

Veleučilište u Rijeci

Poslovni odjel

## **PLATFORMA ZA PRONALAZAK SMJEŠTAJA**

Gabrijel Lukić

Rijeka, 2022.



Veleučilište u Rijeci

Poslovni odjel

## **PLATFORMA ZA PRONALAZAK SMJEŠTAJA**

Programsko inženjerstvo

(projektna dokumentacija)

MENTOR:

Dr. sc. Marin Kaluža, prof. visoke škole

STUDENT:

Gabrijel Lukić

MBS: 2422000121/21

Rijeka, lipanj, 2022.

## Sadržaj

1. Opis projekta .....	6
1.1 Specifikacija zahtjeva: .....	6
2. Arhitektura sustava .....	7
3. Dijagrami .....	8
3.1 Dijagrami Akcija .....	8
3.2 Sekvencijalni dijagram .....	18
3.3 Dijagram Klasa .....	23
3.4 Use-Case Diagram .....	26
4. WireFrame .....	27
5. Struktura mapa (Frontend i Backend) .....	30
5.1 Prijava korisnika .....	30
5.2 Registracija korisnika .....	30
5.3 Dodavanje/Izmijena smjestaja .....	30
5.4 Razgovori .....	30
5.5 Izmijena osobnih podataka .....	31
5.6 Prikaz smještaja .....	31
6. Prikaz izgleda završene aplikacije .....	31
7. Prikaz dijelova koda .....	33
7.1 Prikaz isječka koda za real-time razgovore .....	33
7.2 Prikaz isječka koda za dodavanje smještaja .....	34
7.3 Prikaz isječka koda za registraciju korisnika .....	34
8. Sigurnosno testiranje .....	35
8.1 CSP: Wildcard Directive i Content Security Policy (CSP) Header Not Set .....	35
8.2 Cross-Domain Misconfiguration .....	35
8.3 Directory Browsing .....	36
8.4 Anti-clickjacking Header .....	36
8.5 Server Leaks Information via "X-Powered-By" HTTP Response Header Field .....	36
9. Testiranje aplikacije .....	36
10. Dovršenost aplikacije .....	37
11. Zaključak .....	38



## 1.Opis projekta

Cilj ovog projekta je izraditi web-stranicu na kojoj će korisnici/klijenti moći pretraživati i dodavati smještajne objekte. Svaki korisnik koji se registrira na platformu će imati mogućnost dodavanja vlastitog objekta. Svaki objekt će sadržavati informacije o samom objektu: Koje usluge nudi, površina, lokacija, slike, recenzije itd...) te informacije o autoru (Ime i prezime). Također korisnik koji je registriran će moći kontaktirati vlasnike objekta koristeći „Chat“ sustavi i dodavati objekte koji mu se sviđaju u favorite.

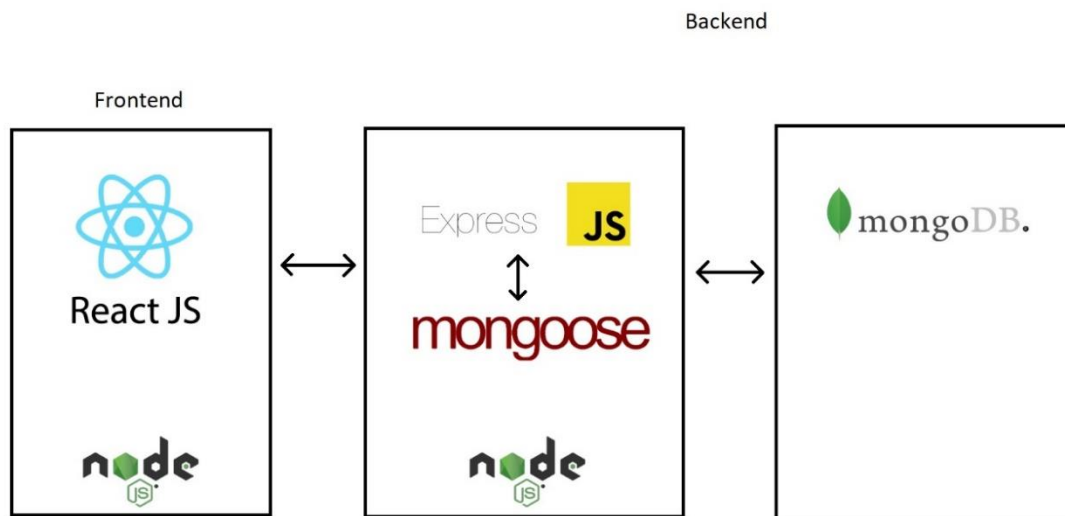
Na stranici će biti omogućeno pregledanje svih objekta bez potrebe prethodne registracije. Također će se moći sve te objekte filtrirati na različite načine. Moći će se filtrirati po dostupnosti za određene datume, po sadržaju koji smještaj nudi (TV, Klima, Wifi itd...), po ocjenama od gostiju i po popularnosti na koju će utjecati ocjena gostiju, „reply-score“ korisnika i broj koliko puta je objekt dodan u favorite.

### 1.1 Specifikacija zahtjeva:

1. Pregled svih smještaja
2. Filtriranje svih smještaja
3. Pregled pojedinog smještaja
4. Pregled pojedinog korisnika i njegovih smještaja koje on nudi na platformi
5. Svaki smještaj:
  - a. Informacije o korisniku koji ga je stavio
  - b. Galerija slika
  - c. Opis objekta
  - d. Recenzije
6. Kalendar Dostupnosti
7. Stavljanje smještaja
8. User Dashboard
9. Reply Score rangiranje
10. Review Score rangiranje
11. mogućnost kontaktiranja kroz chat
12. Google Maps integracija, pokazivanje smještaja na mapi
13. Login/Register za svakog korisnika
14. Reaktivnost

## 2.Arhitektura sustava

Za izradu ovog projekta koristit će se React za Front-end kako bi omogućili brzo i responzivno sučelje za korisnike, a podatke ćemo dohvaćati putem RestAPI koristeći axios. Što se tiče Back-Enda on će biti realiziran u obliku RestAPI koji će se izraditi koristeći Node.js sa Express.js frameworkom i Mongoose-om koji će nam omogućiti da spojimo Node.js sa MongoDB bazom podataka.

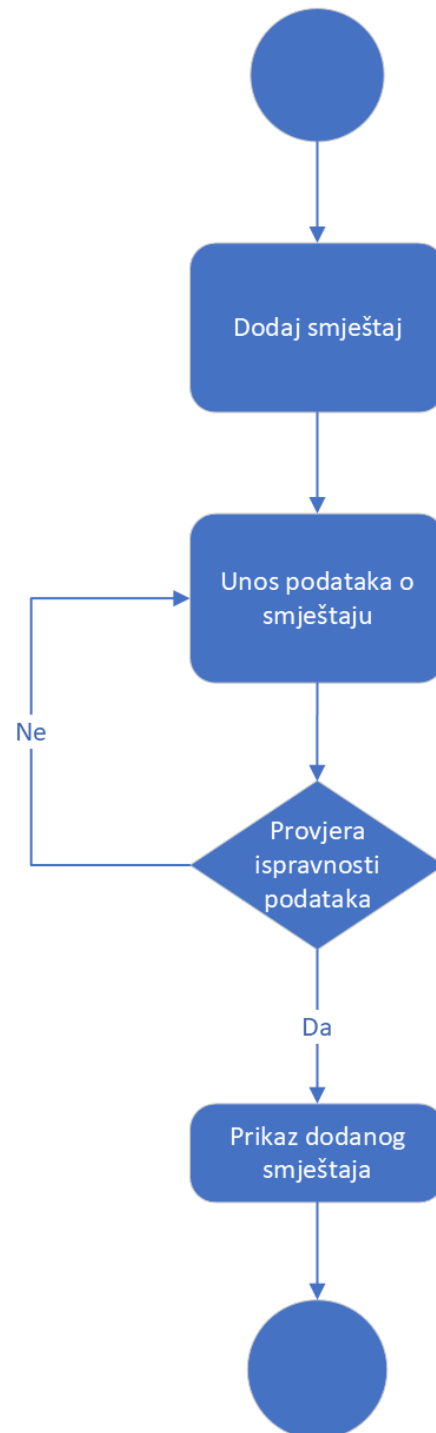


Za razvojno okruženje izabran je VS Code s obzirom da nudi dosta ekstenzija koji olakšavaju razvijanje i testiranje REST API i React aplikacije. Npr. (ES7 React Snipepts, Thunder Client)

Pregled Tehnologija	
Front-end	React
Back-end	Node.js / Express.js / Mongoose
Baza-podataka	MongoDb
Razvojno-Okruženje	VS Code

### 3. Dijagrami

#### 3.1 Dijagrami Akcija



Slika 1: Dodavanje Smještaja



**Identifikacijski sažetak:** Dodavanje smještaja

**Akteri:** Korisnik

**Preduvjeti:** Korisnik mora biti registriran

**Tijek događaja:**

**Glavni uspješni scenarij (G):**

1. Korisnik odabire gumb dodaj smještaj
2. Unosi podatke o smještaju
3. Provjera podataka
4. Prikazuje se dodani smještaj na platformi

