pouzdanost aplikacije.

## Organizacija sustava na visokoj razini

### 1. Klijent - poslužitelj

- sustav je organiziran prema klijent-poslužitelj arhitekturi, što omogućava fleksibilnost, modularnost i skalabilnost
- responzivna web aplikacija koju klijent koristi za unos, pregled i upravljanje podacima (npr. virtualnim ormarima, artiklima..)
- poslužitelj je backend dio sustava odgovoran je za obradu zahtjeva, autentifikaciju korisnika, logiku aplikacije i komunikaciju s bazom podataka

### 2. Baza podataka

- koristi se za pohranu svih relevantnih podataka i upravljanje informacijama o korisnicima, artiklima i oglasima
- omogućava komunikaciju među korisnicima i oglašivačima

### 3. Grafičko sučelje

Aplikacija ima intuitivno grafičko sučelje koje omogućava:

- neregistriranim korisnicima: Pregled i pretraživanje dijeljenih artikala
- registriranim korisnicima: Upravljanje virtualnim ormarima, dodavanje artikala, prijedloge kombinacija i dijeljenje
- oglašivačima: Upravljanje galerijama i interakciju s korisnicima

## Organizacija aplikacije

### frontend

- odgovoran za korisničko sučelje i interakciju korisnika s aplikacijom
- funkcionalnosti: omogućuje registriranim i neregistriranim korisnicima pretraživanje i pregled artikala, sučelje za kreiranje virtualnih ormara, dodavanje artikala, uređivanje ormara i dijeljenje artikala, integracija s oglasima i interaktivni prikaz galerija oglašivača
- koristi RESTful API za slanje i dohvaćanje podataka od backenda putem HTTP(S) zahtjeva (npr. GET, POST, PUT, DELETE)

### backend

- upravlja poslovnom logikom, obradom zahtjeva, sigurnošću i komunikacijom s bazom podataka
- funkcionalnosti: autentifikacija i autorizacija, upravljanje podacima, upravljanje oglasima, notifikacije
- koristi MVC za odvajanje slojeva podataka, prezentacije i logike

### MVC arhitektura

Aplikacija koristi Model-View-Controller (MVC) arhitekturu kako bi se razdvojila prezentacija, poslovna logika i podaci, čime se povećava modularnost i održivost sustava.

#### 1.model

- predstavlja podatke i poslovnu logiku sustava
- spremanje, dohvaćanje i manipulacija podacima

#### 2. view

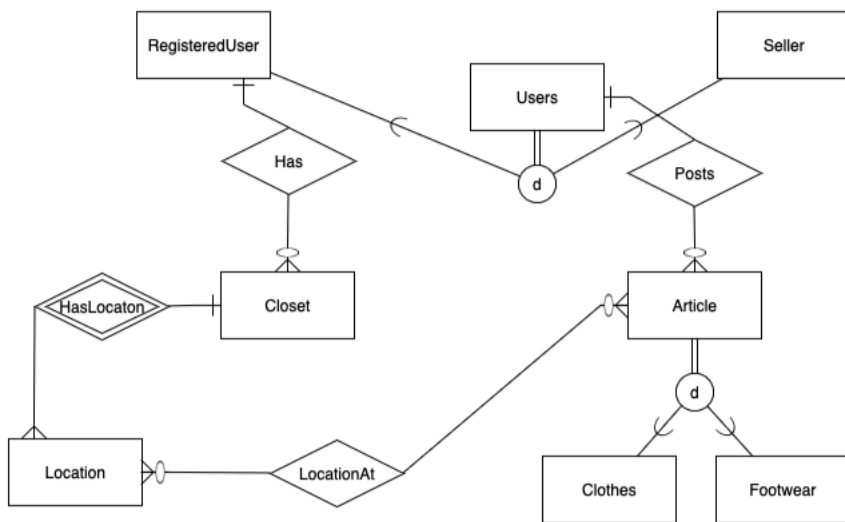
- korisnički sloj koji korisnici vide i koriste
- dohvaćanje podataka putem API-ja i prikaz u stvarnom vremenu
- pružanje vizualne povratne informacije korisnicima (npr. uspješna registracija ili greška pri unosu podataka)

#### 3. controller

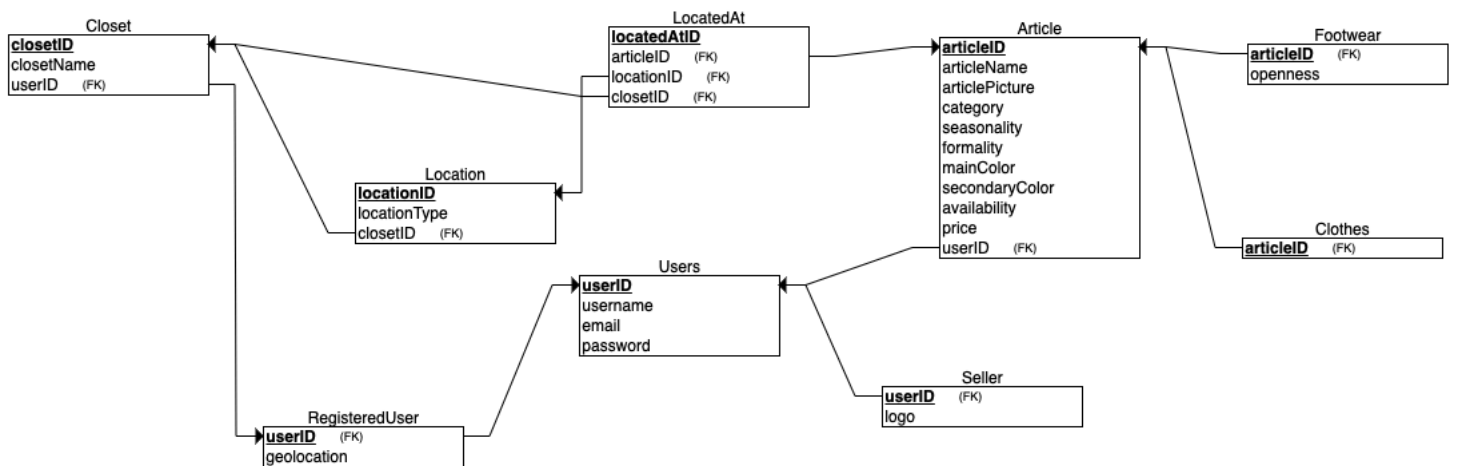
- posreduje između modela i pogleda, upravlja korisničkim unosima i poslovnom logikom
- obrada poslovnih pravila
- povezivanje s modelima za dohvaćanje ili spremanje podataka
- vraćanje odgovora frontendu

