

**VISOKA TEHNIČKA ŠKOLA
STRUKOVNIH STUDIJA
SUBOTICA**

**PROJEKAT
iz predmeta E-mobil aplikacije**

KANDIDAT

Boris Evetović 16119219

Stevan Kazi 16119234

MENTOR

dr Zlatko Čović

SUBOTICA 2022. god.

SADRŽAJ

OPIS ZADATKA	1
Web sajt - turističke destinacije	1
Mobilna aplikacija	2
REALIZACIJA ZADATKA	3
OPIS FUNKCIONALNOSTI	10
Tourism	10
Korisnik	10
Admin	13
Mobilna aplikacija	16
KORIŠĆENA LITERATURA	21

OPIS ZADATKA

Web sajt - turističke destinacije

Ideja informacionog sistema je da se kroz aplikaciju, na što jednostavniji način vrši pregled i pretraga velikog broja dostupnih turističkih destinacija. Sam pristup web aplikaciji je omogućen samo registrovanim korisnicima, a registracija se vrši kroz vrlo jednostavan registracioni formular i verifikaciju email-a klikom na verifikacioni link koji se dostavlja putem email-a nakon popunjavanja formulara. Sama aplikacija razlikuje dve vrste korisnika: admin korisnike i “obične” korisnike, odnosno ciljnu grupu.

Administracija je dostupna samo admin korisnicima (u daljem tekstu “admin”) i putem njenog interfejsa može da se vrše različite manipulacije nad samim destinacijama i entitetima vezanim za destinaciju. Admin može da dodaje nove destinacije, definišući osnovne informacije kao što su naziv, adresa, geografski položaj (latituda i longituda - potrebno za kasniji prikaz na mapama i potencijalnu pretragu), opis, kategoriju i grad u kojoj se nalazi određena destinacija. Lista gradova je unapred definisana (gradovi Srbije), dok se kategorije takođe definišu putem administracije od strane admina. Kategorija ima svoj naziv i ikonicu koja smisleno simboliše kategoriju, a koristi se za kasniji prikaz na interfejsu aplikacije (ikonice su unapred definisane). Takođe, mogu da se vrše i izmene podataka postojećih destinacija i kategorija. Moguće je i obrisati određenu destinaciju ili kategoriju, s tim što, u sličaju brisanja kategorije, treba voditi računa da nijedna destinacija nije povezana za kategoriju koja se briše. Brisanje destinacije može da se uradi “bezbrizno”. Pored osnovnih (za funkcionalnost aplikacije) informacija o destinacijama, potrebno je dodati i slike pojedinačnih destinacija, u svrhu vernijeg vizuelnog prikaza destinacije.

Na samom početku, registrovani korisnik (u daljem tekstu “korisnik”) se nalazi na početnoj stranici na kojoj se prikazuju: neke od najpopularnijih destinacija u Srbiji (recimo “top 10”), neke od najposećenijih destinacija u Srbiji (recimo “top 10”), najpopularnije ili najposećenije (ili oba) destinacije u njegovom gradu (koji je prvobitno definisan putem registracionog formulara, sa kasnijom mogućnošću izmene) Nakon toga, korisnik, putem menija sa leve strane aplikacije (side menu), može da navodi interakciju sa aplikacijom ka raznim stranicama. Jedna od najbitnijih stranica jeste pretraga, gde korisnik može da vrši pretragu destinacija definišući sledeće karakteristike destinacije (u međusobnom odnosu ne “isključive”, odnosno višestruka pretraga) : naziv destinacije (delimično poklapanje), grad ili više gradova, kategorija ili više kategorija.

U pretragu su takođe uključene i opcije sortiranja po popularnosti i po posećenosti destinacija, u rastućem ili opadajućem redosledu. Na kraju, u pretrazi postoji još jedan

filter (“u blizini” ili “u mom gradu”) pomoću kojeg se na prethodnu pretragu dodaje uslov da prikazane destinacije moraju biti iz grada iz kojeg je korisnik. Ova opcija isključuje opciju (polje) pretrage po gradu ili po više gradova. Svaka destinacija koja je iz istog grada iz kojeg je i korisnik, prilikom samog prikaza na interfejsu nosi markicu (zastavicu, en. flag) “u blizini”.

Prikaz destinacija kroz pretragu, odnosno bilo koji višestruki prikaz destinacija je u kraćem obliku, odnosno prikazuju se neke osnovne informacije o destinaciji, dok se sve informacije prikazuju ulaskom na samu stranicu pojedinačne destinacije. Na stranici pojedinačne destinacije, korisnik ima mogućnost, pored detaljnog pregleda informacija, dodavanje review-a (komentara) na destinaciju (neograničeno) i “lajkovanja” ili “dislajkovanja” destinacije (ukoliko nije “lajkovana” destinacija prikazuje se like opcija, a ukoliko je “lajkovana” prikazuje se dislike opcija).

Korisnik ima mogućnost dodavanja destinacija u liste “omiljeno” i “za posetiti”. To može uraditi na stranici prikaza pojedinačne destinacije, a takođe ima i opciju da izbacuje određene destinacije iz listi. Liste i imaju i posebnu stranicu za prikaz.

Korisnik ima i opciju jednostavnog podešavanja profila, jer nije potreban veliki broj korisničkih informacija za funkcionalnost aplikacije. Korisnik može da menja ime, prezime i grad u kom živi. Dodatno može da promeni predefinisanu sliku/avatare koju nakon registracije ima svaki korisnik.

