

Dokumentacija za projekat **ChargingStation**

Na temu “Punjači za električna vozila”

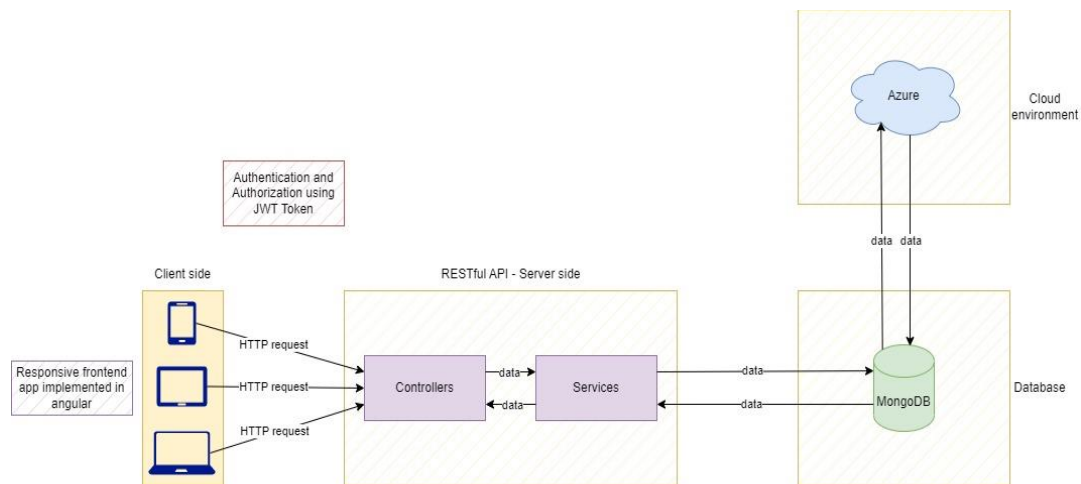
Sadržaj

1. Opis projekta	3
2. Arhitektura aplikacije	4
3. Korišćene tehnologije	5
3.1 Angular	5
3.2 Nest.js	5
3.3 MongoDB	6
4. Struktura baze podataka.....	7
4.1 Modeli u JSON zapisu	7
5. Korišćeni upiti ka bazi	9
5.1 Upiti za korisnika	9
5.2 Upiti za punjače.....	11
5.3 Upiti za termine punjenja	12
5.4 Upiti za vozila	12
5.5 Upiti za zakazivanje punjenja.....	13
5.6 Upiti za istoriju punjenja.....	14
6. Korisnički interfejs	15
6.1 Uloga Administratora	15
6.2 Uloga Vozača.....	23
7. Funkcionalnosti.....	28
7.1 Funkcionalnosti administratora	28
7.2 Funkcionalnosti vozača.....	29
7.3 Algoritam za nalaženje najpogodnijeg punjača	29
Parametri koji se uzimaju u obzir tokom izbora punjača:.....	30
7.4 Slučajevi korišćenja	30
8. Dokumentacija koda	Error! Bookmark not defined.
8.1 Backend deo projekta.....	Error! Bookmark not defined.
8.1.5 Šeme	Error! Bookmark not defined.
8.1.9 Podržani HTTP zahtevi	39
9. Potencijalna unapređenja.....	Error! Bookmark not defined.
9.1. Optimizacija algoritma za nalaženje najpogodnijeg punjača	Error! Bookmark not defined.
9.2. Implementacija napredne analitike i izveštavanja	Error! Bookmark not defined.
9.3. Povezivanje sa drugim sistemima	Error! Bookmark not defined.
9.4. Povećanje sigurnosti sistema.....	Error! Bookmark not defined.
9.5. Unapređenje korisničkog interfejsa (UI) i korisničkog iskustva (UX)	Error! Bookmark not defined.
9.6. Dodavanje funkcionalnosti za društvenu interakciju	Error! Bookmark not defined.
9.7. Napredne funkcionalnosti za administratore ...	Error! Bookmark not defined.

1. Opis projekta

Potrebno je napraviti web aplikaciju koja će omogućiti vozačima da imaju uvid u sve punjače na određenoj teritoriji i jednostavnim unosom lokacije moći će da ispituju putanje do istih. Takođe će postojati administrator koji će imati mogućnost da nadzoriše sve entitete u sistemu. Svaki punjač će imati unapred definisane termine u kojima se može puniti, tako da vozač prilikom prihvatanja punjenja na tom punjaču mora selektovati i termin u kojem će doći. Kada vozač potvrdi termin, aplikacija treba da blokira dalje zauzimanje tog punjača u tom terminu. Iz gore navedenog se može primetiti da će aplikacija imati 2 tipa korisnika: **Vozač** i **Administrator**, dakle potrebno je implementirati i mehanizam za autentifikaciju i autorizaciju.

2. Arhitektura aplikacije



2.1. Prikaz arhitekture aplikacije

3. Korišćene tehnologije

3.1 Angular

Angular je okvir za izradu aplikacija na strani klijenta koji koristi TypeScript. Omogućava razvoj dinamičkih web aplikacija i pruža sledeće ključne funkcionalnosti:

- 1) **Komponente:** Osnovni gradivni blokovi Angular aplikacija, što omogućava modularnost i ponovnu upotrebu koda.
- 2) **Servisi i Dependency Injection:** Angular omogućava kreiranje servisa za rukovanje podacima i logikom, koji se mogu lako koristiti u različitim delovima aplikacije putem dependency injection mehanizma.
- 3) **Router:** Angular router omogućava navigaciju između različitih delova aplikacije bez potrebe za osvežavanjem stranice.
- 4) **Forme i Validacija:** Angular pruža moćan alat za rad sa formama i njihovom validacijom, što je ključno za autentifikaciju i registraciju korisnika.
- 5) **HttpClient:** Modul za HTTP komunikaciju omogućava jednostavno slanje zahteva ka serveru i obradu odgovora.

3.2 Nest.js

Nest.js je okvir za izradu aplikacija na strani servera koji koristi Node.js i TypeScript. Osmišljen je tako da bude fleksibilan, skalabilan i lako održiv. Ključne karakteristike uključuju:

- 1) **Modularna Arhitektura:** Aplikacije su organizovane u module, što omogućava bolju organizaciju koda i jednostavnije upravljanje složenim projektima.
- 2) **Kontroleri:** Kontroleri su zaduženi za rukovanje HTTP zahtevima i definisanje ruta koje korisnici mogu koristiti za interakciju sa aplikacijom.
- 3) **Servisi:** Servisi sadrže poslovnu logiku aplikacije i mogu se koristiti u različitim kontrolerima.
- 4) **ORM Podrška:** Nest.js se može lako integrisati sa različitim ORM alatima, kao što su TypeORM, Sequelize, i Prisma, za rad sa bazama podataka.
- 5) **Autentifikacija i Autorizacija:** Nest.js pruža podršku za implementaciju raznih strategija autentifikacije i autorizacije, uključujući JWT (JSON Web Token) i OAuth.

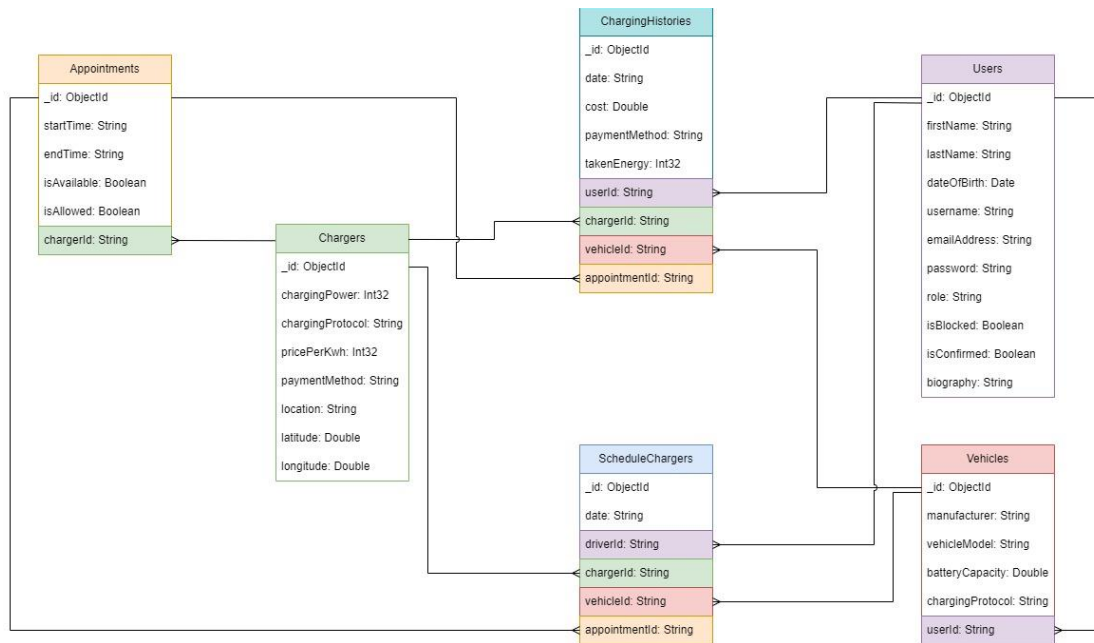
3.3 MongoDB

MongoDB je NoSQL baza podataka koja koristi fleksibilan, dokumentno-orijentisan model podataka. Dizajnirana je za skalabilnost, performanse i lakoću korišćenja. Sledeće ključne funkcionalnosti izdvajaju MongoDB:

- 1) **Dokumentno-orijentisan model:** Podaci se skladište u JSON-sličnim dokumentima, što omogućava prirodnu i intuitivnu mapu između objekata u aplikaciji i podataka u bazi.
- 2) **Šematska fleksibilnost:** MongoDB omogućava skladištenje podataka bez unapred definisane šeme, što pruža fleksibilnost za promene u strukturi podataka bez potrebe za migracijama baze podataka.
- 3) **Indeksiranje:** Napredne opcije indeksiranja omogućavaju brzu i efikasnu pretragu podataka.
- 4) **Replikacija:** MongoDB podržava replikaciju podataka putem replika setova, što obezbeđuje visoku dostupnost i otpornost na greške.
- 5) **Sharding:** MongoDB omogućava horizontalno skaliranje putem shardinga, što omogućava distribuciju podataka preko više servera i podršku za velike količine podataka i visoke performanse.
- 6) **Agregacioni okvir:** Moćan agregacioni okvir omogućava kompleksne upite i analizu podataka koristeći pipelajnovski pristup.
- 7) **Podrška za transakcije:** MongoDB podržava ACID transakcije na više dokumenata, omogućavajući konzistentnost podataka čak i u složenim operacijama.

