

Programsko inženjerstvo
Ak. god. 2020./2021.

Online galerija
Dokumentacija, Rev. 2

Grupa: *Raketa*
Voditelj: *Alen Kurtović*

Datum predaje: *14. siječnja. 2021.*

Nastavnik: *Eugen Vušak*

Sadržaj

1 Dnevnik promjena dokumentacije	3
2 Opis projektnog zadatka	5
3 Specifikacija programske potpore	8
3.1 Funkcionalni zahtjevi	8
3.1.1 Obrasci uporabe	10
3.1.2 Sekvencijski dijagrami	23
3.2 Ostali zahtjevi	28
4 Arhitektura i dizajn sustava	29
4.1 Baza podataka	30
4.1.1 Opis tablica	31
4.1.2 Dijagram baze podataka	36
4.2 Dijagram razreda	37
4.3 Dijagram stanja	39
4.4 Dijagram aktivnosti	41
4.5 Dijagram komponenti	43
5 Implementacija i korisničko sučelje	44
5.1 Korištene tehnologije i alati	44
5.2 Ispitivanje programskog rješenja	46
5.2.1 Ispitivanje komponenti	46
5.2.2 Ispitivanje sustava	50
5.3 Dijagram razmještaja	51
5.4 Upute za puštanje u pogon	52
6 Zaključak i budući rad	54
Popis literature	55
Indeks slika i dijagonama	57

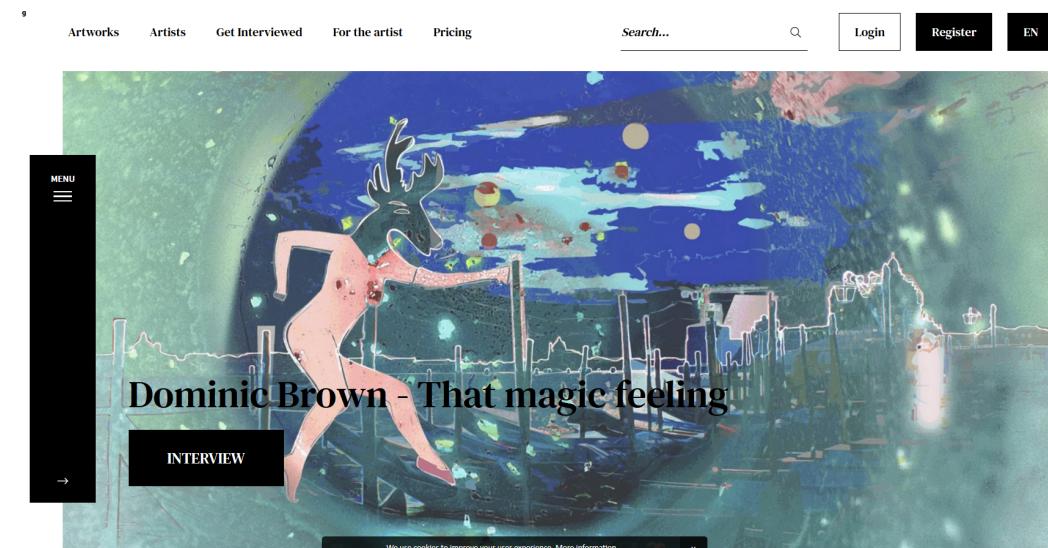
1. Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodataka	Autori	Datum
0.1	Napravljen predložak, dodan prvi dio opisa obrazaca uporabe.	Kurtović	21.10.2020.
0.2	Dodani aktori i njihovi funkcionalni zahtjevi.	Novinc	21.10.2020.
0.3	Dodan drugi dio opisa obrazaca uporabe.	Blagaić	21.10.2020.
0.4	Dodan treći dio opisa obrazaca uporabe.	Mahović	21.10.2020.
0.5	Dodan četvrti dio opisa obrazaca uporabe.	Šestak	21.10.2020.
0.5.1	Izmjene u obrascima uporabe.	Šestak	21.10.2020.
0.6	Dodan peti dio opisa obrazaca uporabe.	Ilić	21.10.2020.
0.7	Dodan šesti dio opisa obrazaca uporabe.	Zubčić	21.10.2020.
0.7.1	Ispravljene greške u obrascima uporabe.	Zubčić	21.10.2020.
0.8	Dodani dijagrami obrazaca uporabe i napravljene izmjene obrazaca.	Ilić	04.11.2020.
0.9	Dodan opis projektnog zadatka.	Šestak	06.11.2020.
0.9.1	Dodane slike u opis projektnog zadatka i ispravljanje pogrešaka.	Šestak	07.11.2020.
0.10	Dodani sekvencijski dijagrami i ostali zahtjevi.	Ilić	08.11.2020.
0.11	Font i tipfeleri	Šestak	08.11.2020
0.12	Dodana slika dijagrama Controller razreda	Zubčić	11.11.2020.
0.13	Dodana slika dijagrama DTO razreda	Novinc	11.11.2020.
0.14	Dodana relacijska shema baze podataka	Kurtović	12.11.2020.
0.15	Dodane tablice baze podataka	Šestak	12.11.2020.
0.16	Dodani opisi baze i njezinih tablica, dodan dio opisa arhitekture, ispravljeni tipfeleri	Kurtović	13.11.2020.
0.17	Dopunjjen opis arhitekture	Novinc	13.11.2020.
0.18	Dodan opis dijagrama razreda	Zubčić	13.11.2020.
1.0	Korigiranje teksta i provjera dokumentacije	Kurtović	13.11.2020.

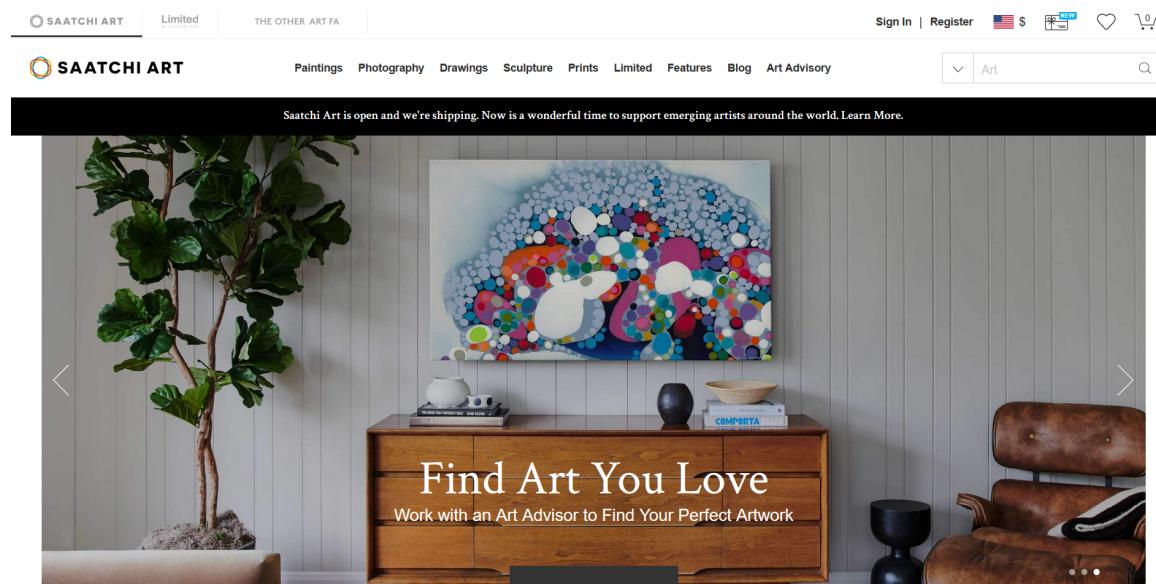
Rev.	Opis promjene/dodataka	Autori	Datum
1.1	Ispravljanje tipfelera, izmjena sekvencijskih i dijagrama obrazaca uporabe	Ilić	02.01.2021.
1.2	Dodani dijagrami aktivnosti i razmještaja s opisima	Ilić	03.01.2021.
1.3	Dodan dijagram stanja s opisima	Kurtović	03.01.2021.
1.4	Dodan opis tehnologija i alata	Kurtović	03.01.2021.
1.5	Dodan zaključak i budući rad	Šestak	12.01.2021.
1.6	Dodan opis ispitivanja komponenti	Zubčić	12.01.2021.
1.7	Dodan dijagram komponenti	Šestak	13.01.2021.
1.8	Dodane upute za puštanje u pogon	Ilić	13.01.2021.
1.9	Ispravljanje tipfelera, dodana literatura	Kurtović	14.01.2021.
1.9	Ažurirani dijagrami razreda	Zubčić	14.01.2021.
1.10	Ispunjeno dio tablice aktivnosti	Novinc	14.01.2021.
1.11	Dodan sastanak, ispunjena tablica aktivnosti do kraja	Šestak	14.01.2021.
1.12	Ažuriran dijagram stanja	Kurtović	14.01.2021.

2. Opis projektnog zadatka

Ovaj projekt će se baviti osmišljavanjem online galerije. Prilikom kupnje ili pregleda slika, zainteresirani ljudi najčešće moraju obići nekoliko različitih galerija te pogledati nekoliko izložbi. Također, najčešće ne mogu odmah kupiti sliku te sama kupnja nije baš jednostavna, pogotovo ako smo zainteresirani za sliku koja je na izložbi. U vrijeme pisanja ovog teksta, na snazi su mjere za suzbijanje virusa COVID-19 pa je ograničen broj ljudi u zatvorenim prostorima te je preporučeno držanje socijalne distance. Galerije su definitivno mjesta gdje se lako može skupiti veći broj ljudi na manjoj površini pa je online galerija ne samo praktična zamjena pravih galerija već i puno sigurnija opcija.



Slika 2.1: Snimka zaslona je slikana 7.11.2020. na stranici:
<https://onlinegallery.art/en/>



Slika 2.2: Snimka zaslona je slikana 7.11.2020. na stranici:
<https://www.saatchiart.com/>

Za razliku od sličnih stranica ovdje navedenih, ovaj projekt će biti baziran na prezentaciji izložbi u kojima se nalaze djela, a ne na prezentaciji samih djela. Također će projekt imati i komentare na pojedina djela što nije podržano na navedenim web stranicama. O samoj implementaciji tih stranica ne možemo govoriti bez da znamo koje su funkcionalnosti dane administratoru.

Online galerija namijenjena je široj publici ljubitelja likovnih djela, neovisno jesu li kupci istih ili samo posjetitelji izložbe.

Web stranici se pristupa sa jednom od četiri uloge: neregistrirani korisnik, umjetnik, registrirani posjetitelj ili administrator. Neregistrirani korisnik ima najmanje ovlasti te on može sortirati izložbe po određenim svojstvima te može vidjeti pregled aktivnih i nadolazećih izložba.

Na početnoj stranici se nalazi popis izložbi. Odabirom kategorije po kojoj se želi sortirati, djela se sortiraju.

Za registraciju u sustav korisniku su potrebni ime, prezime, email i PayPal račun.

Registracijom u sustav neregistrirani korisnik postaje registrirani posjetitelj te dobiva ovlasti za ulazak u izložbu te pregled, komentiranje i kupnju djela. On također može pregledavati vlastite izvršene transakcije.

Nakon ulaska u početnu stranicu, klikom na izložbu ulazi se u istu. Otvara se stra-

nica izložbe te se u njemu nalaze sva djela te izložbe.

Prelaskom miša preko slike djela, pojavljuje se slika košarica, a njenim odabirom otvara se stranica za plaćanje. Kupnja se zatim potvrđuje te se korisnik prijavljuje na PayPal račun te uplaćuje novac.

Klikom na djelo otvara se nova stranica djela u kojem je moguće komentirati.

Umjetnik prilikom registracije prilaže i svoj portfolio u .pdf formatu. On ima sve ovlasti registriranog posjetitelja, ali on može i dodavati, pregledavati i uređivati kolekcije radova na svojem računu te prijaviti jednu ili više svojih kolekcija na natječaj.

Administrator je uloga s najvišim ovlastima te on može stvarati natječaje, odlučivati o prijavi kolekcije na natječaj, stvarati izložbu, pregledavati sve izvršene transakcije, pregledavati i uklanjati sve račune, zatvarati izložbe te brisati komentare.

Iako je web stranica namjenjena galeriji, uz minimalne modifikacije stranica bi mogla služiti online dućan. U tom slučaju korisnici bi otvarali kategorije artikla primjerice kućanske potrepštine, hranu, piće itd... Nakon otvaranja kategorije pojavljuju se artikli koje je moguće dodati u košaricu. Komentari na artiklima bi bili mogući u smislu recenzije proizvoda. Jedine promjene na stranici bi trebale biti promjene naziva jer su sve ostale funkcije već implementirane.

U budućnosti bi se na stranici moglo dodati više različitih opcija plaćanja poput plaćanja bankovnim i kreditnim karticama te bi bilo korisno podržati mobilne korisnike razvojem mobilne stranice.

