



PROJEKT

Maribor, 2024

Avtorji: Lena Bojc, Vivien Štampfer, Nik Fišer

Vsebina

PROJEKT	1
O naši aplikaciji	3
Namestitev in zagon aplikacije	4
Diagram primera uporabe	5
Seznam primerov uporabe	6
ER- diagram	17
Diagram aktivnosti	18
Uporabniške zgodbe	19
Oblikovalni vzorci	23

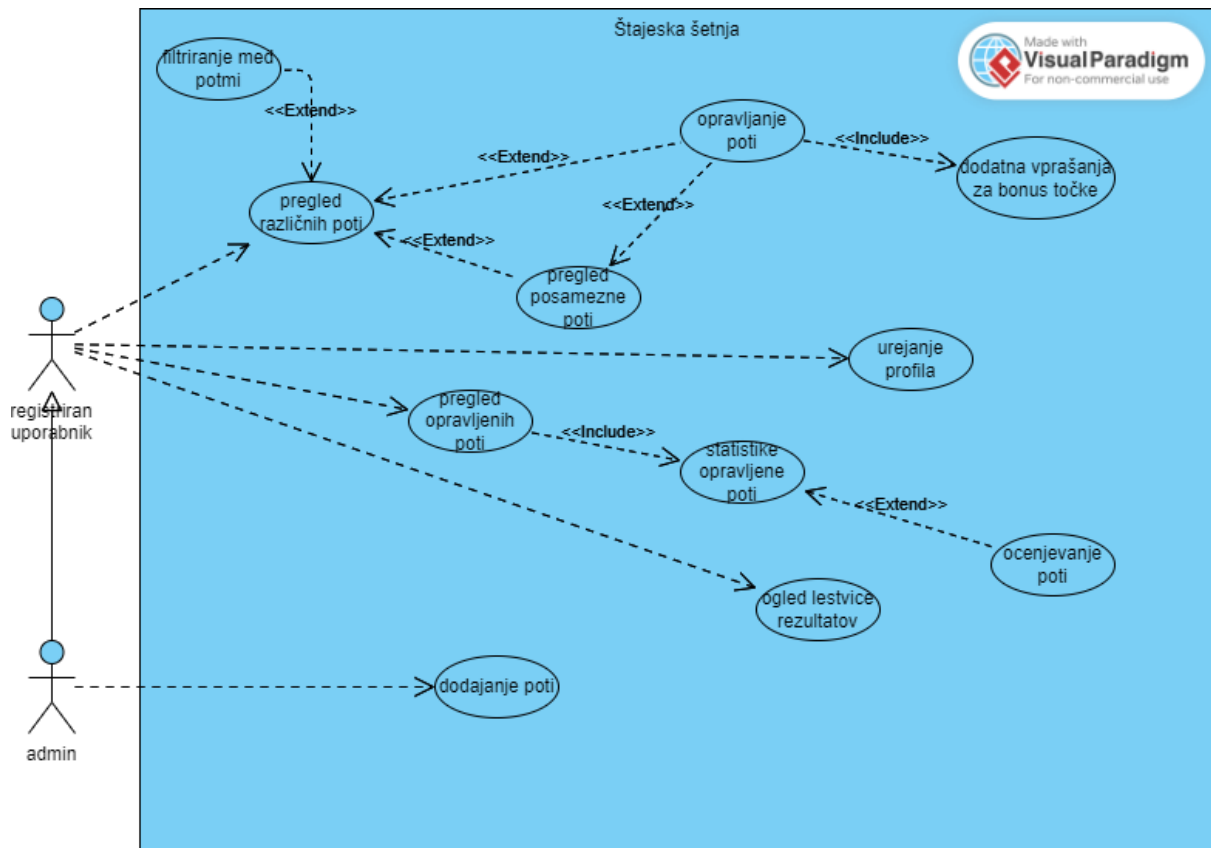
O naši aplikaciji

Naša mobilna aplikacija uporabnikom omogoča odkrivanje znamenitosti na edinstven in interaktiven način. Ko uporabnik začne svojo pot, mu aplikacija postavi vprašanje, ki ga usmerja k določeni znamenitosti. Njegova naloga je, da to znamenitost poišče in ob tem uživa v raziskovanju okolice. Med raziskovanjem lahko uporabnik koristi tudi možnost dodatnih vprašanj, s katerimi pridobi dodatne točke. Rezultate svojega raziskovanja lahko primerja z drugimi uporabniki na lestvici najboljših. Aplikacija združuje zabavo in izobraževanje ter spodbuja k raziskovanju kulturnih in zgodovinskih točk na zanimiv in interaktiven način.