MongoDB je pogodan za širok spektar aplikacija, uključujući web i mobilne aplikacije, analitičke platforme i sisteme za upravljanje sadržajem, pružajući visoku fleksibilnost i skalabilnost.

4. Struktura baze podataka



4.1 Prikaz strukture MongoDB baze podataka

4.1 Modeli u JSON zapisu

Appointment:

```
{
  "startTime": "string",
  "endTime": "string",
  "isAvailable": "boolean",
  "isAllowed": "boolean",
  "chargerId": "string"
}
```

Charger:

```
{
  "chargingPower": "number",
  "chargingProtocol": "string",
  "pricePerKwh": "number",
  "paymentMethod": "string",
  "location": "string",
  "latitude": "number",
  "longitude": "number"
}
```

ScheduleCharger:

```
{
  "driverId": "string",
  "chargerId": "string",
  "appointmentId": "string",
  "vehicleId": "string",
  "date": "string"
}
```

ChargingHistory:

```
{
  "date": "string",
  "cost": "number",
  "paymentMethod": "string",
  "takenEnergy": "number",
  "userId": "string",
  "chargerId": "string",
  "vehicleId": "string",
  "appointmentId": "string"
}
```

User:

```
{
  "firstName": "string",
  "lastName": "string",
  "dateOfBirth": "Date",
  "biography": "string",
  "username": "string",
  "emailAddress": "string",
  "password": "string",
  "role": "string",
  "gender": "string",
  "isBlocked": "boolean",
  "isConfirmed": "boolean",
  "refreshTokenHash": "string"
}
```

Vehicle:

```
{
  "manufacturer": "string",
  "vehicleModel": "string",
  "batteryCapacity": "number",
  "chargingProtocol": "string",
  "userId": "string"
}
```


5. Korišćeni upiti ka bazi

U ovom poglavlju su predstavljeni upiti ka MongoDB bazi za svaki entitet posebno.

5.1 Upiti za korisnika

Dobaviti korisnika sa određenim id-em:

```
const user = await this.userModel.findById(model._id);
```

Ažurirati korisnika sa određenim id-em

```
await this.userModel.findOneAndUpdate(  
  {  
    _id: model._id,  
  },  
  {  
    ...model  
  }  
);
```

Dobaviti sve korisnike koji su u ulozi vozača:

```
const users = await this.userModel.find({ role: 'driver' });
```

Dobaviti sve korisnike koji nisu blokirani, potvrđeni su i imaju ulogu vozača:

```
const users = await this.userModel.find({ isBlocked: false, isConfirmed: true,  
role: 'driver' });
```

Dobaviti sve korisnike koji su potvrđeni i imaju ulogu vozača:

```
const users = await this.userModel.find({ isConfirmed: true, role: 'driver' });
```

Dobaviti sve potvrđene korisnike čije ime, prezime, biografija, korisničko ime ili pol sadrže prosleđeni parametar query:

```
const confirmedUsers = await this.userModel.find(  
  $and: [  
    { isConfirmed: true },  
    {  
      $or: [  
        { firstName: new RegExp(query, 'i') },  
        { lastName: new RegExp(query, 'i') },  
        { biography: new RegExp(query, 'i') },  
        { username: new RegExp(query, 'i') },  
        { gender: new RegExp(query, 'i') }  
      ]  
    }  
  ]  
);
```

```
    ]  
  });
```

Dobaviti sve nepotvrđene korisnike koji imaju ulogu vozača:

```
const unconfirmedUsers = await this.userModel.find({ isConfirmed: false, role:  
'driver' });
```

Dobaviti sve nepotvrđene korisnike čije ime, prezime, biografija, korisničko ime ili pol sadrže prosleđeni parametar query:

```
const unconfirmedUsers = await this.userModel.find({  
  $and: [  
    { isConfirmed: false },  
    {  
      $or: [  
        { firstName: new RegExp(query, 'i') },  
        { lastName: new RegExp(query, 'i') },  
        { biography: new RegExp(query, 'i') },  
        { username: new RegExp(query, 'i') },  
        { gender: new RegExp(query, 'i') }  
      ]  
    }  
  ]  
});
```

Dobaviti sve blokirane vozače:

```
const blockedDrivers = await this.userModel.find({ isBlocked: true, role:  
'driver' });
```

Dobaviti sve neblokirane vozače:

```
const unblockedDrivers = await this.userModel.find({ isBlocked: false, role:  
'driver' });
```

Blokirati korisnika sa određenim id-em:

```
await this.userModel.findByIdAndUpdate(userId, { isBlocked: true });
```

Odblokirati korisnika sa određenim id-em:

```
await this.userModel.findByIdAndUpdate(userId, { isBlocked: false });
```

Potvrditi korisnika sa određenim id-em:

```
await this.userModel.findByIdAndUpdate(userId, { isConfirmed: true });
```

Obrisati korisnika sa određenim id-em:

```
await this.userModel.findByIdAndDelete(userId);
```

Pronaći korisnika koji ima korisničko ime ili elektronsku adresu zadatu u promenljivoj userNameOrEmail:

```
const user = await this.userModel.findOne({ $or: [  
  { username: userNameOrEmail },  
  { email: userNameOrEmail }  
]
```

```
});
```

Sačuvati novog korisnika sa prosleđenim podacima uz dodatno hash-ovanje lozinke:

```
const newUser = await new this.userModel({
  ...model,
  password: await this.hashData(model.password)
}).save();
```

Pronaći vozača sa određenim id-em:

```
const existingDriver = await this.userModel.findOne({ _id: model.driverId, role:
'driver' });
```

5.2 Upiti za punjače

Pronaći punjač sa određenim id-em:

```
const charger = await this.chargerModel.findById(chargerId);
```

Sačuvati punjač sa prosleđenim podacima:

```
const savedCharger = await new this.chargerModel(model).save();
```

Ažurirati punjač sa određenim id-em:

```
await this.chargerModel.findByIdAndUpdate(model._id, model);
```

Dobaviti sve punjače:

```
const chargers = await this.chargerModel.find();
```

Dobaviti sve punjače čiji protokol, metoda plaćanja, lokacija, snaga ili cena po KWH sadrže prosleđeni parametar query:

```
const chargers = await this.chargerModel.find({
  $or: [
    { chargingProtocol: new RegExp(query, 'i') },
    { paymentMethod: new RegExp(query, 'i') },
    { location: new RegExp(query, 'i') },
    { chargingPower: parseFloat(query) || 0.0 },
    { pricePerKwh: parseFloat(query) || 0.0 },
  ]
});
```

Obrisati punjač sa određenim id-em:

```
await this.chargerModel.findByIdAndDelete(chargerId);
```

5.3 Upiti za termine punjenja

Sačuvati termin sa prosleđenim podacima:

```
await new this.appointmentModel(model).save();
```

Pronaći termin sa prosleđenim id-em:

```
const appointment = this.appointmentModel.findById(model._id);
```

Ažurirati termin sa određenim id-em:

```
await this.appointmentModel.findByIdAndUpdate(model._id, model);
```

Dozvoliti termin sa određenim id-em za punjač sa određenim id-em:

```
await this.appointmentModel.findByIdAndUpdate(  
  { _id: model.appointmentId, chargerId: model.chargerId },  
  { isAllowed: true, isAvailable : true }  
);
```

Zabraniti termin sa određenim id-em za punjač sa određenim id-em:

```
await this.appointmentModel.findByIdAndUpdate(  
  { _id: model.appointmentId, chargerId: model.chargerId },  
  { isAllowed: false, isAvailable : false }  
);
```

Dobaviti sve termine za određeni punjač:

```
const appointments = await this.appointmentModel.find({ chargerId });
```

Obrisati termin sa određenim id-em:

```
await this.appointmentModel.findByIdAndDelete(appointmentId);
```

Obrisati sve termine za punjač sa određenim id-em:

```
await this.appointmentModel.deleteMany({ chargerId });
```

Pronaći termin sa određenim id-em vezan za određeni punjač:

```
const existingAppointment = await this.appointmentModel.findOne(  
  {  
    _id: model.appointmentId,  
    chargerId: model.chargerId  
  });
```

5.4 Upiti za vozila

Pronaći vozilo sa određenim id-em:

```
const vehicle = await this.vehicleModel.findById(vehicleId);
```

Ažurirati vozilo sa određenim id-em:

```
await this.vehicleModel.findByIdAndUpdate(model._id, model);
```

Dobaviti sva vozila od određenog vozača:

```
const vehicles = await this.vehicleModel.find({ userId });
```

Dobaviti sva vozila:

```
const vehicles = await this.vehicleModel.find();
```

Dobaviti sva vozila čiji proizvođač, model, protokol ili kapacitet baterije sadrže prosleđeni parametar query:

```
const vehicles = await this.vehicleModel.find({
  $or: [
    { manufacturer: new RegExp(query, 'i') },
    { vehicleModel: new RegExp(query, 'i') },
    { chargingProtocol: new RegExp(query, 'i') },
    { batteryCapacity: parseFloat(query) || 0.0 },
  ]
});
```

Obrisati vozilo sa određenim id-em:

```
await this.vehicleModel.findByIdAndDelete(vehicleId);
```