3. Specifikacija programske potpore

3.1 Funkcionalni zahtjevi

Dionici:

1. Neregistrirani korisnici
2. Umjetnici
3. Registrirani posjetitelji
4. Administratori

Aktori i njihovi funkcionalni zahtjevi:

1. Neregistrirani korisnik (inicijator) može:
 - (a) vidjeti aktivne i nadolazeće izložbe
 - (b) sortirati izložbe po različitim svojstvima
 - (c) se registrirati u sustav, stvoriti novi korisnicki račun za koji su mu potrebni ime, prezime, email i PayPal račun, a umjetnici dodatno prilažu i svoj portfolio u obliku PDF datoteke
2. Umjetnik (inicijator) može:
 - (a) vidjeti aktivne i nadolazeće izložbe
 - (b) sortirati izložbe po različitim svojstvima
 - (c) dodavati, pregledavati i uređivati kolekcije radova na svojem računu
 - (d) pregledavati aktivne natječaje
 - (e) prijaviti jednu ili više svojih kolekcija na natječaj čiji stilovi odgovaraju stilu natječaja
 - (f) ući u izložbu i pregledavati djela
 - i. komentirati djela
 - ii. kupiti neprodana djela
 - (g) pregledavati vlastite izvršene transakcije

3. Posjetitelj (inicijator) može:

- (a) vidjeti aktivne i nadolazeće izložbe
- (b) sortirati izložbe po različitim svojstvima
- (c) ući u izložbu i pregledavati djela
 - i. komentirati djela
 - ii. kupiti neprodana djela
- (d) pregledavati vlastite izvršene transakcije

4. Administrator (inicijator) može:

- (a) stvarati natječaje
- (b) vidjeti prijave kolekcija na natječaje te iste odbiti ili prihvati
- (c) stvoriti izložbu nakon odabiranja kolekcija
- (d) pregledavati sve izvršene transakcije
- (e) pregledavati i uklanjati sve račune
- (f) zatvarati izložbe prije završetka
- (g) brisati komentare

5. Baza podataka (sudionik):

- (a) pohranjuje sve podatke o korisnicima i njihovim ovlastima
- (b) pohranjuje sve podatke o natječajima, izložbama i kolekcijama

3.1.1 Obrasci uporabe

Opis obrazaca uporabe

UC1 - Registracija korisnika

- **Glavni sudionik:** Neregistrirani korisnik
- **Cilj:** Registrirati korisnika na platformu
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju za registraciju
 2. Korisnik unosi ime, prezime, e-mail adresu i PayPal račun
 3. Korisnik prima obavijest o uspješnoj registraciji
- **Opis mogućih odstupanja:**
 - 2.a Odabir već zauzetog korisničkog imena i/ili e-maila, unos korisničkog podatka u nedozvoljenom formatu, pružanje neispravnoga e-maila ili neispravnog PayPal računa
 1. Sustav obavještava korisnika o neuspjelom upisu i vraća ga na stranicu za registraciju
 2. Korisnik mijenja potrebne podatke te završava unos ili odustaje od registracije

UC2 - Registracija umjetnika

- **Glavni sudionik:** Neregistrirani korisnik
- **Cilj:** Registracija umjetnika na platformu
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju za registraciju
 2. Korisnik unosi ime, prezime, e-mail adresu, PayPal račun i portfolio u obliku PDF datoteke
 3. Korisnik prima obavijest o uspješnoj registraciji
- **Opis mogućih odstupanja:**
 - 2.a Odabir već zauzetog korisničkog imena i/ili e-maila, unos korisničkog podatka u nedozvoljenom formatu, pružanje neispravnoga e-maila ili neispravnog PayPal računa

1. Sustav obavještava korisnika o neuspjelom upisu i vraća ga na stranicu za registraciju
2. Korisnik mijenja potrebne podatke te završava unos ili odustaje od registracije

UC3 - Prijava u sustav

- **Glavni sudsionik:** Posjetitelj, umjetnik, administrator
- **Cilj:** Dobivanje pristupa korisničkom sučelju
- **Sudsionici:** Baza podataka
- **Preduvjet:** Registracija
- **Opis osnovnog tijeka:**
 1. Unos korisničkog imena i lozinke
 2. Potvrda o ispravnosti unesenih podataka
 3. Pritup korisničkim funkcijama
- **Opis mogućih odstupanja:**
 - 2.a Neispravno korisničko ime i/ili lozinka
 1. Sustav obavještava korisnika o neuspjelom upisu i vraća ga na stranicu za registraciju

UC4 - Pregled osobnih podataka

- **Glavni sudsionik:** Posjetitelj
- **Cilj:** Pregled osobnih podataka
- **Sudsionici:** Baza podataka
- **Preduvjet:** Prijava u sustav
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju "Osobni podaci"
 2. Aplikacija prikazuje osobne podatke korisnika

UC5 - Promjena osobnih podataka

- **Glavni sudsionik:** Posjetitelj
- **Cilj:** Promjena osobnih podataka
- **Sudsionici:** Baza podataka
- **Preduvjet:** Prijava u sustav
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju za promjenu osobnih podataka
 2. Korisnik mijenja podatke

3. Korisnik spremi promjene
 4. Baza podataka se ažurira
- **Opis mogućih odstupanja:**
 - 2.a Korisnik mijenja svoje osobne podatke, ali ne odabire opciju "Spremi promjene"
 1. Sustav obavještava korisnika da nije spremio podatke prije izlaska iz prozora

UC6 - Brisanje računa

- **Glavni sudionik:** Posjetitelj
- **Cilj:** Izbrisati korisnički račun
- **Sudionici:** Baza podataka
- **Preduvjet:** Prijava u sustav
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju "Izbriši račun"
 2. Korisnički račun se briše iz baze podataka
 3. Otvara se naslovna stranica

UC7 - Dodavanje kolekcija

- **Glavni sudionik:** Umjetnik
- **Cilj:** Dodavanje nove kolekcije na korisnikov račun
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav kao umjetnik
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju za dodavanje kolekcija
 2. Korisnik dodaje novu kolekciju na svoj račun
 3. korisnik spremi promjene
 4. Baza podataka se ažurira
- **Opis mogućih odstupanja:**
 - 2.a Korisnik dodaje novu kolekciju, ali ne odabere opciju "Spremi"
 1. Sustav obavještava korisnika da nije spremio podatke prije izlaska iz prozora
 - 2.b Pri stvaranju nove kolekcije korisnik ne unosi jedan od sljedećih podataka o kolekciji: naziv, opis, stil te jedno ili više djela u nekom slikovnom formatu

1. Sustav obavještava korisnika da određeni podaci o kolekciji nedostaju
2. Korisnik mijenja potrebne podatke te završava unos ili odustaje od dodavanja nove kolekcije

UC8- Pregled kolekcija

- **Glavni sudionik:** Umjetnik
- **Cilj:** Pregled kolekcije radova na svojem računu
- **Sudionici:** Baza podataka
- **Preduvjet:** Umjetnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Umjetnik odabire opciju "Moje kolekcije"
 2. Prikazuju se umjetnikove kolekcije radova

UC9 - Uređivanje kolekcija

- **Glavni sudionik:** Umjetnik
- **Cilj:** Uređivanje kolekcije radova na svojem računu
- **Sudionici:** Baza podataka
- **Preduvjet:** Umjetnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Umjetnik odabire opciju "Moje kolekcije"
 2. Prikazuju se umjetnikove kolekcije radova
 3. Umjetnik odabire kolekciju koju želi urediti
 4. Nakon uređivanja potvrđuje izmjenu
 5. Promjene se upisuju u bazu podataka

UC10 - Stvaranje natječaja

- **Glavni sudionik:** Administrator
- **Cilj:** Stvaranje natječaja za umjetnike
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je registriran i dodijeljena su mu prava administratora
- **Opis osnovnog tijeka:**
 1. Administrator odabire opciju stvaranja natječaja
 2. Unosi se minimalni skup informacija o natječaju
 3. Administrator odabire opciju "Spremi"
 4. Natječaj postaje vidljiv unutar aplikacije

UC11 - Pregled natječaja

- **Glavni sudionik:** Umjetnik, administrator
- **Cilj:** Pregled aktivnih natječaja
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen kao umjetnik ili su mu dodijeljena prava administratora
- **Opis osnovnog tijeka:**
 1. Umjetnik odabire opciju "Natječaji"
 2. Umjetniku se prikazuje lista svih aktivnih natječaja
 3. Umjetnik odabire natječaj
 4. Prikazuju se podatci o natječaju

UC12 - Prijava na natječaj

- **Glavni sudionik:** Umjetnik
- **Cilj:** Prijava umjetnika na natječaj
- **Sudionici:** Baza podataka
- **Preduvjet:** Umjetnik je prijavljen i postoji aktivni natječaj
- **Opis osnovnog tijeka:**
 1. Umjetnik odabire opciju "Aktivni natječaji"
 2. Prikazuje se lista trenutno aktivnih natječaja
 3. Umjetnik odabire natječaj na koji se želi prijaviti
 4. Nudi se lista umjetnikovih kolekcija čiji stil odgovara stilu traženom u natječaju
 5. Umjetnik odabire jednu ili više kolekcija
 6. Umjetnik odabire opciju "Potvrdi prijavu"
 7. Promjena se pohranjuje u bazu podataka
- **Opis mogućih odstupanja:**
 - 4.a Umjetnik nema nijednu kolekciju čiji stil odgovara stilu traženom u natječaju
 1. Sustav obavještava umjetnika da nema nijednu kolekciju u traženom stilu

UC13 - Pregled prijava

- **Glavni sudionik:** Administrator
- **Cilj:** Pregled prijave
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je registriran i dodijeljena su mu prava administratora
- **Opis osnovnog tijeka:**
 1. Administrator odabire opciju "Završeni natječaji"
 2. Prikazuje se lista završenih natječaja
 3. Administrator odabire natječaj za koji želi pregledati dobivene prijave
 4. Prikazuje se lista prijava

UC14 - Odabir kolekcija za izložbu

- **Glavni sudionik:** Administrator
- **Cilj:** Odabir kolekcije za izložbu
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je registriran i dodijeljena su mu prava administratora
- **Opis osnovnog tijeka:**
 1. Umjetnik odabire natječaj za koji želi razmotriti prijave
 2. Prikazuje se lista prijava
 3. Administrator pregledava prijavljene kolekcije umjetnika
 4. Administrator prihvata određenu kolekciju odabirom opcije "Prihvati kolekciju"
 5. Promjena se pohranjuje u bazu podataka

UC15 - Stvaranje izložbe

- **Glavni sudionik:** Administrator
- **Cilj:** Stvaranje izložbe
- **Sudionici:** Baza podataka
- **Preduvjet:**
 - Korisnik je registriran i dodijeljena su mu prava administratora
 - Administrator je odabrao kolekcije za izložbu
- **Opis osnovnog tijeka:**
 1. Administrator upisuje naziv i opis izložbe, vrijeme početka i trajanja, te iznos provizije
 2. Administrator stvara izložbu odabirom opcije "Stvori izložbu"

3. Promjena se pohranjuje u bazu podataka

UC16 - Pregled aktivnih i nadolazećih izložbi

- **Glavni sudionik:** Neregistrirani korisnik, posjetitelj
- **Cilj:** Pregled trenutno aktivnih i nadolazećih izložbi
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju "Izložbe"
 2. Prikaže se popis izložbi s osnovnim informacijama

UC17 - Ulazak u izložbu

- **Glavni sudionik:** Posjetitelj
- **Cilj:** Ulazak u odabranu izložbu
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik odabire izložbu
 2. Otvara se stranica izložbe

UC18 - Pregled odabranog djela i komentara

- **Glavni sudionik:** Posjetitelj
- **Cilj:** Pregled svih informacija i komentara o djelu
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju "Prikaži više o djelu"
 2. Otvara se stranica sa svim detaljima i komentarima o djelu

UC19 - Komentiranje djela

- **Glavni sudionik:** Posjetitelj
- **Cilj:** Komentiranje djela
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik odabire polje za komentare

2. Korisnik upisuje svoj komentar u polje
3. Korisnik potvrđuje svoj komentar
4. . Baza podataka se ažurira
5. . Korisnikov komentar se nalazi u polju pokraj slike

UC20 - Kupnja djela

- **Glavni sudionik:** Posjetitelj
- **Cilj:** Kupnja određenog djela
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik odabire kupnju djela
 2. Korisnik se prijavljuje na stranicu partnera (PayPal)
 3. Korisnik preko protokola za plaćanje uplaćuje sredstva
 4. Baza podataka se ažurira
 5. Pokreće se proces slanja slike
- **Opis mogućih odstupanja:**
 1. Korisnik zbog nekog razloga nije u mogućnosti prijaviti se na stranicu partnera
 2. Korisnik zbog nekog razloga nije u mogućnosti obaviti uplatu

UC21 - Pregled vlastitih izvršenih transakcija

- **Glavni sudionik:** Posjetitelj, umjetnik
- **Cilj:** Pregled svih vlastitih izvršenih transakcija
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik odabire popis svih transakcija
 2. Prikaže se lista svih izvršenih transakcija

UC22 - Pregled svih transakcija

- **Glavni sudionik:** Administrator
- **Cilj:** Pregled svih transakcija na stranici
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik ima prava administratora
- **Opis osnovnog tijeka:**

1. Korisnik odabire popis svih transakcija
2. Prikaže se lista svih transakcija

UC23 - Pregledavanje računa

- **Glavni sudionik:** Administrator
- **Cilj:** Pregled svih korisničkih računa
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je registriran i dodijeljena su mu prava administratora
- **Opis osnovnog tijeka:**
 1. Administrator odabire opciju pregledavanja korisnika
 2. Prikaže se lista svih ispravno registriranih korisnika s osobnim podacima

UC24 - Uklanjanje računa

- **Glavni sudionik:** Administrator
- **Cilj:** Uklanjanje korisničkog računa
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je registriran i dodijeljena su mu prava administratora
- **Opis osnovnog tijeka:**
 1. Administrator odabire opciju uklanjanja korisnika
 2. Administrator pronalazi željenog korisnika
 3. Administrator uklanja željenog korisnika i njegove podatke iz baze podataka

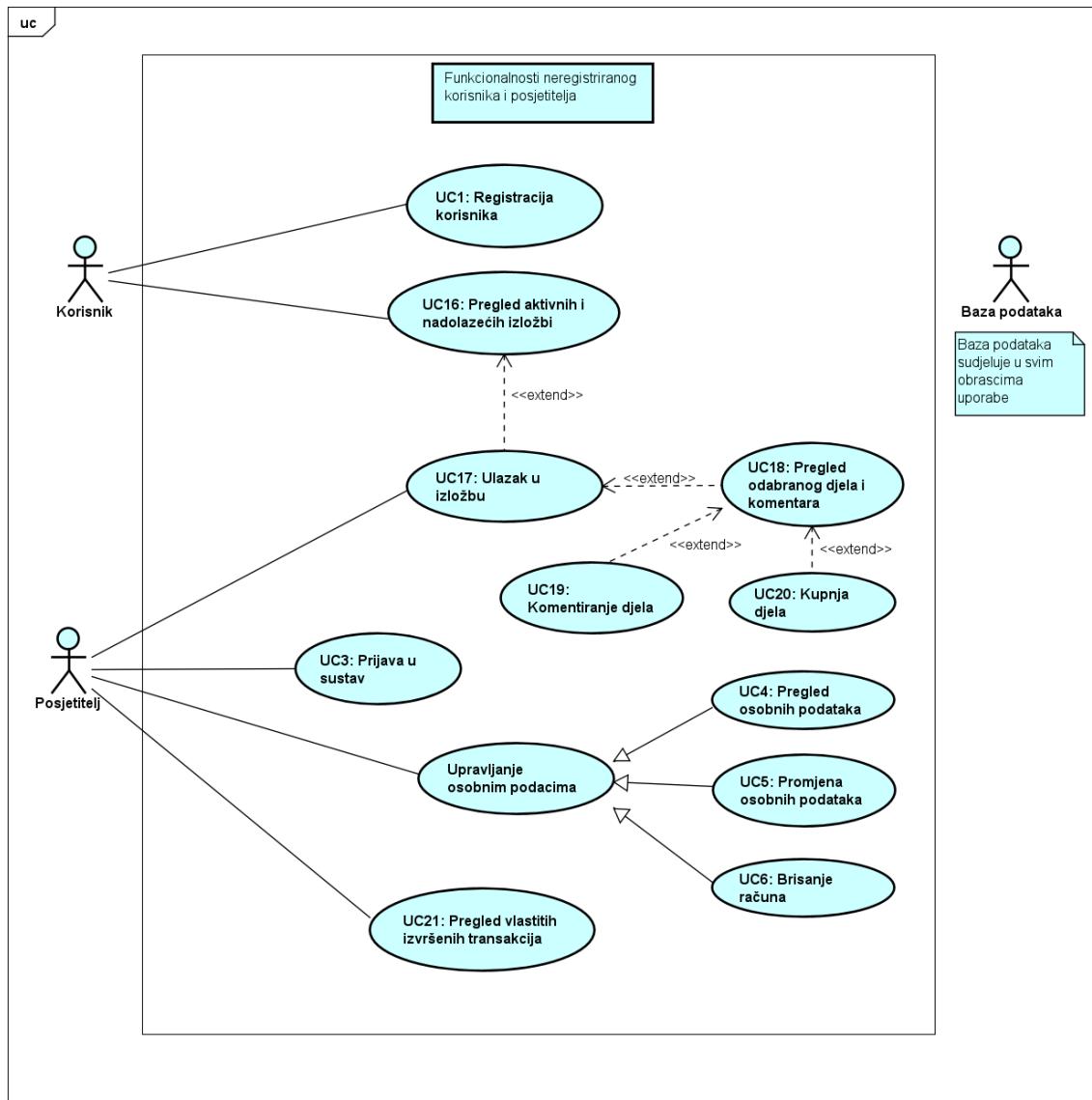
UC25 - Zatvaranje izložbi prije kraja

- **Glavni sudionik:** Administrator
- **Cilj:** Zatvaranje izložbe prije kraja
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je registriran i dodijeljena su mu prava administratora, izložba je trenutno aktivna
- **Opis osnovnog tijeka:**
 1. Administrator odabire opciji "Aktivne izložbe"
 2. Prikaže se lista trenutno aktivnih izložbi
 3. Odabire željenu izložbu
 4. Odabire opciju "Zatvoriti izložbu prije kraja"
 5. Promjene se upisuju u bazu podataka