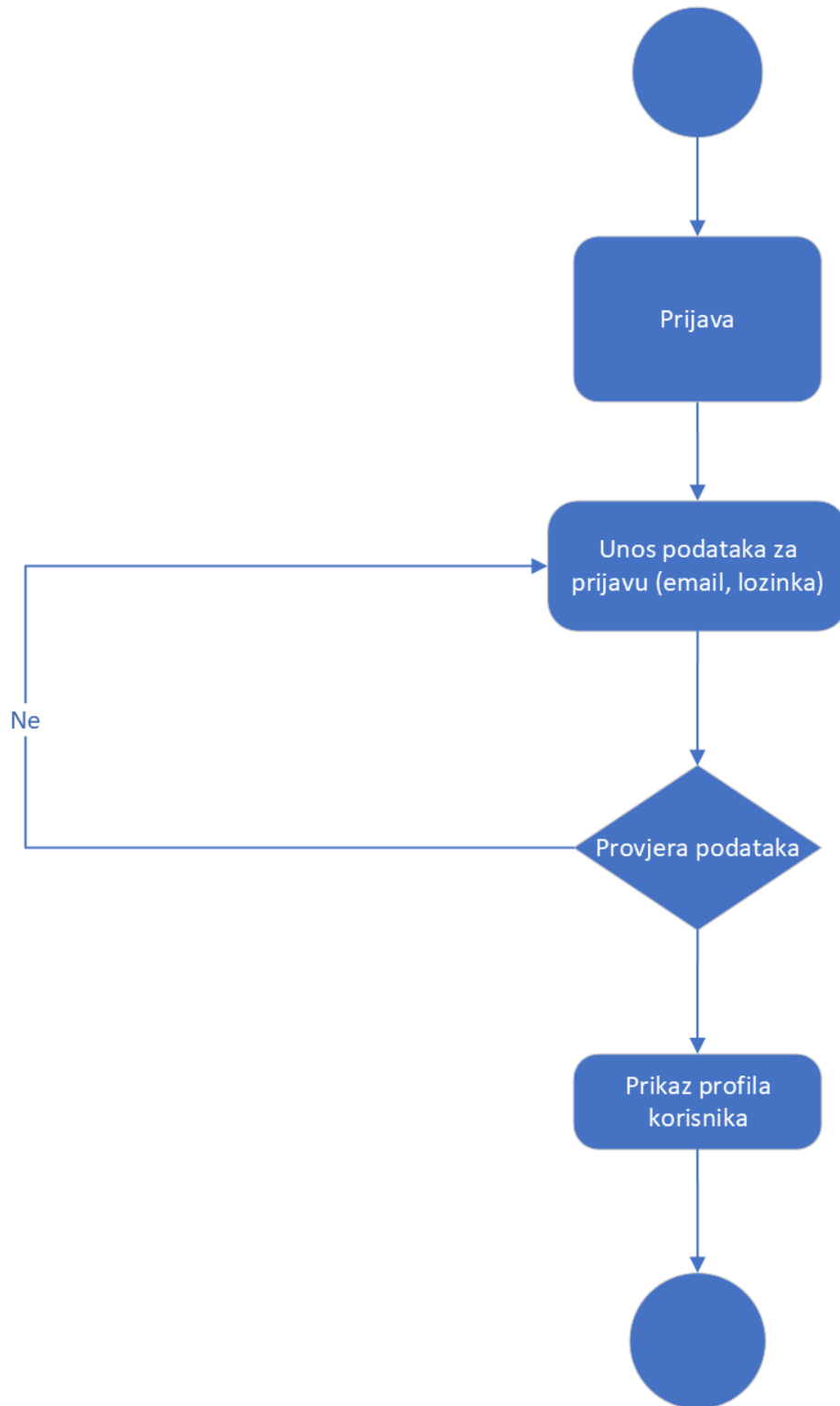
**Alternativne sekvence:**

**A1: nema**

**Sekvence s greškom:**

**E1: Neuspješna prijava u sustav – ponovni unos**

1. Iz koraka G3 – Sustav javlja korisniku da ima neispravne podatke
2. Korisnik upisuje ispravne podatke o smještaju
3. Prikazuje se dodani smještaj na platformi



Slika 2: Dijagram akcija prijava

**Identifikacijski sažetak:** Prijava u sustav

**Akteri:** Korisnik

**Preduvjeti:** Korisnik mora biti registriran

**Tijek događaja:**

**Glavni uspješni scenarij (G):**

1. Korisnik odabire gumb Prijava
2. Unosi podatke za prijavu
3. Provjera podataka
4. Prikazuje se profil korisnika

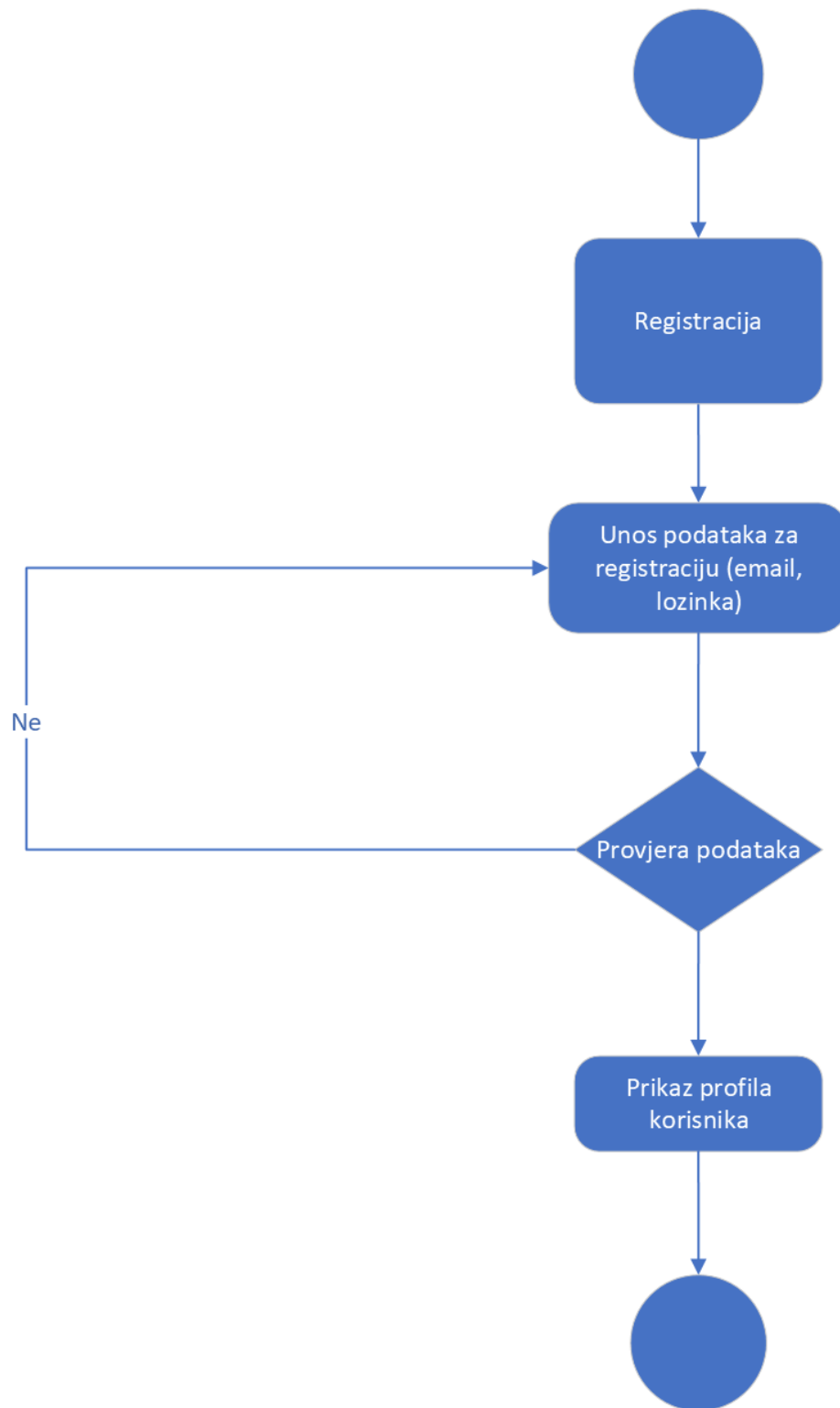
**Alternativne sekvence:**

**A1: nema**

**Sekvence s greškom:**

**E1: Neuspješna prijava u sustav – ponovni unos**

1. Iz koraka G3 – Sustav javlja korisniku da je unio neispravne podatke
2. Korisnik upisuje ispravne podatke za prijavu
3. Prikazuje se profil korisnika



Slika 3: Dijagram akcija registracija

**Identifikacijski sažetak:** Registracija korisnika

**Akteri:** Korisnik

**Preduvjeti:** -

**Tijek događaja:**

**Glavni uspješni scenarij (G):**

1. Korisnik odabire gumb Registracija
2. Unosi podatke za registraciju
3. Provjera podataka
4. Prikazuje se profil korisnika

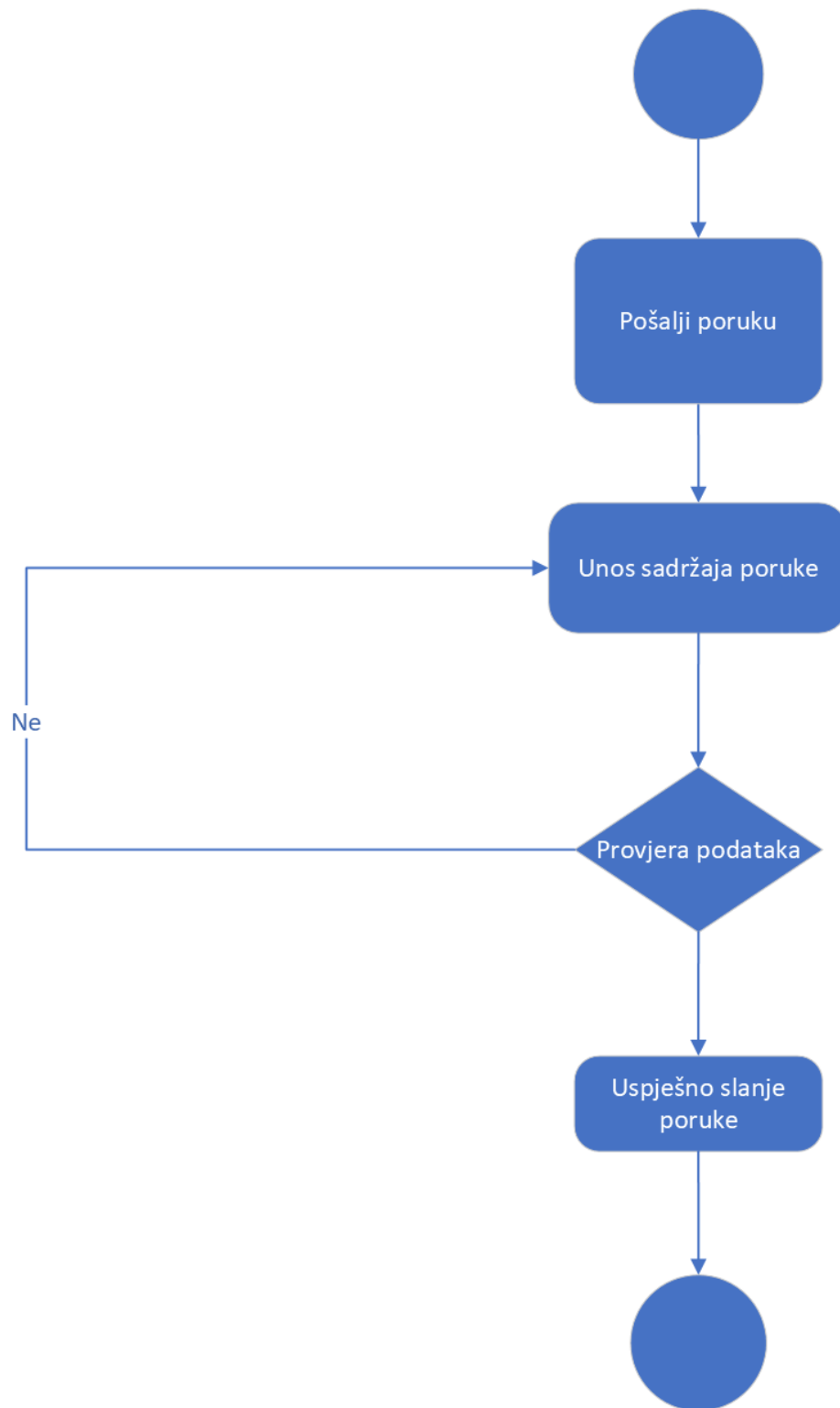
**Alternativne sekvence:**

**A1: nema**

**Sekvence s greškom:**

**E1: Neuspješna prijava u sustav – ponovni unos**

1. Iz koraka G3 – Sustav javlja korisniku da je unio neispravne podatke
2. Korisnik upisuje ispravne podatke za prijavu
3. Prikazuje se profil korisnika



Slika 4: Dijagram akcija slanje poruke

**Identifikacijski sažetak:** Slanje poruka

**Akteri:** Korisnik

**Preduvjeti:** Korisnik mora biti registriran

**Tijek događaja:**

**Glavni uspješni scenarij (G):**

1. Korisnik odabire gumb Pošalji poruku
2. Unosi sadržaj poruke
3. Provjera podataka (dali korisnik kojemu se šalje poruka postoji)
4. Slanje poruke