Mobilna aplikacija

Korisnik, na samom početku kada otvori mobilnu aplikaciju mora da popuni login formu i da se loguje. Na samoj mobilnoj aplikaciji korisnik ima dve opcije, jedna je da pregleda destinacije na mapi u blizini njegove trenutne lokacije, a druga je da pregleda istoriju svojih lokacija. Ukoliko korisnik želi da pregleda destinacije u blizini svoje lokacije, prvo mora da odobri pristup korišćenju lokacije mobilnog telefona od strane mobilne aplikacije. Kada korisnik pokuša da uđe u pregled destinacija, iskočiće mu prozor na kome može da odobri ili da odbije korišćenje lokacije telefona. Nakon toga, korisniku se na mapi prikazuje njegova trenutna lokacija i destinacije oko njega u krugu od 2 km. Marker koji označava korisnikovu lokaciju je drugačije boje u odnosu na markere koji označavaju destinacije u krugu. Destinacije van kruga nisu vidljive na mapi. Korisnik može da pomera svoju lokaciju zadržavanjem prsta na markeru svoje lokacije i pomeranjem markera na određeno mesto na mapi. Nakon toga se pomera i krug i osvežavaju se destinacije. Takođe, svakih 10 sekundi se proverava da li se korisnik pomerio, i ukoliko jeste, osvežava se i sama lokacija korisnika i destinacije u krugu oko korisnikove lokacije. Zbog boljih performansi, lokacija se ne uzima sa 100%-tnom preciznošću prilikom osvežavanja lokacije, što znači da korisnik treba da se pomeri za određenu fizičku razdaljinu da bi se mapa osvežila. Prilikom samog ulaza na pregled destinacija u blizini, beleži se trenutna lokacija korisnika na mapi, dok se nakon toga ne beleže daljnje lokacije, zbog validnosti podataka (da korisnik ne bi imao hiljade zabeleženih lokacija u razmaku od nekoliko metara). Druga funkcionalnost mobilne aplikacije je pregled istorije

korisnikovih lokacija. Lokacije se prikazuju u tabeli. Svaka lokacija je dostupna i na mapi klikom na link u poslednjoj koloni tabele. Za svaku lokaciju u istoriji lokacija se beleži i broj “poseta”, odnosno koliko puta je korisnik pristupio mapi da destinacijama u blizini sa identične lokacije.

REALIZACIJA ZADATKA

Na samom početku, korisnik mora da se ulogiju u mobilnu aplikaciju putem login forme i klikom na dugme uloguj se. Klik na dugme poziva */api/login* endpoint koji je definisan na backend aplikaciji pomoću php jwt bundle. Validacija login forme na samoj mobilnoj aplikaciji:

```
const validateEmail = (text) => {
  let reg = /^\\w+([\\.-]?\\w+)*@\\w+([\\.-]?\\w+)*\\.\\w+\\w+)$/;

  if(text.length == 0){
    setEmailError("Obavezno polje");
  }
  else if(text.indexOf(' ') >= 0){
    setEmailError('Nepravilan email');
  }
  else if(reg.test(credentials.email) === false) {
    setEmailError('Nepravilan email');
  } else{
    setEmailError("");
    setEmailValid(true);
  }
}

const validatePassword = (text) => {
  if(text.length == 0){
    setPasswordError("Obavezno polje");
  }
  else if(text.length < 8){
    setPasswordError("Lozinka mora da sadrži bar 8 karaktera");
  }
  else if(text.indexOf(' ') >= 0){
    setPasswordError('Lozinka ne sme da sadrži razmak');
  }
  else{
    setPasswordError("");
    setPasswordValid(true);
  }
}
```

Funkcije `validateEmail` i `validatePassword` se pozivaju na svaku promenu sadržaja u odgovarajućim tekstualnim poljima u samom login formularu.

```
<Text>Email</Text>
  {emailError.length > 0 && <Text style={styles.error}>{emailError}</Text>}
  <TextInput
    style={styles.input}
    placeholder="email"
    onChangeText={(newText) => {
      validateEmail(newText);
      setCredentials({email: newText, password: credentials.password});
    }}
  />
<Text>Lozinka</Text>
  {passwordError.length > 0 && <Text style={styles.error}>{passwordError}</Text>}
  <TextInput
    style={styles.input}
    secureTextEntry={true}
    placeholder="lozinka"
    onChangeText={(newText) => {
      validatePassword(newText);
      setCredentials({email: credentials.email, password: newText})
    }}
  />
```

Na sam klik na dugme uloguj se u login formularu, poziva se endpoint za login i ukoliko se od backend-a dobije response sa access tokenom, access token se setuje u async storage kako bi kasnije mogao da se koristi prilikom poziva ostalih endpointa.

```
<Button
  title='Uloguj se'
  onPress={async () => {
    try {
      if (emailValid && passwordValid) {
        let response = await fetch(API_URL + '/api/login', {
          method: 'POST',
          headers: {
            Accept: 'application/json',
            'Content-Type': 'application/json'
          },
          body: JSON.stringify(credentials)
        });

        response = await response.json();

        if (response.hasOwnProperty('message')) {
          setLoginError(response.message);
        }
        if (response.hasOwnProperty('token') &&
response.hasOwnProperty('refresh_token')) {
```

```

    try {
      await AsyncStorage.setItem('@token', response.token);
      await AsyncStorage.setItem('@refresh_token',
response.refresh_token);

      navigation.navigate('Menu');
    } catch (error) {

      console.log(error);
    }
  }
} catch (e) {

}

}}
/>

```

Nakon uspešnog loginovanja korisnik bira pregled destinacija ili pregled istorije lokacija. Na samom početku se od korisnika traži dozvola za korišćenje lokacije mobilnog telefona. Ukoliko je lokacija odobrena, koordinate lokacije se šalju na `/api/destinations/radius` endpoint koji će da vrati sve destinacije u blizini u odnosu na prosleđenu lokaciju. Zatim se upisuje trenutna korisnička lokacija u istoriju lokacija pozivanjem endpoint-a `/api/users/location`. Nakon toga se prikazuju destinacije na mapi, zajedno sa lokacijom korisnika.

```
React.useEffect(() => {
  (async () => {
    let { status } = await Location.requestForegroundPermissionsAsync();

    if (status !== 'granted') {
      setErrorMsg('Permission to access location was denied');
      return;
    }

    let location = await Location.getCurrentPositionAsync({});

    try {
      const token = await AsyncStorage.getItem('@token');

      let response = await fetch(API_URL + '/api/destinations/radius',
{
  method: 'POST',
  headers: {
    Accept: 'application/json',
    'Content-Type': 'application/json',
  },
  body: JSON.stringify({
    lat: location.coords.latitude,
    lon: location.coords.longitude,
    radius: radius,
  })
});
    } catch (error) {
      console.log(error);
    }
  })();
});
```

```

        Authorization: 'Bearer ' + token
      },
      body: JSON.stringify(location.coords),
    });

    response = await response.json();
    setDestinations(response);