Namestitev in zagon aplikacije

- Prenos projekta: Najprej prenesite projekt na svoj računalnik. Prepričajte se, da imate nameščen Node.js.
- Namestitev Expo aplikacije: Na svojem pametnem telefonu namestite aplikacijo Expo, ki bo omogočila delovanje naše aplikacije na vaši napravi.
- Namestitev paketov: Odprite terminal v korenski mapi projekta in zaženite ukaz "npm install", da prenesete vse potrebne pakete za projekt.
- Nastavitev backend strežnika: Pojdite v mapo backend in znova zaženite ukaz "npm install", da zagotovite pravilno delovanje backend strežnika.
- Zagon aplikacije: V mapi backend zaženite ukaz "npm start". Vrnite se v mapo Setanje in v terminal vpišite "npx expo start". Ta ukaz bo zagnal aplikacijo ter prikazal QR kodo.
- Povezava z mobilno napravo: Odprite Expo aplikacijo na svojem telefonu in skenirajte prikazano QR kodo. Aplikacijo lahko po želji preizkusite tudi v emulatorju na računalniku.

Diagram primera uporabe



Seznam primerov uporabe

PRIJAVA

PU#	Prijava	
Cilj in kontekst	Kot registriran uporabnik se želim vpisati v svoj račun, da lahko uporabljam aplikacijo	
Pogoj uspešnega zaključka	Uporabnik je že registriran	
Pogoj neuspešnega zaključka	Uporabnik še ni registriran	
Primarni in sekundarni akterji	Registriran uporabnik	
Scenarij	korak	Akcija
	1	Odprtje aplikacije
	2	Vnos uporabniškega imena in gesla
	3	klik na gumb Vpis
	4	
Razširitve	korak	Vejitvena akcija
	3a	Napačno uporabniško ime ali geslo
	3b	Možnost spremembe gesla ali ponovne registracije

REGISTRACIJA

PU#	Registracija uporabnika	
Cilj in kontekst	Uporabnik se lahko registrira v sistem	
Pogoj uspešnega zaključka	Veljaven e-mail račun	
Pogoj neuspešnega zaključka		
Primarni in sekundarni akterji	Uporabnik	
Scenarij	korak	Akcija
	1	Odprtje aplikacije
	2	Klik na gumb Registracija
	3	Vnos podatkov (e-mail, geslo, uporabniško ime)
	4	Klik na gumb Registracija
Razširitve	korak	Vejitvena akcija
	3a	nepravilna oblika ali dolžina podatka

PREGLED RAZLIČNIH POTI

PU#	Pregled različnih poti	
Cilj in kontekst	Registriran uporabnik si želi ogledati seznam vseh možnih poti	
Pogoj uspešnega zaključka	Prijavljen uporabnik	
Pogoj neuspešnega zaključka	Ni vnesenih poti	
Primarni in sekundarni akterji	Prijavljen uporabnik	
Scenarij	korak	Akcija
	1	Uporabnik klikne na seznam poti
	2	Prikažejo se mu seznam poti z osnovnimi informacijami
	3	
	4	
Razširitve	korak	Vejitvena akcija

FILTRIRANJE MED POTMI

PU#	Filtriranje med potmi	
Cilj in kontekst	registriran uporabnik lahko filtriram med potmi iz seznama po najnovejši, najstarejši, najtežji, najlažji poti in po abecednem vrstnem redu.	
Pogoj uspešnega zaključka	Registriran uporabnik	
Pogoj neuspešnega zaključka		
Primarni in sekundarni akterji		
Scenarij	korak	Akcija
	1	Uporabnik se premakne na podstran Seznam Poti
	2	na vrhu strani je prikazan filter, kjer izbere željene filtre
	3	prikaz filtriranega seznama
Razširitve	korak	Vejitvena akcija
	1a	