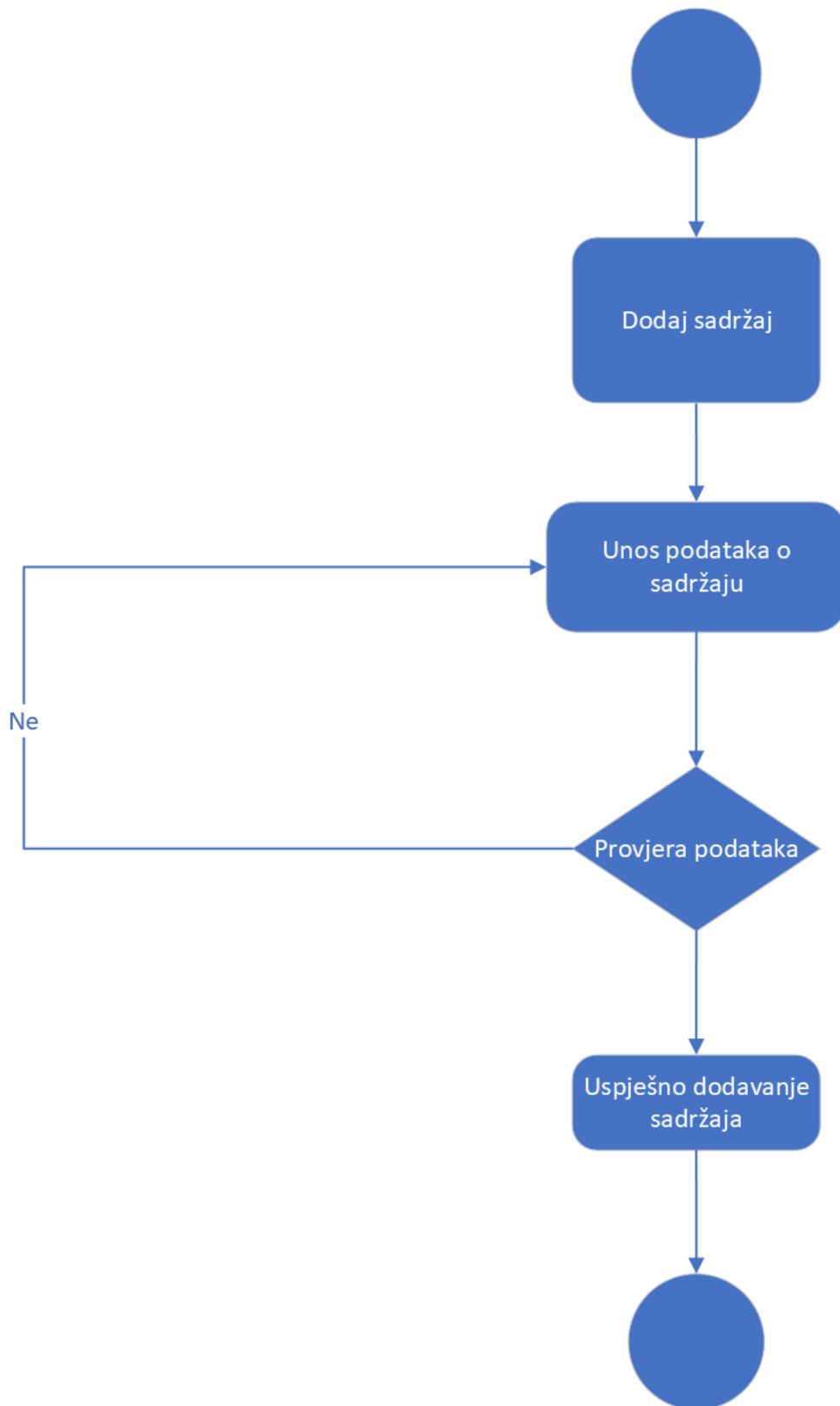
**Alternativne sekvence:**

**A1: nema**

**Sekvence s greškom:**

**E1: Neuspješna prijava u sustav – ponovni unos**

1. Iz koraka G3 – Sustav javlja korisniku da je unio neispravne podatke
2. Korisnik upisuje ispravne podatke za slanje poruke
3. Poruka se šalje drugom korisniku



Slika 5: Dijagram akcija dodavanje sadržaja



**Identifikacijski sažetak:** Dodavanje sadržaja

**Akteri:** Administrator

**Preduvjeti:** Mora biti administrator

**Tijek događaja:**

**Glavni uspješni scenarij (G):**

1. Korisnik odabire gumb Dodaj sadržaj
2. Unosi podatke o sadržaju
3. Provjera unesenih podataka o sadržaju
4. Sadržaj se dodaje na platformu

**Alternativne sekvence:**

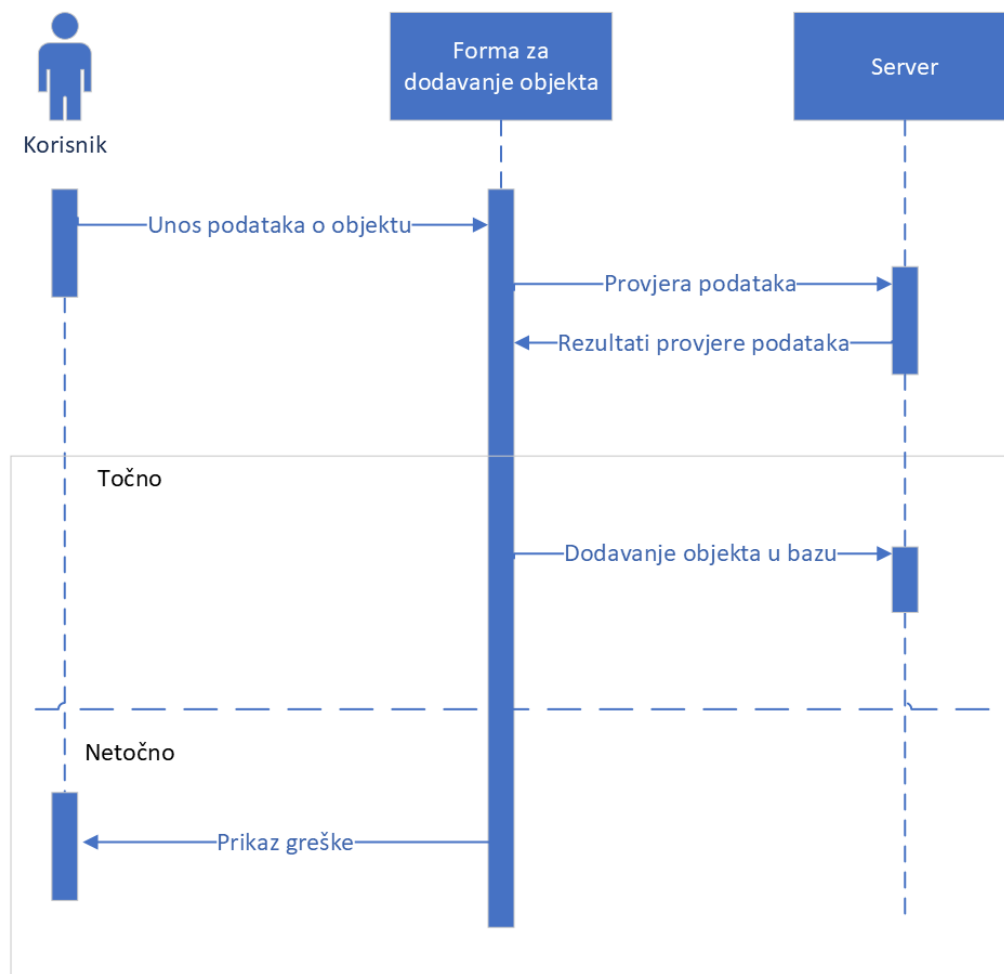
**A1: nema**

**Sekvence s greškom:**

**E1: Neuspješna prijava u sustav – ponovni unos**

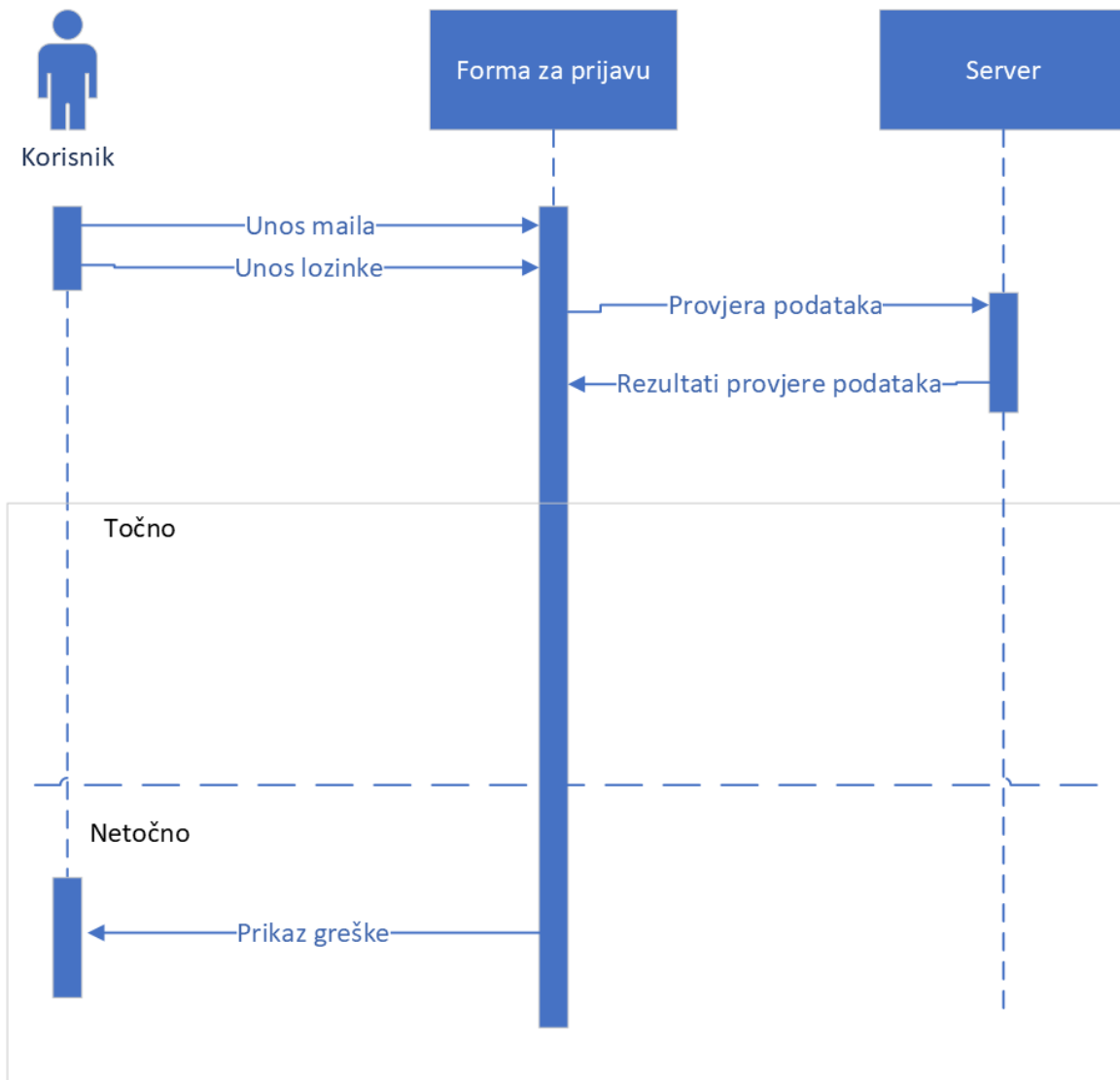
1. Iz koraka G3 – Sustav javlja korisniku da je unio neispravne podatke
2. Korisnik upisuje ispravne podatke za dodavanje sadržaja
3. Prikazuje se uspješno dodani sadržaj