```

```

    let locationLog = {
      latitude: String(location.coords.latitude),
      longitude: String(location.coords.longitude),
    };
    await fetch(API_URL + '/api/users/locations', {
      method: 'POST',
      headers: {
        Accept: 'application/json',
        'Content-Type': 'application/json',
        Authorization: 'Bearer ' + token
      },
      body: JSON.stringify(locationLog)
    });
  } catch (e) {

    return;
  }

  setPin({
    latitude: location.coords.latitude,
    longitude: location.coords.longitude
  });

  })();
}, []);

```

Na svakih 10 sekundi se osvežava lokacija korisnika i mapa sa destinacijama u blizini uz pomoć endpointa za dobijanje desitnacija u određenom radijusu i korisničkog access tokena kojeg je korisnik dobio prilikom loginovanja.

```

userLocationUpdateInterval={10000}
onUserLocationChange={async (e) => {
  e.persist();
  try {
    const token = await AsyncStorage.getItem('@token');
    let response = await fetch(API_URL + '/api/destinations/radius', {
      method: 'POST',
      headers: {
        Accept: 'application/json',
        'Content-Type': 'application/json',
        Authorization: 'Bearer ' + token
      },

```



```

        body: JSON.stringify(e.nativeEvent.coordinate),
    });

    response = await response.json();
    setDestinations(response);
} catch (e) {

    return;
}
setPin({
    latitude: e.nativeEvent.coordinate.latitude,
    longitude: e.nativeEvent.coordinate.longitude
}))

```

Druga opcija je pregled istorije korisničkih lokacija. Korisničke lokacije se prikazuju u tabeli koja sadrži vreme, koordinate, broj ponavljanja i link ka prikazu lokacije na mapi. Lista korisničkih lokacija se dobija pozivanjem endpoint-a *api/users/locations*, ovog puta sa GET metodom.

```

React.useEffect(() => {
    (async () => {
        const token = await AsyncStorage.getItem('@token');

        let response = await fetch(API_URL + '/api/users/locations', {
            method: 'GET',
            headers: {
                Accept: 'application/json',
                'Content-Type': 'application/json',
                Authorization: 'Bearer ' + token
            }
        });

        response = await response.json();
        setLocationLog(response);
    })();
}, []);

```

Prikazvinje tabele sa istorijom lokacija, svaka ćelija se generiše custom funkcijama za generisanje određene vrste ćelije u tabeli.

```

return (
    <View style={styles.container}>
    <Table borderStyle={{ borderWidth: 1 }}>
        <Row
            data={[
                renderCellHead('Vreme'),
                renderCellHead('Lokacija'),

```

```

        renderCellHead('Broj poseta'),
        renderCellHead('Prikaži na mapi')
    ]}
    flexArr={[1, 1, 1]}
    style={styles.head}
  />

  {locationLog.map((location, index) => {
    return (
      <Row
        data={[
          renderCell(String(new Date(location.createdAt))),
          renderCell(location.latitude + ' ' + location.longitude),
          renderCell(location.times),
          renderCellLink(location.latitude, location.longitude)
        ]}
        flexArr={[1, 1, 1, 1]}
        style={styles.row}
        key={index}
      />
    )
  })}
</Table>
</View>
);

```

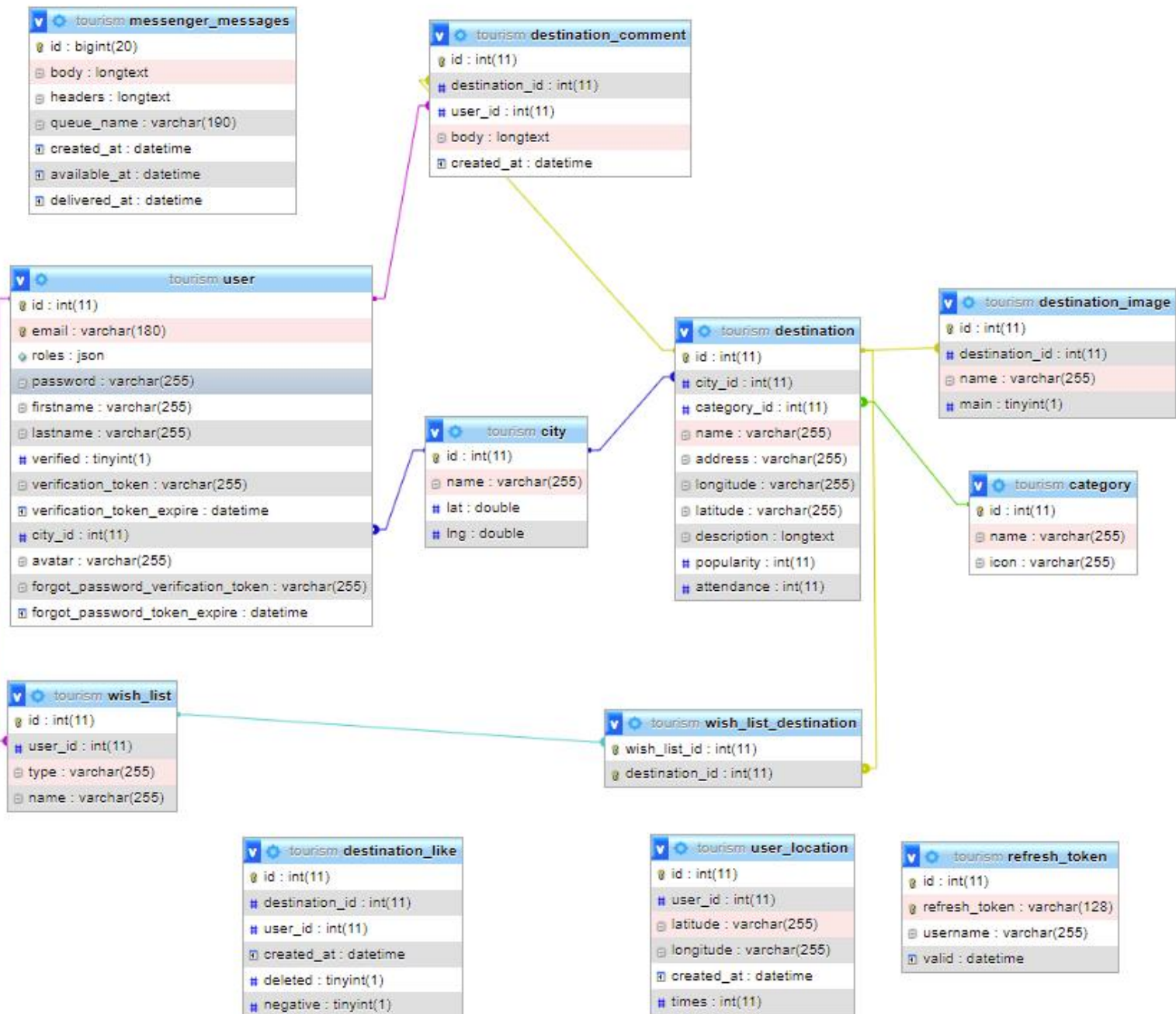
Funkcija `renderCellLink` sadrži link ka prikazu određene korsiničke lokacije na mapi. Na screen `SingleDestination` se prosleđuju koordinate lokacije koja se nalazi u tom redu tabele, kao parametri.

```

const renderCellLink = (latidute, longitude) => {
  return (
    <Text
      onPress={() => navigation.navigate('SingleDestination', {
        latitude: latidute,
        longitude: longitude
      })}
      style={{textAlign: "center"}}
    >
      mapa
    </Text>
  );
}