Pronaći vozilo sa određenim id-em koje pripada tačno određenom vozaču:

```
const existingVehicle = await this.vehicleModel.findOne({ _id: model.vehicleId,
userId: model.driverId });
```

5.5 Upiti za zakazivanje punjenja

Dobaviti termine punjenja za određeni datum i punjač:

```
const scheduleChargers = await this.scheduleChargerModel.find({
  chargerId: chargerId,
  date: date
});
```

Dobaviti termine punjenja za određeni datum, termin i punjač:

```
const scheduleChargings = await this.scheduleChargerModel.findOne({
  chargerId: model.chargerId,
  appointmentId: model.appointmentId,
  date: model.date
});
```

Dobaviti termine punjenja za određeni datum i određeni punjač:

```
await new this.scheduleChargerModel(model).save();
```

Dobaviti termine punjenja za određenog vozača:

```
const scheduleChargers = await this.scheduleChargerModel.find({ driverId });
```

Dobaviti termine punjenja od određenog vozača čiji datum sadrži vrednost promenljive query:

```
const scheduleChargers = await this.scheduleChargerModel.find({
  date: new RegExp(query, 'i'),
  driverId: driverId
});
```

5.6 Upiti za istoriju punjenja

Sačuvati punjenje u istoriju punjenja:

```
await new this.chargingHistoryModel(model).save();
```

Pronaći istoriju punjenja sa određenim id-em:

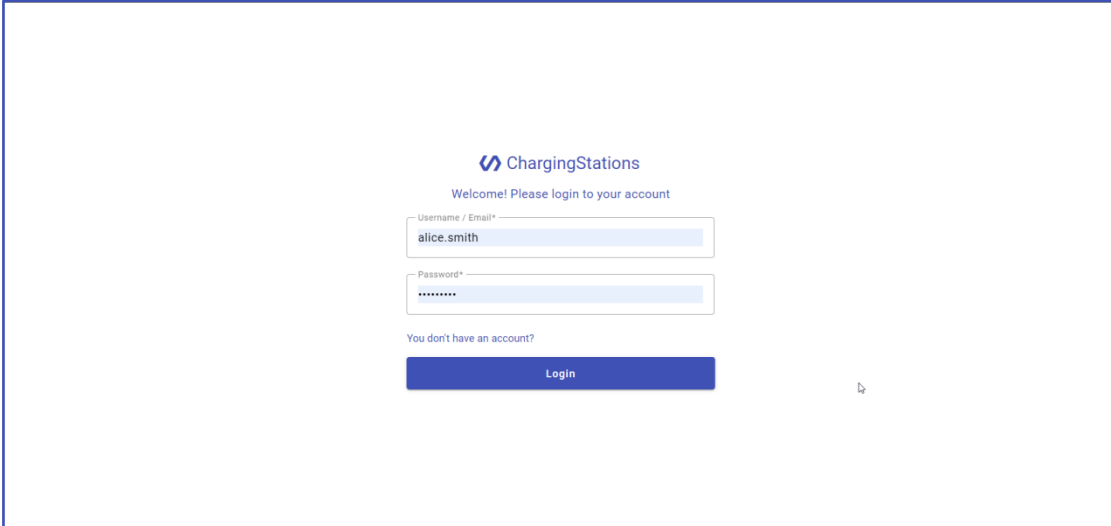
```
const chargingHistory = await
this.chargingHistoryModel.findById(chargingHistoryId);
```

Dobaviti sve istorije punjenja za određenog vozača:

```
const chargingHistories = await this.chargingHistoryModel.find({ userId });
```

6. Korisnički interfejs

6.1 Uloga Administratora



The screenshot shows the login interface for 'ChargingStations'. At the top, there is a logo and the text 'Welcome! Please login to your account'. Below this, there are two input fields: 'Username / Email*' containing 'alice.smith' and 'Password*' with masked characters. A link 'You don't have an account?' is positioned below the password field. At the bottom, there is a blue 'Login' button.

ChargingStations

Welcome! Please login to your account

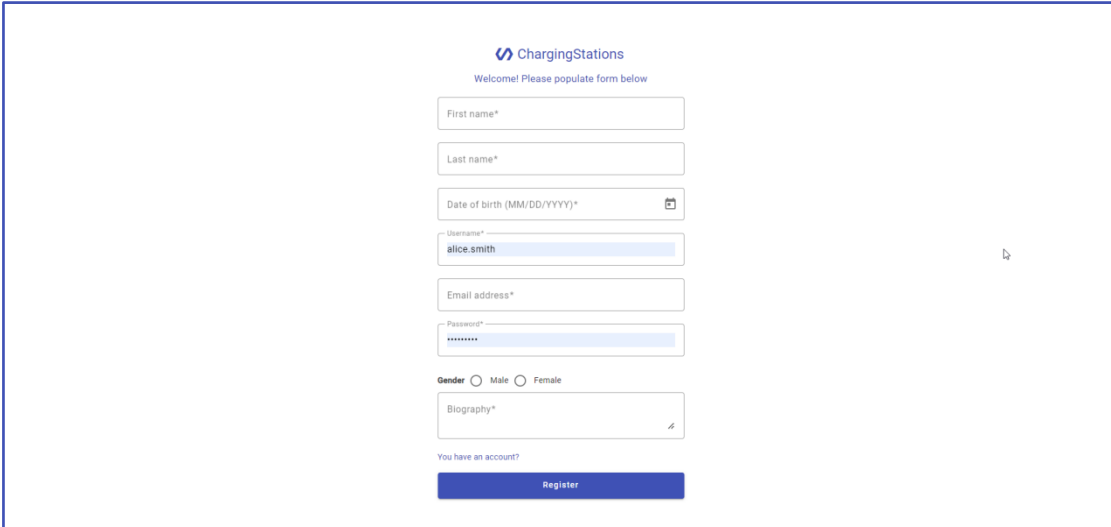
Username / Email*
alice.smith

Password*

[You don't have an account?](#)

Login

6.1.1. Prikaz stranice za prijavu



The screenshot shows the registration interface for 'ChargingStations'. At the top, there is a logo and the text 'Welcome! Please populate form below'. Below this, there are several input fields: 'First name*', 'Last name*', 'Date of birth (MM/DD/YYYY)*' with a calendar icon, 'Username*' containing 'alice.smith', 'Email address*', and 'Password*' with masked characters. There are also radio buttons for 'Gender' (Male and Female) and a 'Biography*' field with a text area icon. A link 'You have an account?' is positioned above the 'Register' button.

ChargingStations

Welcome! Please populate form below

First name*

Last name*

Date of birth (MM/DD/YYYY)*

Username*
alice.smith

Email address*

Password*

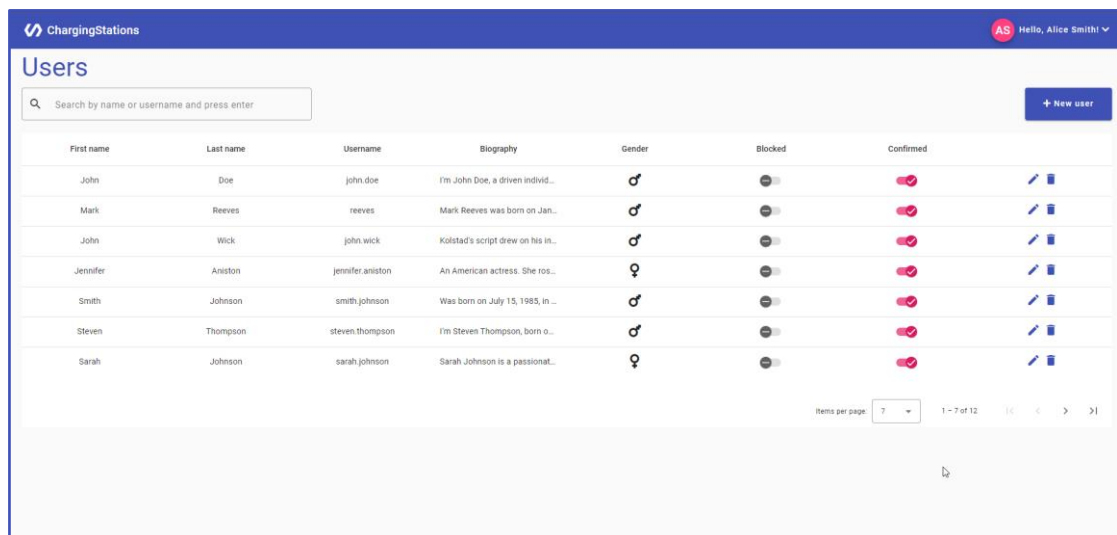
Gender ☐ Male ☐ Female

Biography*

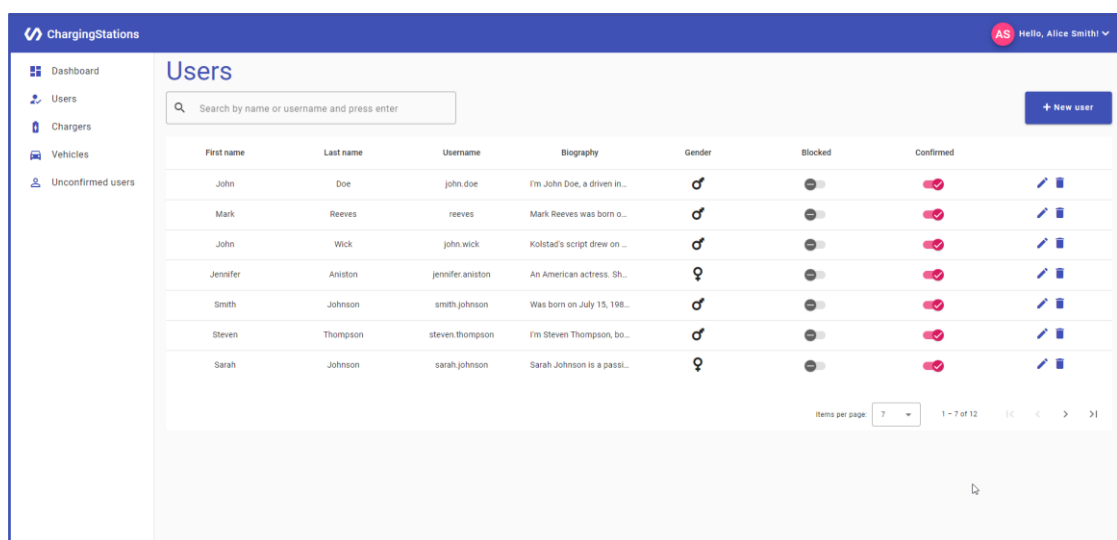
[You have an account?](#)