PREGLED POSAMEZNE POTI

PU#	Pregled posamezne poti	
Cilj in kontekst	Registriran uporabnik si lahko ogleda podrobnosti poti, da se lažje odloči, katero pot bo izvajal.	
Pogoj uspešnega zaključka	Registriran uporabnik	
Pogoj neuspešnega zaključka		
Primarni in sekundarni akterji		
Scenarij	korak	Akcija
	1	Uporabnik je na zavihku Seznam Poti in klikne na eno izmed poti
	2	prikaz naziva poti, pričakovano časovno dolžino, pričakovano dolžino v km, težavnost, nekoliko daljši opis poti in njegove tematike, gumb Začni in gumb za izhod iz podrobnega ogleda
	3	
	4	
Razširitve	korak	Vejitvena akcija
	1a	

OPRAVLJANJE POTI

PU#	Opravljanje poti	
Cilj in kontekst	Kot registriran uporabnik želim izvajati pot, da si lahko pridobi točke.	
Pogoj uspešnega zaključka		
Pogoj neuspešnega zaključka		
Primarni in sekundarni akterji		
Scenarij	korak	Akcija
	1	Klik na gumb Začni
	2	Prikaz zemljevida, kjer je vidna njegova trenutna lokacija, možnost vpisa naslednje lokacije (integracija Google Maps) in prikaz uganke, ki jo je potrebno rešiti, za naslednjo lokacijo.
	3	Slika lokacije, za prepoznavanje pravilnosti vmesnega cilja
	4	
Razširitve	korak	Vejitvena akcija
	3a	uporabnik ni na pravi lokaciji (GPS in slika se ne ujemata)
	3b	prikaz namiga na pravilno lokacijo (do 3x, potem tudi možnost prikaza naslova lokacije)

DODATNA VPRAŠANJA ZA BONUS TOČKE

PU#	Dodatna vprašanja za bonus točke	
Cilj in kontekst	Kot uporabnik želim med potjo dobivati dodatna vprašanja, ki mi omogočajo zbiranje dodatnih točk.	
Pogoj uspešnega zaključka		
Pogoj neuspešnega zaključka		
Primarni in sekundarni akterji		
Scenarij	korak	Akcija
	1	med izvajanjem poti (premik iz ene lokacije na drugo) se prikažejo dodatna vprašanja, kjer ima uporabnik možnost vpisati odgovor ali gumb Preskoči)
	2	prikaz dodatnih vprašanj, ko prispe na pravo lokacijo o tej lokaciji, na katera lahko odgovori ali preskoči
	3	
	4	
Razširitve	korak	Vejitvena akcija
	1a	

PREGLED OPRAVLJENIH POTI

PU#	Pregled opravljenih poti	
Cilj in kontekst	Kot uporabnik želim imeti dostop do poti, ki sem jih že opravil.	
Pogoj uspešnega zaključka		
Pogoj neuspešnega zaključka		
Primarni in sekundarni akterji		
Scenarij	korak	Akcija
	1	odprtje zavihka Seznam opravljenih poti.
	2	prikaz seznama vseh opravljenih poti
Razširitve	korak	Vejitvena akcija
	1a	

STATISTIKA OPRAVLJENE POTI

PU#	Statistika opravljene poti	
Cilj in kontekst	Kot uporabnik bi rad videl statistiko opravljene poti	
Pogoj uspešnega zaključka	registriran uporabnik izvede pot	
Pogoj neuspešnega zaključka		
Primarni in sekundarni akterji		
Scenarij	korak	Akcija
	1	premik na Seznam opravljenih poti
	2	klik na eno izmed opravljenih poti
	3	prikaz podrobnosti poti (prikaz naziva poti, pričakovano časovno dolžino + dejanski porabljen čas, pričakovano dolžino v km + opravljena dolžina), težavnost, nekoliko daljši opis poti in njegove tematike, gumb za izhod iz podrobnega ogleda, pridobljene točke)
Razširitve	korak	Vejitvena akcija
	1a	