### 3.2 Sekvencijalni dijagram



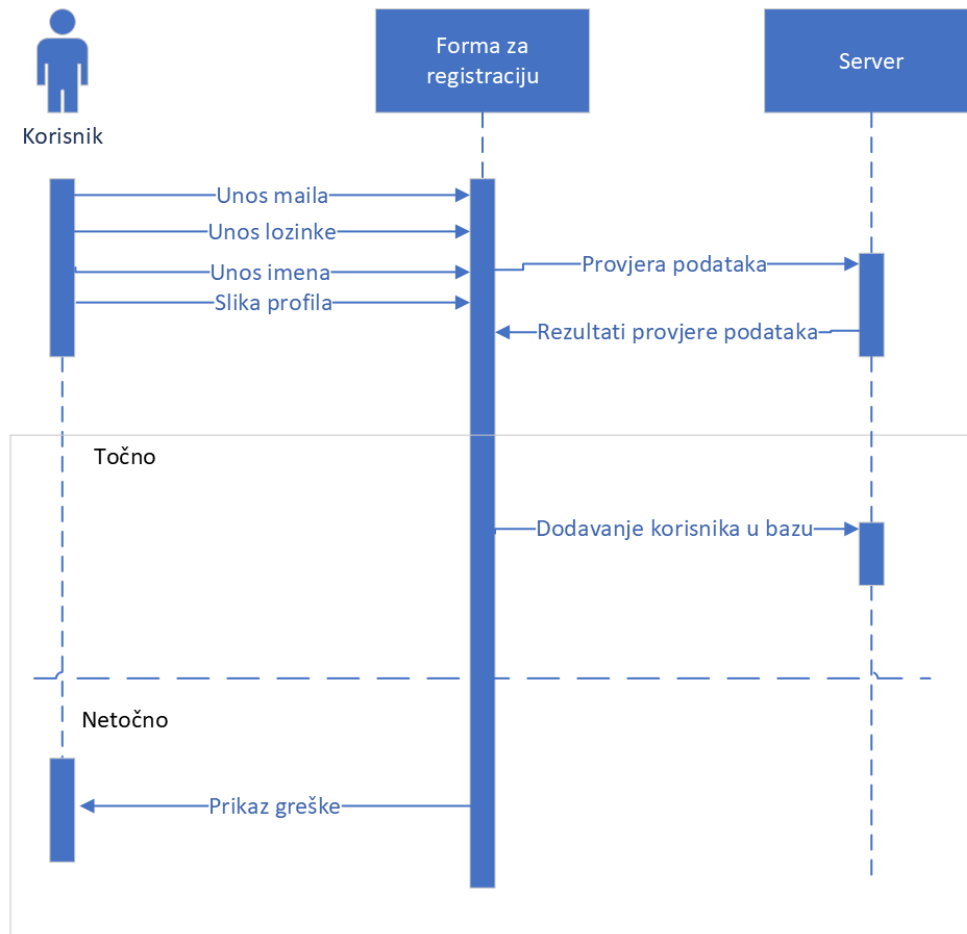
Slika 6: Sekvencijalni dijagram dodavanje objekta

Kako bih korisnik dodao objekt mora prvo otvoriti formu za dodavanje objekta. Unutar te forme unosi podatke o objektu (Ime, lokaciju, sadržaj itd.). Kada je ispunio sve potrebne informacije i želi dodati objekt na platformu server vrši provjeru dali su podaci ispravno uneseni te ako jesu objekt se dodaje u bazu, a ako pak nisu korisniku se prikazuje greška.



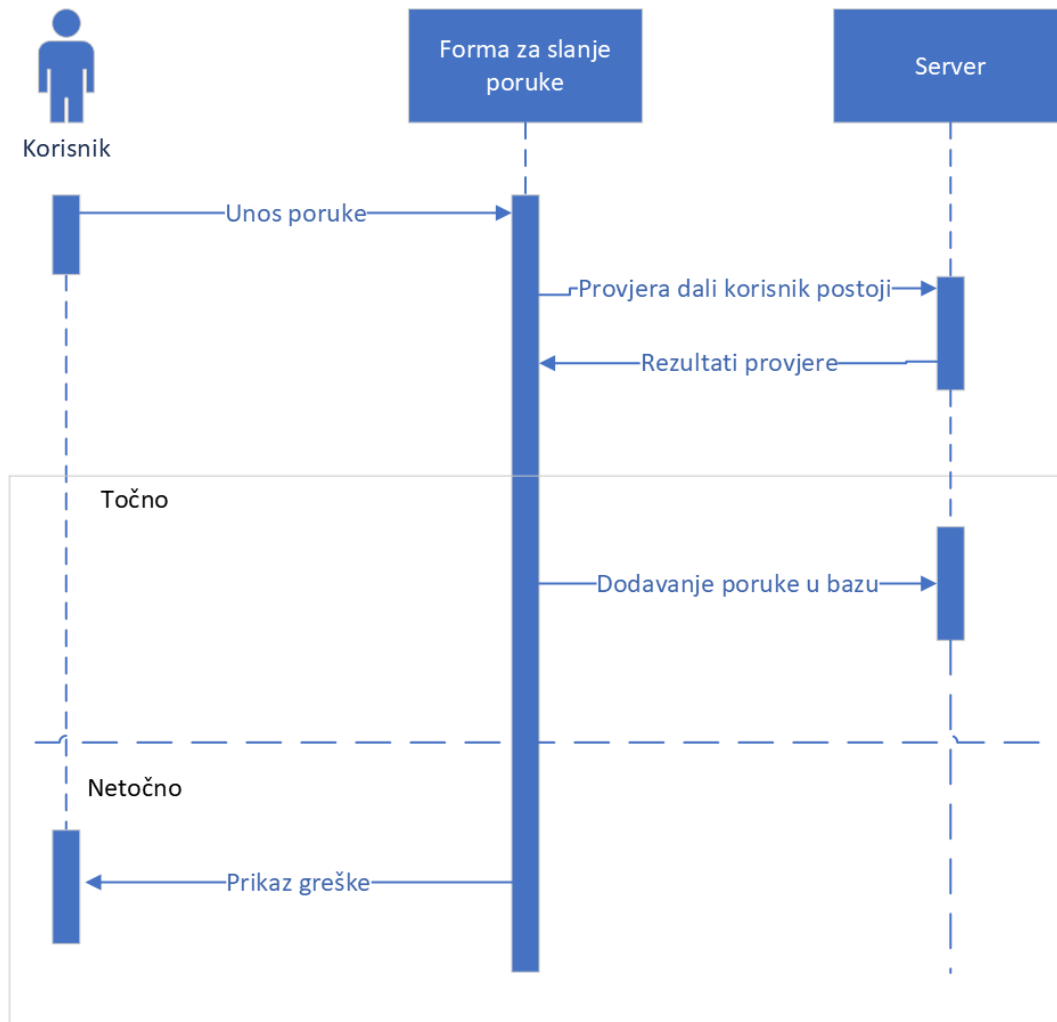
Slika 7: Sekvencijalni dijagram Prijava

Kako bih se korisnik prijavio u sustav mora prvo otvoriti formu za prijavu. Unutar te forme unosi svoje pristupne podatke (korisničko ime i šifru). Kada je ispunio sve potrebne informacije i želi se prijaviti na platformu server vrši provjeru dali su podaci ispravno uneseni te ako jesu korisnika se prijavljuje u sustav, a ako pak nisu korisniku se prikazuje greška.



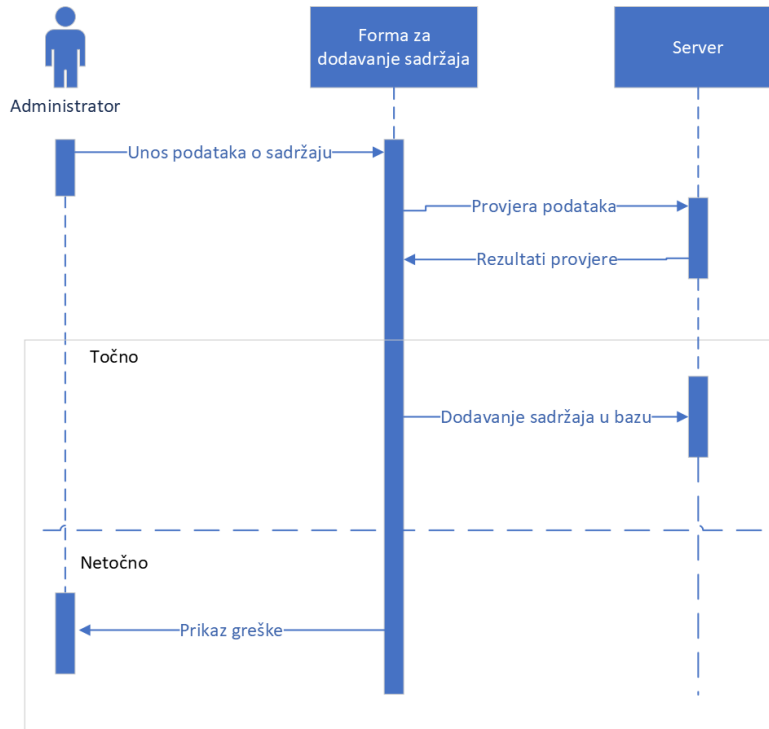
Slika 8: Sekevencijalni dijagram registracija

Kako bih se korisnik registrirao na platformu potrebno je otvoriti formu za registraciju. Unutar te forme unosi podatke o sebi (Email, ime, lozinku, sliku profila itd.). Kada je ispunio sve potrebne informacije i želi se registrirati server vrši provjeru dali su podaci ispravno uneseni te ako jesu korisnik se dodaje u bazu te mu je omogućen pristup platformi.