## Baza Podataka

### ER dijagram



## Relacijski dijagram



## Opis tablica

### Users

atribut	tip
userID	INT
userName	VARCHAR(200)
email	VARCHAR(200)
password	VARCHAR(50)

### Closet

atribut	tip
closetID	INT
closetName	VARCHAR(200)
userID	INT

### Registered User

atribut	tip
userID	INT
geolocation	VARCHAR(200)

### Seller

atribut	tip
userId	INT
logo	VARCHAR(200)

### Location

atribut	tip
locationID	INT
closetId	INT
locationType	VARCHAR(20)

Article

atribut	tip
articleId	INT
articleName	VARCHAR(250)
articlePicture	VARCHAR(250)
category	VARCHAR(50)
seasonality	VARCHAR(50)
formality	VARCHAR(50)
mainColor	VARCHAR(20)
secondaryColor	VARCHAR(20)
availabilty	VARCHAR(50)
price	NUMERIC(7, 2)
userId	INT

Footwear

atribut	tip
articleId	INT
openness	VARCHAR(50)

Clothes

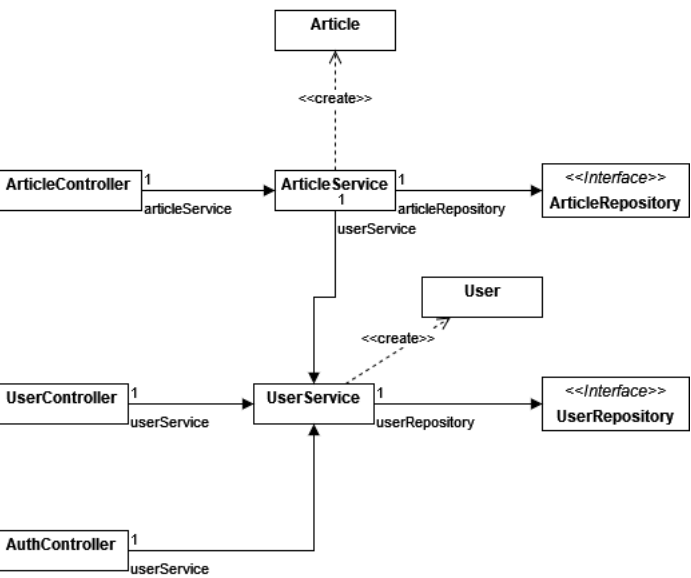
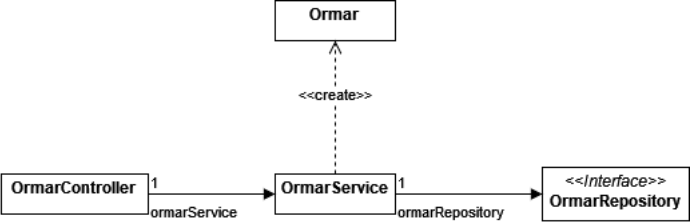
atribut	tip
articleId	INT