OGLED LESTVICE REZULTATOV

PU#	Ogled lestvice rezultatov	
Cilj in kontekst	Kot uporabnik bi rad videl lestvico rezultatov, da lahko vidim, kdo je vodilni na lestvici	
Pogoj uspešnega zaključka		
Pogoj neuspešnega zaključka		
Primarni in sekundarni akterji		
Scenarij	korak	Akcija
	1	Klik na zavihek Lestvica
	2	prikaz lestvice z uporabniškimi imeni in njihovimi rezultati
	4	
Razširitve	korak	Vejitvena akcija
	1a	

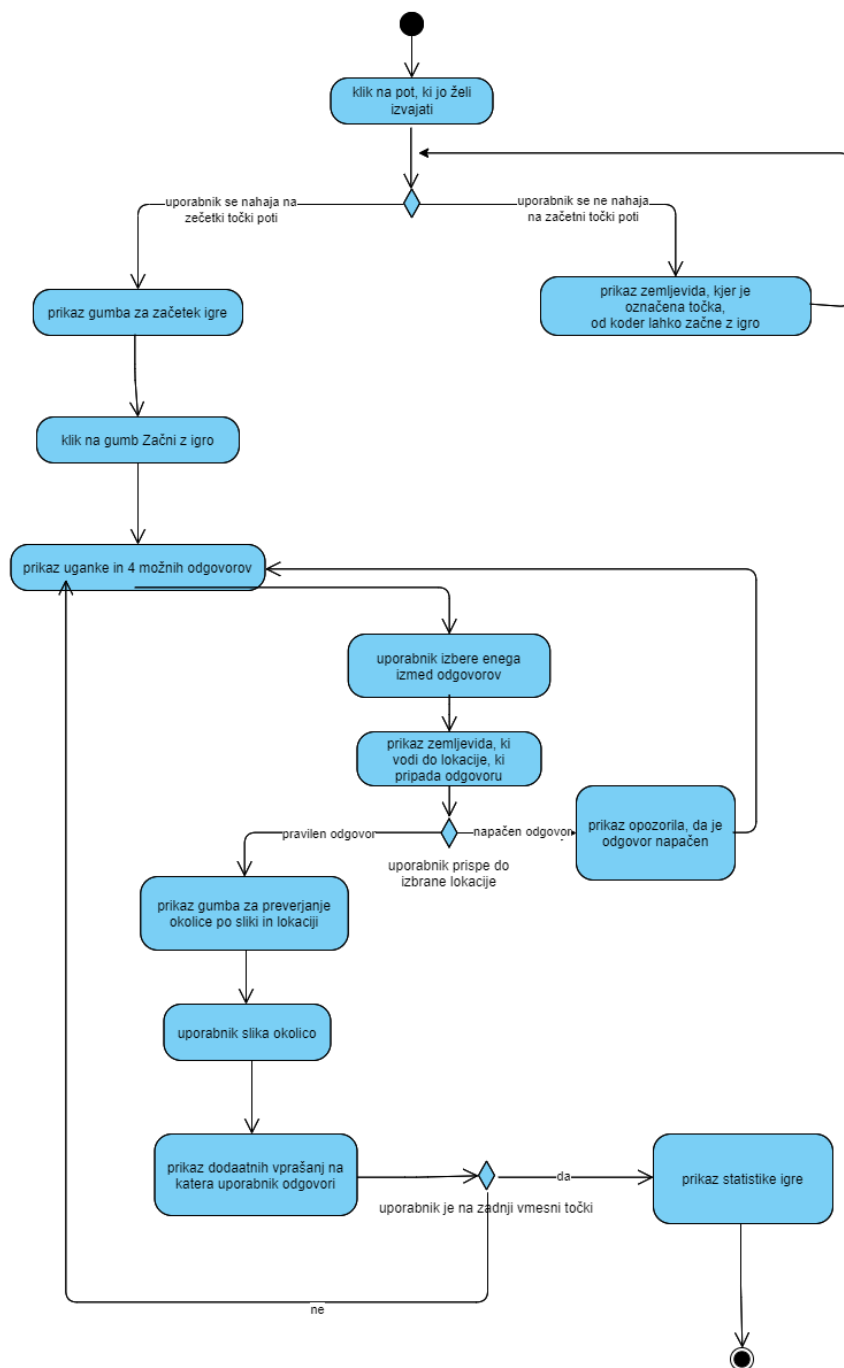
DODAJANJE POTI

PU#	Dodajanje poti	
Cilj in kontekst	Kot administrator bi rad dodajal nove poti, ki jih lahko registrirani uporabniki izvajajo	
Pogoj uspešnega zaključka		
Pogoj neuspešnega zaključka		
Primarni in sekundarni akterji		
Scenarij	korak	Akcija
	1	klik na zavihek Dodaj pot
	2	vnos vmesnih točk
	3	vnos vprašanj za dodatne točke
	4	klik na gumb Objavi
Razširitve	korak	Vejitvena akcija
	1a	

17



Diagram aktivnosti



Uporabniške zgodbe

PRIJAVA

NAZIV: Prijava	
PRIORITETA	ZADOLŽENI
OPIS Kot registriran uporabnik se želim vpisati v svoj račun, da lahko dostopam do aplikacije.	
SPREJEMNI KRITERIJI <ul style="list-style-type: none">• given: uporabnik odpre aplikacijo na svojem telefonu• when: uporabnik vpiše uporabniško ime, geslo in klikne na gumb Prijava• then: uporabniku se odpre domača stran aplikacije	

REGISTRACIJA

NAZIV: Registracija	
PRIORITETA:	ZADOLŽENI:
OPIS: Kot neregistriran uporabnik bi se rad registriral, da lahko uporabljam aplikacijo.	