Register

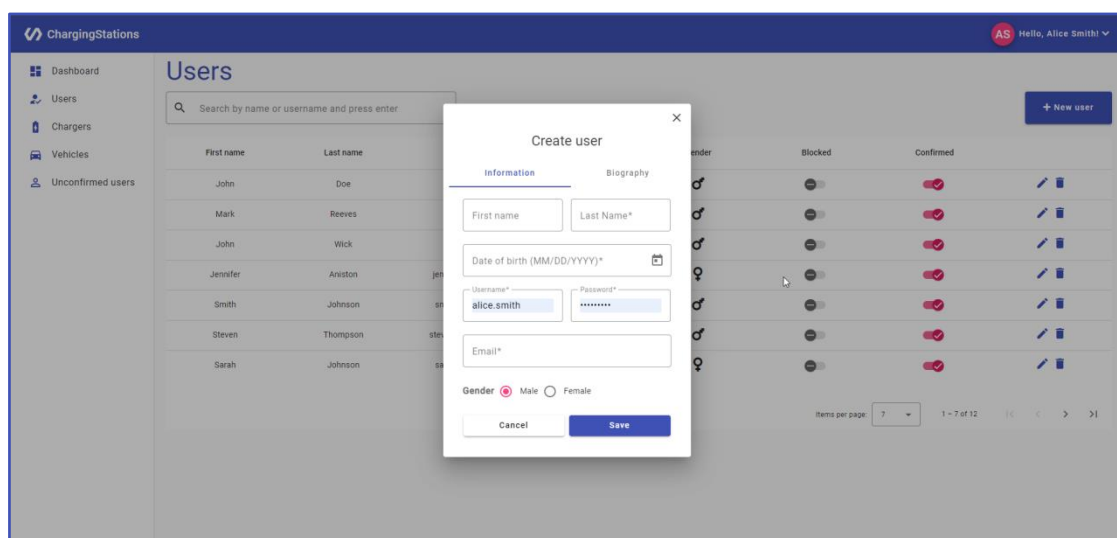
6.1.2. Prikaz stranice za registraciju



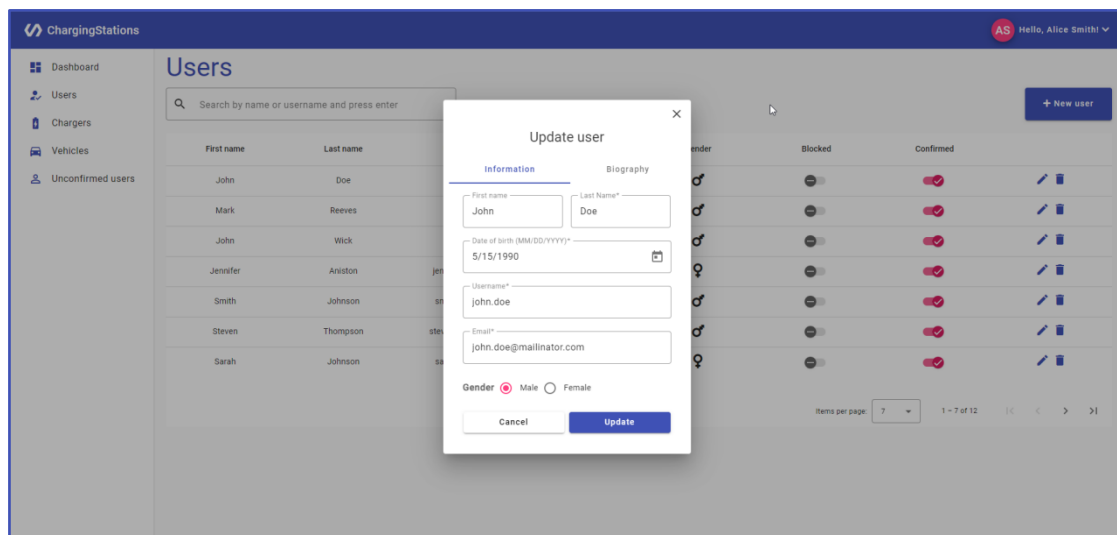
6.1.3. Prikaz stranice svih korisnika



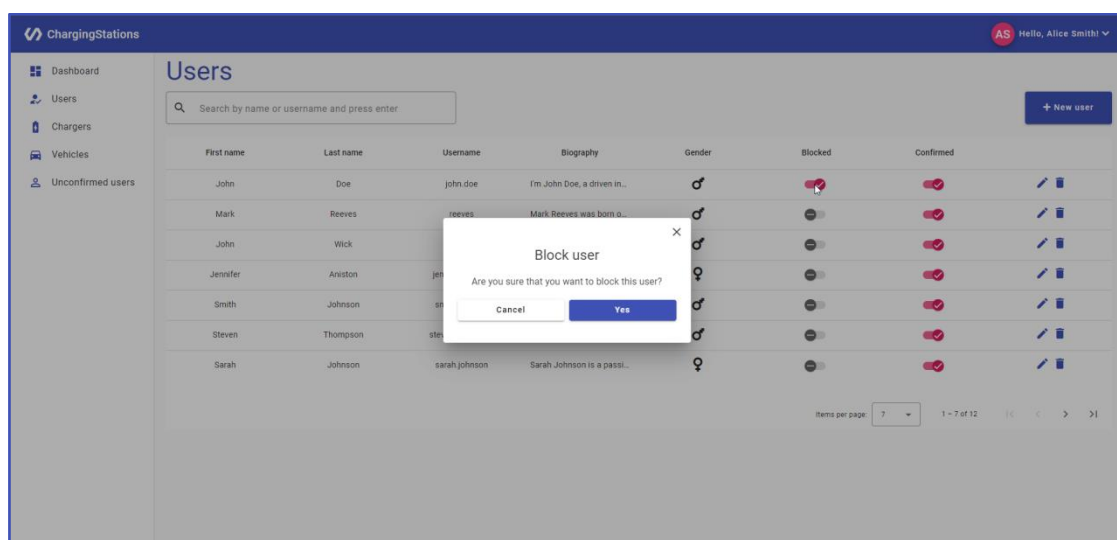
6.1.4. Prikaz mogućnosti otkrivanja opcija sa strane



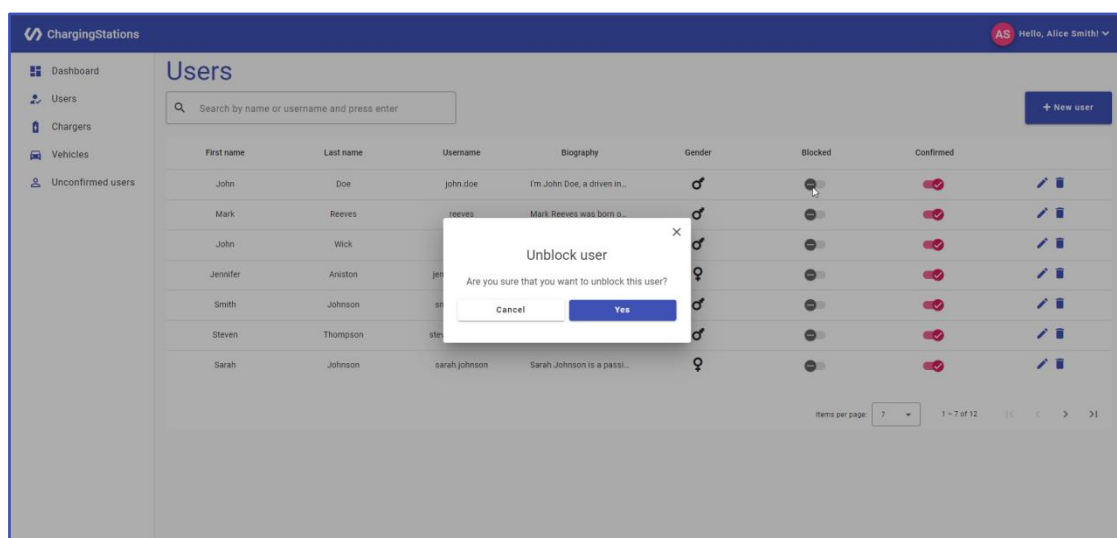
6.1.5. Prikaz modala za kreiranje korisnika



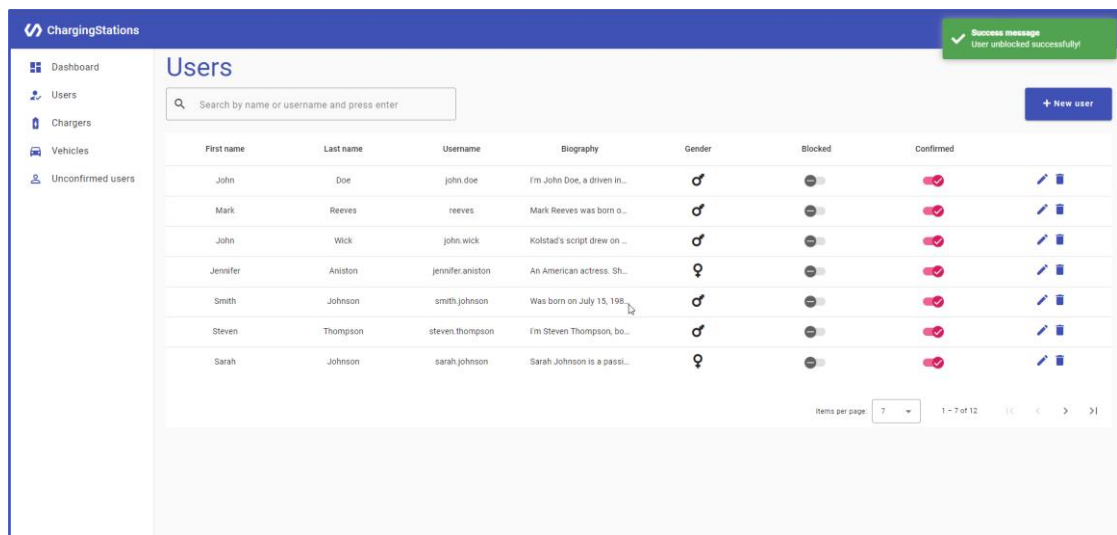
6.1.6. Prikaz modala za ažuriranje korisnika