LocatedAt

atribut	tip
locatedAtID	INT
articleId	INT
locationID	INT
closetId	INT

Dijagram razreda

Generička funkcionalnost



Razredi

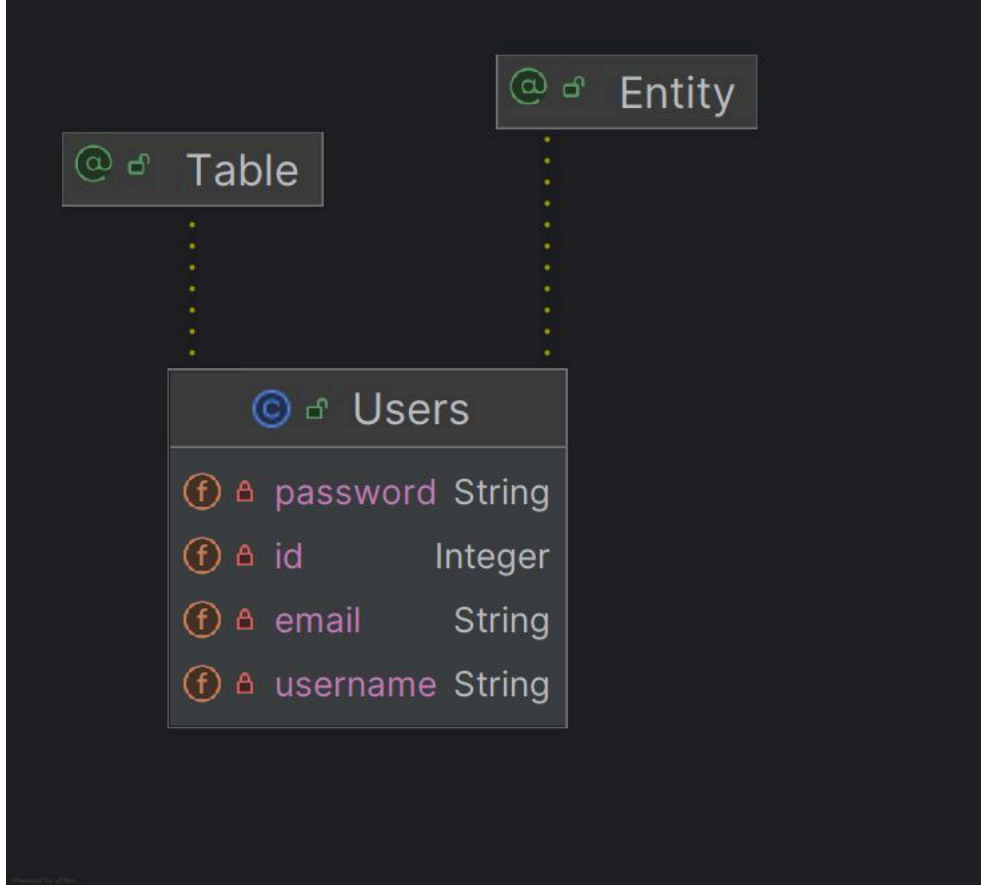
Article

@ Entity Table

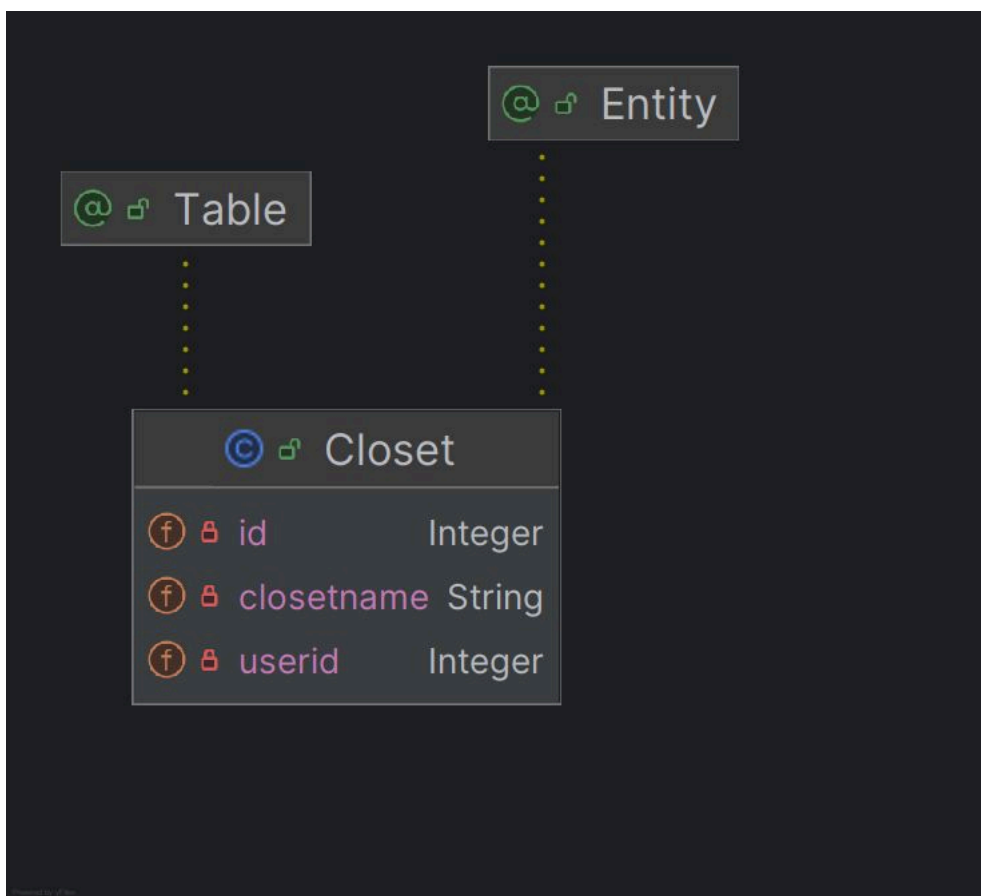
@ Entity Entity

Article		
Ⓜ Ⓜ	articlepicture	String
Ⓜ Ⓜ	id	Integer
Ⓜ Ⓜ	category	String
Ⓜ Ⓜ	maincolor	String
Ⓜ Ⓜ	formality	String
Ⓜ Ⓜ	price	BigDecimal
Ⓜ Ⓜ	userid	Integer
Ⓜ Ⓜ	articlename	String
Ⓜ Ⓜ	availability	String
Ⓜ Ⓜ	seasonality	String
Ⓜ Ⓜ	secondarycolor	String