```

STRUKTURA BAZE PODATAKA

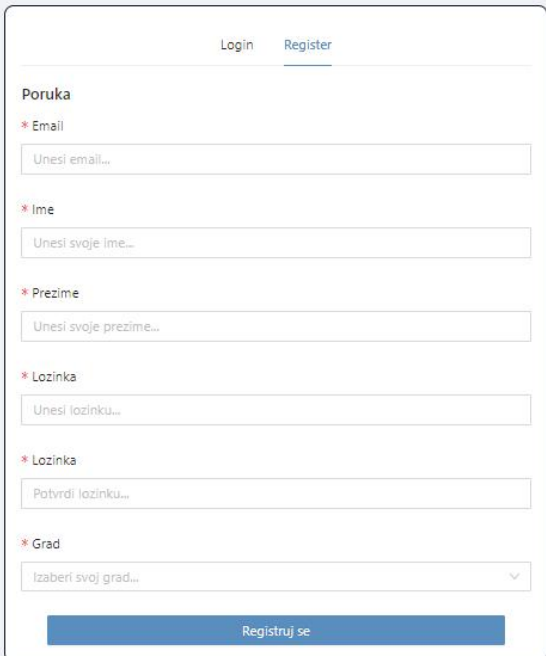


OPIS FUNKCIONALNOSTI

Tourism

Korisnik

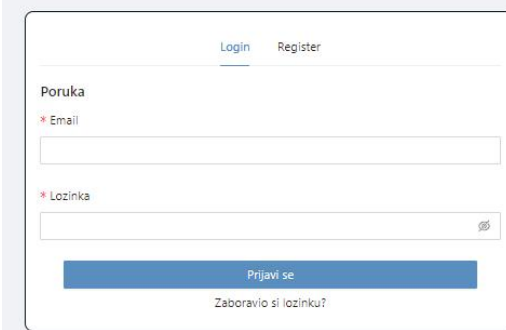
Korisnik prvo mora da bude registrovan da bi imao pristup web sajtu. Prema tome prva funkcionalnost je registrationi formular. Korisnik popunjava registrationi formular navodeći neke osnovne informacije o sebi. Nakon toga mu stiže verifikacioni email sa verifikacionim linkom. Klikom na link korisnik verifikuje svoj email. Zatim može da se uloguje pomoću login formulara (email i šifra).



The registration form is titled "Register" and includes a "Login" link. It contains the following fields:

- * Email**: Text input with placeholder "Unesi email..."
- * Ime**: Text input with placeholder "Unesi svoje ime..."
- * Prezime**: Text input with placeholder "Unesi svoje prezime..."
- * Lozinka**: Text input with placeholder "Unesi lozinku..."
- * Lozinka**: Text input with placeholder "Potvrdi lozinku..."
- * Grad**: Dropdown menu with placeholder "Izaberi svoj grad..."

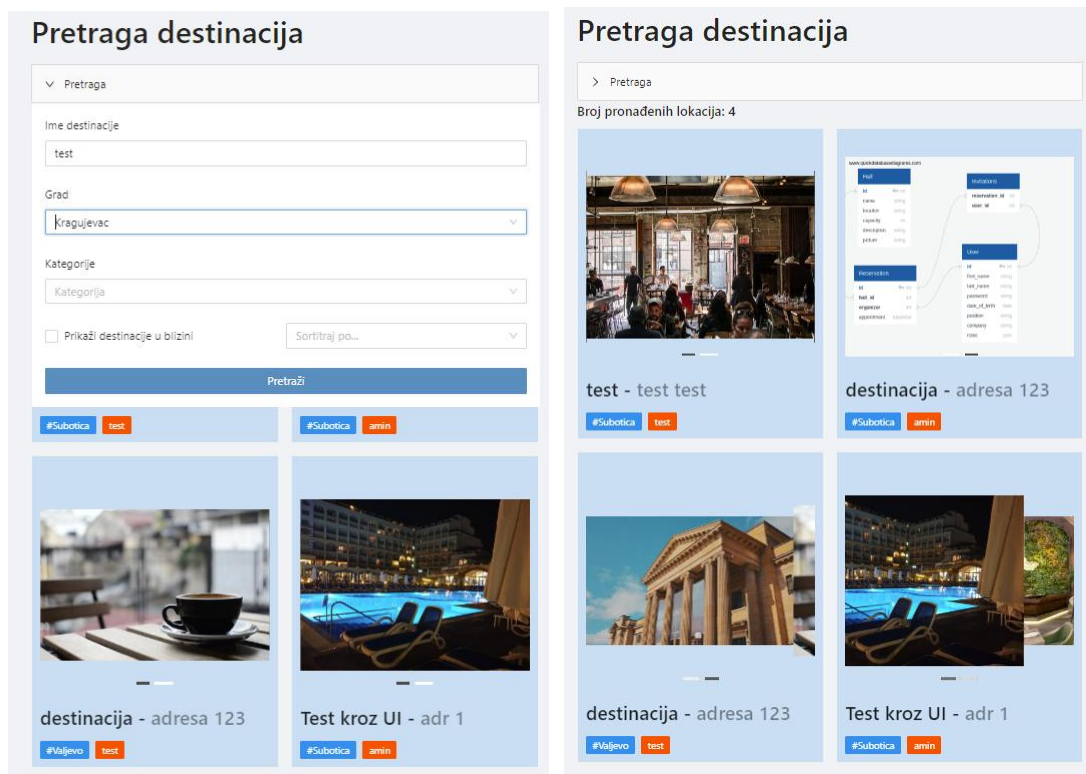
A blue button labeled "Registruj se" is at the bottom.