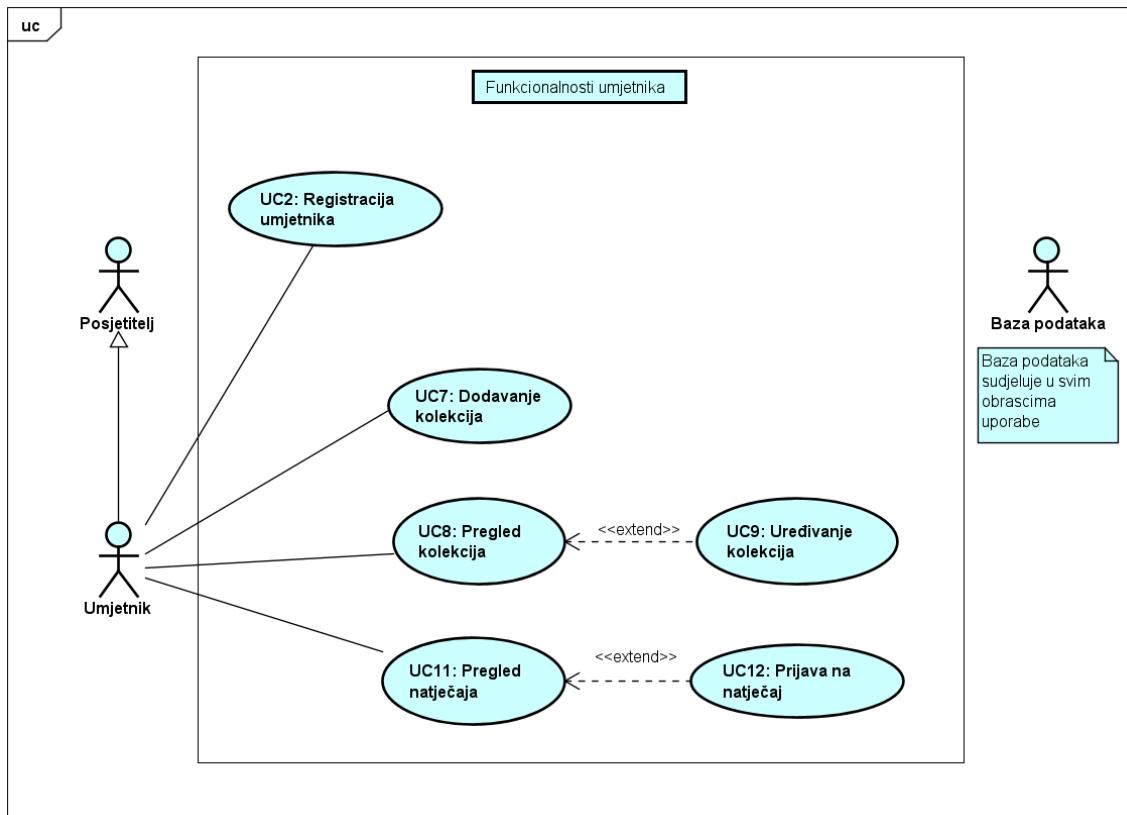
UC26 - Brisanje komentara

- **Glavni sudionik:** Administrator
- **Cilj:** Brisanje komentara
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je registriran, dodijeljena su mu prava administratora i postoji komentar
- **Opis osnovnog tijeka:**
 1. Administrator odabire komentar
 2. Odabire opciju "Izbriši"
 3. Komentar se uklanja iz baze podataka