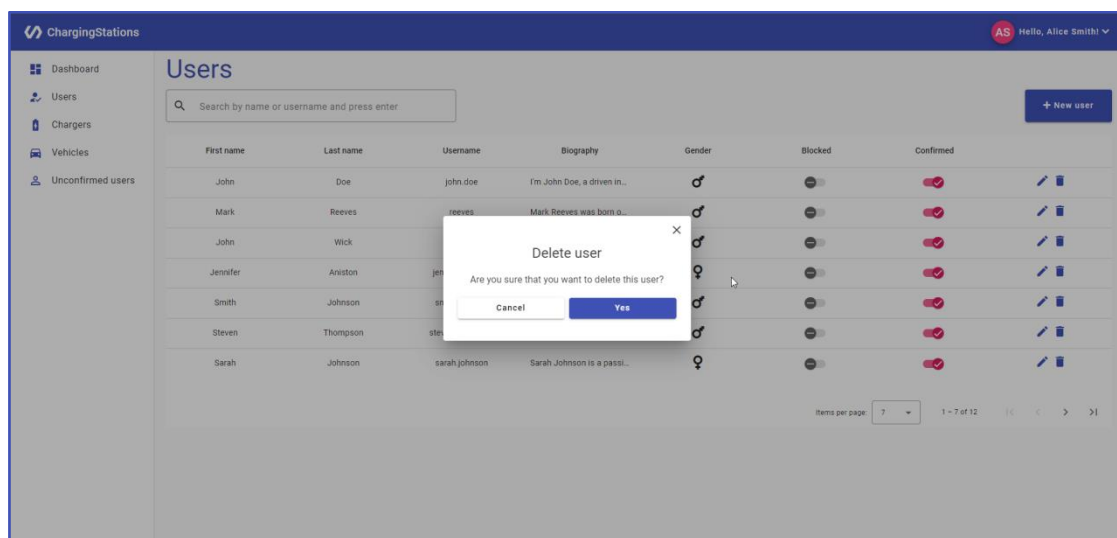
6.1.7. Prikaz modala za blokiranje korisnika



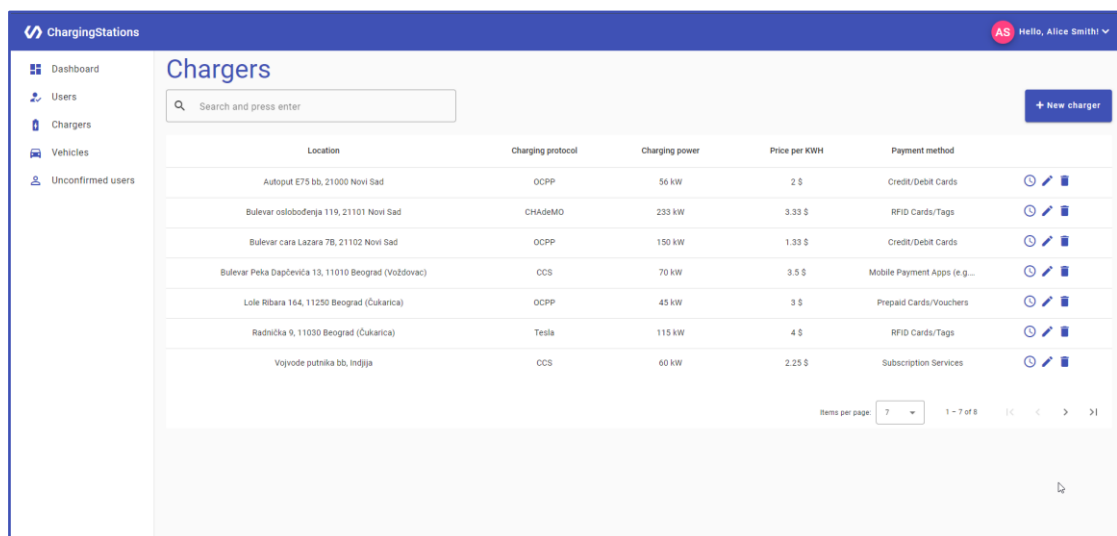
6.1.8. Prikaz modala za odblokiranje korisnika



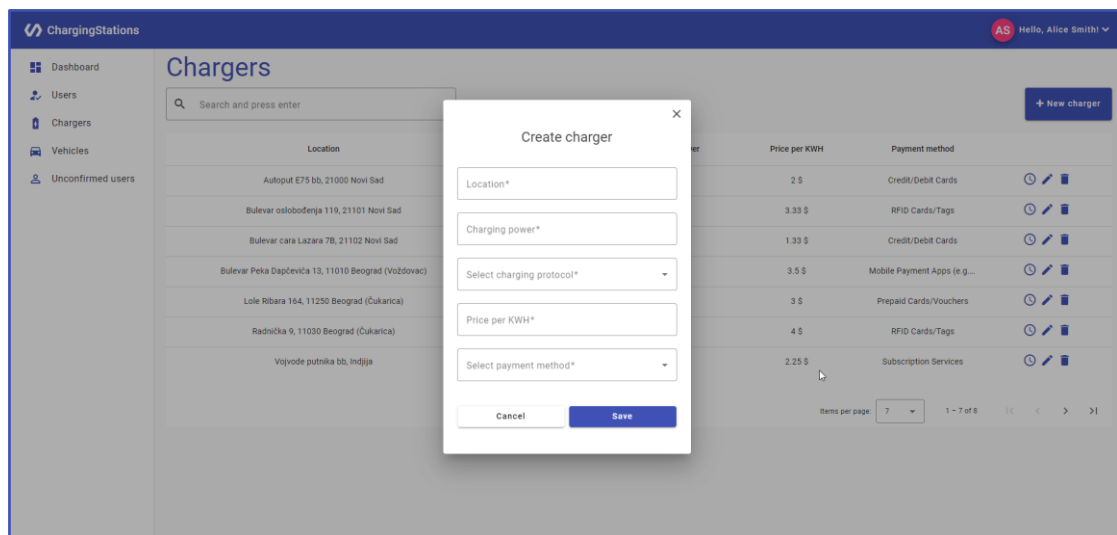
6.1.9. Prikaz notifikacione poruke nakon uspešne operacije



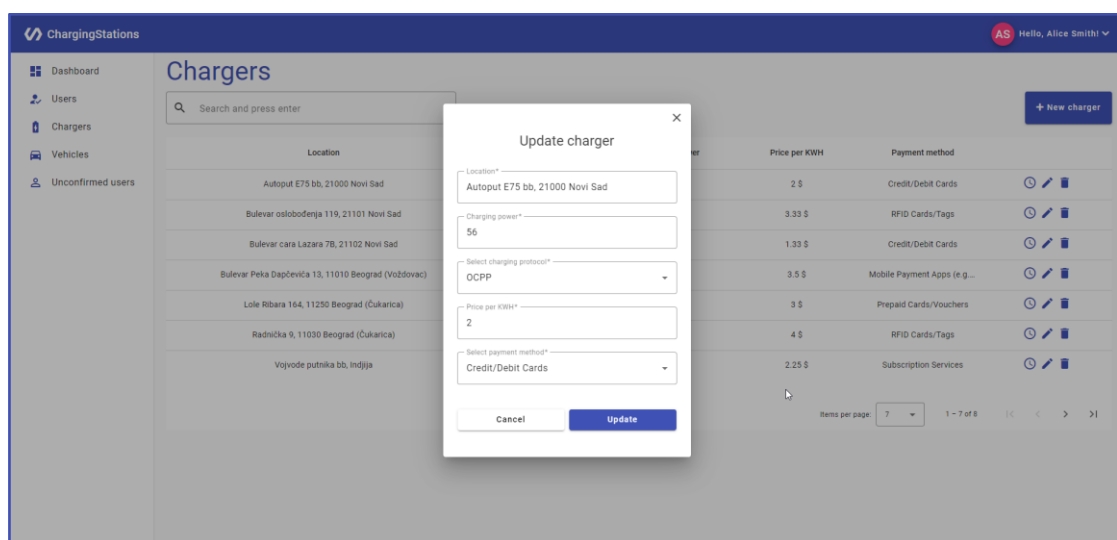
6.1.10. Prikaz modala za brisanje korisnika