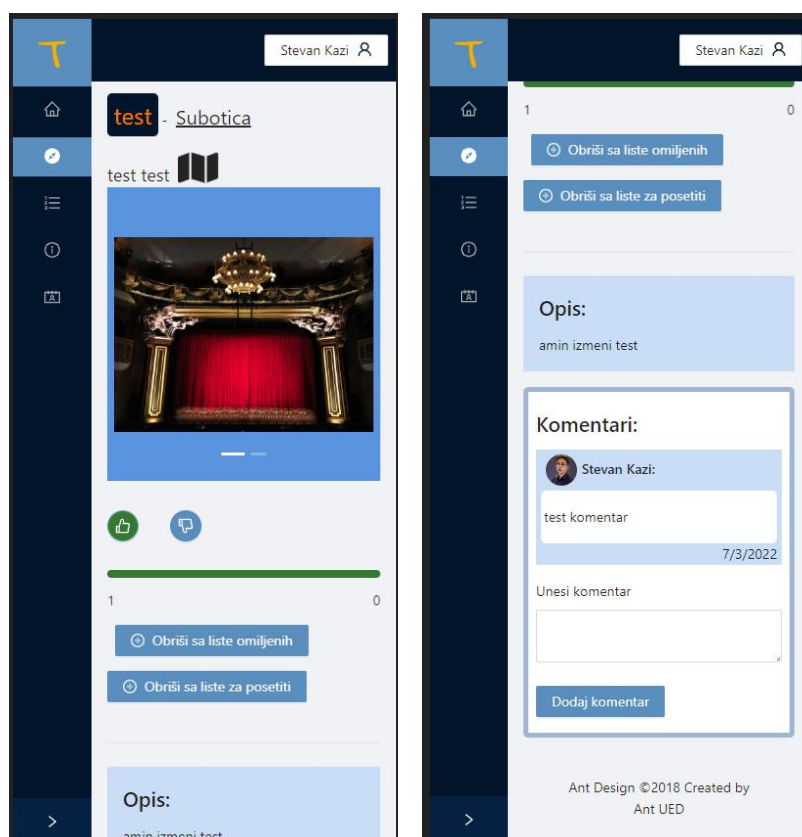
The login form is titled "Login" and includes a "Register" link. It contains the following fields:

- * Email**: Text input
- * Lozinka**: Text input with a toggle icon for password visibility

A blue button labeled "Prijavi se" is at the bottom, with a link "Zaboravio si lozinku?" below it.

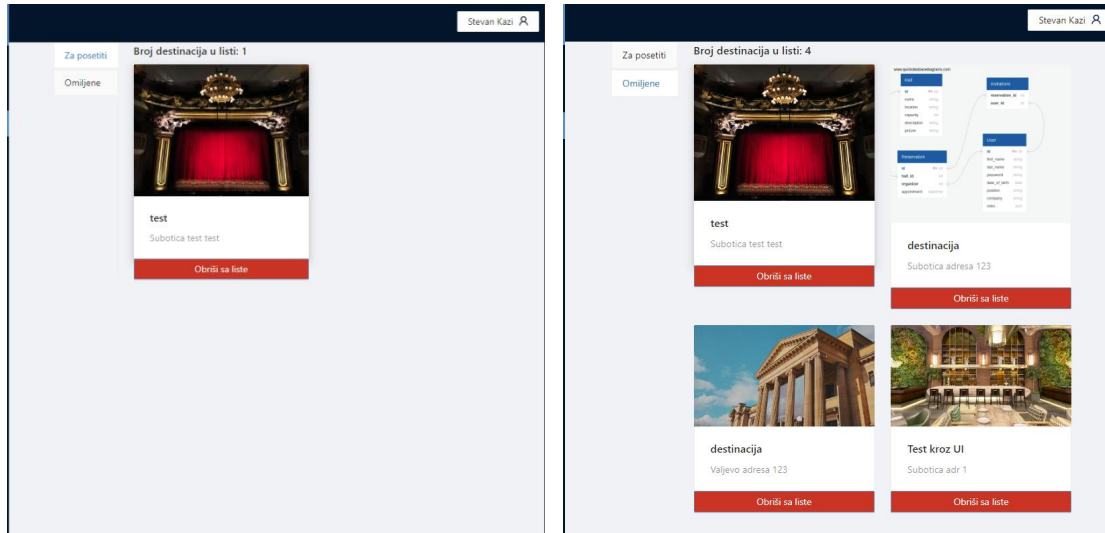


Na stranici za pretragu, korisnik može da popuni kriterijume za pretragu, a zatim će dobiti rezultate pretrage. Nakon toga, klikom na određenu destinaciju, korisnik može da ode na stranicu pojedinačne destinacije gde se vide sve dostupne informacije o destinaciji.