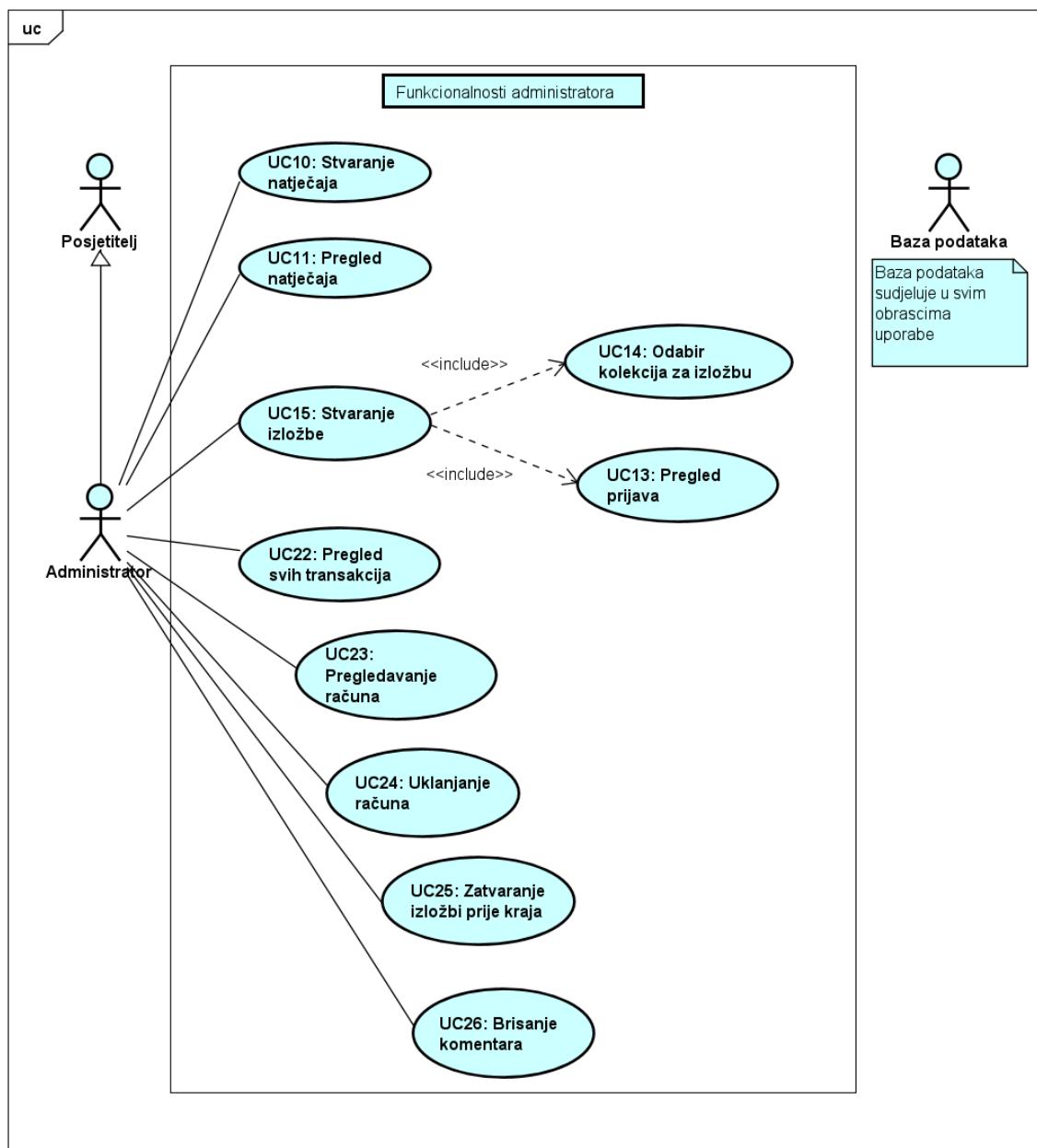
Dijagrami obrazaca uporabe



Slika 3.1: Dijagram obrazaca uporabe: funkcionalnosti korisnika i posjetitelja



Slika 3.2: Dijagram obrazaca uporabe: funkcionalnosti umjetnika

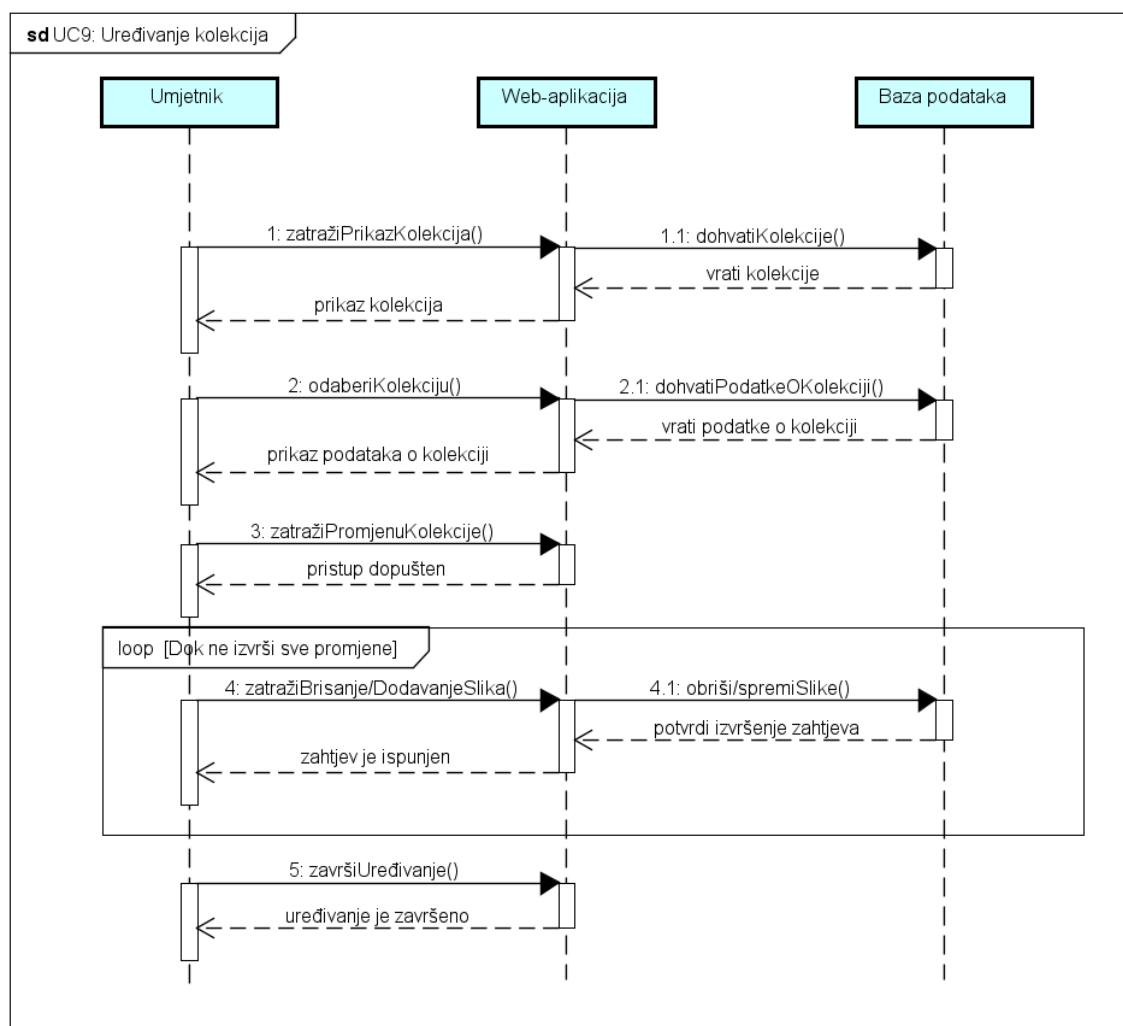


Slika 3.3: Dijagram obrazaca uporabe: funkcionalnosti administratora

3.1.2 Sekvencijski dijagrami

Obrazac uporabe UC9 - Uređivanje kolekcija

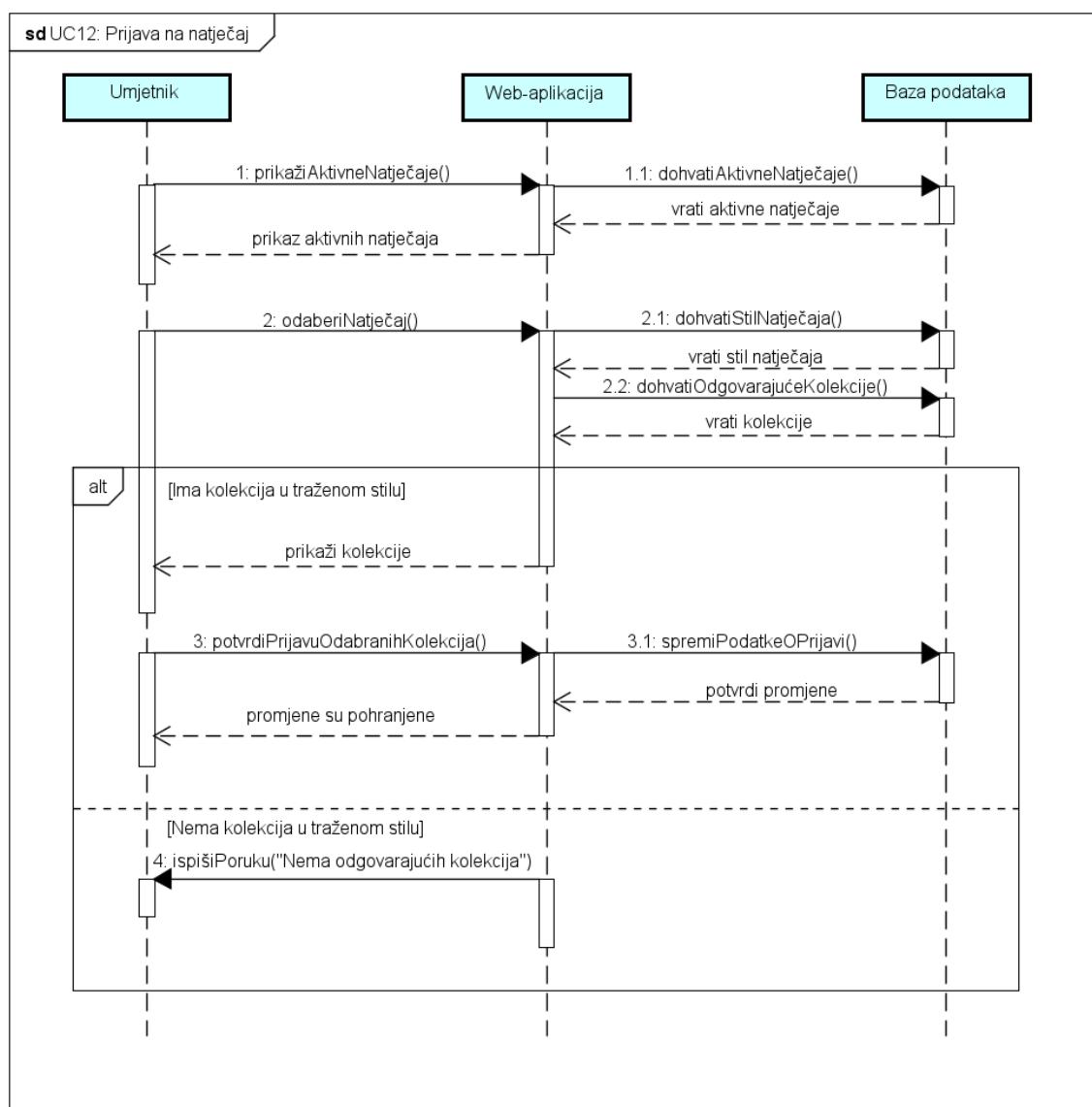
Umjetnik odabire opciju "Moje kolekcije". Poslužitelj dohvaća popis umjetnikovih kolekcija iz baze podataka te ih prikazuje umjetniku. Umjetnik odabire kolekciju koju želi urediti. Poslužitelj dohvaća podatke o kolekciji iz baze podataka i prikazuje ih umjetniku. Umjetnik šalje zahtjev za promjenu kolekcije. Poslužitelj mu dopušta izmjenu kolekcije. Umjetnik unosi nove podatke podatke o kolekciji, briše i\ili dodaje slike. Poslužitelj unosi izmjene u bazu podataka. Baza podataka potvrđuje promjenu poslužitelju, a poslužitelj umjetniku.



Slika 3.4: Sekvencijski dijagram za UC9

Obrazac uporabe UC12 - Prijava na natječaj

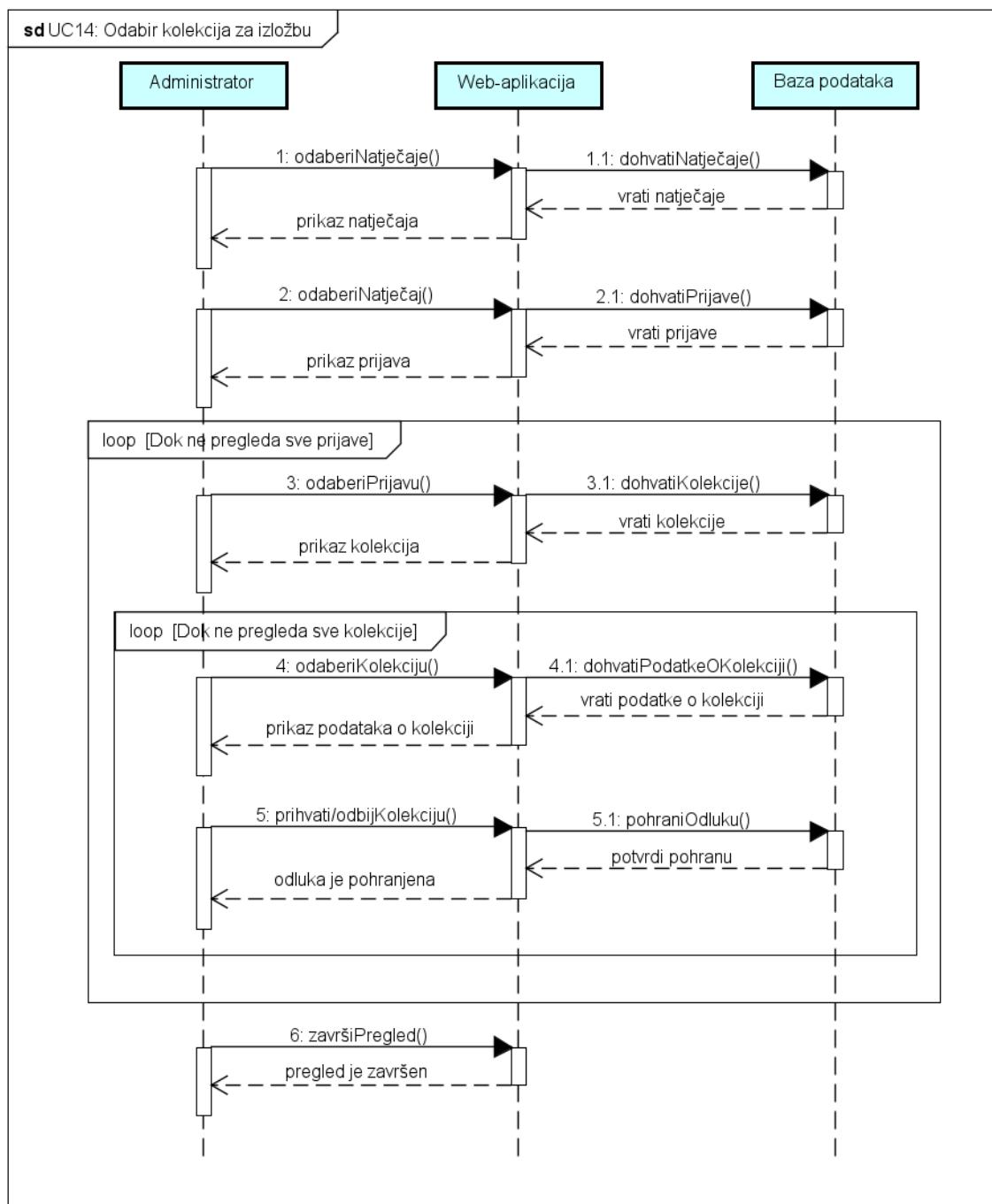
Umjetnik odabire opciju "Aktivni natječaji". Poslužitelj dohvaća popis aktivnih natječaja iz baze te ih prikazuje umjetniku. Umjetnik odabire natječaj na koji se želi prijaviti. Poslužitelj iz baze podataka dohvaća umjetnikove kolekcije. U slučaju da umjetnik nema nijednu kolekciju čiji stil odgovara stilu traženom u natječaju, sustav ga obavještava o tome. U suprotnom, poslužitelj prikazuje kolekcije umjetniku. Umjetnik odabire jednu ili više kolekcija te šalje zahtjev za potvrđivanje prijave. Poslužitelj sprema prijavu u bazu podataka. Baza podataka potvrđuje izmjene.



Slika 3.5: Sekvencijski dijagram za UC12

Obrazac uporabe UC14 - Odabir kolekcija za izložbu

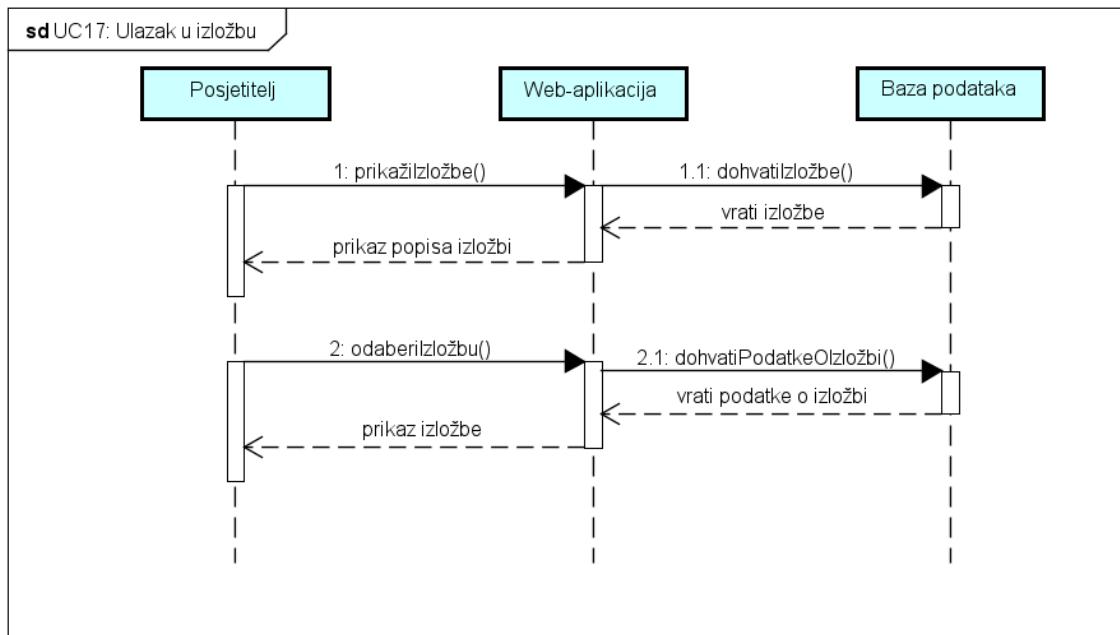
Administrator odabire opciju "Natječaji". Poslužitelj iz baze podataka dohvaća natječaje i prikazuje ih. Administrator odabire natječaj za koji želi razmotriti prijave. Poslužitelj iz baze dohvaća prijave na natječaj i prikazuje ih administratoru. Administrator odabire prijavu,a poslužitelj iz baze podataka dohvaća prijavljene kolekcije i prikazuje ih. Administrator odabire kolekciju koju želi razmotriti. Poslužitelj iz baze podataka dohvaća podatke o kolekciji i prikazuje ih. Administrator odabire opciju prihvati/odbij kolekciju. Poslužitelj odluku pohranjuje u bazu podataka. Postupak odabira prijave i pregleda kolekcija ponavlja se sve dok administrator ne razmotri sve prijave. Na kraju, administrator odabire opciju "Završi pregled".



Slika 3.6: Sekvencijski dijagram za UC14

Obrazac uporabe UC17 - Ulazak u izložbu

Posjetitelj odabire opciju "Izložbe". Poslužitelj dohvaća popis izložbi iz baze podataka i prikazuje ih umjetniku. Posjetitelj s liste trenutno aktivnih izložbi odabire izložbu koju želi pogledati. Poslužitelj iz baze podataka dohvaća podatke o izložbi i prikazuje ih posjetitelju.



Slika 3.7: Sekvencijski dijagram za UC17

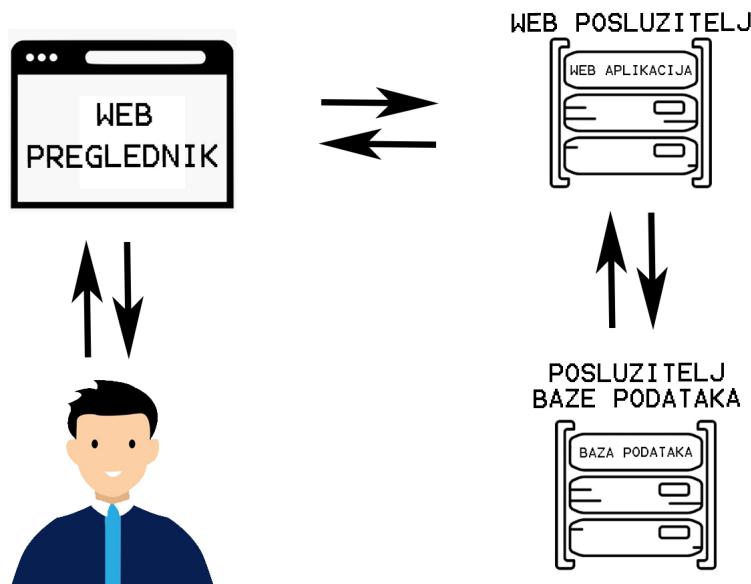
3.2 Ostali zahtjevi

- Sustav pri implementaciji mora koristiti objektno-orientiranu paradigmu
- Sustav treba sadržavati upute za rad
- Rad u sustavu treba biti jednostavan i intuitivan te osigurati pozitivno iskustvo korisnika
- Neispravno korištenje ne smije utjecati na rad sustava
- Korisnik u projektu na odgovor ne bi trebao čekati duže od jedne sekunde
- Nadogradnja sustava ne smije utjecati na postojeće funkcionalnosti
- Korisničko sučelje mora podržavati hrvatske dijakritičke znakove
- Sustav mora korisiti HTTPS protokol
- Veza s bazom podataka mora biti adekvatno zaštićena, pouzdana i efikasna
- Privatni podaci korisnika moraju biti zaštićeni
- Sustav mora osigurati pouzdanost i točnost podataka
- Sustav mora upravljati ulogama i pravima korisnika te omogućiti obavljanje samo dozvoljenih akcija
- Sustav mora poštovati zakone i regulacije
- Sustav kao valutu koristi HRK

4. Arhitektura i dizajn sustava

Arhitektura se može podijeliti na tri podsustava:

- Web poslužitelj
- Web aplikacija
- Baza podataka



Slika 4.1: Arhitektura sustava

Web preglednik je program koji korisniku omogućuje pregled web-stranica i multimedijalnih sadržaja vezanih uz njih. Njime gledamo trenutnu stranicu koja je pisana u kodu koji preglednik prevodi u nešto svakome razumljivo. Korištenjem preglednika šalju se zahtjevi web poslužitelju.

Web poslužitelj omogućava komunikaciju klijenta s aplikacijom. Za tu komunikaciju najčešće se koristi HTTP protokol, a poslužitelj zahtjev napisan u tom protokolu šalje web aplikaciji.

Web aplikacija obrađuje zahtjeve i ovisno o zahtjevu, komunicira s bazom podataka te korisniku vraća odgovor u obliku HTML dokumenta koji web preglednik prevodi.

Za izradu web stranice odlučili smo se za korištenje PostgreSQL baze podataka s kojom ćemo komunicirati putem Klijent-Server modela.

Za frontend tj. klijentski dio aplikacije koristit ćemo Vue.js. Vue.js je open-source JavaScript framework za izradu web aplikacija. On je dizajniran s ciljem da komplikirane aplikacije podijelimo na manje dijelove koje možemo ponovno koristiti. Razvojno okruženje za Vue.js je Microsoft Visual Studio.