Slika 9: Sekevencijalni dijagram slanja poruke

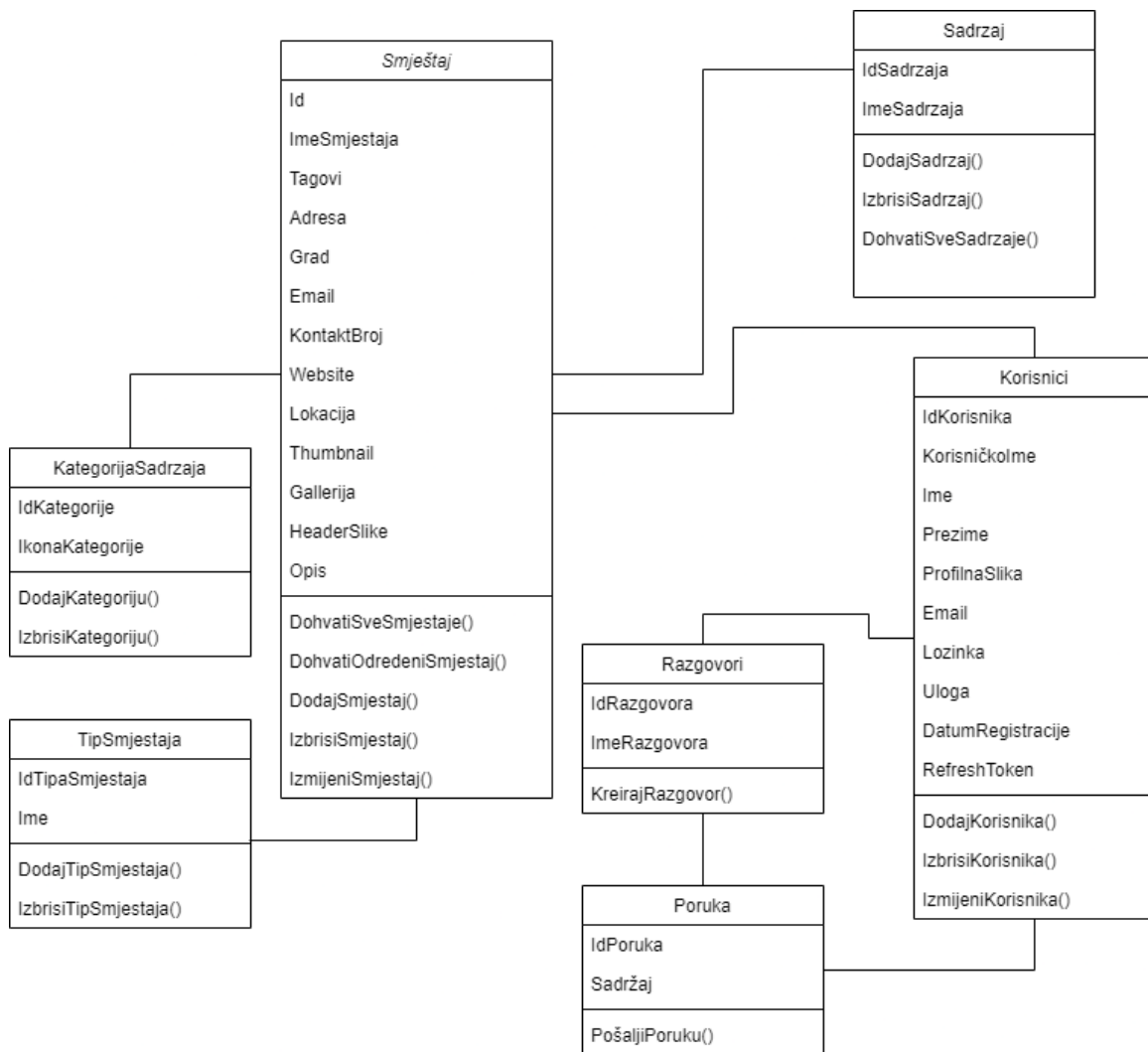
Kako bih korisnik poslao poruku drugom korisniku na platformi, potrebno je unutar njegovog smještaja otvoriti formu za slanje poruka. Unutar te forme unosi sadržaj poruke te kada klikne na gumb „Pošalji“, server provjerava dali drugi korisnik uopće postoji te ako postoji mu dostavlja poruku inače izbacuje grešku.



Slika 10: Sekevencijalni dijagram dodavanje sadržaja

Za dodavanje sadržaja koje kasnije korisnici mogu izabrati kada dodaju vlastite objekte administratora mora ići na formu za dodavanje sadržaja. Zatim unijeti podatke o svakom od sadržaja. Zatim server provjerava dali su podaci ispravno uneseni te ako jesu se taj sadržaj dodaje u bazu podataka i postaje dostupan ostalim korisnicima pri dodavanju smještaja. U slučaju neispravnosti podataka poakzuje se greška.

### 3.3 Dijagram Klasa



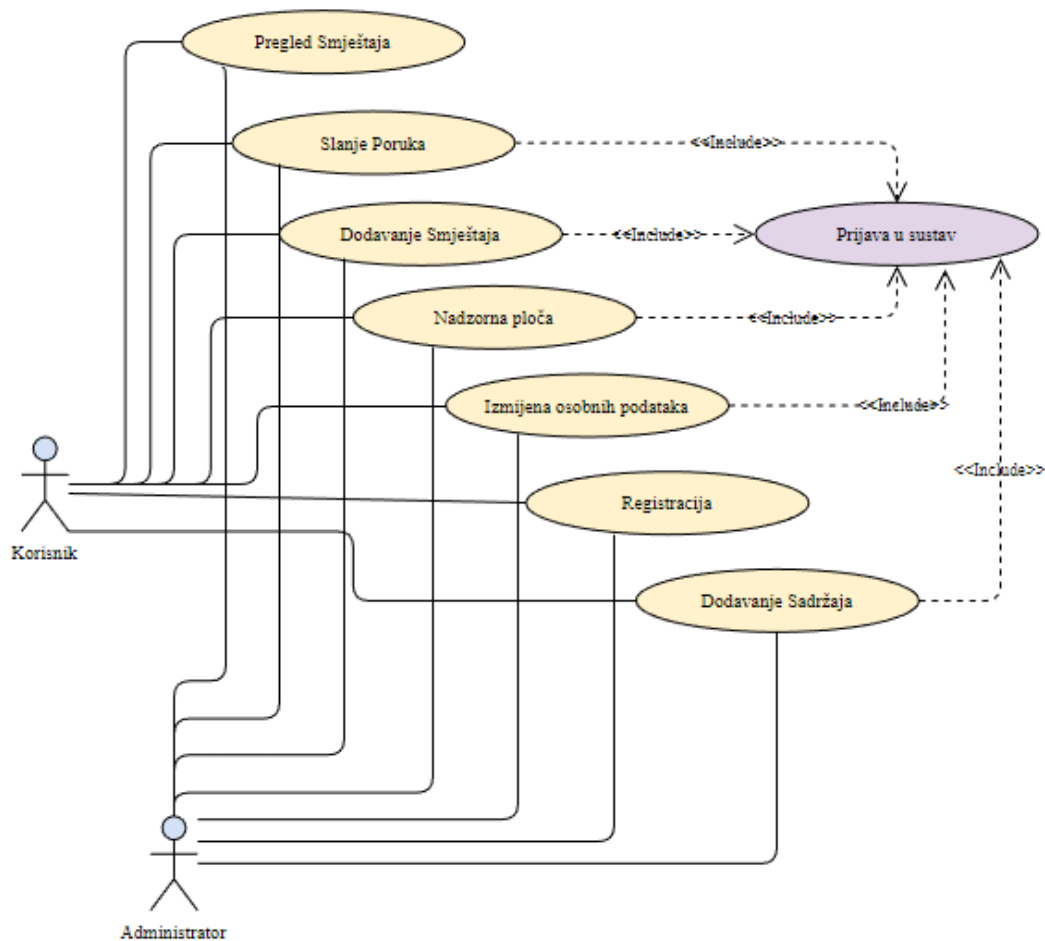
Slika 11: Dijagram Klasa

Naziv Klase	Naziv Metode	Opis Metode
Smještaj	DohvatiSveSmjestaje	Služi za dohvaćanje svih smještaja u bazi. Koristi se za prikaz smještaja.
	DohvatiOdređeniSmjestaj	Služi kako bih dohvatili samo jedan određeni smještaj. Koristi se kod prikaza pojedinog smještaja.
	DodajSmjestaj	Služi za dodavanje smještaja u bazu podataka. Koristi se kada korisnik želi dodati smještaj na platformu
	IzbrisiSmjestaj	Služi za brisanje smještaja iz baze podataka. Koristi se kada korisnik želi izbrisati svoj smještaj sa platforme.
	IzmijeniSmjestaj	Služi za izmijenu podataka smještaja koji se trenutno nalazi na platformi. Koristi se kada korisnik želi izmijeniti podatke već postojećeg smještaja.
KategorijaSadržaja	DodajKategoriju	Služi za dodavanje kategorije sadržaja. Koristi se kada administrator želi dodati kategorije sadržaja.
	IzbrisiKategoriju	Služi za brisanje kategorije sadržaja. Koristi se kada administrator želi izbrisati kategoriju sadržaja.
	DohvatiKategorijeSadržaja	Služi za dohvaćanje svih kategorija sadržaja. Koristi se kod dodavanja sadržaja.
TipSmjestaja	DodajTipSmjestaja	Služi za dodavanje tipa smještaja. Koristi se od strane administratora.
	IzbrisiTipSmjestaja	Služi za brisanje Tipa smještaja. Koristi se od strane administratora.
	DohvatiTipoveSmjestaja	Služi za dohvaćanje svih tipova objekta. Koristi se pri dodavanju smještaja na platformu.



Sadržaj	DodajSadržaj	Služi za dodavanje sadržaja koji se kasnije nude na izbor pri dodavanju objekta na stranicu.
	IzbrisiSadržaj	Služi za brisanje određenog sadržaja. Koristi se od strane admina.
	DohvatiSveSadržaje	Služi za dohvaćanje svih sadržaja. Služi kako bih dohvatili sav sadržaj i prikazali ga korisniku na raznim lokacijama (Dodavanje objekta, pregled objekta itd.)
Korisnici	DodajKorisnika	Služi za dodavanje korisnika u bazu. Koristi se pri registraciji
	IzbrisiKorisnika	Služi za brisanje korisnika. Koristi se pri brisanju profila određenog korisnika. Koristi se od strane administratora.
	IzmijeniKorisnika	Služi za izmijenu podataka korisnika. Koristi se od strane korisnika za promjenu vlastitih podataka.
Razgovori	DohvatiRazgovore	Služi za dohvaćanje svih razgovora koje je određeni korisnik pokrenio.
	KreirajRazgovor	Služi za kreiranje razgovora između dva korisnika. Koristi se kada jedan korisnik želi stupiti u kontakt s drugim korisnikom.
Poruka	DohvatiPoruke	Služi za dohvaćanje svih poruka između dva korisnika.
	PosaljiPoruku	Služi za slanje poruke određenom korisniku.