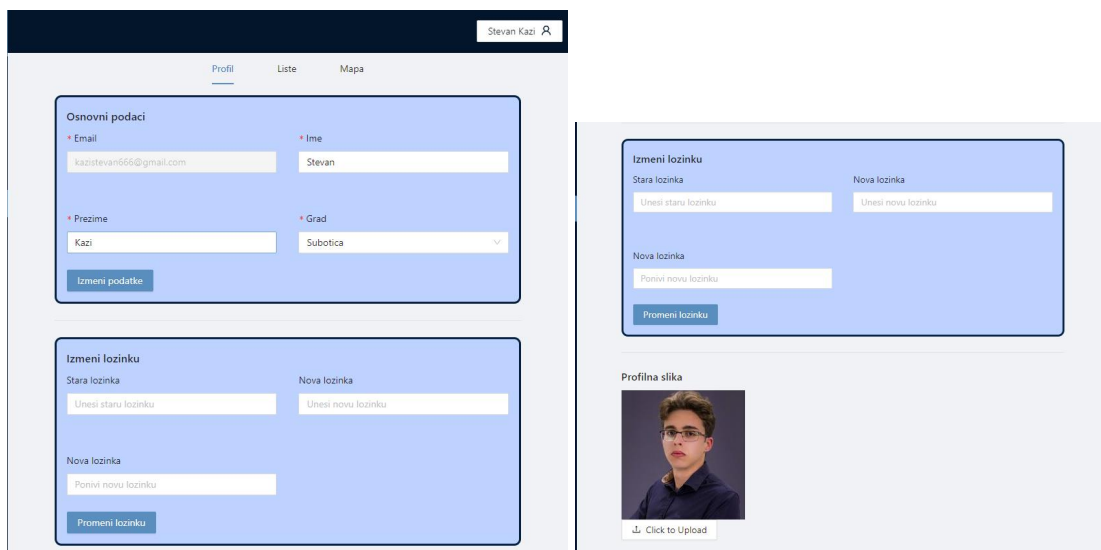


Na stranici pojedinačne destinacije, korisnik može da doda destinaciju u određenu listu, doda lajk ili dislajk, i doda komentar i pregleda komentare drugih korisnika.

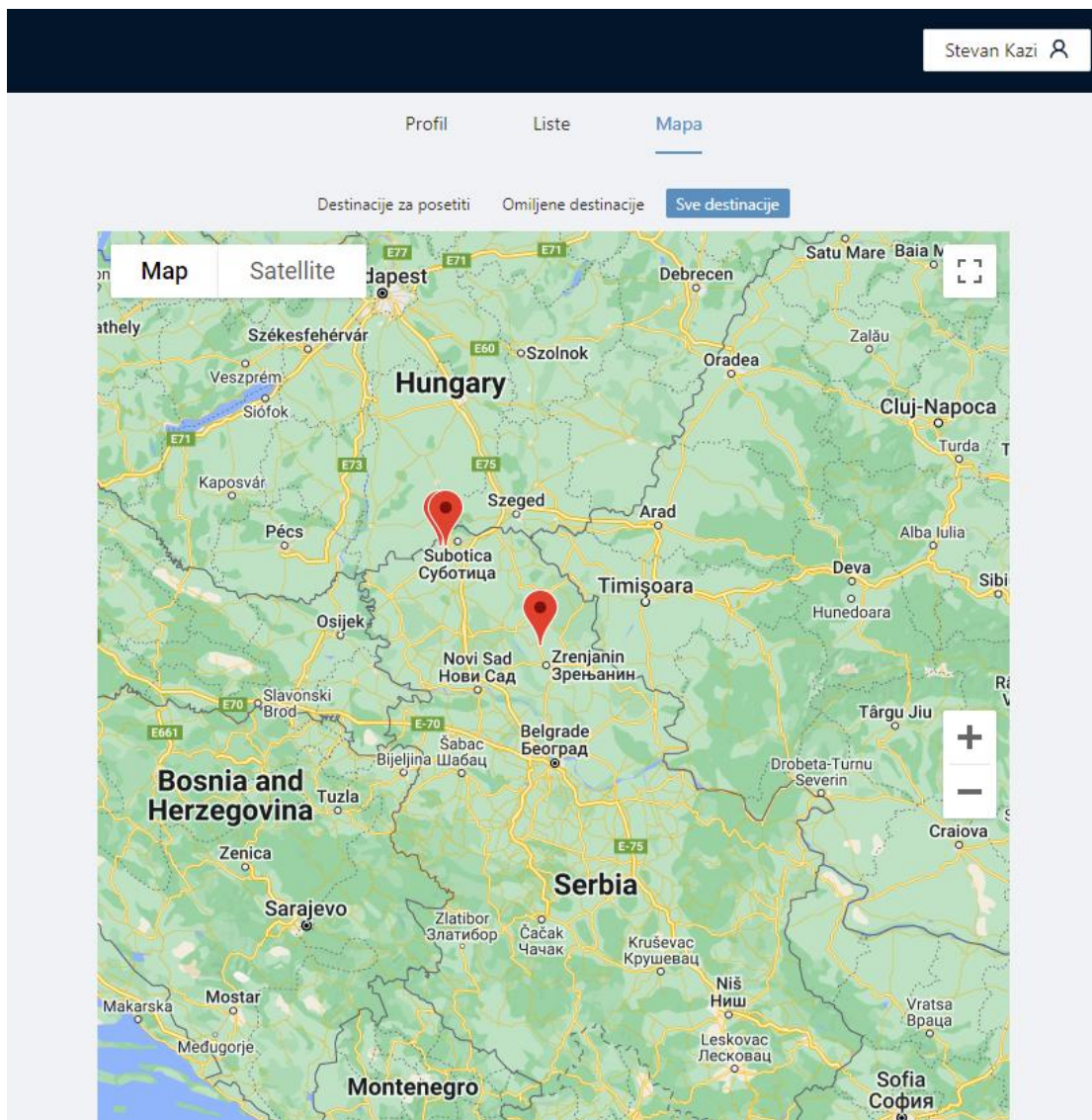
Zatim, postoji i stranica za korisnikove liste, gde korisnik može da pregleda sve destinacije koje je prethodno dodao u listu omiljenih ili za posetiti. Takođe može i da skine destinaciju sa liste.



Zatim idemo na profil korisnika. Korisnik u svom profilu može da menja neke osnovne informacije o sebi, da promeni šifru i da promeni profilnu sliku.