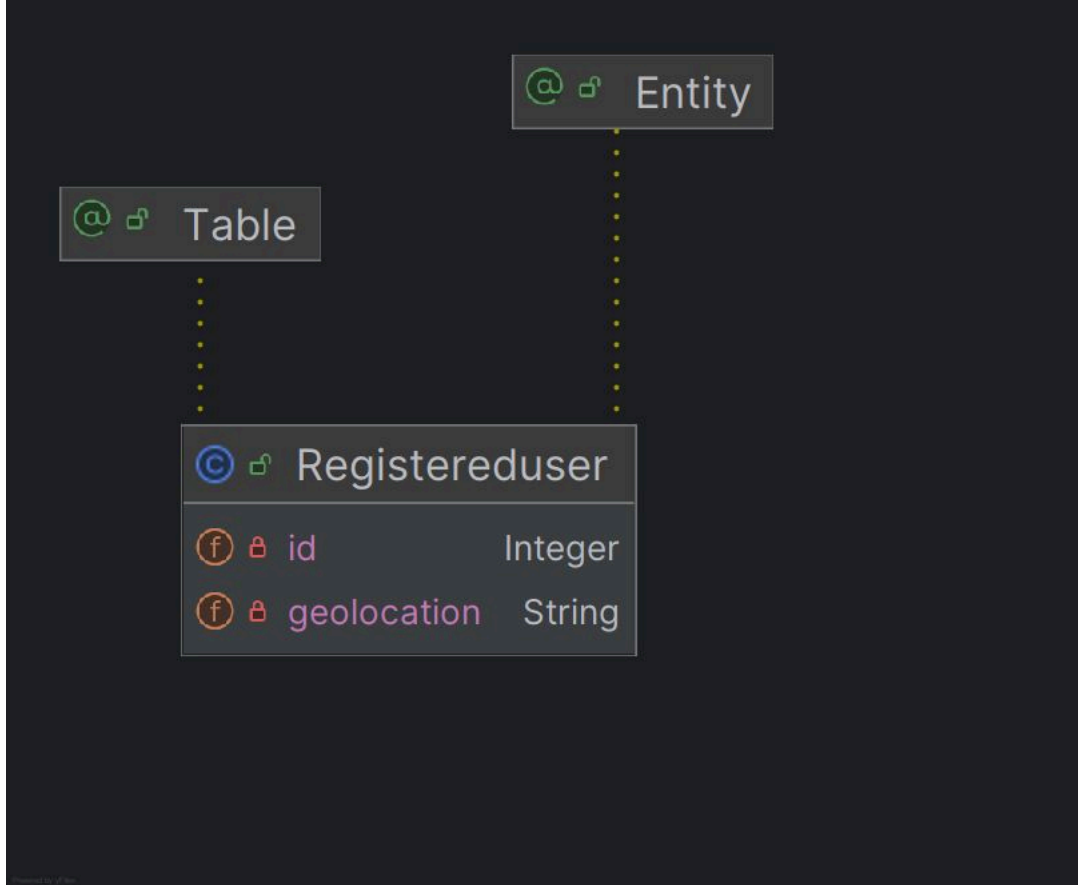
Users



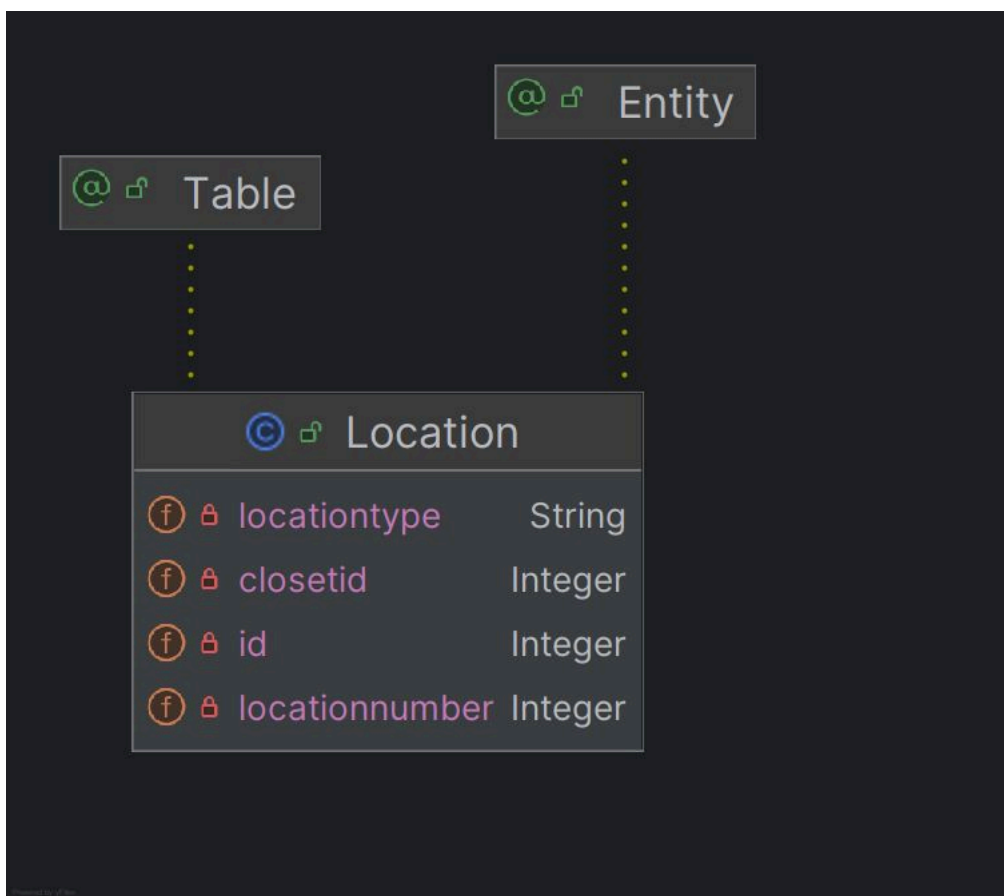
## Closet



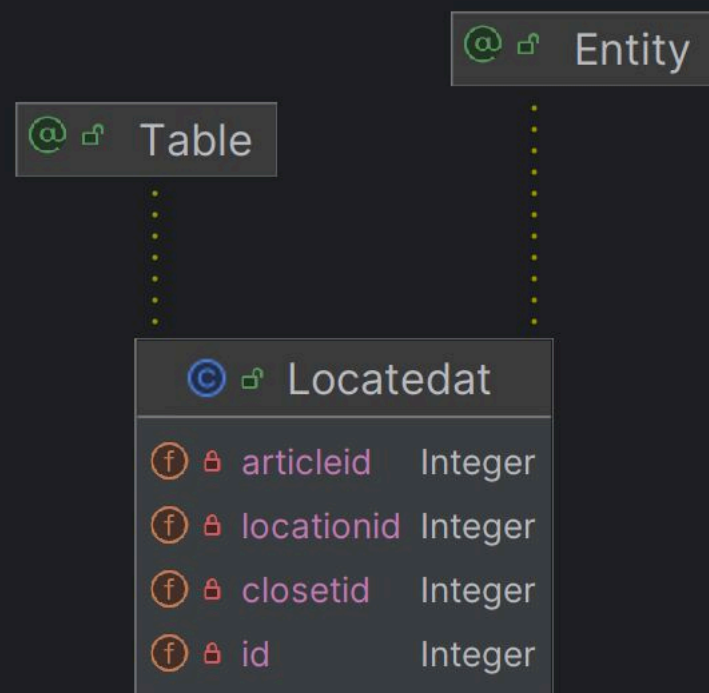
## RegisteredUser



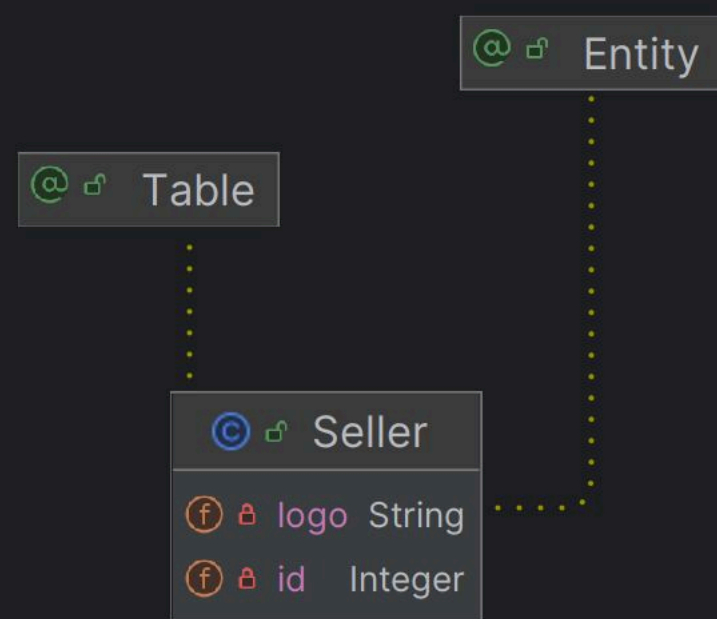
## Location



## LocationAt

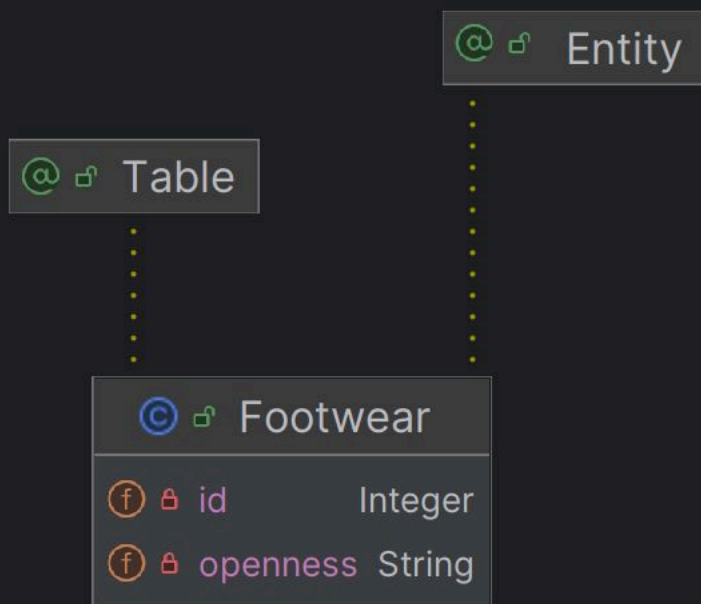