SPREJEMNI KRITERIJI: <ul style="list-style-type: none">• Given: Uporabnik odpre aplikacijo.• When: Uporabnik vpiše uporabniško ime, email in geslo (ki je dovolj dolgo in vsebuje znake ter male in velike črke), in klikne na gumb Registriraj se• Then: uporabniku se prikaže prijava, se prijavi in uporabniku se odpre domača stran aplikacije.	

PREGLED SEZNAMA POTI

NAZIV: Pregled seznama poti	
PRIORITETA:	ZADOLŽENI:
OPIS: Kot registriran uporabnik želim imeti seznam vseh poti, med katerimi lahko izbiram za izvajanje.	

SPREJEMNI KRITERIJI:

- **Given:** Uporabnik je prijavljen v aplikacijo
- **When:** Uporabnik klikne na zavihek Poti (znak seznam)
- **Then:** Prikaže se podstran s seznamom poti, z zmožnostjo filtriranja.

PREGLED POSAMEZNE POTI

NAZIV: Pregled posamezne poti

PRIORITETA:

ZADOLŽENI:

OPIS:

Kot prijavljen uporabnik si želim ogledati podrobnosti o poti, da se lažje odločim za izvajanje le-te.

SPREJEMNI KRITERIJI:

- **Given:** Uporabnik se nahaja na seznamu poti
- **When:** Uporabnik klikne na eno izmed poti
- **Then:** Prikaže se mu podroben opis te poti, ki vsebuje naziv, sliko (glede na temo), dolžino (čas, km), kratek opis, težavnost, tematika in gumb za začetek izvajanja poti.

IZVAJANJE POTI

NAZIV: Izvajanje poti

PRIORITETA:

ZADOLŽENI:

OPIS:

Kot registriran uporabnik želim izvajati pot, da si lahko pridobim točke.