Za backend tj. serverski dio aplikacije koristit ćemo programski jezik Java i Spring Boot framework. Radno okruženje za backend je IntelliJ IDEA.

Arhitektura Spring Boot framework-a koristi više slojeva:

- Sloj korisničke strane - implementiran u Vue.js
- Controller - povezuje korisnika i poslužitelja
- Service - obavlja poslovnu logiku i izračune
- Domain - implementira model podataka domene
- DAO (data access object) - obavlja sve operacije nad bazom podataka
- Sloj baze podataka - obavlja pohranu podataka u neku bazu

4.1 Baza podataka

Za našu web stranicu koristit ćemo relacijsku bazu podataka PostgreSQL. Baza podataka se sastoji od tablica, odnosno relacija koje su definirane svojim imenom i skupom atributa. Zadaća baze podataka je brza i jednostavna pohrana, izmjena i dohvata podataka za daljnju obradu.

Baza podataka ove aplikacije sastoji se od sljedećih tablica:

- Korisnik
- Kolekcija
- Komentar
- Transakcija
- KolekcijaNatjecaj
- KolekcijaIzlozba

- PosjetiteljIzlozba
- Djelo
- Izlozba
- Natjecaj

4.1.1 Opis tablica

Korisnik Ovaj entitet sadržava sve informacije o korisniku stranice. Sadrži atribute: Id, korisničko ime, ime, prezime, e-mail, lozinka, e-mail adresa PayPal računa, razina ovlasti korisnika te njegov portfolio u PDF-u. Ovaj entitet je u vezi *Many-to-One* s entitetom Komentar preko atributa Id korisnika, u vezi *Many-to-Many* s entitetom PosjetiteljIzlozba preko atributa Id korisnika, u vezi *One-to-Many* s entitetom Transakcija u ulozi primatelja i pošiljatelja preko atributa Id korisnika i u vezi *One-to-Many* s entitetom Kolekcija preko atributa Id korisnika.

Korisnik		
Id (PK)	INT	Primarni ključ za korisnika
KorisnickoIme (K)	VARCHAR	Korisničko ime korisnika
Ime	VARCHAR	Ime korisnika
Prezime	VARCHAR	Prezime korisnika
Email (K)	VARCHAR	E-mail korisnika
Lozinka	VARCHAR	Lozinka korisnika
PayPalEmail (K)	VARCHAR	Lozinka korisnika
RazinaOvlasti	VARCHAR	Oznaka razine ovlasti korisnika
PdfPortfolio	BYTEA	Portfolio umjetnika u obliku pdfa, samo za umjetnike

Kolekcija Ovaj entitet sadržava sve informacije o kolekcijama na stranici. Sadrži atribute: Id kolekcije, naziv kolekcije, opis kolekcije, stil kolekcije te Id umjetnika čija je kolekcija. Ovaj entitet je u vezi *One-to-Many* s entitetom KolekcijaNatjecaj preko atributa Id kolekcije, u vezi *One-to-Many* s entitetom KolekcijaIzlozba preko atributa Id kolekcije, u vezi *One-to-Many* s entitetom Djelo preko atributa Id kolekcije i u vezi *Many-to-One* s entitetom Korisnik preko atributa Id umjetnika.

Kolekcija		
Id (PK)	INT	Primarni ključ za kolekciju
Naziv	VARCHAR	Naziv kolekcije
Opis	VARCHAR	Opis kolekcije
Stil	VARCHAR	Stil kolekcije
IdUmjetnik (FK)	INT	Strani ključ vezan uz korisnika s ulogom umjetnik

Komentar Ovaj entitet sadržava sve informacije o komentarima djela. Sadrži atribute: Id komentara, tekst komentara, vrijeme komentara, Id djela koje je komentirano napisan te Id autora koji je to djelo komentirao. Ovaj entitet je u vezi *Many-to-One* s entitetom Djelo preko atributa Id djelo, u vezi *Many-to-One* s entitetom Korisnik preko atributa Id autor.

Komentar		
Id (PK)	INT	Primarni ključ komentara
Tekst	VARCHAR	Tekst komentara
Vrijeme	TIMESTAMP	Vrijeme kada je korisnik ostavio komentar
IdDjelo(FK)	INT	Strani ključ vezan uz djelo
IdAutor(FK)	INT	Strani ključ vezan uz korisnika koji je komentirao

Transakcija Ovaj entitet sadržava sve informacije o pojedinoj transakciji kupnje djela. Sadrži atribute: Id transakcije, iznos, vrijeme transakcije, Id djela koje je kupljeno, Id platitelja koji je kupio djelo te Id korisnika koji prima novce. Ovaj entitet je u vezi *Many-to-One* s entitetom Korisnik preko atributa Id platitelj i Id primatelj i u vezi *One-to-One* s entitetom Djelo preko atributa Id djelo.

Transakcija		
Id(PK)	INT	Primarni ključ transakcije
Iznos	FLOAT	Iznos plaćene cijene u kunama
Vrijeme	TIMESTAMP	Vrijeme kada je transakcija tj. uplata sredstava završena
IdDjelo(FK)	INT	Strani ključ vezan uz kupljeno djelo

Transakcija		
IdPlatitelj(FK)	INT	Strani ključ vezan uz korisnika koji je uplatio sredstva
IdPrimatelj(FK)	INT	Strani ključ vezan uz korisnika koji prima novce

KolekcijaNatjecaj Ovaj entitet sadržava sve informacije o odnosu kolekcije i natječaja. Sadrži atribute: Id kolekcije te Id natječaja u kojem je kolekcija. Ovaj entitet je u vezi *Many-to-One* s entitetom Kolekcija preko atributa Id kolekcija i u vezi *Many-to-One* s entitetom Natjecaj preko atributa Id natjecaj.

KolekcijaNatjecaj		
IdKolekcija(PK,FK)	INT	Dio primarnog ključa vezan uz kolekciju, ujedno i strani ključ
IdNatjecaj(PK,FK)	INT	Dio primarnog ključa vezan uz natječaj, ujedno i strani ključ

KolekcijaIzložba Ovaj entitet sadržava sve informacije o odnosu kolekcije i izložbe. Sadrži atribute: Id izložbe te Id kolekcije u kojem je izložba. Ovaj entitet je u vezi *Many-to-One* s entitetom Kolekcija preko atributa Id kolekcija i u vezi *Many-to-One* s entitetom Izložba preko atributa Id izložba.

KolekcijaIzložba		
IdKolekcija(PK,FK)	INT	Dio primarnog ključa vezan uz kolekciju, ujedno i strani ključ
IdIzložba(PK,FK)	INT	Dio primarnog ključa vezan uz izložbu, ujedno i strani ključ

PosjetiteljIzložba Ovaj entitet sadržava sve informacije o odnosu posjetitelja i izložbe. Sadrži atribute: Id posjetitelja te Id izložbe koju posjetitelj pregledava. Ovaj entitet je u vezi *Many-to-One* s entitetom Posjetitelj preko atributa Id posjetitelj i u vezi *Many-to-One* s entitetom Izložba preko atributa Id izložba.

PosjetiteljIzložba		
IdIzložba(PK,FK)	INT	Dio primarnog ključa vezan uz izložbu, ujedno i strani ključ

PosjetiteljIzlozba		
IdPosjetitelj(PK,FK)	INT	Dio primarnog ključa vezan uz korisnika koji je posjetio izložbu, ujedno i strani ključ

Djelo Ovaj entitet sadržava sve informacije o pojedinom djelu. Sadrži atribute: Id djela, naziv, opis, cijena, stil, djelo u binarnom formatu te Id kolekcije u kojem je djelo. Ovaj entitet je u vezi *Many-to-One* s entitetom Kolekcija preko atributa Id kolekcija, u vezi *One-to-One* s entitetom Transakcija preko atributa Id djelo.

Djelo		
Id(PK)	INT	Primarni ključ djela
Naziv	VARCHAR	Naziv djela
Opis	VARCHAR	Autorov tj. umjetnikov opis djela
Cijena	FLOAT	Cijena djela u kunama
Stil	VARCHAR	Stil djela
blob	BYTEA	Djelo u pdf formatu
IdKolekcija(FK)	INT	Strani ključ vezan uz kolekciju u kojem je djelo

Izlozba Ovaj entitet sadržava sve informacije o izložbama. Sadrži atribute: Id izložbe, naziv, opis, vrijeme početka izložbe, vrijeme trajanja, provizija prilikom kupnje djela, stil te Id natječaja unutar kojeg je izloženo djelo. Ovaj entitet je u vezi *One-to-Many* s entitetom PosjetiteljIzlozba preko atributa Id izlozba, u vezi *One-to-Many* s entitetom KolekcijaIzlozba preko atributa Id izlozba i u vezi *Many-to-One* s entitetom Natjecaj preko atributa Id natjecaj.

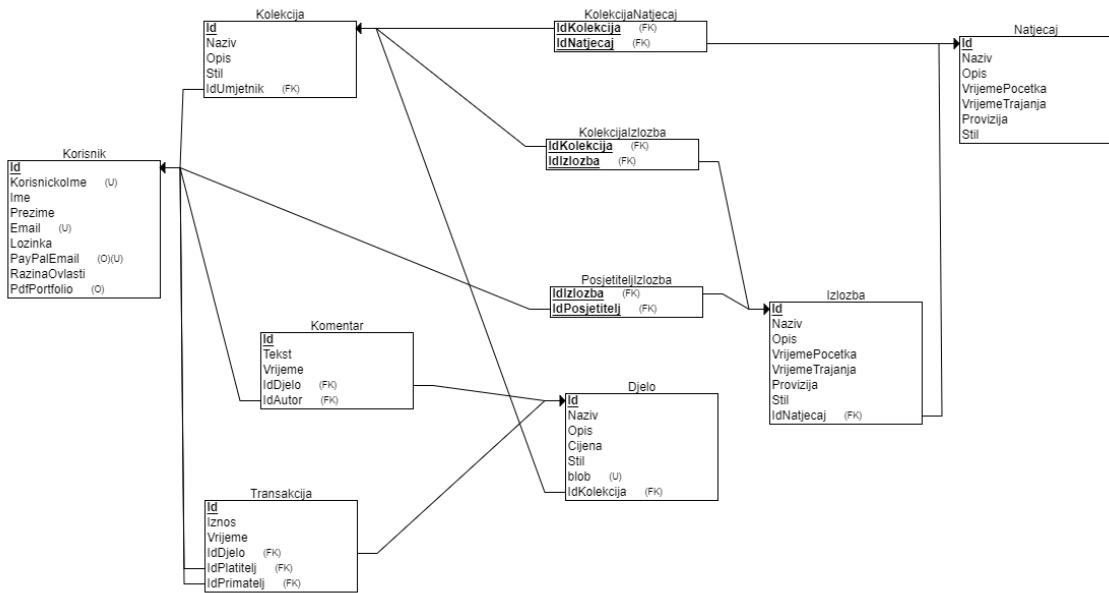
Izlozba		
Id(PK)	INT	Primarni ključ izložbe
Naziv	VARCHAR	Naziv djela
Opis	VARCHAR	Autorov tj. umjetnikov opis djela
VrijemePocetka	TIMESTAMP	Vrijeme u kojem počinje izložba
VrijemeTrajanja	INTERVAL	Interval u kojem izložba traje

Izložba		
Provizija	FLOAT	Broj izražen u postocima o kojem ovisi koliko stranica uzima proviziju od cijene djela tj. koliko će od ukupne cijene djela novaca dobiti korisnik. Korisnik dobiva $(100-\text{Provizija})/100 * \text{CijenaDjela}$ kuna dok ostatak novaca ide stranici.
Stil	VARCHAR	Stil izložbe
IdNatjecaj(FK)	INT	Strani ključ vezan uz natječaj u kojem je djelo

Natjecaj Ovaj entitet sadržava sve informacije o natječajima. Sadrži atribute: Id natječaja, naziv, opis, vrijeme početka natječaja, vrijeme trajanja, provizija prilikom kupnje djela te stil natječaja. Ovaj entitet je u vezi *One-to-One* s entitetom Izložba preko atributa Id natjecaj i u vezi *One-to-Many* s entitetom KolekcijaNatjecaj preko atributa Id natjecaj.

Natjecaj		
Id(PK)	INT	Primarni ključ natječaja
Naziv	VARCHAR	Naziv natječaja
Opis	VARCHAR	Administratorov opis natječaja
VrijemePocetka	TIMESTAMP	Vrijeme u kojem počinje natječaj
VrijemeTrajanja	INTERVAL	Interval u kojem natječaj traje
Provizija	FLOAT	Broj izražen u postocima o kojem ovisi koliko stranica uzima proviziju od cijene djela tj. koliko će od ukupne cijene djela novaca dobiti korisnik. Korisnik dobiva $(100-\text{Provizija})/100 * \text{CijenaDjela}$ kuna dok ostatak novaca ide stranici.
Stil	VARCHAR	Stil natječaja

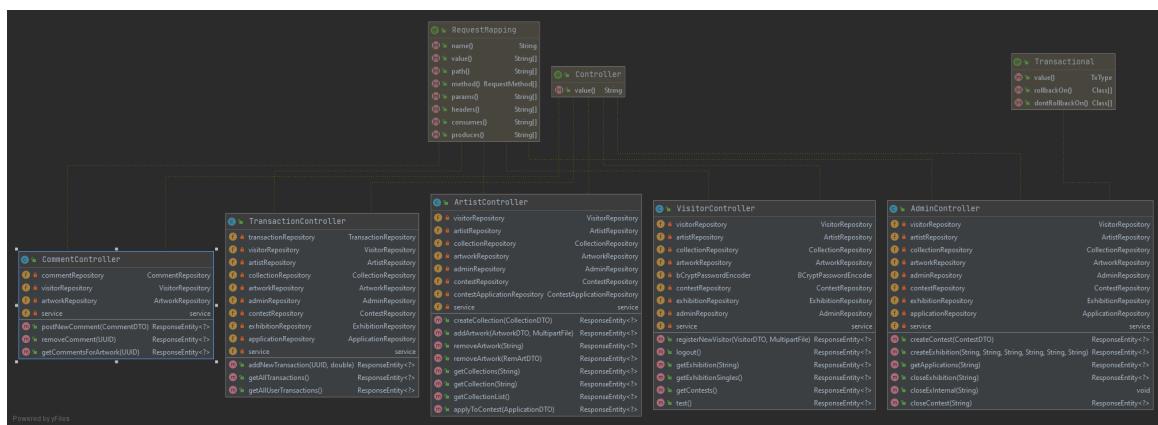
4.1.2 Dijagram baze podataka



Slika 4.2: Relacijska shema baze podataka

4.2 Dijagram razreda

Na slikama 4.2 i 4.3 prikazani su dijagrami razreda u backend dijelu arhitekture. Razredi na slici 4.2 nasljeđuju razred Controller. Metode tih razreda manipuliraju DTO-ima (Data transfer object) koji su dohvaciени pomoću metoda implementiranih u Model razredima. Metode implementirane u Controller razredima vracaju JSON datoteke s HTTP statusnim kodom. Slika 4.3 prikazuje dijagram DTO razreda.