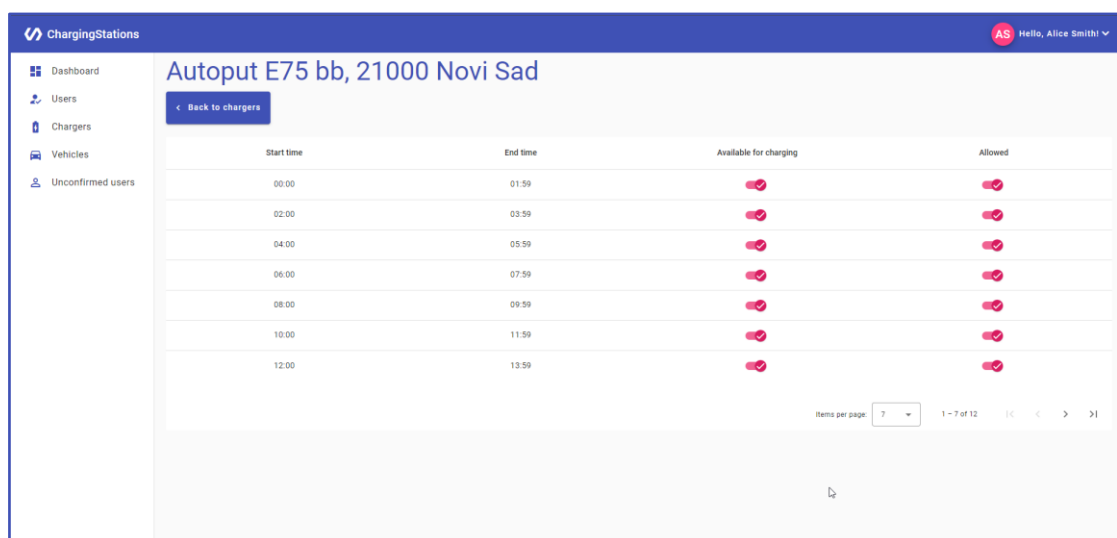
6.1.11. Prikaz stranice svih punjača



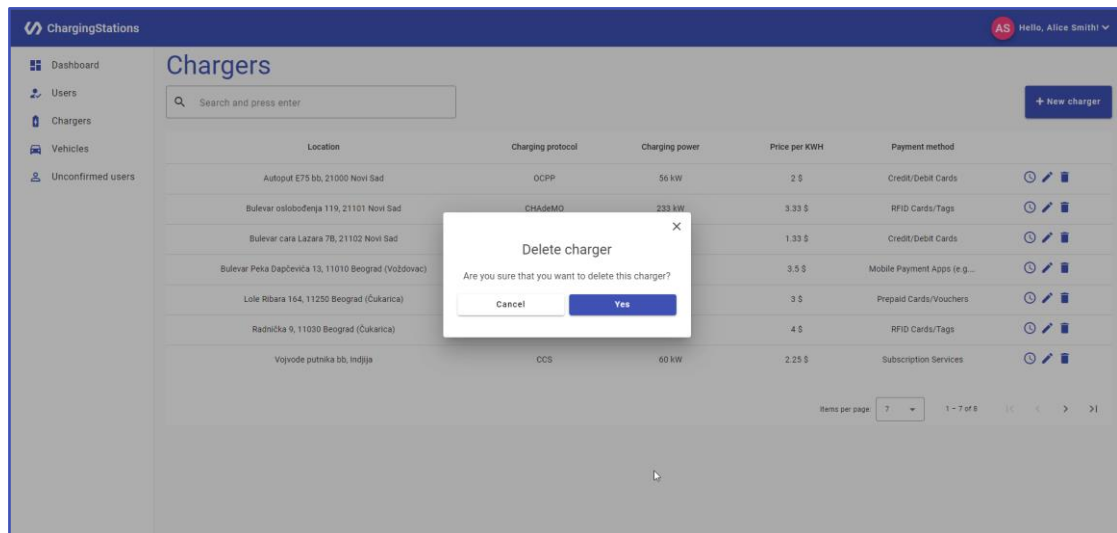
6.1.12. Prikaz modala za kreiranje punjača



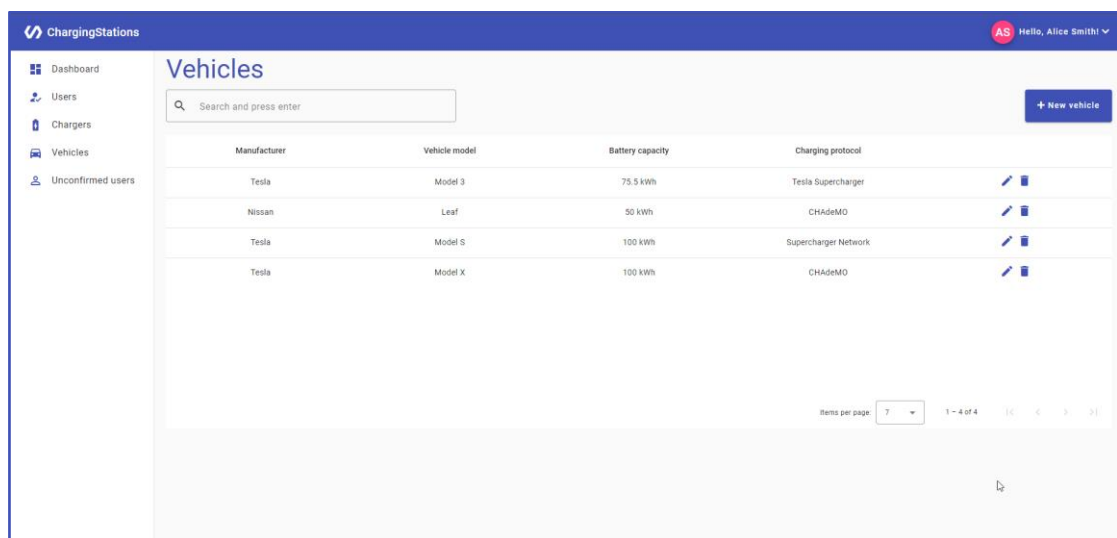
6.1.13. Prikaz stranice za ažuriranje punjača



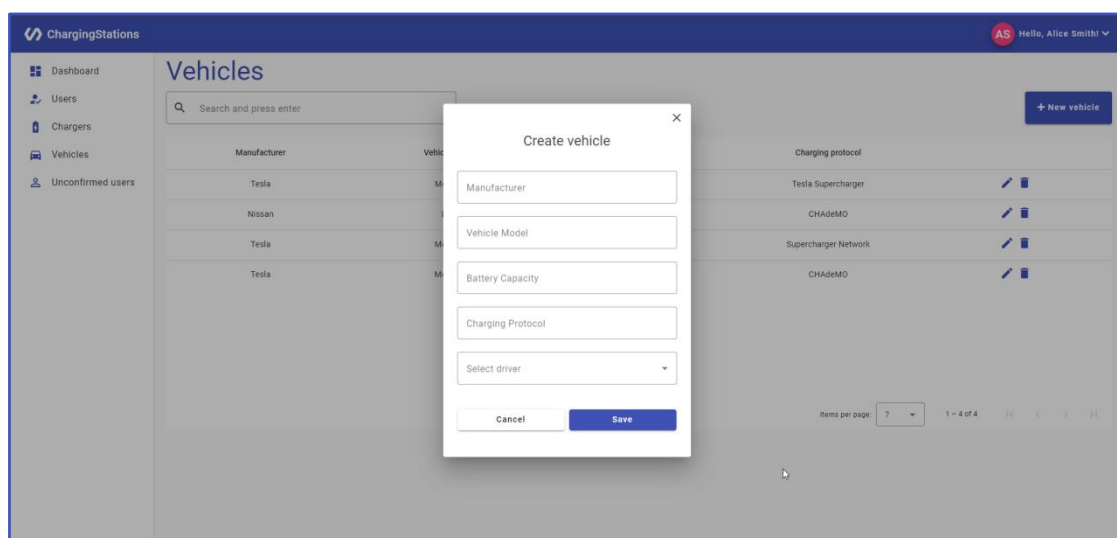
6.1.14. Prikaz stranice svih termina za određeni punjač



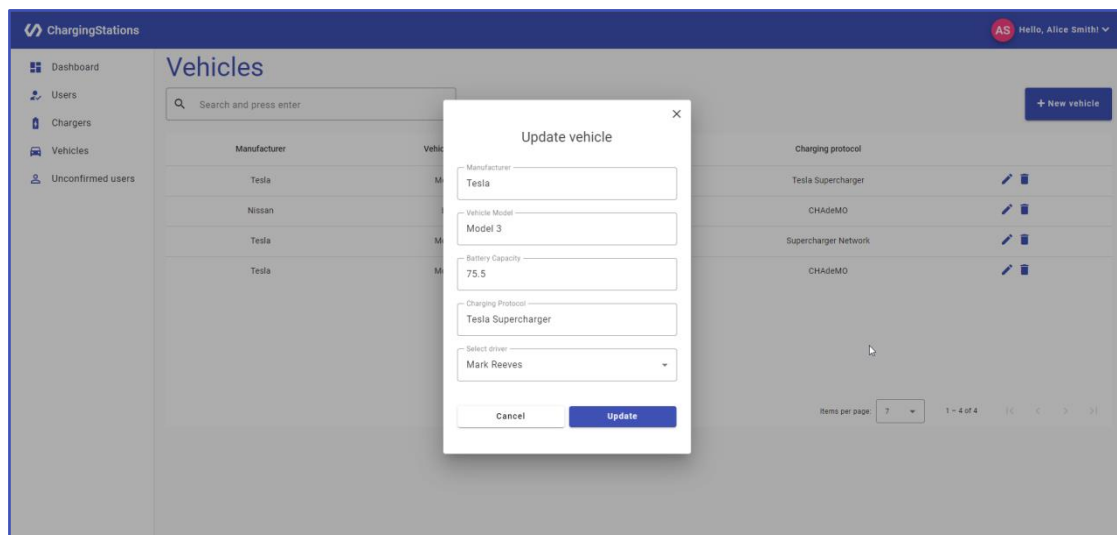
6.1.15. Prikaz modala za brisanje punjača