SPREJEMNI KRITERIJI:

- **Given:** Uporabnik je na podrobnem ogledu poti.
- **When:** Klikne na gumb Začni
- **Then:** Prikaže se prva uganka, za naslednjo lokacijo.
- **When:** Uporabnik vpiše lokacijo.
- **Then:** Prikaz zemljevida do njegove naslednje lokacije (uganka vedno dostopna med izvajanjem)
- **When:** Uporabnik se premika do naslednje lokacije
- **Then:** Prikaže se vprašanje za dodatne točke
- **When:** uporabnik klikne na eno izmed možnosti
- **Then:** ob pravilnem odgovoru, se prištejejo dodatne točke (konstanten števec točk)
- **When:** uporabnik slika lokacijo znotraj aplikacije
- **Then:** prikaz naslednje uganke ali namig za pravo lokacijo
- ponavljanje lokacije in dodatnih vprašanj do končne lokacije

- 3x možnost namiga, potem naslov prave lokacije (lahko si že prej ogleda pravo lokacijo z odbitkom točk)

POVEZOVANJE Z AI

NAZIV: Povezovanje z AI	
PRIORITETA:	ZADOLŽENI:
OPIS: Kot registriran uporabnik želim fotografirati okolico, da pridobim naslednje vprašanje.	
SPREJEMNI KRITERIJI: <ul style="list-style-type: none"> • Given: Uporabnik je na končni lokaciji vmesne ali končne točke • When: Klikne na gumb za slikanje okolice • Then: Prikaže se kamera. 	

PREGLED SEZNAMA OPRAVLJENIH POTI

NAZIV: Pregled seznama opravljenih poti	
PRIORITETA:	ZADOLŽENI:
OPIS: Kot prijavljen uporabnik želim videti svoje pretekle poti, da vidim svoj score in podrobnosti opravljene poti.	
SPREJEMNI KRITERIJI: <ul style="list-style-type: none"> • Given: Uporabnik se nahaja na zavihku seznam opravljenih poti. • When: pregleda posamezno pot. • Then: Prikaz seznama vseh opravljenih poti 	

PREGLED POSAMEZNE OPRAVLJENE POTI

NAZIV: Pregled posamezne opravljene poti
--

PRIORITETA:	ZADOLŽENI:
<p>OPIS:</p> <p>Kot prijavljen uporabnik želim videti podrobnosti opravljene poti, da vidim, kako je moja pot potekala.</p>	
<p>SPREJEMNI KRITERIJI:</p> <ul style="list-style-type: none"> • Given: Uporabnik je na zavihku Opravljene poti • When: Uporabnik klikne na eno izmed opravljenih poti • Then: Prikažejo se podrobnosti poti, kot tudi njegova pot (GPS sledenje) na zemljevidu, njegove dosežene točke pri tej poti, porabljen čas in opravljeni kilometri. 	

OGLED LESTVICE REZULTATOV

NAZIV: Ogled lestvice rezultatov	
PRIORITETA:	ZADOLŽENI:
<p>OPIS:</p> <p>Kot prijavljen uporabnik želim videti lestvico rezultatov, da vidim, kdo je boljši od mene.</p>	
<p>SPREJEMNI KRITERIJI:</p> <ul style="list-style-type: none"> • Given: Uporabnik ima odprto aplikacijo • When: Uporabnik klikne na Lestvica rezultatov. • Then: Prikaz skupno število točk in njegova razvrstitev, pod tem pa lestvica drugih uporabnikov, ki so razvrščeni od najboljšega do 10. najboljšega. 	

DODAJANJE POTI

NAZIV: Dodajanje poti	
PRIORITETA:	ZADOLŽENI:
<p>OPIS:</p> <p>Kot administrator bi rad dodajal pot, da je uporabniki lahko izvajajo.</p>	
<p>SPREJEMNI KRITERIJI:</p> <ul style="list-style-type: none"> • Given: Admin je prijavljen v aplikacijo in na zavihku Profil uporabnika. • When: Klikne na gumb Dodaj pot. • Then: Prikaz vnosnih polj (začetna lokacija, končna lokacija, vmesne lokacije, dodatna vprašanja in možni odgovori, naziv, opis, težavnost, ...) • When: Vnos v vnosna polja in klik na gumb Dodaj • Then: Seznam vseh poti, kjer je dodana tudi nova lokacija. 	

UREJANJE POTI

NAZIV: Urejanje poti	
PRIORITETA:	ZADOLŽENI:
OPIS: Kot admin želim urejati podatke o poti, če je prišlo do kakšnih napak.	
SPREJEMNI KRITERIJI: <ul style="list-style-type: none"> • Given: Admin se nahaja na ogledu seznama poti. • When: Klik na uredi pot. • Then: Možnost urejanja vnosa v vnosna polja. 	