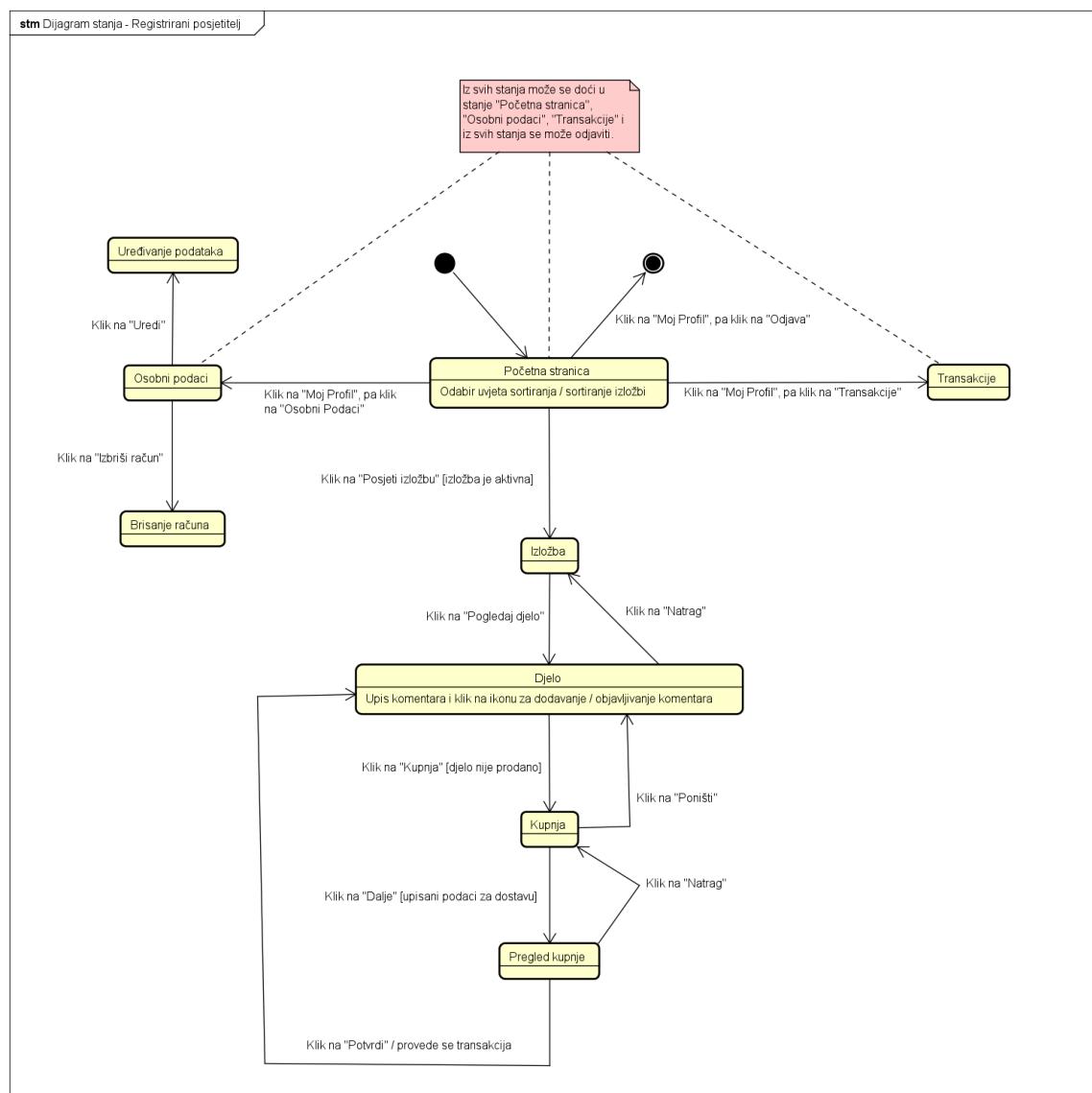
Slika 4.3: Dijagram Razreda Controller



Slika 4.4: Dijagram DTO razreda

4.3 Dijagram stanja

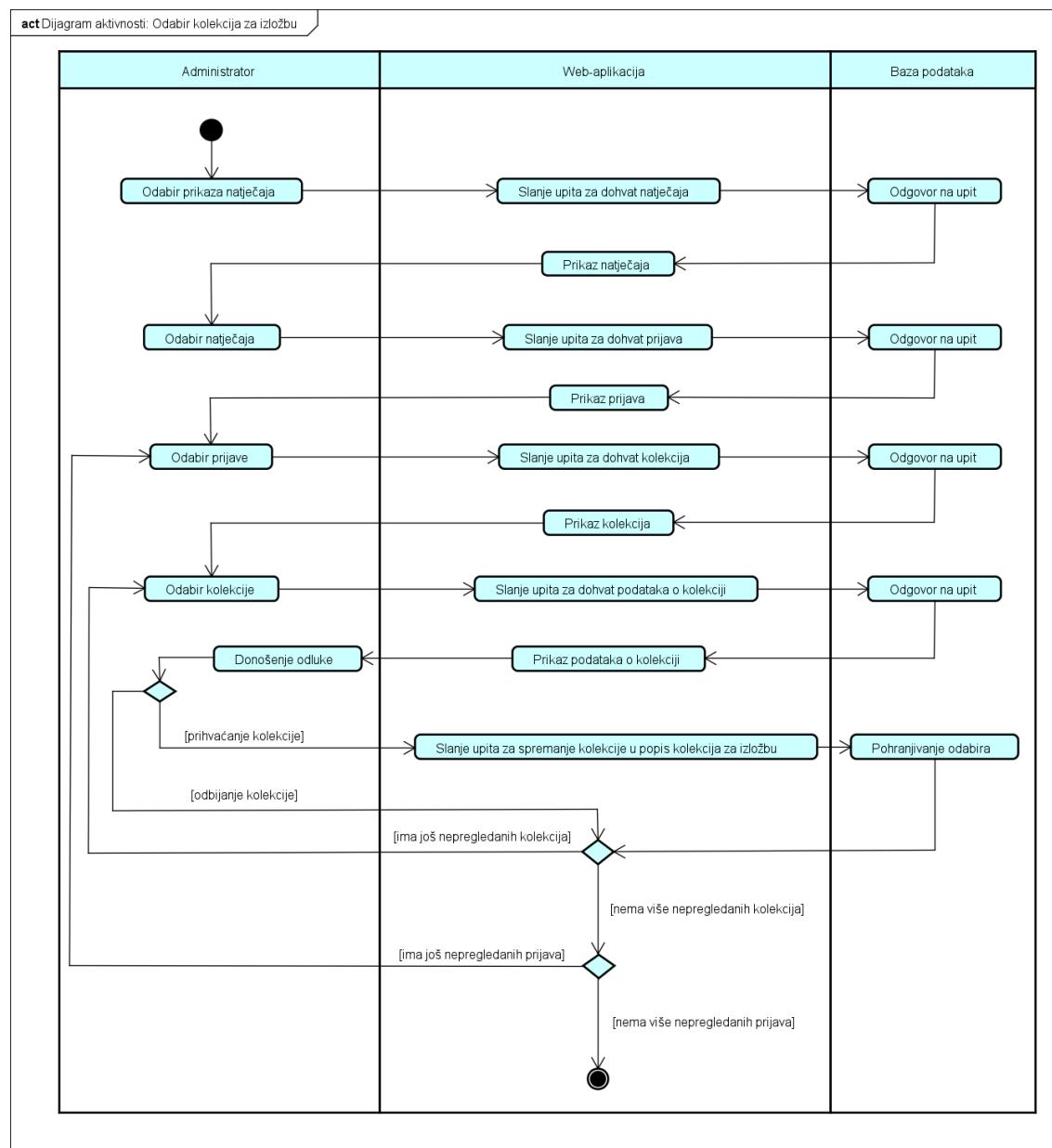
Dijagram stanja prikazuje stanja objekta te prijelaze iz jednog stanja u drugo temeljene na događajima. Na slici prikazan je dijagram stanja za registriranog posjetitelja. Nakon prijave, posjetitelju se prikazuje početna stranica na kojoj može vidjeti aktivne i buduće izložbe. Za aktivne izložbe ima opciju posjećivanja izložbe gdje za svako djelo može dodati komentar. Djelo može kupiti samo ako ono nije prodano. Posjetitelj može u padajućem izborniku odabrati "Osobni Podaci" što će mu prikazati stranicu s njegovim podacima. U izborniku također može odabrati "Transakcije" gdje može vidjeti transakcije u kojima je sudjelovao.



Slika 4.5: Dijagram stanja: Registrirani posjetitelj

4.4 Dijagram aktivnosti

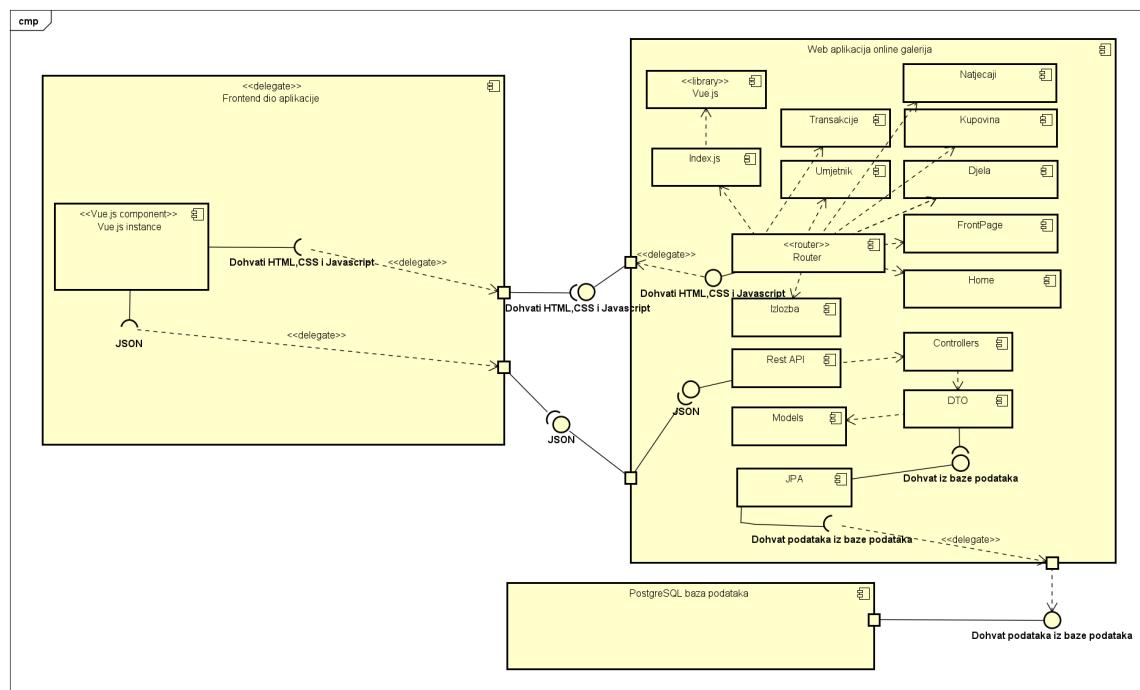
Slika prikazuje dijagram aktivnosti za proces odabira kolekcija za izložbu. Dijagram je vertikalno podijeljen u particije po aktorima koji uključuju administratora, web-aplikaciju i bazu podataka. Administrator odabire prikaz natječaja. Web-aplikacija bazi podataka šalje upit za dohvrat natječaja i prikazuje ih. Administrator odabire natječaj i pregledava prijave. Iz svake prijave odabire nula ili više kolekcija koje želi uključiti u izložbu. Ovaj proces se ponavlja sve dok ima nepregledanih prijava.



Slika 4.6: Dijagram aktivnosti: Odabir kolekcija za izložbu

4.5 Dijagram komponenti

Dijagram komponenti prikazuje organizaciju komponenti prema okolini. Sustavu se pristupa preko sučelja za dohvat HTML, CSS i Javascript datoteke te preko REST API komponenti. Router je komponenta Vue.js-a koja određuje koju će datoteku sučelje dobiti. Preko sučelja za JSON podatke pristupa se REST API-u. REST API je zadužen za backend podatke aplikacije. JPA služi za dohvat podataka iz baze podataka dok se podaci koji se šalju Java Springom šalju u obliku DTO (Data Transfer Object).



Slika 4.7: Dijagram komponenti

5. Implementacija i korisničko sučelje

5.1 Korištene tehnologije i alati

Komunikacija u timu realizirana je korištenjem aplikacija WhatsApp¹ i Discord². Za izradu UML dijagrama korišten je alat Astah UML³, a kao sustav za upravljanje izvornim kodom Git⁴. Udaljeni rezitorij projekta je dostupan na web platformi GitLab⁵.

Kao razvojno okruženje za *frontend* korišten je Microsoft Visual Studio⁶ - integrirano je razvojno okruženje (IDE) tvrtke Microsoft. Prvenstveno se koristi za razvoj računalnih programa za operacijski sustav Windows, kao i za web-stranice, web-aplikacije, web-usluge i mobilne aplikacije. Visual Studio za razvoj softvera koristi razne Microsoftove platforme.

Kao razvojno okruženje za *backend* korišten je IntelliJ IDEA⁷ - integrirano je razvojno okruženje (IDE) tvrtke JetBrains. Prvenstveno se koristi za razvoj programa u programskom jeziku Java, ali podržava i druge jezike. Dostupno je u dvije verzije: besplatna Community Edition i komercijalna Ultimate Edition.

Aplikacije je napisana koristeći radni okvir Spring Boot⁸ i jezik Java⁹ za izradu *backenda* te Vue.js¹⁰ i jezik Javascript¹¹ za izradu *frontenda*. Vue.js je open-source MVC *frontend* radni okvir pisan u Javascriptu za izgradnju korisničkih sučelja i jednostraničnih aplikacija gdje se glavni dokument dinamički mijenja. Radni okvir Spring Boot je specijalizacija radnog okvira Spring¹² s ciljem jednostavnijeg i bržeg oblikovanja web aplikacija. U svojoj automatskoj konfiguraciji olakšava posao programeru jer više toga uobičajenog u web aplikacijama već ima podešeno.

¹<https://www.whatsapp.com/>

²<https://discord.com/>

³<https://astah.net/products/astah-uml/>

⁴<https://git-scm.com/>

⁵<https://gitlab.com/>

⁶<https://visualstudio.microsoft.com/>

⁷<https://www.jetbrains.com/idea/>

⁸<https://spring.io/projects/spring-boot>

⁹<https://www.java.com/>

¹⁰<https://vuejs.org/>

¹¹<https://www.javascript.com/>

¹²<https://spring.io/projects/spring-framework>

Za bazu podataka koristili smo PostgreSQL¹³.