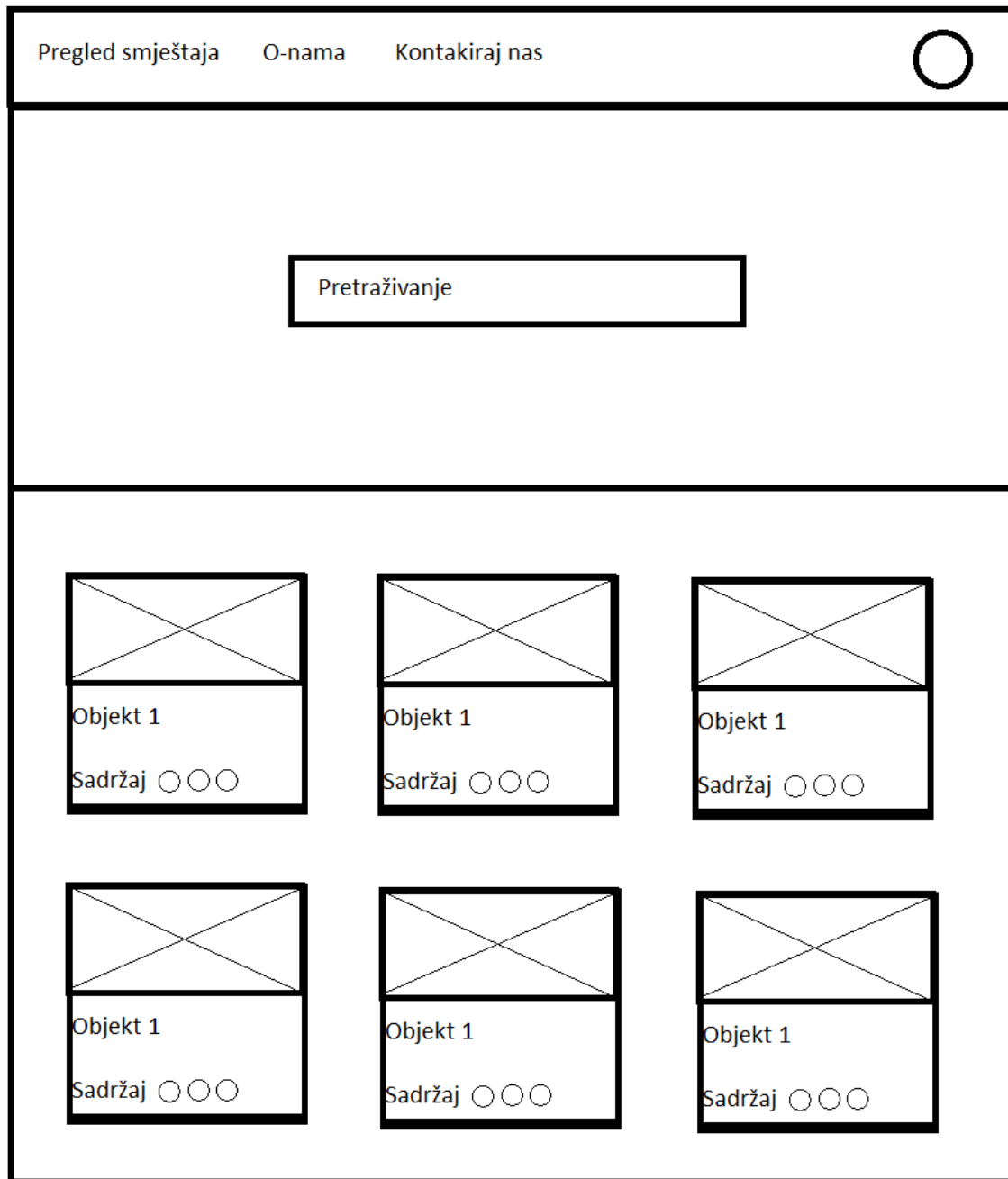
### 3.4 Use-Case Diagram



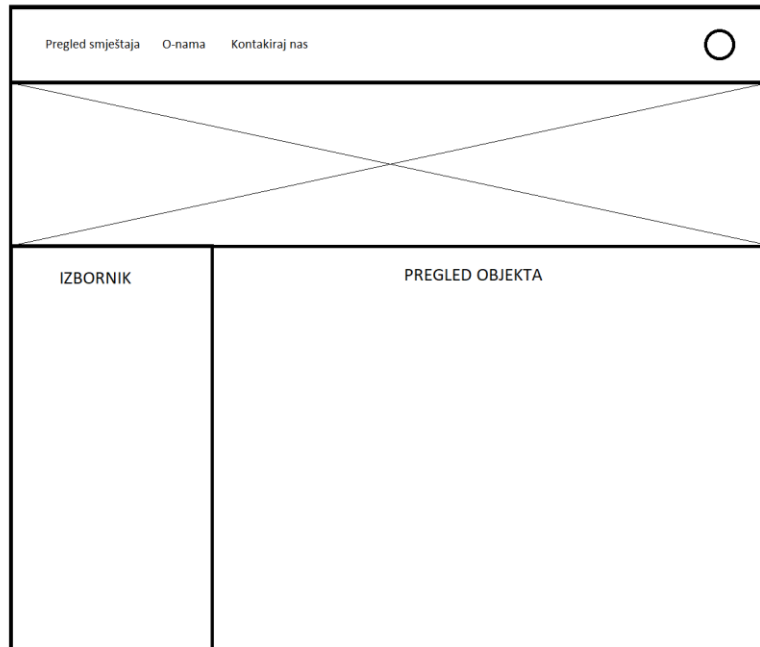
Slika 12: Use-case Diagram

Iz ovog Use-Case diagrama vidljivo je da Administrator i Korisnik imaju gotovi pa isti pristup funkcionalnostima. Korisnik nema jedino pristup za dodavanje sadržaja koji se nudi na izbor pri dodavanju smještaja. Također samo dvije funkcionalnosti ne zahtijevaju da se bude prijavljen u sustav (Registracija, Pregled smještaja) dok ostale to zahtijevaju (Slanje Poruka, Dodavanje Smještaja, Nadzorna Ploča, Izmijena osobnih podataka, Dodavanje Smještaja).