Seller

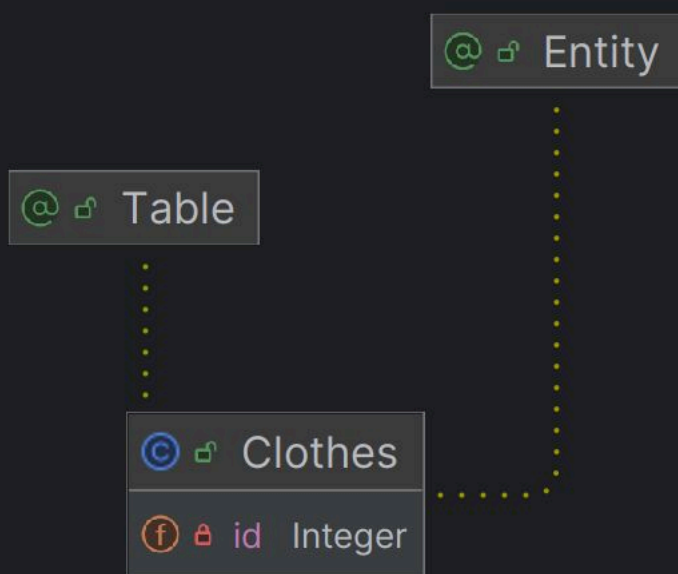


Footwear



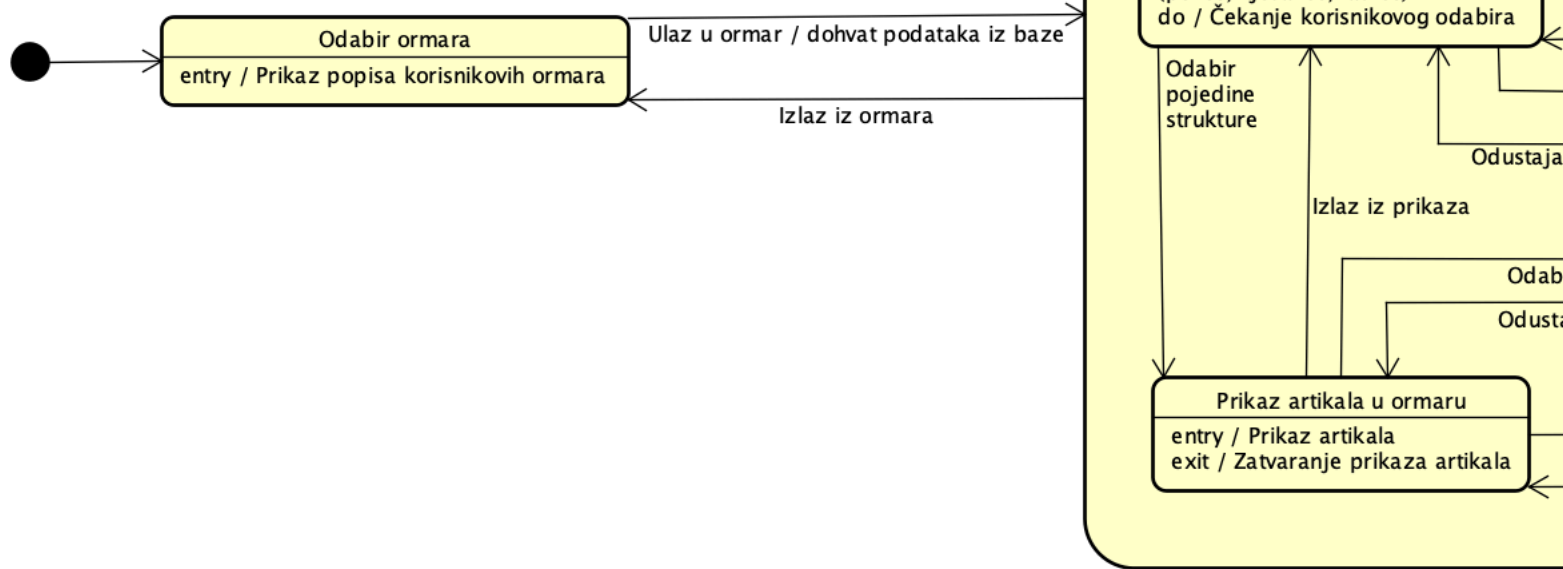


## Clothes

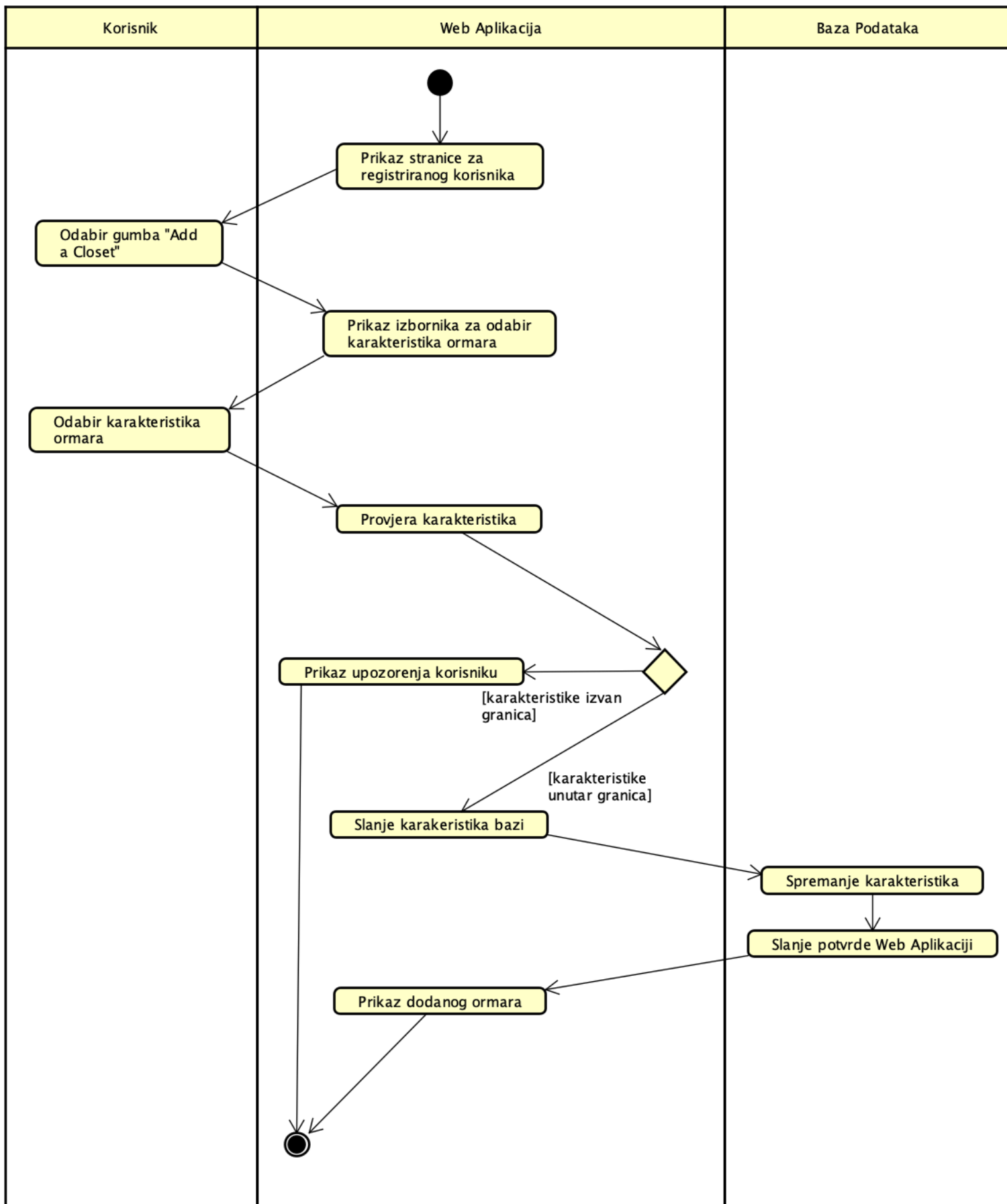


## Dinamičko ponašanje aplikacije

### UML dijagrami stanja



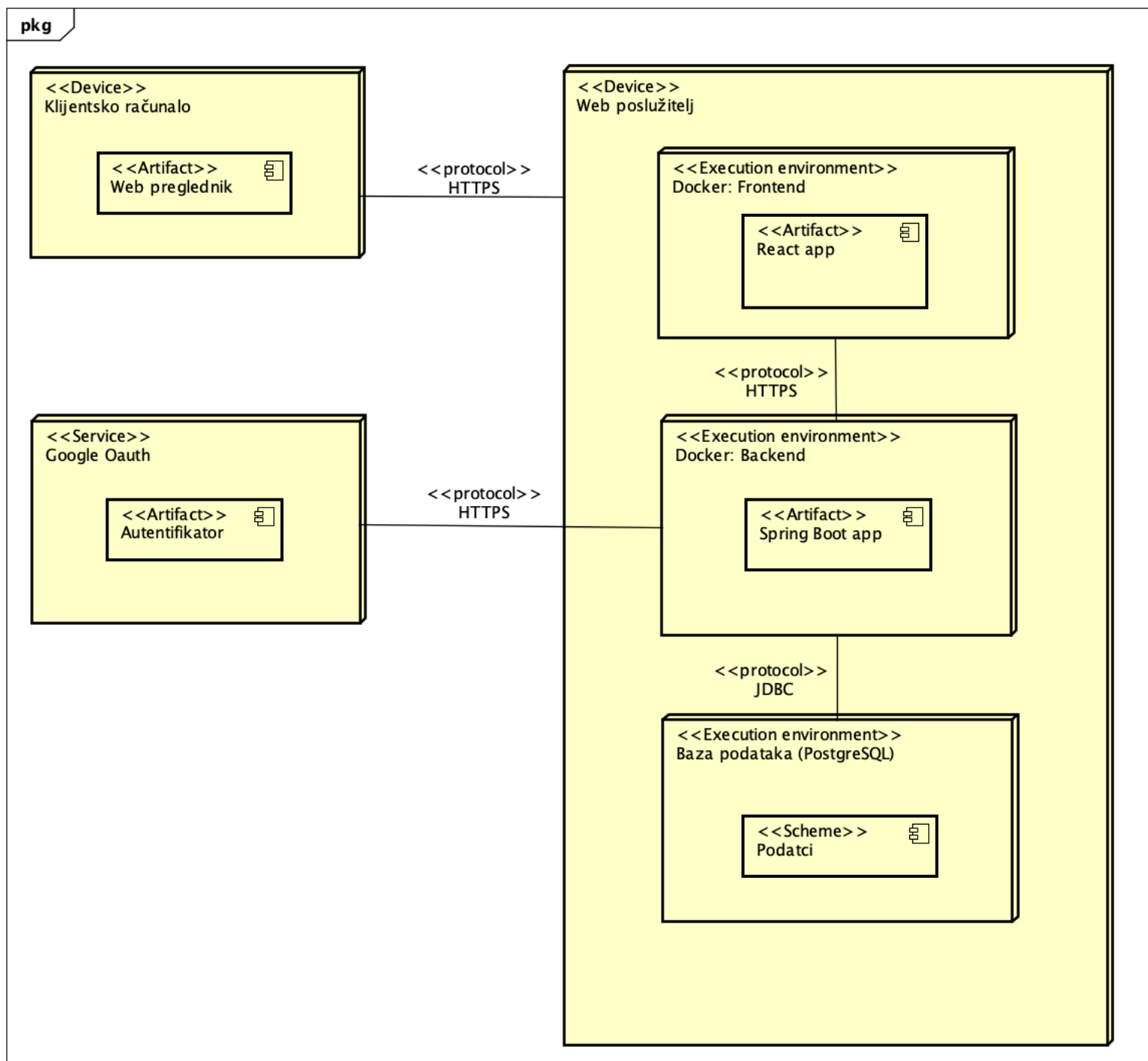
## UML dijagrami aktivnosti



## 5. Arhitektura komponenata i razmještaja

### Dijagram komponenti

# Dijagram razmještaja



## 6. Ispitivanje programskog rješenja

### Ispitivanje komponenti

Document s testovima: [testiranje.pdf](#)