¹³<https://www.postgresql.org/>

5.2 Ispitivanje programskog rješenja

5.2.1 Ispitivanje komponenti

Sljedeći jedinični testovi su korišteni za ispitivanje jedinica nad razredima koji implementiraju neke od temeljnih funkcionalnosti.

```
@Test
public void testVisitorRegistration() throws Exception {
    VisitorDTO visitorDTO = new VisitorDTO();
    visitorDTO.setName("Pero");
    visitorDTO.setSurname("Perić");
    visitorDTO.setEmail("peroperic@gmail.com");
    visitorDTO.setPassword("password123");
    visitorDTO.setPaypalMail("peroperic@gmail.com");
    visitorDTO.setFlag(false);

    String json = new ObjectMapper().writeValueAsString(visitorDTO);
    MockMultipartFile jsonFile = new MockMultipartFile("json", "", "application/json", json.getBytes());
    MockMultipartFile nullFile = new MockMultipartFile("file", "", "Auto", (byte[]) null);

    mockMvc.perform(MockMvcRequestBuilders.multipart("/visitor/registration").file(jsonFile).file(nullFile))
        .andDo(print())
        .andExpect(status().isOk());
}

@Test
public void testArtistRegistration() throws Exception {
    VisitorDTO visitorDTO = new VisitorDTO();
    visitorDTO.setName("Pero");
    visitorDTO.setSurname("Umjetnik");
    visitorDTO.setEmail("peroumjetnik@gmail.com");
    visitorDTO.setPassword("password123");
    visitorDTO.setPaypalMail("peroumjetnik@gmail.com");
    visitorDTO.setFlag(true);

    String json = new ObjectMapper().writeValueAsString(visitorDTO);
    MockMultipartFile jsonFile = new MockMultipartFile("json", "", "application/json", json.getBytes());
    MockMultipartFile nullFile = new MockMultipartFile("file", "", "Auto", (byte[]) null);

    mockMvc.perform(MockMvcRequestBuilders.multipart("/visitor/registration").file(jsonFile).file(nullFile))
        .andDo(print())
        .andExpect(status().isOk());
}
```

Slika 5.1: Jedinični testovi koji testiraju registraciju novih posjetitelja i novih umjetnika

```
@Test
public void testVisitorLogin() throws Exception {
    String json = new ObjectMapper().writeValueAsString(new LoginForm( email: "peroperic@gmail.com", password: "password123"));
    mockMvc.perform(MockMvcRequestBuilders.post( urlTemplate: "/Login").contentType(MediaType.APPLICATION_JSON).content(json))
        .andDo(print())
        .andExpect(status().isOk());
}

@Test
@Disabled
public void testArtistLogin() throws Exception {
    String json = new ObjectMapper().writeValueAsString(new LoginForm( email: "peroumjetnik@gmail.com", password: "password123"));
    mockMvc.perform(MockMvcRequestBuilders.post( urlTemplate: "/login").contentType(MediaType.APPLICATION_JSON).content(json))
        .andDo(print())
        .andExpect(status().isOk());
}
```

Slika 5.2: Jedinični testovi koji ispituju prijavu posjetitelja i umjetnika

```
@Test
public void registrationWithSameEmailShouldThrow() throws Exception {
    VisitorDTO visitorDTO = new VisitorDTO();
    visitorDTO.setName("Pero");
    visitorDTO.setSurname("Sistimailom");
    visitorDTO.setEmail("peroperic@gmail.com");
    visitorDTO.setPassword("perinasifra2");
    visitorDTO.setPaypalMail("peroperic@gmail.com");
    visitorDTO.setFlag(false);

    String json = new ObjectMapper().writeValueAsString(visitorDTO);
    MockMultipartFile jsonFile = new MockMultipartFile( name: "json", originalFilename: "", contentType: "application/json", json.getBytes());
    MockMultipartFile nullFile = new MockMultipartFile( name: "file", originalFilename: "", contentType: "Auto", (byte[]) null);

    mockMvc.perform(MockMvcRequestBuilders.multipart( urlTemplate: "/visitor/registration").file(jsonFile).file(nullFile))
        .andDo(print())
        .andExpect(status().is4xxClientError());
}
```

Slika 5.3: Jedinični test koji ispituje da se vraća prikladni kod za grešku kada se pri registraciji koristi e-mail postojećeg korisnika

```

@Test
public void testPostArtwork() throws Exception {
    ArtworkDTO artworkDTO = new ArtworkDTO();
    artworkDTO.setCollectionName("Kolekcija za testiranje 2");
    artworkDTO.setDescription("Ovo predivno djelo za jeftinu cijenu od 100 kuna je koristeno za potrebe unit testiranja.");
    artworkDTO.setName("Unit test djelo");
    artworkDTO.setStyle("OLJAKA");
    artworkDTO.setPrice("100");

    String json = new ObjectMapper().writeValueAsString(new LoginForm( email: "peroumjetnik@gmail.com", password: "password123"));
    String artworkJson = new ObjectMapper().writeValueAsString(artworkDTO);

    MockMultipartFile jsonFile = new MockMultipartFile( name: "json", originalFilename: "", contentType: "application/json", artworkJson.getBytes());
    MockMultipartFile nullFile = new MockMultipartFile( name: "file", originalFilename: "", contentType: "Auto", (byte[]) null);

    MvcResult mvcResult = mockMvc.perform(MockMvcRequestBuilders.post( uriTemplate: "/login").contentType(MediaType.APPLICATION_JSON).content(json)).andReturn();

    String header = mvcResult.getResponse().getHeader( name: "Authorization");

    mockMvc.perform(MockMvcRequestBuilders.multipart( uriTemplate: "/artist/addArtwork")
        .file(jsonFile)
        .file(nullFile)
        .header( name: "Authorization", header))
        .andDo(print())
        .andExpect(status().isOk());
}

```

Slika 5.4: Jedinični test koji ispituje dodavanje nove slike u kolekciju umjetnika

```

@Disabled
public void testCreateCollection() throws Exception {

    String json = new ObjectMapper().writeValueAsString(new LoginForm( email: "peroumjetnik@gmail.com", password: "password123"));

    MvcResult mvcResult = mockMvc.perform(MockMvcRequestBuilders.post( uriTemplate: "/login").contentType(MediaType.APPLICATION_JSON).content(json)).andReturn();
    String header = mvcResult.getResponse().getHeader( name: "Authorization");
    System.out.println("\n\n Ovo je header " + header);

    CollectionDTO collectionDTO = new CollectionDTO();
    collectionDTO.setName("Kolekcija za testiranje 2");
    collectionDTO.setDescription("Opis Kolekcije za testiranje");
    collectionDTO.setStyle("OLJAKA");

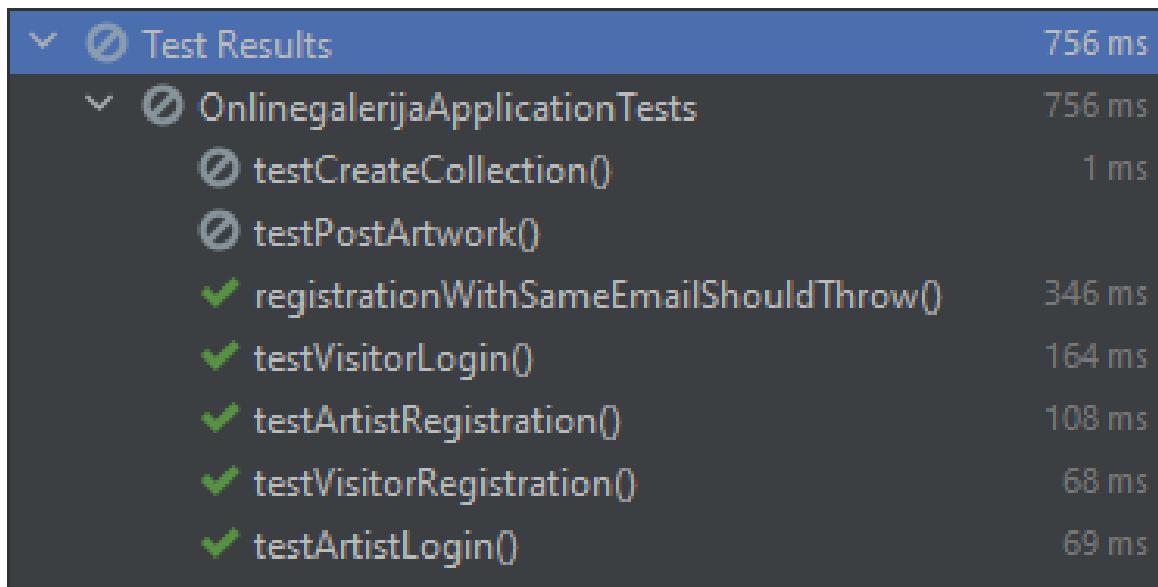
    String jsonColl = new ObjectMapper().writeValueAsString(collectionDTO);

    mockMvc.perform(MockMvcRequestBuilders.post( uriTemplate: "/artist/createCollection").header( name: "Authorization", header)
        .contentType(MediaType.APPLICATION_JSON).content(jsonColl))
        .andDo(print())
        .andExpect(status().isOk());
}

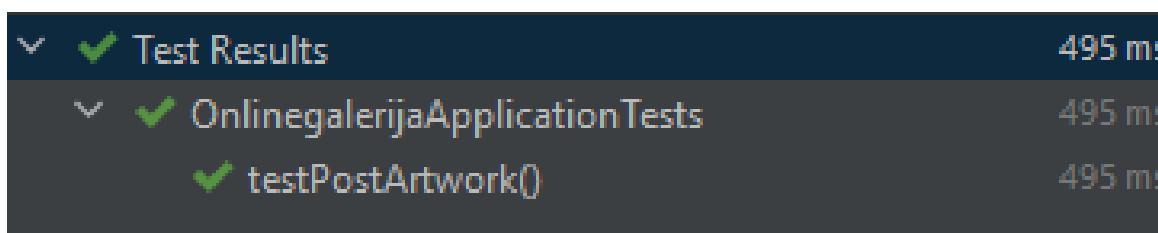
```

Slika 5.5: Jedinični test koji ispituje stvaranje nove umjetnikove kolekcije

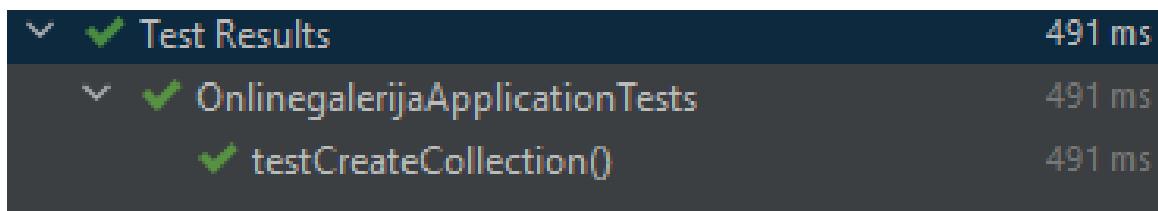
Sljedeće slike prikazuju rezultate pokretanja jediničnih testova. Zbog načina na koji su pisani nisu mogli biti pokrenuti svi istovremeno.



Slika 5.6: Rezultati prvih 5 jediničnih testova, zadnja dva nisu bili pokretani



Slika 5.7: Rezultat jediničnog testa koji ispituje dodavanje slike u kolekciju umjetnika



Slika 5.8: Rezultat jediničnog testa koji ispituje stvaranje nove kolekcije

5.2.2 Ispitivanje sustava

Potrebno je provesti i opisati ispitivanje sustava koristeći radni okvir Selenium¹⁴. Razraditi **minimalno 4 ispitna slučaja** u kojima će se ispitati redovni slučajevi, rubni uvjeti te poziv funkcionalnosti koja nije implementirana/izaziva pogrešku kako bi se vidjelo na koji način sustav reagira kada nešto nije u potpunosti ostvareno. Ispitni slučaj se treba sastojati od ulaza (npr. korisničko ime i lozinka), očekivanog izlaza ili rezultata, koraka ispitivanja i dobivenog izlaza ili rezultata.

Izradu ispitnih slučajeva pomoću radnog okvira Selenium moguće je provesti pomoću jednog od sljedeća dva alata:

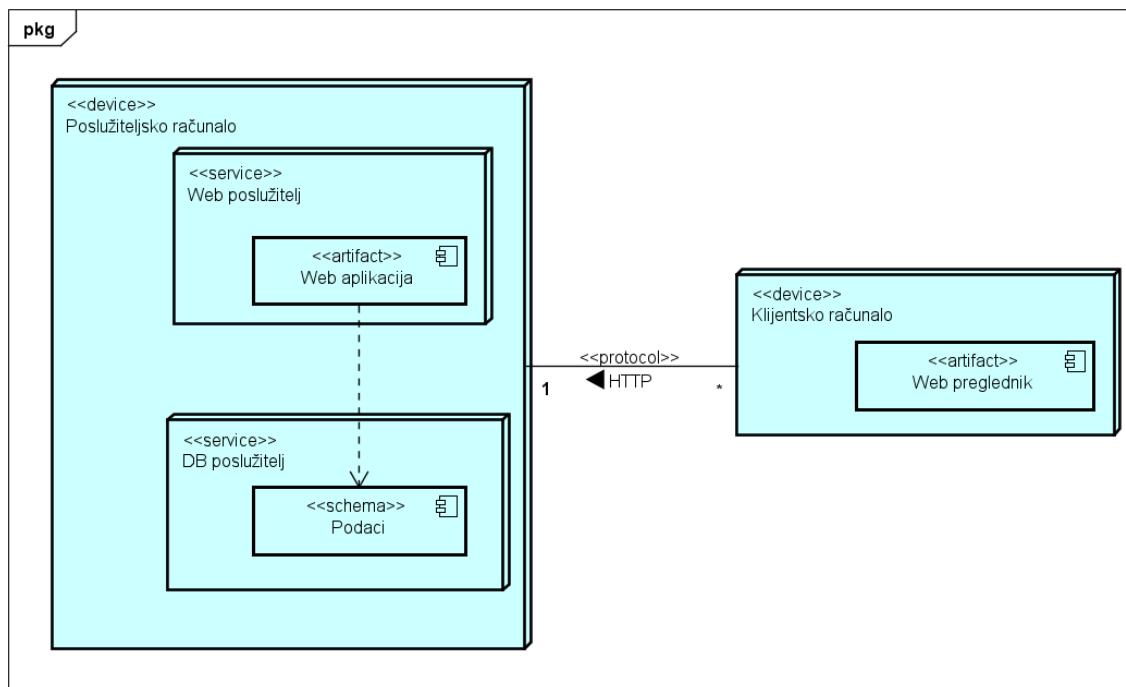
- dodatak za preglednik **Selenium IDE** - snimanje korisnikovih akcija radi automatskog ponavljanja ispita
- **Selenium WebDriver** - podrška za pisanje ispita u jezicima Java, C#, PHP koristeći posebno programsко sučelje.