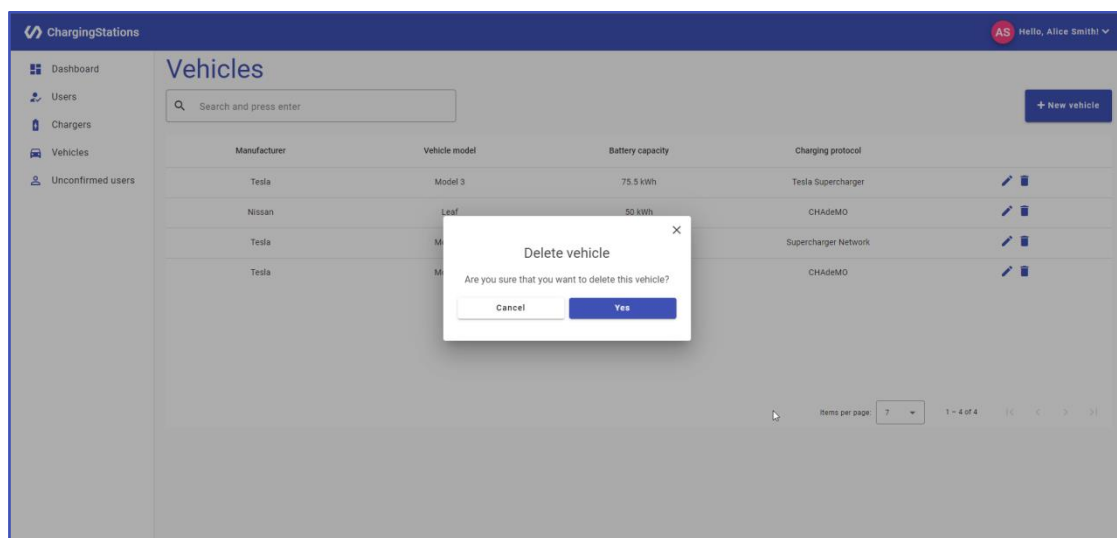
6.1.16. Prikaz stranice svih vozila



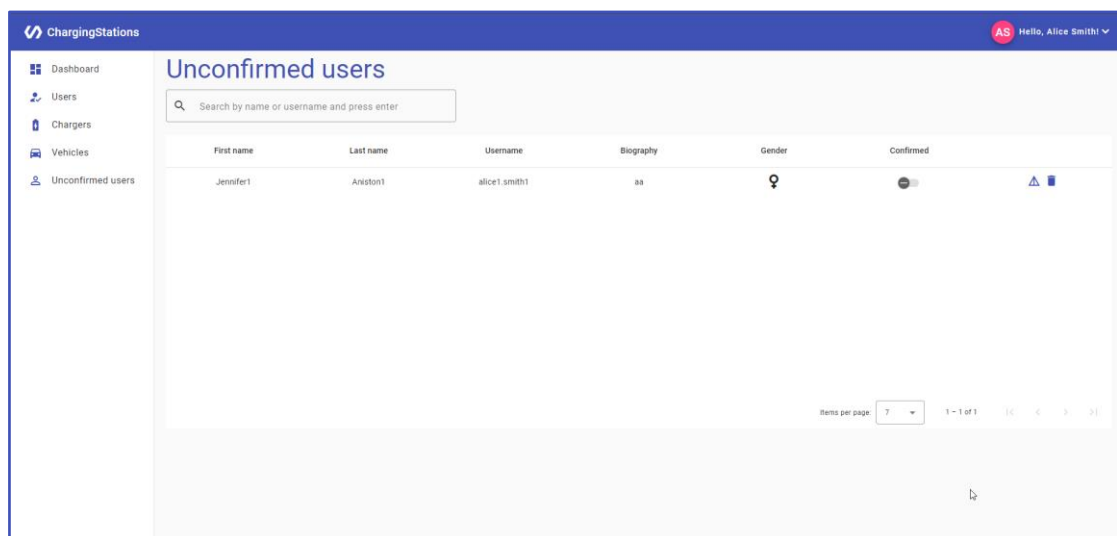
6.1.17. Prikaz modala za kreiranje vozila



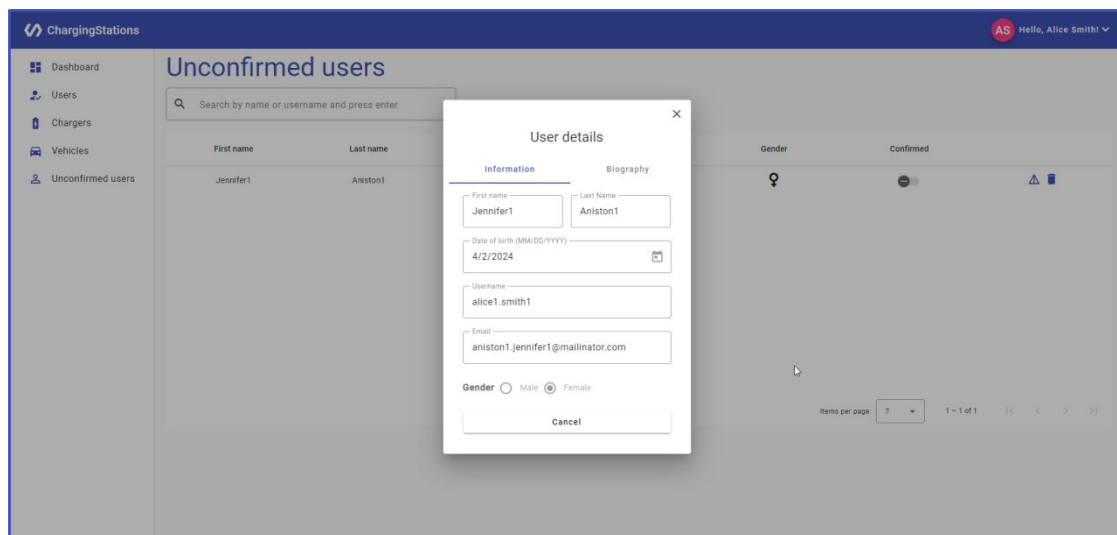
6.1.18. Prikaz modala za ažuriranje vozila



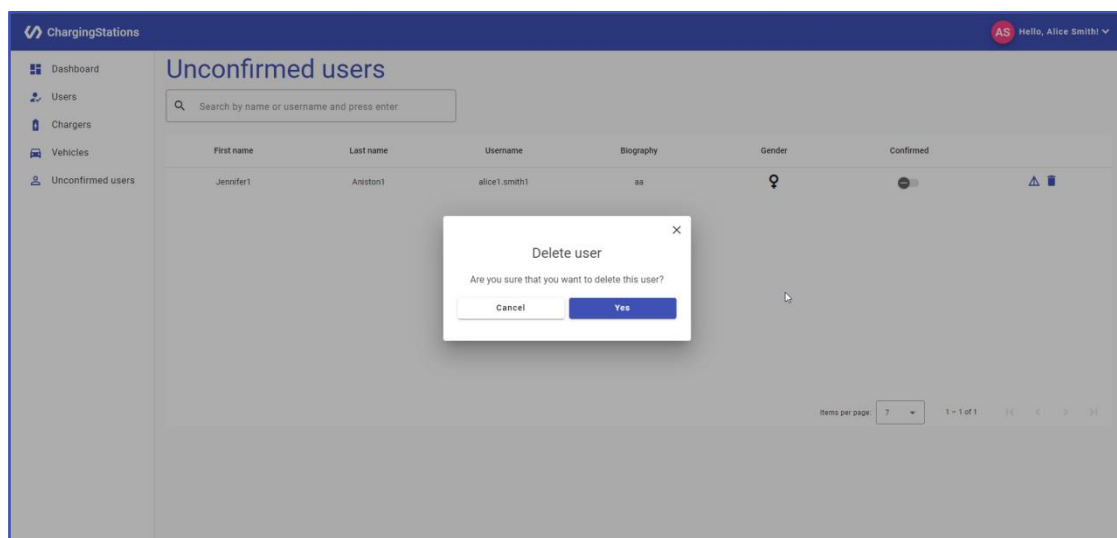
6.1.19. Prikaz modala za brisanje vozila



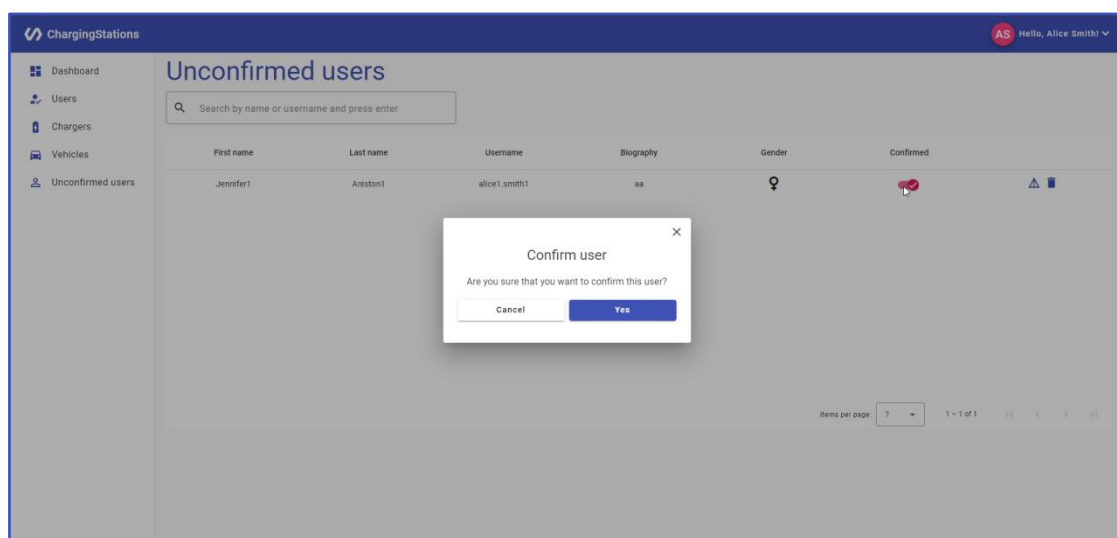
6.1.20. Prikaz stranice svih nepotvrđenih korisnika