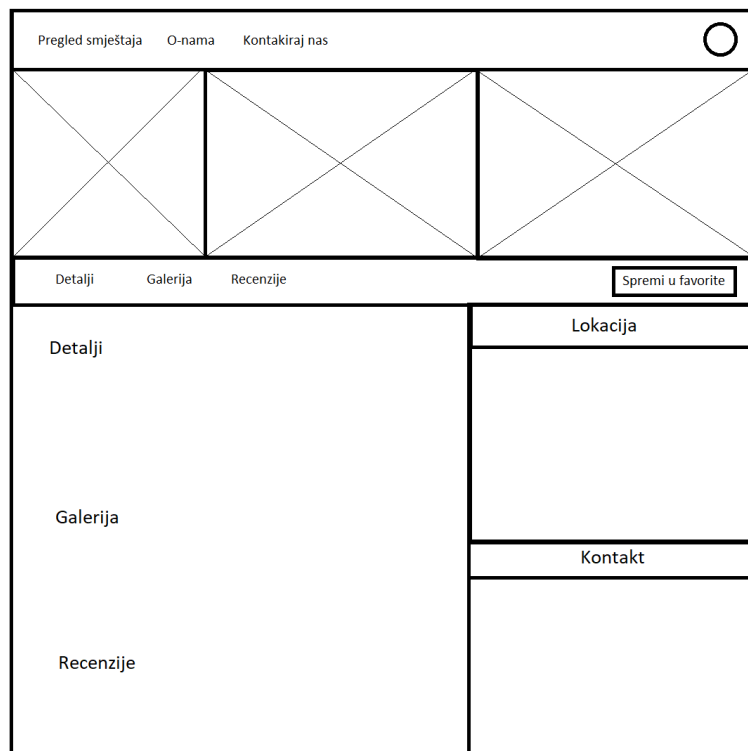
## 4. WireFrame



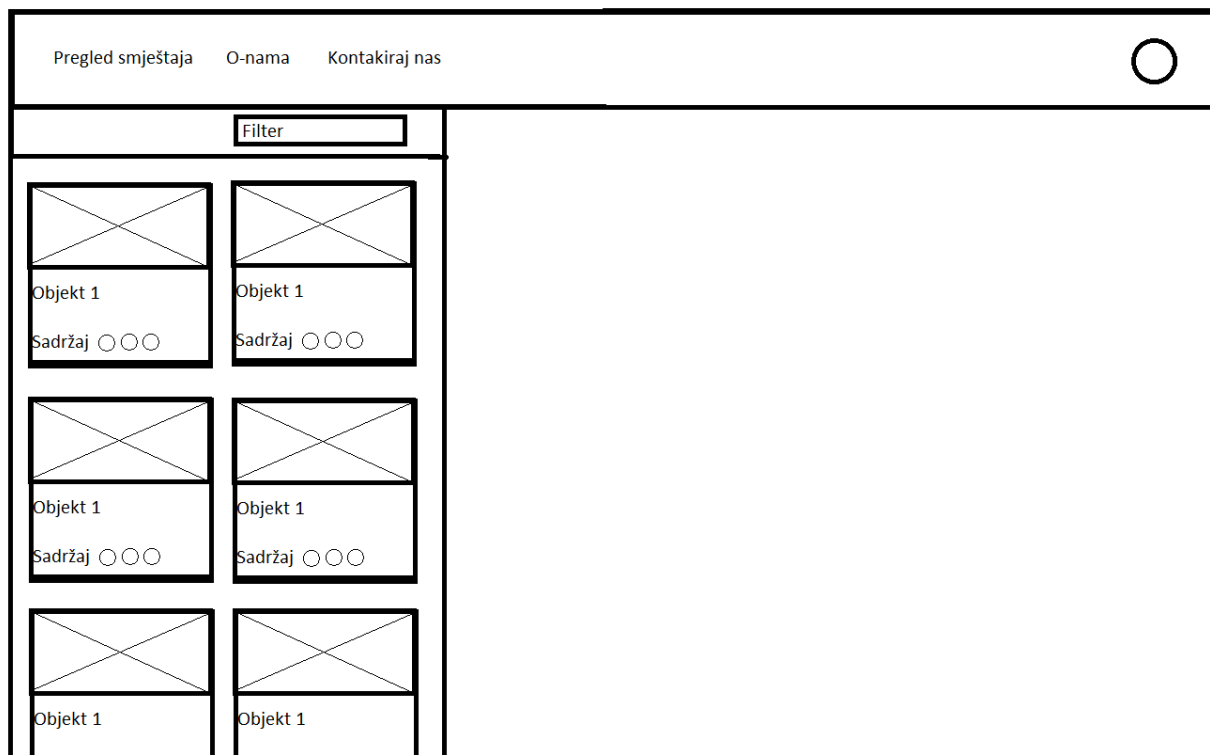
Slika 13: Wireframe – Naslovna



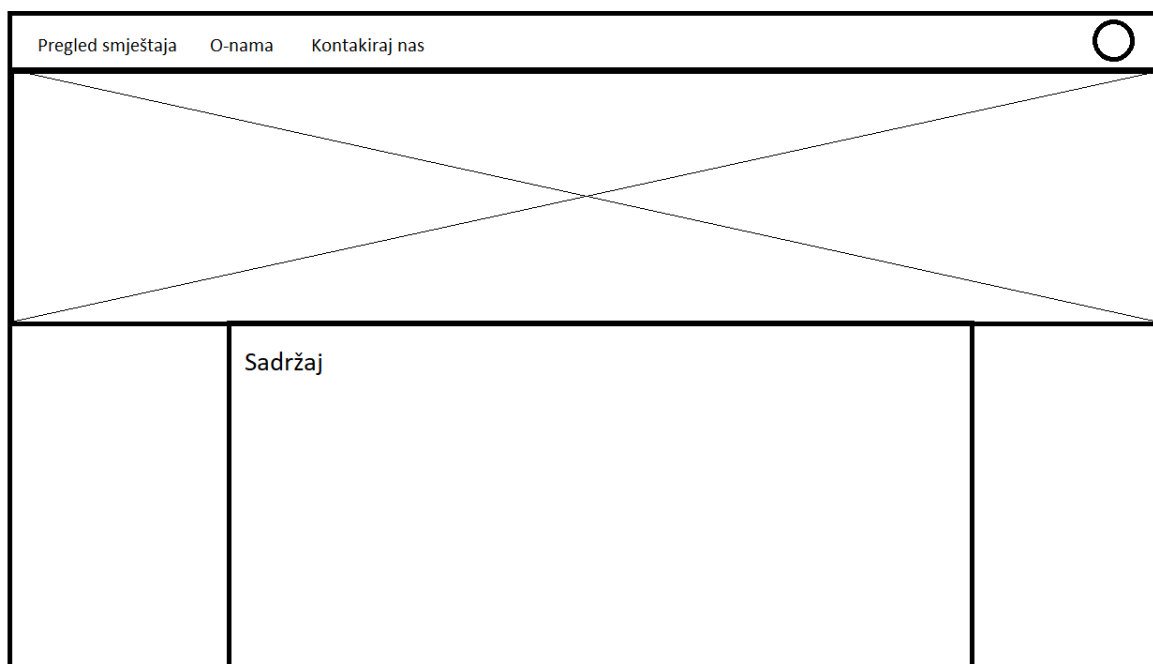
Slika 14: Wireframe – Profil



Slika 15: Wireframe – Objekt

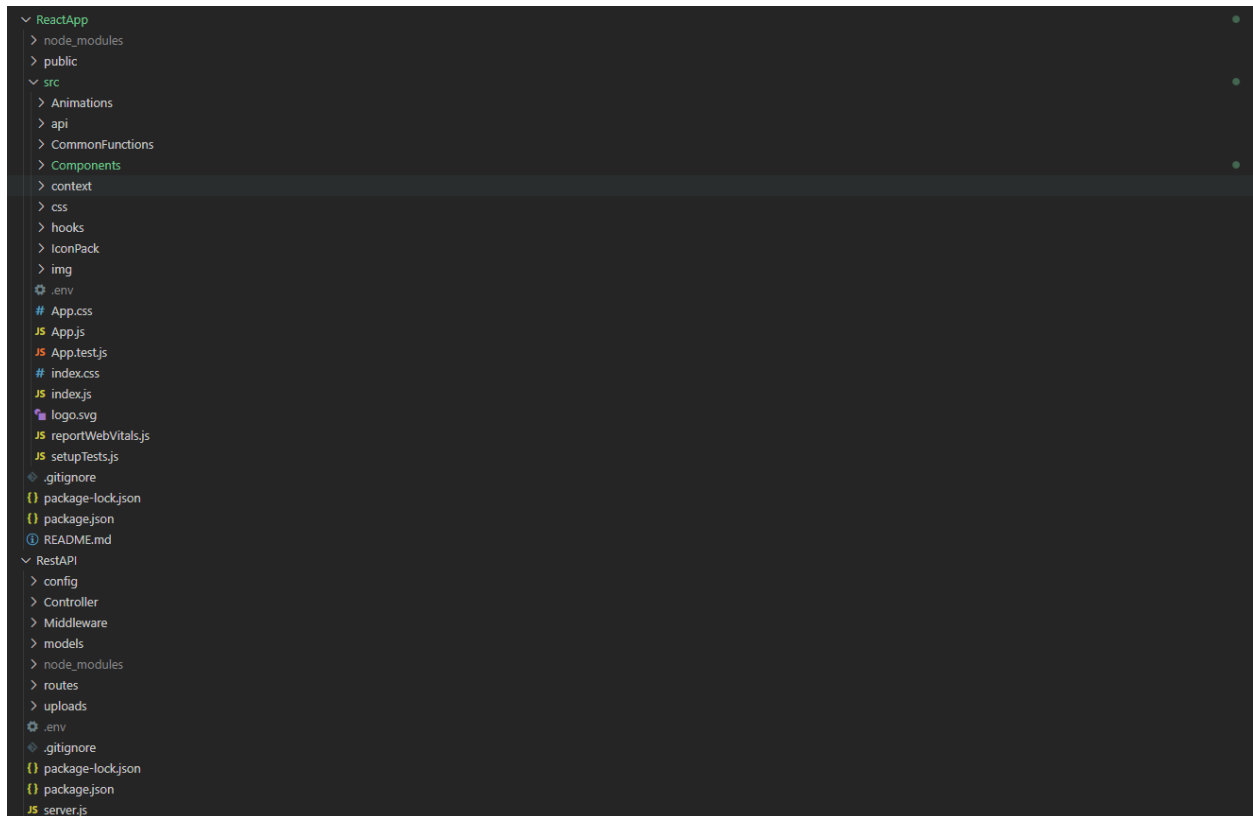


Slika 16: Wireframe – Pretraživanje



Slika 17: Wireframe - Osnovna Stranica (O-nama, politika privatnosti itd.)

## 5. Struktura mapa (Frontend i Backend)



Slika 18: Prikaz Strukutre mapa

### 5.1 Prijava korisnika

<b>Prikaz stranice za prijavu</b>	<b>ReactApp-&gt;src-&gt;Login.js</b>
<b>Backend za prijavu</b>	RestAPI->routes->authRoutes.js

### 5.2 Registracija korisnika

<b>Prikaz stranice za registraciju</b>	<b>ReactApp-&gt;src-&gt;Register.js</b>
<b>Backend za registraciju</b>	RestAPI->routes->authRoutes.js

### 5.3 Dodavanje/Izmijena smjestaja

<b>Prikaz stranice za dodavanje smjestaja</b>	<b>ReactApp-&gt;src-&gt;Components-&gt;AddAccommodation-&gt;AddAccommodationForm.js</b>
<b>Backend za dodavanje smjestaja</b>	RestAPI->routes->accommodationRoutes.js

### 5.4 Razgovori

<b>Prikaz stranice za razgovore</b>	<b>ReactApp-&gt;src-&gt;ChatPage.js</b>
<b>Backend za razgovore</b>	RestAPI->routes->chatRoutes.js

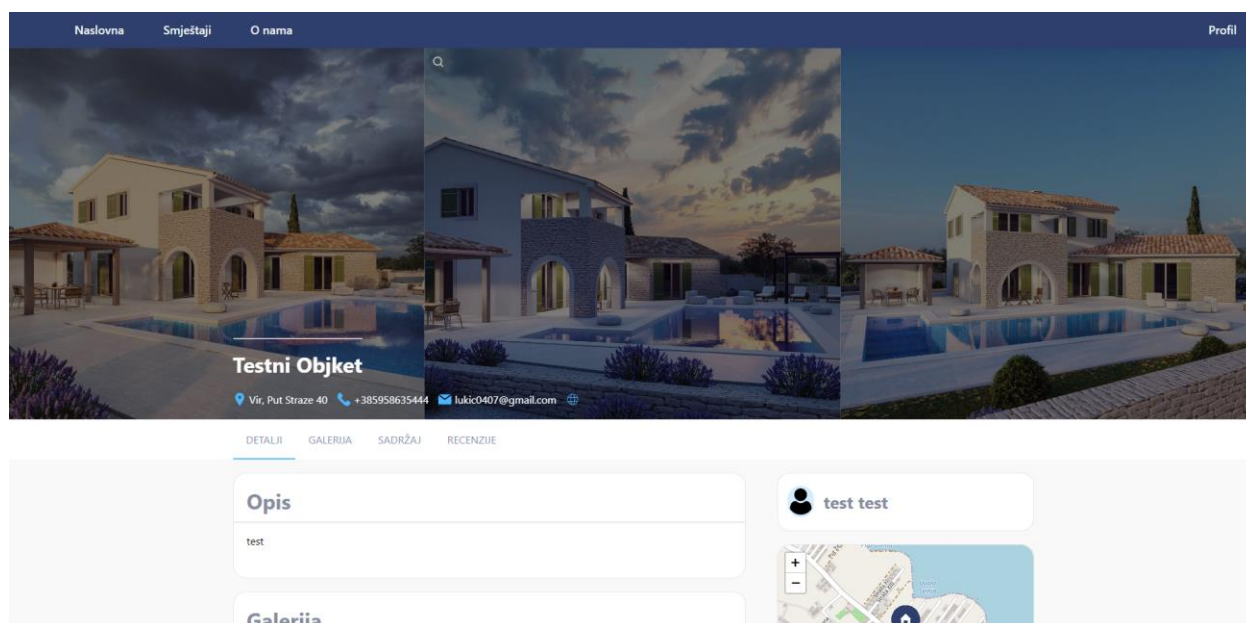
## 5.5 Izmijena osobnih podataka

<b>Prikaz stranice za izmijenu osobnih podataka</b>	<b>ReactApp-&gt;src-&gt;MyProfile.js</b>
<b>Backend za izmijenu osobnih podataka</b>	RestAPI->routes->userRoute.js

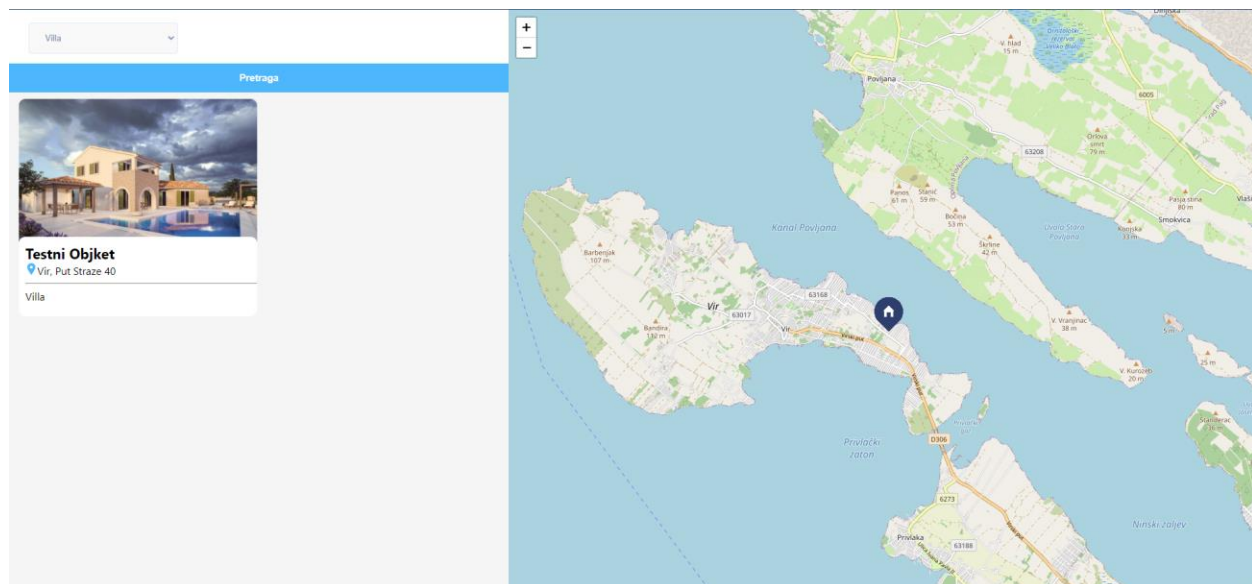
## 5.6 Prikaz smještaja

<b>Prikaz stranice određenog smještaja</b>	<b>ReactApp-&gt;src-&gt; AccommodationSingle.js</b>
<b>Prikaz stranice svih smještaja</b>	ReactApp->src-> Accommodation.js
<b>Backend za prikaz određenog smještaja</b>	RestAPI->routes->accommodationRoutes.js
<b>Backend za prikaz svih smještaja</b>	RestAPI->routes->accommodationRoutes.js

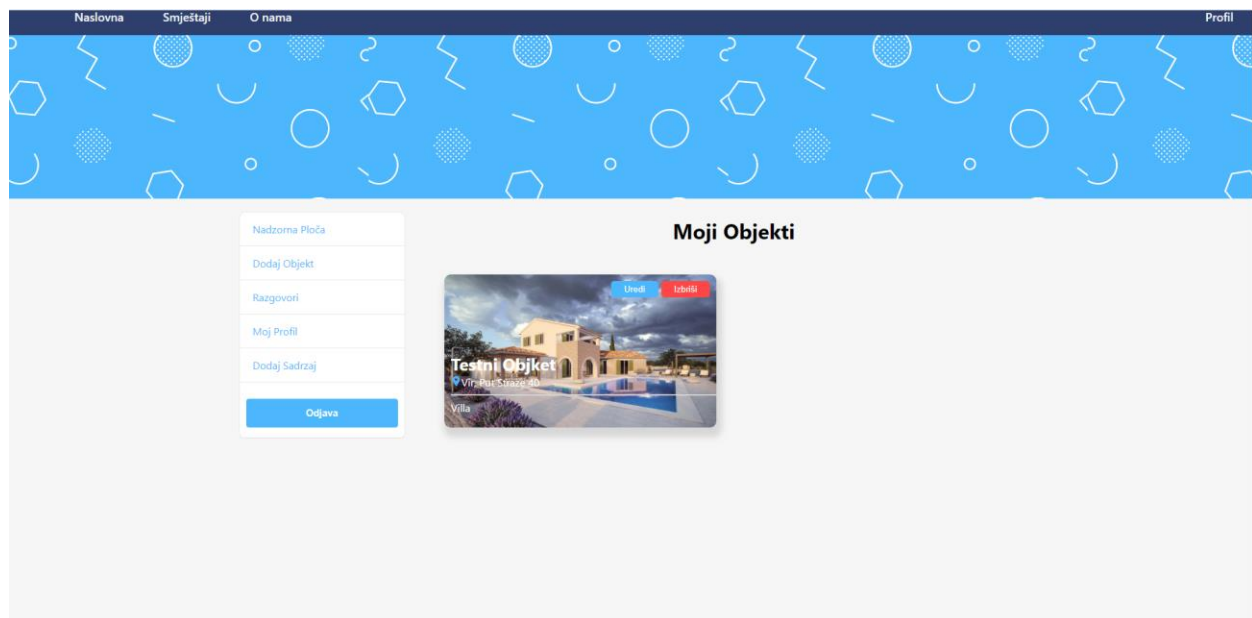
## 6. Prikaz izgleda završene aplikacije