## 7. Tehnologije za implementaciju aplikacije

### Korištene tehnologije i alati

#### Programski jezici:

- JavaScript: za razvoj klijentskog dijela (frontend)
- Java: za razvoj poslužiteljske strane (backend)

#### Radni okviri:

- React: za razvoj klijentskog dijela aplikacije
- CSS i Bootstrap: za stiliziranje korisničkog sučelja
- Spring Boot: za razvoj i upravljanje poslužiteljskim dijelom aplikacije

## Baza podataka:

- PgAdmin

## Razvojni alati:

- VS Code: glavni alat za razvoj u JavaScriptu i Reactu
- IntelliJ IDEA: koristi se za razvoj poslužiteljske strane u Javi

## Alati za ispitivanje

- JUnit: alat za ispitivanje komponenti i jedinica u Javi
- Selenium IDE: alat za automatizirano ispitivanje korisničkog sučelja i funkcionalnosti sustava

## Alati za razmještaj

- Docker: koristi se za razmještaj aplikacije

## Komunikacija:

- Discord

# 8. Upute za puštanje u pogon

## Upute

### 1. Instalacija

- Preduvjeti: Node.JS 20, JDK 21
- Preuzimanje:

```
git clone https://github.com/LukasMarovic/Slagaci-ormara
```

```
git switch develop
```

- Instalacija ovisnosti: frontend: npm i

```
backend: maven -> reload project
```

### 2. Postavke

#### Konfiguracijske datoteke:

- frontend: vite.config.js za automatsko preusmjeravanje API callova prema backendu u razvojnom okruženju

nginx.conf za konfiguriranje komunikacije između backenda i frontenda na deployanoj aplikaciji

- backend: .env za environment varijable u razvojnom okruženju, pri deploymentu se varijable zadaju na Render postavkama

application.properties za konfiguraciju postavka backenda

### 3. Pokretanje aplikacije

- Razvojno okruženje npm run dev

```
mvn spring-boot:run (ili kroz IDE)
```

- Provjera rada <http://localhost:5173> - frontend

<http://localhost:8080/api> - backend

### 4. Deployment platforma

Koristi se besplatna verzija Render platforme za deployment frontenda, backenda i baze.

- Priprema repozitorija frontend i backend imaju Dockerfileove u kojima je sve opisano za deployment. Koriste se yml skripte build-and-push-frontend.yml i build-and-push-backend.yml za automatsko redeployanje u slučaju novog pusha na granu. trigger\_redeploy.yml skripta se koristi za ručno redeployanje frontend i backenda.

Za redeployment skripte se koriste GitHub Actions, za koje se u postavkama repozitorija mora postaviti read and write permissions

- Postavljanje na Render Povezati github sa Renderom

Podesiti environment varijable i postaviti hookove za github actions

- Environment varijable BACKEND\_URL - za dohvaćanje backenda

# Opis prisutpa aplikaciji na javnom poslužitelju

- front-end: <https://closetly-721y.onrender.com/>
- back-end: <https://ormarbackend.onrender.com/api/>

## Zaključak i budući rad

### Zaključak

Naš ovogodišnji zadatak u sklopu kolegija „programsko inženjerstvo“ bio je napraviti „virtualni ormar“ tj. web aplikaciju pomoću koje korisnik može lakše organizirati svoju obuču i odjeću te ju također oglašavati ili dijeliti s drugima. Za savladavanje ovog zadatka oformljen je tim od sedam članova, a sami zadatak podijeljen na dvije faze rada.

U prvoj fazi nam je dan sami zadatak, pojašnjeno što se od nas očekuje i u kojem bismo smjeru trebali krenuti. Nakon što je tim bio formiran, bile su dodijeljene uloge svakom članu tima. Često smo imali sastanke gdje smo dogovarali raspodjelu posla, razgovore oko alata i razvojnih okolina koje ćemo koristiti te raspravljali o problemima (funkcionalnostima) koje moraju biti riješene do kraja prve faze. Na kraju prve faze dizajn stranice je bio napravljen, riješili smo svu potrebnu dokumentaciju za prvi dio zadatka i izradili smo i objavili prvu verziju web aplikacije koja je imala probu bazu podataka te nekolicinu funkcionalnosti.

U drugoj fazi broj sastanaka se povećao i počeli smo intenzivnije raditi na samom završavanju projekta. Baza je implementirana do kraja i popunjena, cijeli dizajni aplikacije je riješen, dokumentacija je bila dopunjena sa svime što smo napravili i mijenjali u drugoj fazi te smo dodali sve funkcionalnosti koje su u prvoj fazi nedostajale.

Sve u svemu, ovaj zadatak bio je jako zanimljivo i korisno iskustvo za sve nas jer smo dobili stvarni dojam tome kako raditi u timu, držati se rokova, kako raspodijeliti posao u timu te na kraju napraviti ciljani zadatak do kraja. Naravno, uvijek ima mjesta za napredak, ali smo zadovoljni aplikacijom koju smo napravili te vjerujemo da će nam stečeno iskustvo pomoći u radu na budućim projektima.

## Prikaz aktivnosti grupe

### Dnevnik sastajanja

#### 1. Sastanak

- Datum: 18. Listopada 2024.
- Prisustvovali: svi članovi
- Teme sastanka
  - raspodjela poslova
  - shvaćanje zadatka

#### 2. Sastanak

- Datum: 24. Listopada 2024.
- Prisustvovali: Toni Bajan, Maja Miličević, Lara Šimić, Dražen Pešut, Filip Vrbanac, Patrik Kapetan
- Teme sastanka
  - Razrada ideje za početnu stranicu

#### 3. Sastanak

- Datum: 30. Listopada 2024.
- Prisustvovali: svi članovi
- Teme sastanka
  - poboljšanje ideje za početni stranicu

#### 4. Sastanak

- Datum: 3. Studenog 2024.
- Prisustvovali: Lara Šimić, Maja Miličević, Lukas Marović
- Teme sastanka
  - osmišljavanje baze podataka