Takođe, može i da pristupi listama klikom na tab "liste". Na poslednjem tabu koji se zove mapa korisnik može da pregleda na mapi destinacije koje je dodao u listu omiljenih ili za posetiti.



Admin

Administrator ima pristup stranicama za dodavanje, editovanje i brisanje kategorija i destinacija.

Unesi novu destinaciju

Id	Ime destinacije	Adresa	Grad	Kategorija	Akcije
1	test	test test	Subotica	test	 
2	destinacija	adresa 123	Subotica	amin	 
3	destinacija	adresa 123	Valjevo	test	 
4	Test kroz UI	adr 1	Subotica	amin	 

* Ime destinacije

Unesi ime destinacije...

* Unesi adresu destinacije

Unesi adresu destinacije

Opis destinacije

Unesi opis...

* Kategorija

Izaberi kategoriju...

* Grad

Izaberi grad

Izaberi lokaciju



Cancel

OK

Stevan Kazi

Unesi novu kategoriju

Id	Ikonica kategorije	Ime kategorije	Akcije
1		test	
2		amin	
3		test	

< 1 >

Mobilna aplikacija

Korisnik, na samom početku kada otvori mobilnu aplikaciju mora da popuni login formu i da se loginuje.

The image displays two side-by-side screenshots of a mobile application's login screen. Both screens show a 'Login' title at the top. The form consists of two input fields: 'Email' and 'Lozinka' (Password). The email field contains the text 'kazistevan666@gmail.com'. The password field contains a series of dots. Below the password field, there is a blue button labeled 'ULOGUJ SE'. In the left screenshot, a red error message 'Lozinka mora da sadrži bar 8 karaktera' is visible below the password field. The right screenshot shows the same form but without the error message. The status bar at the top of both screens shows the time as 19:05 and various system icons. The bottom of each screen features a black navigation bar with three icons: a square, a circle, and a triangle.

19:05 19:05

VoLTE1 VoLTE1

19:05 19:05

VoLTE1 VoLTE1

Login Login

Email Email

kazistevan666@gmail.com kazistevan666@gmail.com

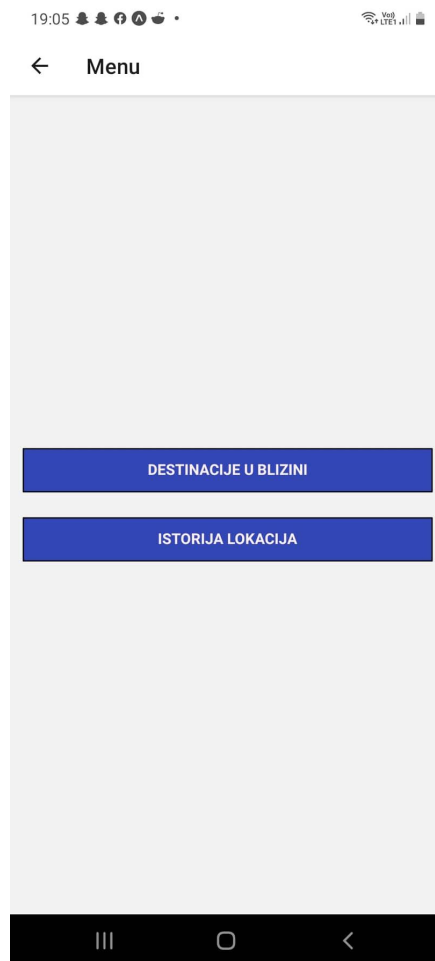
Lozinka Lozinka

Lozinka mora da sadrži bar 8 karaktera

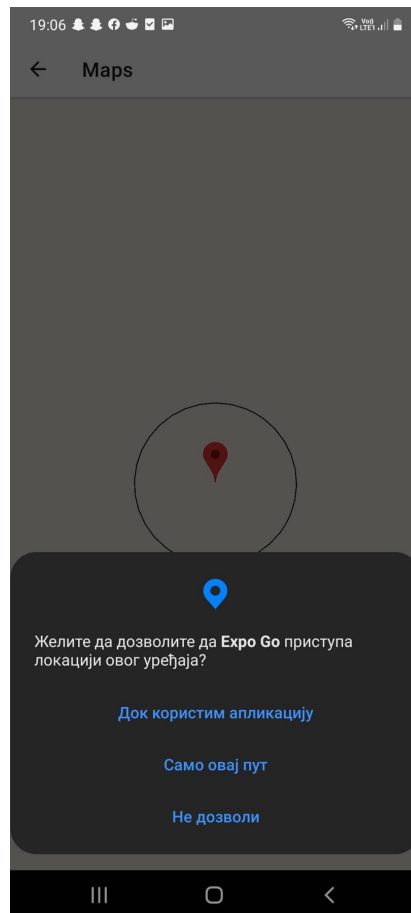
.....

ULOGUJ SE ULOGUJ SE

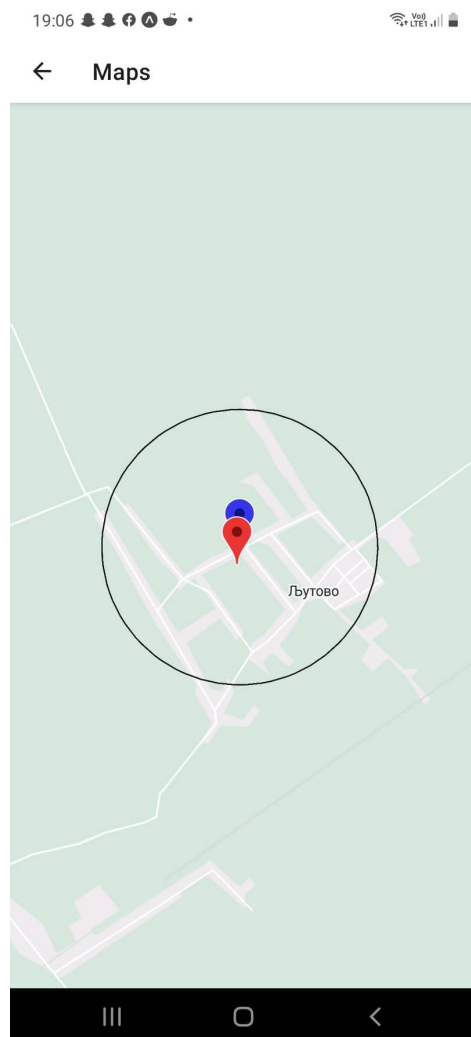
Na samoj mobilnoj aplikaciji korisnik ima dve opcije, jedna je da pregleda destinacije na mapi u blizini njegove trenutne lokacije, a druga je da pregleda istoriju svojih lokacija.