Slika 19: Prikaz dodanog objekta



Slika 20: Prikaz objekata na karti



Slika 21: Prikaz nadzorne ploče



Slika 22: Prikaz uređivanja profila

## 7. Prikaz dijelova koda

### 7.1 Prikaz isječka koda za real-time razgovore

```

47
48 const io = require('socket.io')(server,{
49   pingTimeout:60000,
50   cors:{corsOptions}
51 })
52
53 io.on("connection",(socket) =>{
54   socket.on('setup', (userData)=>{
55     socket.join(userData);
56     socket.emit('connected');
57   })
58
59   socket.on('join chat',(room)=>{
60     socket.join(room);
61     console.log('User joinded room:' + room);
62   })
63
64   socket.on('typing',(room)=>socket.in(room).emit('typing'))
65   socket.on('stop typing',(room)=>socket.in(room).emit('stop typing'))
66
67   socket.on('new message',(newMessageRecived)=>{
68     var chat = newMessageRecived.chat;
69     if(!chat.users) return console.log('chat.users not defined');
70
71     chat.users.forEach(user => {
72       if (user._id == newMessageRecived.sender._id) return;
73       socket.in(user._id).emit('message received', newMessageRecived);
74     });
75   })
76 })

```

Slika 23:Socket.io

## 7.2 Prikaz isječka koda za dodavanje smještaja

```
51 {
52   } = req.body
53   const user_id = req?.user_id;
54   console.log(req.body);
55   if(!user_id || !name || !type || !street || !city || !contactNumber ){
56     return res.status(400).json("message: Bad user data")
57   }
58   const accommodation = new accommodationModel({
59     ownedby_id:user_id,
60     name:name,
61     type:type,
62     tags:tags,
63     street:street,
64     city:city,
65     email:email,
66     contactNumber:contactNumber,
67     website:website,
68     dateOfAddition:dateOfAddition,
69     thumbnail:thumbnail,
70     gallery:gallery,
71     headerphotos:headerphotos,
72     location:JSON.parse(location),
73     services:services,
74     content:JSON.parse(content),
75     description:description
76   })
77   try{
78     const newAccommodation = await accommodation.save();
79     //status 201 - objekt kreiran
80     res.status(201).json(newAccommodation);
81   }catch(err){
82     //status 400 - User dao krive podatke
83     res.status(500).json({message: err.message})
84   }
85 }
86
```

Slika 24: Dodavanje smještaja

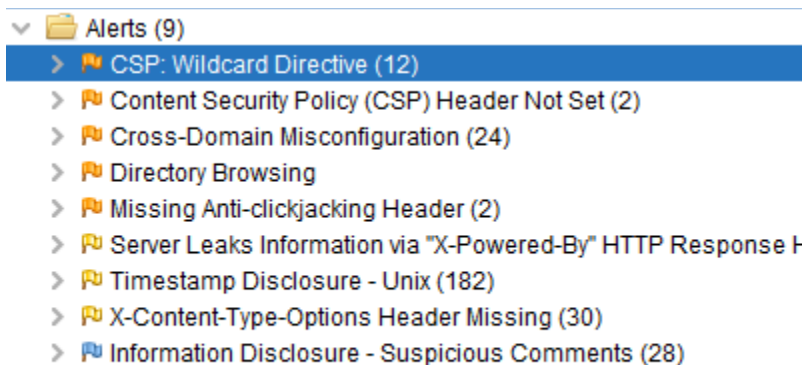
## 7.3 Prikaz isječka koda za registraciju korisnika

```
6 const PWD_REGEX = /^(?=.*[a-z])(?=.*[A-Z])(?=.*[0-9])(?=.*[!@#34])-.{8,24}$/;
7
8 const registerUser = async (req,res)->{
9   const {username,email,password,name,surname} = req.body
10   if(!username || !email || !password || !name || !surname){
11     console.log("here");
12     return res.status(400).json("message: Bad user data")
13   }else if(!USER_REGEX.test(username) || !PWD_REGEX.test(password)){
14     console.log("there");
15     return res.status(400).json("message: Bad user data")
16   }
17   const duplicateUser = await userModel.find({username: username})
18   if(duplicateUser.length){
19     return res.status(409).json("message: User already exists")
20   }
21   const hashedpassword = await bcrypt.hash(password,11)
22   const user = new userModel({
23     username:username,
24     name:name,
25     surname:surname,
26     email: email,
27     password: hashedpassword,
28     roles: {
29       "owner": Number(process.env.OWNER_ROLE)
30     }
31   })
32   try{
33     //Kreiranje osobe i spremanje u bazu podataka i u varijablu newPerson kako bi mogli poslati nazad
34     const newUser = await user.save();
35     //status 201 - objekt kreiran
36     res.status(201).json(newUser);
37   }catch(err){
38     //status 400 - User dao krive podatke
39     res.status(500).json({message: err.message})
40   }
41 }
```

Slika 25: Registracija korisnika

## 8. Sigurnosno testiranje

Sigurnosnim testiranjem koristeći OWASP ZAP utvrđeno je 9 problema na web stranici. Najvažniji će biti opisani u nadolazećim poglavljima.



Slika 26: Sigurnosno testiranje koristeći OWASP ZAP

### 8.1 CSP: Wildcard Directive i Content Security Policy (CSP) Header Not Set

Politika sigurnosti sadržaja (CSP) dodatni je sloj sigurnosti koji pomaže u otkrivanju i ublažavanju određenih vrsta napada. Uključujući (ali ne ograničavajući se na) Cross Site Scripting (XSS) i napade ubrizgavanjem podataka. Ti se napadi koriste za sve, od krađe podataka do uništavanja stranice ili distribucije zlonamjernog softvera. CSP pruža skup standardnih HTTP zaglavlja koji vlasnicima web-mjesta omogućuju da deklariraju odobrene izvore sadržaja koje bi preglednicima trebalo dopustiti učitavanje na tu stranicu - pokrivene vrste su JavaScript, CSS, HTML okviri, fontovi, slike i objekti koji se mogu ugraditi kao što su Java apleti, ActiveX, audio i video datoteke.

### 8.2 Cross-Domain Misconfiguration

Pogrešna konfiguracija CORS-a na web-poslužitelju dopušta zahtjeve za čitanje na više domena iz proizvoljnih domena trećih strana, koristeći neovlaštene API-je na ovoj domeni. Međutim, implementacije web preglednika ne dopuštaju proizvoljnim trećim stranama čitanje odgovora provjerenih API-ja. To donekle smanjuje rizik. Ovu pogrešnu konfiguraciju napadač bi mogao upotrijebiti za pristup podacima koji su dostupni na neovlašteni način, ali koji koristi neki drugi oblik sigurnosti, kao što je popis IP adresa.

### 8.3 Directory Browsing

Pregledavanje datoteka je kada web-stranici pristupate putem web-preglednika i umjesto web-stranice vidite popis datoteka i mapa. To se događa zato što web-poslužitelj na kojem se nalazi vaša stranica ne može samo prikazivati web-stranice već i sadržaj datoteka.

### 8.4 Anti-clickjacking Header

Clickjacking, također poznat kao je kada napadač koristi više prozirnih ili neprozirnih slojeva kako bi prevario korisnika da klikne gumb ili vezu na drugoj stranici kada je namjeravao kliknuti na stranicu najviše razine. Dakle, napadač "otima" klikove namijenjene njihovoj stranici i usmjerava ih na drugu stranicu, najvjerojatnije u vlasništvu druge aplikacije, domene ili oboje.

Koristeći sličnu tehniku, tipke koje se pritisnu također se mogu „oteti“. Uz pažljivo izrađenu kombinaciju stilova, iframeova i tekstualnih okvira, korisnika se može navesti da povjeruje da upisuje lozinku na svoj e-mail ili bankovni račun, ali umjesto toga upisuje u nevidljivi okvir koji kontrolira napadač.

### 8.5 Server Leaks Information via "X-Powered-By" HTTP Response Header Field

Web/aplikacijski poslužitelj propušta informacije putem jednog ili više "X-Powered-By" HTTP zaglavlja odgovora. Pristup takvim informacijama može olakšati napadačima da identificiraju druge okvire/komponente na koje se vaša web aplikacija oslanja i ranjivosti kojima takve komponente mogu biti podložne.

## 9. Testiranje aplikacije

Testiranje aplikacije vršeno je ručno. Mogli su se koristiti automatizirani alati poput npr. Katalon, ali nisu zbog vrlo brzog razvoja aplikacije i čestih promjenama i specifičnosti nekakvih komponenti to nije bilo moguće izvršiti. Stoga se aplikacija testirala ručno nakon svake inačice.

## 10. Dovršenost aplikacije

Naziv funkcionalnosti	Dovršeno	U tijeku	Nedovršeno
Pregled svih smještaja			
Filtriranje svih smještaja			
Pregled pojedinog smještaja			
<i>Pregled pojedinog korisnika i njegovih smještaja koje on nudi na platformi</i>			
<i>Svaki smještaj:</i> <ul style="list-style-type: none"><li>◦ <i>Informacije o korisniku koji ga je stavio</i></li><li>◦ <i>Galerija slika</i></li><li>◦ <i>Opis objekta</i></li><li>◦ <i>Recenzije</i></li></ul>			
<i>Stavljanje smještaja</i>			
<i>User Dashboard</i>			
<i>Google Maps integracija, pokazivanje smještaja na mapi</i>			
<i>Login/Register za svakog korisnika</i>			
<i>Reaktivnost</i>			
<i>Kalendar Dostupnosti</i>			
<i>Reply Score rangiranje</i>			
<i>Review Score rangiranje</i>			
<i>Mogućnost kontaktiranja kroz chat</i>			

## 11. Zaključak

Izradom ovog projekta sam prodbuio svoje znanje izrade i planiranja samog razvoja aplikacije. Kroz ovaj projekt naišao sam na razne izazove kod razvijanja same aplikacije, ali na sreću većinu sam uspio razriješiti bez većih poteškoća. Za razvoj aplikacije sam koristio MERN stack (MongoDB, Express, React i Node.js). Nikada se dosada nisam koristio tome tehnologijom, ali smatram da sam je uspio relativno brzo savladati i vjerujem da ću je nastaviti dalje koristit u svojim budućim projektima.