#### 5. Sastanak

- Datum: 4. Studenog 2024.
- Prisustvovali: svi članovi
- Teme sastanka
  - sklapanje svih dijelova u cjelinu
  - popravci završnih detalja

#### 6. Sastanak

- Datum: 5. Siječnja 2025.
- Prisustvovali: Filip Vrbanac, Patrik Kapetan, Dražen Pešut
- Teme sastanka

- stranica za registriranog korisnika

## 7. Sastanak

- Datum: 9. Siječnja 2025.
- Prisustvovali: Lara Šimić, Lukas Marović, Toni Bajan, Filip Vrbanac, Maja Miličević
- Teme sastanka

- općeniti dogovori za nastavak rada na projektu

## 8. Sastanak

- Datum: 12. Siječnja 2025.
- Prisustvovali: Lara Šimić, Lukas Marović, Toni Bajan, Filip Vrbanac, Dražen Pešut, Maja Miličević
- Teme sastanka

- frontend pokazao što su napravili
- razgovori o tome što još treba napraviti

## 9. Sastanak

- Datum: 13. Siječnja 2025.
- Prisustvovali: Dražen Pešut, Maja Miličević
- Teme sastanka

- usklađivanje filtera za pretraživanje s bazom

## 10. Sastanak

- Datum: 21. Siječnja 2025.
- Prisustvovali: svi
- Teme sastanka

- rješavanje problema na front-endu
- rješavanje problema na back-endu
- raspodjela ostatka posla

# Izvještaj rada članova

## Lukas Marović

Povezao back-end i front-end, deployao web aplikaciju na render, napravio wiki i doradio dokumentaciju, napravio workflowove za redeployment, dodao funkcionalnost za login/register i prikazivanje 8 nasumičnih artikala na backend, implementirao bazu u back-end. Dodao queryje za bazu kojom se koriste servisi i kontroleri. Dodao i podesio fetch requestove s front-enda na back-end radi komunikacije. Implementirao sessione za autentifikaciju.

## Toni Bajan

Detaljno proučio Spring Boot te napravio prvu inačicu koda za back-end na koju se kasnije nadograđivalo. Izrada Article i Ormar klase, postavljanje ArticleController-a, OrmarController-a i njihovih servisa. Implementirao bazu u back-end i prikazivanje 8 nasumičnih artikala

Rad na backendu. Stvaranje klase za svaku tablicu iz baze podataka, stvaranje servisa i kontrolera za skoro svaku klasu, uređivanje baze podataka da bolje podnosi edge caseove. Također sam napravio Scrapper klasu za povlačenje artikala odjeće i obuće sa interneta kako bi se dodali u bazu podataka, napravio UsersGenerator klasu koja generira 500 jedinstvenih korisnika, oglašivača i registriranih korisnika, također za popunjenje baze i napravio sam ClosetGenerator za stvaranje ormara za registrirane korisnike koji se također popune artiklima koji su dodani sa Scrapperom. Rad na testovima za Servise.

## Maja Miličević

Izrada baze u pgAdminu, izrada inicijalnih korisnika i oglašivača te njihovo dodavanje u bazu, osmišljavanje kategorije za pretraživanje, punjenje baze

Rad i popravci nad bazom podataka. Rjesavanje vecine dokumentacije i uređivanje wiki. Praćenje i bilježenje sastanaka. Ispravak ER dijagrama.

## Lara Šimić

Osmišljavanje baze podataka, izrada baze u pgAdminu, izrada inicijalnih korisnika i oglašivača te njihovo dodavanje u bazu i osmišljavanje kategorije za pretraživanje. Praćenje rada i sastanaka. Rad na opisu projekta i ER dijagramima.

Rad na većini dokumentacije i uređivanje wiki. Popravak početne dokumentacije, Kreiranje dijagrama.

## Dražen Pešut

Rad na front-endu: Izrada svih komponenti naslovne stranice osim advertiser i advertiser cards

Napravio stranicu za neregistriranog korisnika sve osim dijela za registraciju oglasivaca, stranica registriranog korisnika sve osim funkcionalnosti otvaranja ormara i dodavanja artikala, stranica za registriranog oglasivaca, stranica za rezultate pretrazivanje artikala, stranica za pregled oglasa od nekog oglasivaca pritiskom na logo.

## Filip Vrbanac

Rad na front-endu: Izrada advertiser i advertiser cards komponenta i stranice za login/register.

Na neregistriranoj stranici kreirao karticu za oglašivača. Na registriranoj stranici implementirao funkcionalnosti koje uključuju generiranje ormara, dodavanje artikala u ormare, pregled artikala unutar ormara te mogućnost izmjene postojećih artikala. Razvio stranice za login i registraciju korisnika.

Dizajn stranice za neregistrirane korisnike i stranice za registraciju i login, dokumentacija za arhitekturu sustava

Dizajn stranica za registriranog korisnika, oglašivača, općenitu pretragu artikala, pretragu oglašivača i pretragu artikala pojedinog oglašivaca. Dizajn prikaza ormara i kartica za artikle unutar ormara. Dokumentacija dijagrama za dinamičko ponašanje aplikacije.

## Literatura

1. Programski inženjerstvo, FER ZEMRIS: <https://www.fer.unizg.hr/predmet/proinz>
2. ChatGPT: <https://chatgpt.com>
3. Baze podataka, FER ZPR: <https://www.fer.unizg.hr/predmet/bazpod>
4. ERDPlus: <https://erdplus.com>
5. Spring Boot tutorial: [https://www.youtube.com/watch?v=QuvS\\_VLbGko&t=1771s](https://www.youtube.com/watch?v=QuvS_VLbGko&t=1771s)
6. Scrapper tutorial: <https://www.youtube.com/watch?v=wrFXBV4MwvI>
7. JUnit I Mockito tutorial: [https://www.youtube.com/watch?v=kXhYu939\\_5s](https://www.youtube.com/watch?v=kXhYu939_5s)
8. React Bootstrap dokumentacija: <https://react-bootstrap.github.io>
9. React tutoriali: <https://www.w3schools.com/react/>, <https://www.youtube.com/watch?v=CgkZ7MvWUAA>
10. CSS Bootstrap dokumentacija: <https://getbootstrap.com>
11. Definiranje principa dizajna projekta: <https://help.figma.com/hc/en-us/articles/14552740206743-Lesson-2-Define-your-design-system>
12. Izgradnja dizajna: <https://help.figma.com/hc/en-us/articles/14548865734679-Lesson-3-Build-your-design-system>