Detalji o korištenju alata Selenium bit će prikazani na posebnom predavanju tijekom semestra.

¹⁴<https://www.seleniumhq.org/>

5.3 Dijagram razmještaja

Slika prikazuje specifikacijski dijagram razmještaja koji opisuje topologiju sustava. Sustav se sastoji od poslužiteljskog računala na kojem se nalaze web poslužitelj i poslužitelj baze podataka. Web aplikacija ovisna je o bazi podataka. Klijenti koriste web preglednik za pristup aplikaciji, a komunikacija klijent – poslužitelj odvija se preko HTTP protokola.



Slika 5.9: Specifikacijski dijagram razmještaja

5.4 Upute za puštanje u pogon

Instalacija poslužitelja baze podataka

Prije instalacije potrebno je preuzeti instalacijski program sa sljedeće poveznice:
<https://www.postgresql.org/download/>.

Napomene za instalaciju:

- kod odabira komponenti sustava koje će biti instalirane, obavezno je označiti komponente **PostgreSQL Server** i **PgAdmin**
- ostaviti predložena vrata za pristup sustavu baze podataka: **5432**
- ostale postavke, poput lozinke postgres korisnika, postaviti proizvoljno

Instalacija Mavena

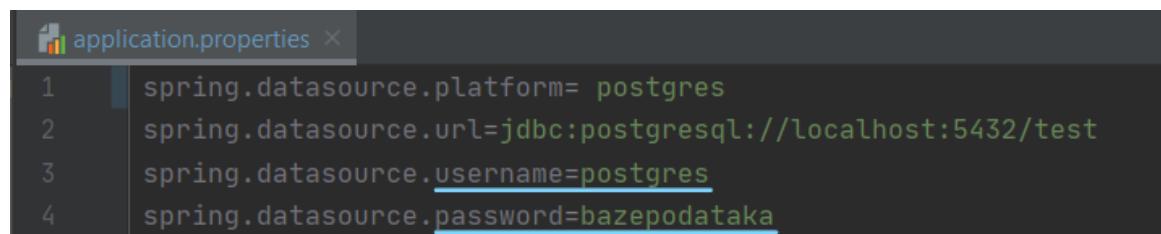
Ukoliko Maven već nije instaliran, potrebno ga je preuzeti s poveznice
<https://maven.apache.org/download.cgi> i instalirati.

Korištenje pgAdmin programa za pristup PostgreSQL sustavu baze podataka

Pri prvom pokretanju programa pgAdmin potrebno je upisati „master“ lozinku – istu onu koju je postavljena za postgres korisnika. Nakon unosa lozinke otvoriti u prozoru „Browser“ listu „Servers“ i kliknuti na znak pored „PostgreSQL 12“. Ponovno unijeti lozinku za postgres korisnika.

Konfiguracija veze s bazom podataka

U datoteci **application.properties** koja se nalazi u direktoriju „backend/src/main/resources“, potrebno je postaviti postgres korisnika i lozinku koja je stavljeni tijekom instalacije poslužitelja baze podataka. Pri prvom pokretanju aplikacije, baza se automatski stvara.



```
application.properties
1 spring.datasource.platform=postgres
2 spring.datasource.url=jdbc:postgresql://localhost:5432/test
3 spring.datasource.username=postgres
4 spring.datasource.password=bazepodataka
```

Pokretanje aplikacije

U terminalu se pozicionirati u direktorij „backend“. Izvesti naredbu:

- mvn spring-boot:run -Drun.arguments=--server.port=8080

Ova naredba pokrenut će Spring Boot aplikaciju na portu 8080. Broj porta može se promijeniti, ali onda je potrebno u datoteci `.env` u direktoriju „frontend“ promijeniti broj porta na koji se šalju zahtjevi.

```
frontend > 🌐 .env
1   VUE_APP_BACKEND_URI=http://localhost:8080
2
```

Pozicionirati u direktorij „frontend“. Izvesti naredbe:

- npm install
- npm run serve -- --port 8081

Prva naredba instalirat će sve potrebne pakete, a druga će pokrenuti Vue JS aplikaciju na portu 8081.

6. Zaključak i budući rad

Zadatak naše grupe bio je razvoj online galerije. Na stranici posjetitelji mogu pregledavati izložbe, ostavljati komentare te kupovati djela, dok umjetnici mogu svoja djela dodavati na izložbu te ista prodavati. Zadatak je uspješno ostvaren te je sam proces podijeljen u dvije faze. U prvoj fazi je okupljen tim te se pretežito bavilo dokumentacijom projekta. Zbog dobre, potpune i cjelovite dokumentacije, ulazak u drugu fazu je s potpuno jasnim ciljevima i dobro definiranim zahtjevima. Pomoć u tome su obrasci uporabe, model baze podataka, sekvencijski dijagrami te model razreda koji su napravljeni tijekom prve faze. Druga faza počela je s proučavanjem alata, sučelja i programske jezike koji će biti korišteni u razvoju aplikacije. Odmah na početku je došlo do podjele u timove za frontend i backend koji su počeli sa izradom. Osim same izrade aplikacije, u drugoj fazi također je dokumentiran projekt i to dijagramima stanja, aktivnosti, komponenti i razmještaja. Komunikacija unutar tima se odvijala pretežito preko WhatsAppa dok su sastanci te komunikacija frontend i backend timova bili preko Discorda. Proširenje aplikacije bilo bi prvenstveno razvojem mobilne aplikacije čime bi aplikacija dobila još više korisnika. Također moguće je i dodavanje novih načina plaćanja te razvoj stranice za dostavu u kojoj bi korisnik mogao birati način preuzimanja slike odnosno dostave iste. Ovaj projekt dao je svim članovima tima iskustvo ne samo u rukovanju s programima poput Asteha, radu s programskim jezicima poput LaTeXa, HTML-a, CSS-a, JS-a te radu s radnim okvirima Vue.js i Spring Boot već i iskustvo timskog rada na dužem i kompleksnijem projektu što će im sigurno biti korisno u budućim projektima tijekom školovanja te unutar radne zajednice na budućim poslovima. Za daljni razvoj aplikacije najvažnije bi bilo razraditi dokumentaciju poput one u prvoj fazi projekta uz koju je puno lakša i jednostavnija izrada bilo kakve nadogradnje aplikacije.

Popis literature

1. Programsko inženjerstvo, FER ZEMRIS, <http://www.fer.hr/predmet/proinzh>
2. I. Sommerville, "Software engineering", 8th ed, Addison Wesley, 2007.
3. The Unified Modeling Language, <https://www.uml-diagrams.org/>
4. Astah Community, <http://astah.net/editions/uml-new>
5. Spring, <https://spring.io/>
6. Vue.js, <https://vuejs.org/>
7. Heroku, <https://devcenter.heroku.com/>
8. Tutorials Point, <https://www.tutorialspoint.com/>
9. Baeldung, <https://www.baeldung.com/>

Indeks slika i dijagrama

2.1 Snimka zaslona je slikana 7.11.2020. na stranici: https://onlinegallery.art/en/	5
2.2 Snimka zaslona je slikana 7.11.2020. na stranici: https://www.saatchiart.com/	6
3.1 Dijagram obrazaca uporabe: funkcionalnosti korisnika i posjetitelja	20
3.2 Dijagram obrazaca uporabe: funkcionalnosti umjetnika	21
3.3 Dijagram obrazaca uporabe: funkcionalnosti administratora	22
3.4 Sekvencijski dijagram za UC9	23
3.5 Sekvencijski dijagram za UC12	24
3.6 Sekvencijski dijagram za UC14	26
3.7 Sekvencijski dijagram za UC17	27
4.1 Arhitektura sustava	29
4.2 Relacijska shema baze podataka	36
4.3 Dijagram Razreda Controller	37
4.4 Dijagram DTO razreda	38
4.5 Dijagram stanja: Registrirani posjetitelj	40
4.6 Dijagram aktivnosti: Odabir kolekcija za izložbu	42
4.7 Dijagram komponenti	43
5.1 Jedinični testovi koji testiraju registraciju novih posjetitelja i novih umjetnika	46
5.2 Jedinični testovi koji ispituju prijavu posjetitelja i umjetnika	47
5.3 Jedinični test koji ispituje da se vraća prikladni kod za grešku kada se pri registraciji koristi e-mail postojećeg korisnika	47
5.4 Jedinični test koji ispituje dodavanje nove slike u kolekciju umjetnika	48
5.5 Jedinični test koji ispituje stvaranje nove umjetnikove kolekcije	48
5.6 Rezultati prvih 5 jediničnih testova, zadnja dva nisu bili pokretani	49
5.7 Rezultat jediničnog testa koji ispituje dodavanje slike u kolekciju umjetnika	49

Dodatak: Prikaz aktivnosti grupe

Dnevnik sastajanja

1. sastanak

- Datum: 5. listopada 2020.
- Prisustvovali: Blagaić, Ilić, Kurtović, Mahović, Novinc, Šestak, Zubčić
- Teme sastanka:
 - sastanak s asistentom
 - detaljnije upoznavanje s temom
 - razrješavanje nedoumica

2. sastanak

- Datum: 13. listopada 2020.
- Prisustvovali: Blagaić, Ilić, Kurtović, Mahović, Novinc, Šestak, Zubčić
- Teme sastanka:
 - okvirna podjela rada
 - odabir tehnologija i alata

3. sastanak

- Datum: 18. listopada 2020.
- Prisustvovali: Blagaić, Ilić, Kurtović, Mahović, Novinc, Šestak, Zubčić
- Teme sastanka:
 - definiranje funkcionalnih zahtjeva
 - razrada obrazaca uporabe
 - dogovor o podjeli rada na dokumentaciji

4. sastanak

- Datum: 21. listopada 2020.
- Prisustvovali: Blagaić, Ilić, Kurtović, Mahović, Novinc, Šestak, Zubčić
- Teme sastanka:
 - revizija napravljene dokumentacije
 - sekvencijski dijagrami
 - daljnja podjela zadatka

5. sastanak

- Datum: 1. studenoga 2020.
- Prisustvovali: Blagaić, Ilić, Kurtović, Mahović, Novinc, Šestak, Zubčić
- Teme sastanka:
 - osmišljavanje dizajna stranice
 - podjela rada frontendu i backendu

6. sastanak

- Datum: 4. studenoga 2020.
- Prisustvovali: Blagaić, Ilić, Kurtović, Mahović, Novinc, Šestak, Zubčić
- Teme sastanka:
 - pregled dokumentacije
 - podjela zadataka

7. sastanak

- Datum: 8. studenoga 2020.
- Prisustvovali: Blagaić, Ilić, Kurtović, Mahović, Novinc, Šestak, Zubčić
- Teme sastanka:
 - priprema za sastanak s asistentom

8. sastanak

- Datum: 9. studenoga 2020.
- Prisustvovali: Blagaić, Kurtović, Mahović, Novinc, Šestak, Zubčić
- Teme sastanka:
 - sastanak s asistencijom - pregled dosadašnjeg rada, razrješavanje nedoumica
 - podjela zadataka (dijagrami klasa, dijagrami baza, opis arhitekture)

9. sastanak

- Datum: 2. prosinca 2020.
- Prisustvovali: Blagaić, Ilić, Kurtović, Mahović, Novinc, Šestak, Zubčić
- Teme sastanka:
 - daljnja podjela rada na implementaciji
 - autentifikacija i autorizacija korisnika

10. sastanak

- Datum: 4. prosinca 2020.
- Prisustvovali: Blagaić, Ilić, Kurtović, Mahović, Novinc, Šestak, Zubčić
- Teme sastanka:
 - sastanak s asistentom - demonstracija dosadašnjeg rada

11. sastanak

- Datum: 22. prosinca 2020.
- Prisustvovali: Blagaić, Ilić, Kurtović, Mahović, Novinc, Šestak, Zubčić
- Teme sastanka:
 - plan dalnjeg napretka
 - podjela zadataka

12. sastanak

- Datum: 27. prosinca 2020.
- Prisustvovali: Blagaić, Ilić, Kurtović, Mahović, Novinc, Šestak, Zubčić
- Teme sastanka:
 - usklađivanje frontenda i backenda, razrješavanje nedoumica

13. sastanak

- Datum: 30. prosinca 2020.
- Prisustvovali: Blagaić, Ilić, Kurtović, Mahović, Novinc, Šestak, Zubčić
- Teme sastanka:
 - plan dalnjeg napretka
 - podjela rada na dokumentaciji
 - pokretanje aplikacije na poslužitelju Heroku

14. sastanak

- Datum: 4. siječnja 2020.
- Prisustvovali: Blagaić, Ilić, Kurtović, Mahović, Novinc, Šestak, Zubčić
- Teme sastanka:
 - pregled dosadašnjeg rada
 - dogovor o završnim detaljima implementacije

15. sastanak

- Datum: 8. siječnja 2020.
- Prisustvovali: Blagaić, Ilić, Kurtović, Mahović, Novinc, Šestak, Zubčić
- Teme sastanka:
 - sastanak s asistentom - demonstracija alfa inačice

16. sastanak

- Datum: 14. siječnja 2020.
- Prisustvovali: Blagaić, Ilić, Kurtović, Mahović, Novinc, Šestak, Zubčić
- Teme sastanka:
 - završni sastanak; revizija dokumentacije i funkcionalnosti aplikacije

Tablica aktivnosti

	Alen Kurtović	Sven Blagaić	Petra Ilić	Bruna Mahović	Dario Novinc	Josip Šestak	Jakov Zubčić
Upravljanje projektom	4				1		1
Opis projektnog zadatka						4	1
Funkcionalni zahtjevi					2		
Opis pojedinih obrazaca	1	2	1	2		2	1
Dijagram obrazaca			3:30				
Sekvencijski dijagrami			4				
Opis ostalih zahtjeva			1				
Arhitektura i dizajn sustava	3	2	2	2	2	3	2
Baza podataka	7				1	4	2
Dijagram razreda					3		3
Dijagram stanja	2						
Dijagram aktivnosti			2				
Dijagram komponenti						5	
Korištene tehnologije i alati	1						
Ispitivanje programskog rješenja	1					1	3
Dijagram razmještaja			0:30				
Upute za puštanje u pogon			1:30				
Dnevnik sastajanja			1				
Zaključak i budući rad						2	
Popis literature	0:30						
Izrada frontend-a		50		30		2	
Izrada backend-a	3		15		30		10
Spajanje frontend-a i backend-a		35		25	35		1:30
Deploy na Heroku	6						

Dijagrami pregleda promjena

dio 2. revizije

Prenijeti dijagram pregleda promjena nad datotekama projekta. Potrebno je na kraju projekta generirane grafove s gitlaba prenijeti u ovo poglavlje dokumentacije. Dijagrami za vlastiti projekt se mogu preuzeti s gitlab.com stranice, u izborniku Repository, pritiskom na stavku Contributors.