Oblikovalni vzorci

1. Edinec

```
let dbInstance: Promise<SQLite.SQLiteDatabase> | null = null;

export const openDatabase = async () => {
  if (!dbInstance) {
    dbInstance = SQLite.openDatabaseAsync('SetanjeDB.db');
  }
  return dbInstance;
};
```

2. MVC- Model-View-Controller

```
// backend/src/models/users.ts
import mongoose, { Schema, Document } from 'mongoose';

export interface IUser extends Document {
  email: string;
  password: string;
  username: string;
  admin: boolean;
}

const UserSchema: Schema = new Schema({
  email: { type: String, required: true, unique: true },
  password: { type: String, required: true },
  username: { type: String, required: true, unique: true },
  admin: { type: Boolean, required: true, default: false }
});

const User = mongoose.model<IUser>('User', UserSchema);

export default User;
```



```

// Setanje/components/UserList/UserList.tsx
import React, { useEffect, useState } from 'react';
import axios from 'axios';

interface IUser {
  _id: string;
  email: string;
  username: string;
  admin: boolean;
}

const UserList: React.FC = () => {
  const [users, setUsers] = useState<IUser[]><[]>;

  useEffect(() => {
    const fetchUsers = async () => {
      try {
        const response = await axios.get('/api/users');
        setUsers(response.data);
      } catch (error) {
        console.error('Error fetching users:', error);
      }
    };

    fetchUsers();
  }, []);

  return (
    <div>
      <h1>User List</h1>
      <ul>
        {users.map(user => (
          <li key={user._id}>
            {user.username} {<user.email>} - {user.admin ? 'Admin' : 'User'}
          </li>
        ))}
      </ul>
    </div>
  );
};

export default UserList;

```

```

// backend/src/controllers/userController.ts
import { Request, Response } from 'express';
import User from '../models/users';

export const createUser = async (req: Request, res: Response) => {
  try {
    const { email, password, username, admin } = req.body;
    const newUser = new User({ email, password, username, admin });
    await newUser.save();
    res.status(201).json(newUser);
  } catch (error) {
    res.status(500).json({ error: error.message });
  }
};

export const getUsers = async (req: Request, res: Response) => {
  try {
    const users = await User.find();
    res.status(200).json(users);
  } catch (error) {
    res.status(500).json({ error: error.message });
  }
};

```

3. Opazovalec

```
// Setanje/services/UserService.ts
import { User, openDatabase, fetchUsers } from '../database';
import { SQLiteDatabase } from 'expo-sqlite';

type Observer = (users: User[]) => void;

class UserService {
  private observers: Observer[] = [];
  private users: User[] = [];

  async init() {
    const db = await openDatabase();
    await this.loadUsers(db);
  }

  async loadUsers(db: SQLiteDatabase) {
    this.users = await fetchUsers(db);
    this.notifyObservers();
  }

  addObserver(observer: Observer) {
    this.observers.push(observer);
  }

  removeObserver(observer: Observer) {
    this.observers = this.observers.filter(obs => obs !== observer);
  }

  private notifyObservers() {
    this.observers.forEach(observer => observer(this.users));
  }
}

export const userService = new UserService();
```

```

// SetanJe/components/UserList/UserList.tsx
import React, { useEffect, useState } from 'react';
import { View, Text, ScrollView } from 'react-native';
import { userService } from '../../../services/UserService';
import { User } from '../../../database';

const UserList: React.FC = () => {
  const [users, setUsers] = useState<User[]>([]);

  useEffect(() => {
    const updateUsers = (newUsers: User[]) => {
      setUsers(newUsers);
    };

    userService.addObserver(updateUsers);
    userService.init();

    return () => {
      userService.removeObserver(updateUsers);
    };
  }, []);

  return (
    <ScrollView>
      <View>
        <Text>User List</Text>
        {users.map(user => (
          <View key={user.id}>
            <Text>{user.name}</Text>
          </View>
        ))}
      </View>
    </ScrollView>
  );
};

export default UserList;

```