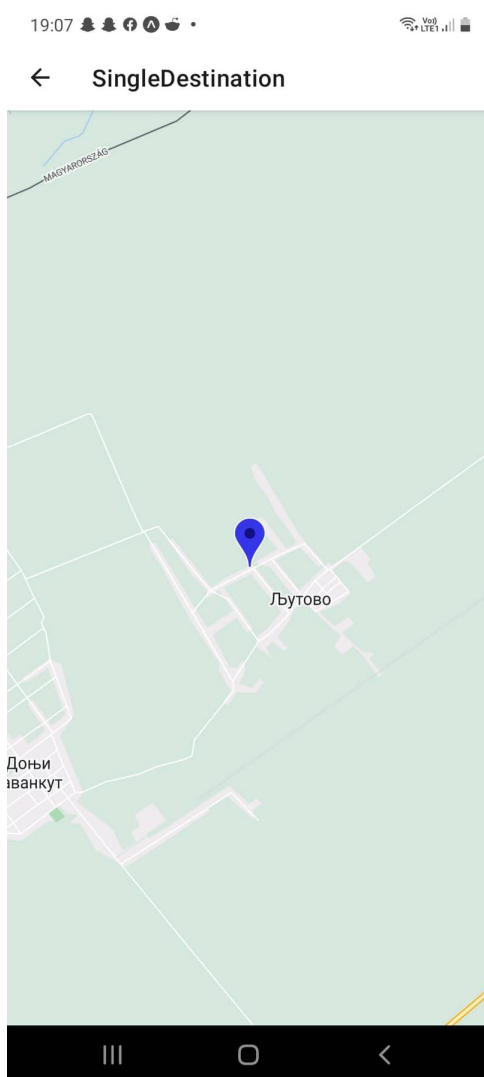
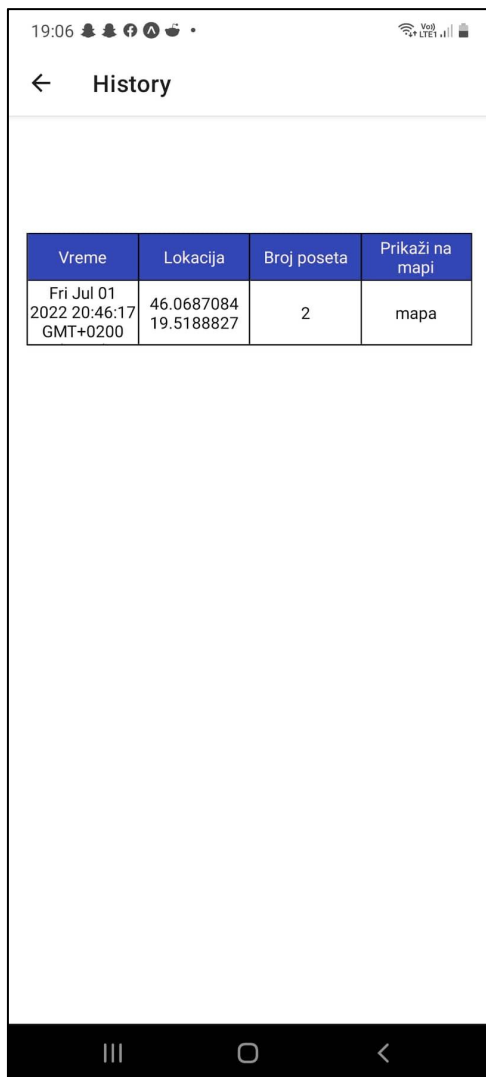
Ukoliko korisnik želi da pregleda destinacije u blizini svoje lokacije, prvo mora da odobri pristup korišćenju lokacije mobilnog telefona od strane mobilne aplikacije. Kada korisnik pokuša da uđe u pregleđ destinacija, iskočiće mu prozor na kome može da odobri ili da odbije korišćenje lokacije telefona.



Nakon toga, korisniku se na mapi prikazuje njegova trenutna lokacija i destinacije oko njega u kurgu od 2 km. Marker koji označava korisnikovu lokaciju je drugačije boje u odnosu na markere koji označavaju destinacije u krugu. Destinacije van kruga nisu vidljive na mapi. Korisnik može da pomera svoju lokaciju zadržavanjem prsta na markeru svoje lokacije i pomeranjem markera na određeno mesto na mapi. Nakon toga se pomera i krug i osvežavaju se destinacije. Takođe, svakih 10 sekundi se proverava da li se korisnik pomerio, i ukoliko jeste, osvežava se i sama lokacija korisnika i destinacije u krugu oko korisnikove lokacije. Zbog boljih performansi, lokacija se ne uzima sa 100%-tnom preciznošću prilikom osvežavanja lokacije, što znači da korisnik treba da se pomeri za određenu fizičku razdaljinu da bi se mapa osvežila. Prilikom samog ulaza na pregled destinacija u blizini, beleži se trenutna lokacija korisnika na mapi, dok se nakon toga ne beleže daljnje lokacije, zbog validnosti podataka (da korisnik ne bi imao hiljade zabeleženih lokacija u razmaku od nekoliko metara).



Druga funkcionalnost mobilne aplikacije je pregled istorije korisnikovih lokacija. Lokacije se prikazuju u tabeli. Svaka lokacija je dostupna i na mapi klikom na link u poslednjoj koloni tabele. Za svaku lokaciju u istoriji lokacija se beleži i broj “poseta”, odnosno koliko puta je korisnik pristupio mapi da destinacijama u blizini sa identične lokacije.



KORIŠĆENA LITERATURA

W1. <https://reactnative.dev/>

W2. <https://expo.dev/>