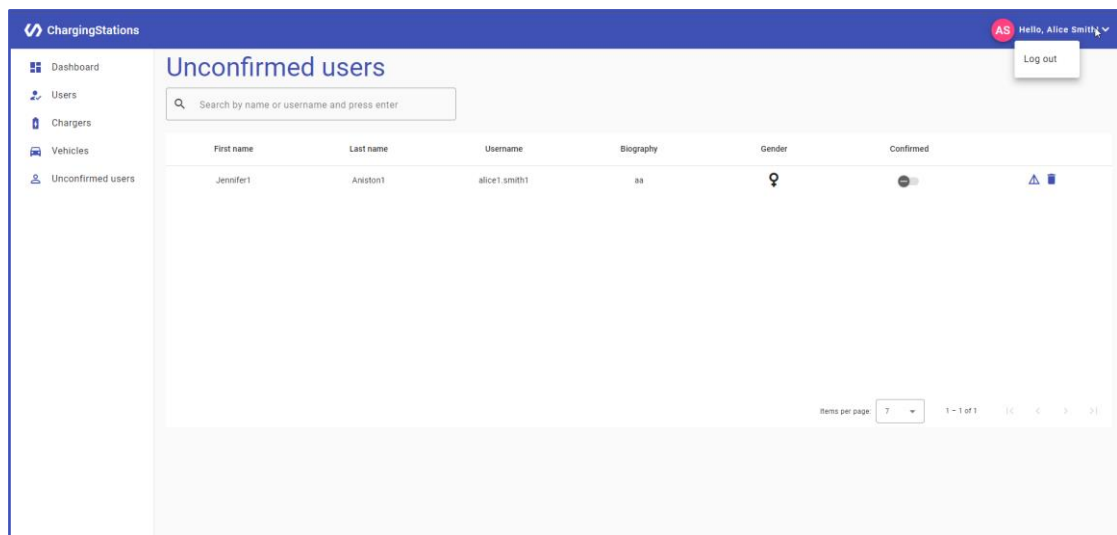
6.1.21. Prikaz modala za detalje nepotvrđenog korisnika



6.1.22. Prikaz modala za brisanje nepotvrđenog korisnika

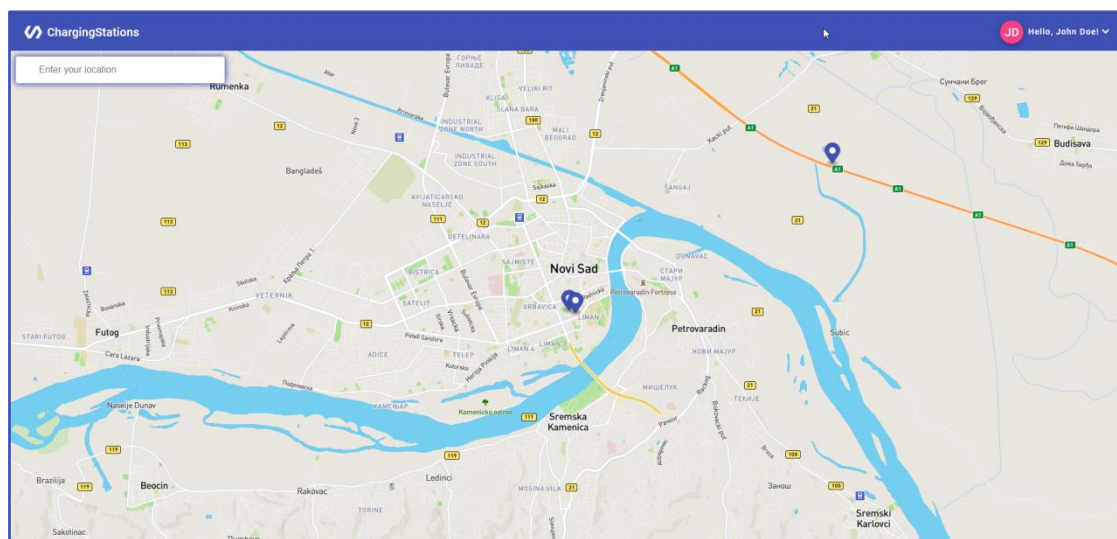


6.1.23. Prikaz modala za potvrdu nepotvrđenog korisnika

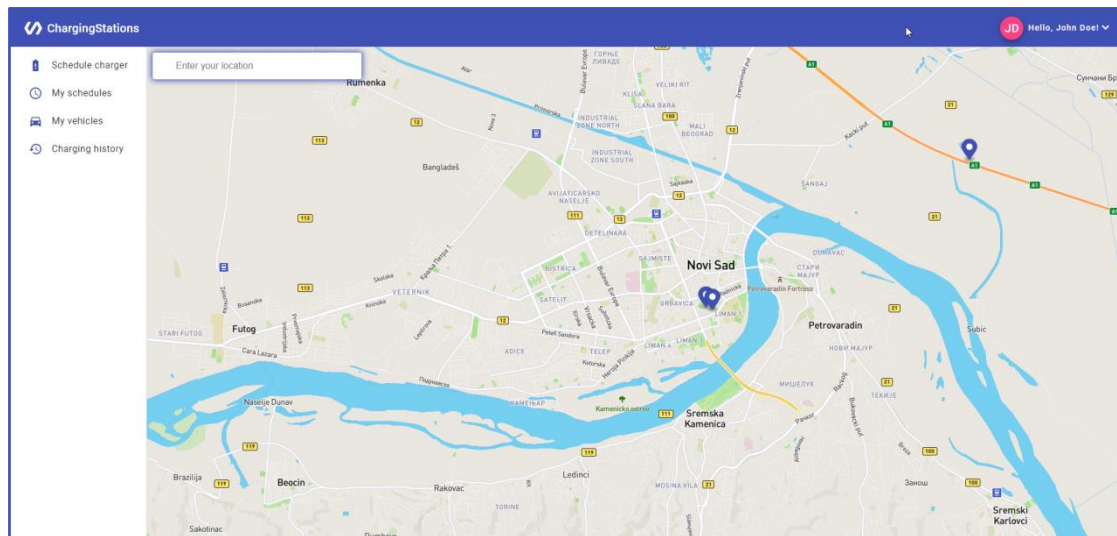


6.1.24. Prikaz opcije za odjavu sa sistema

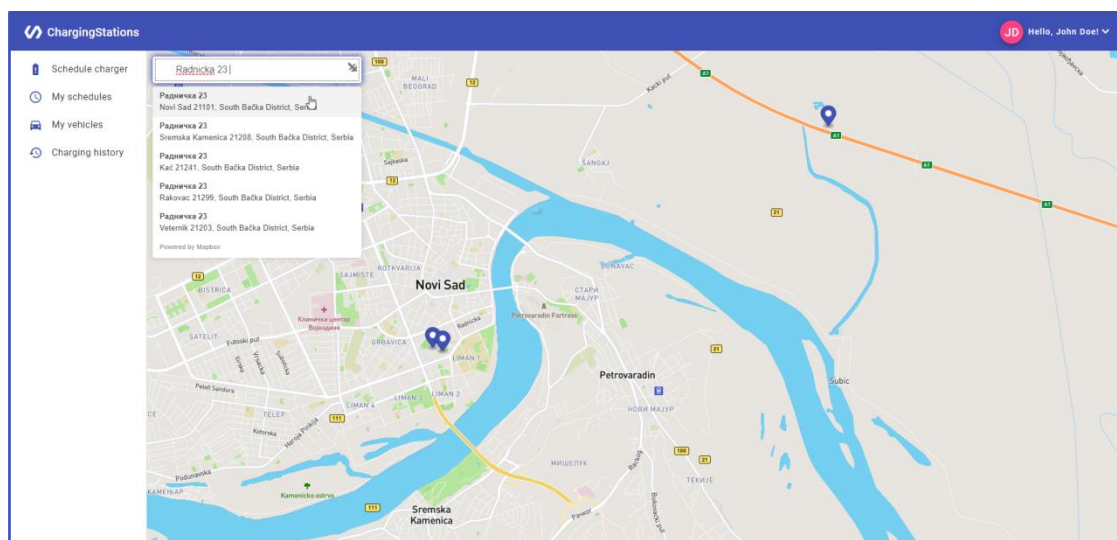
6.2 Uloga Vozača



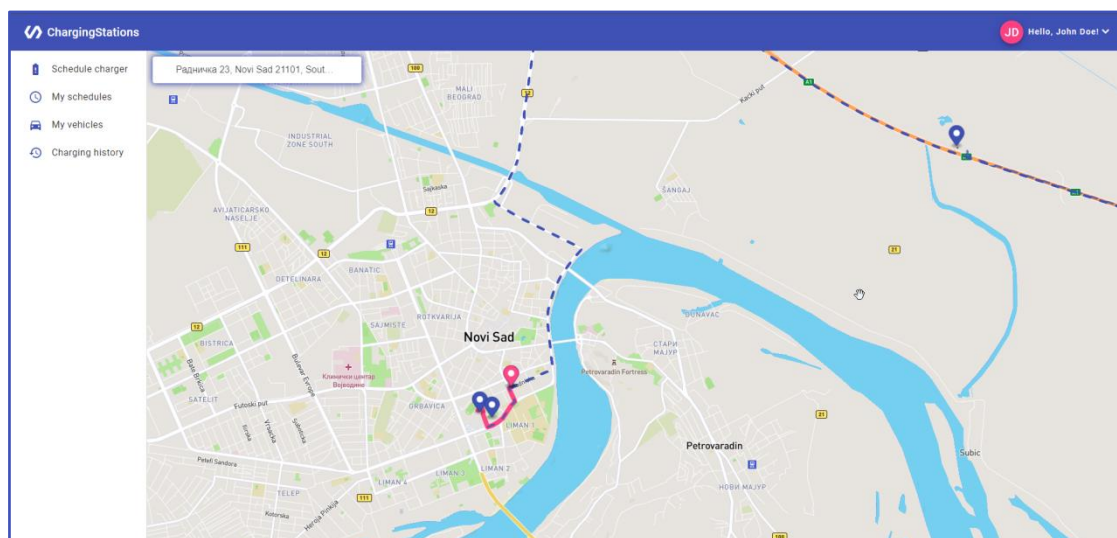
6.2.1. Prikaz mape za pretragu punjača



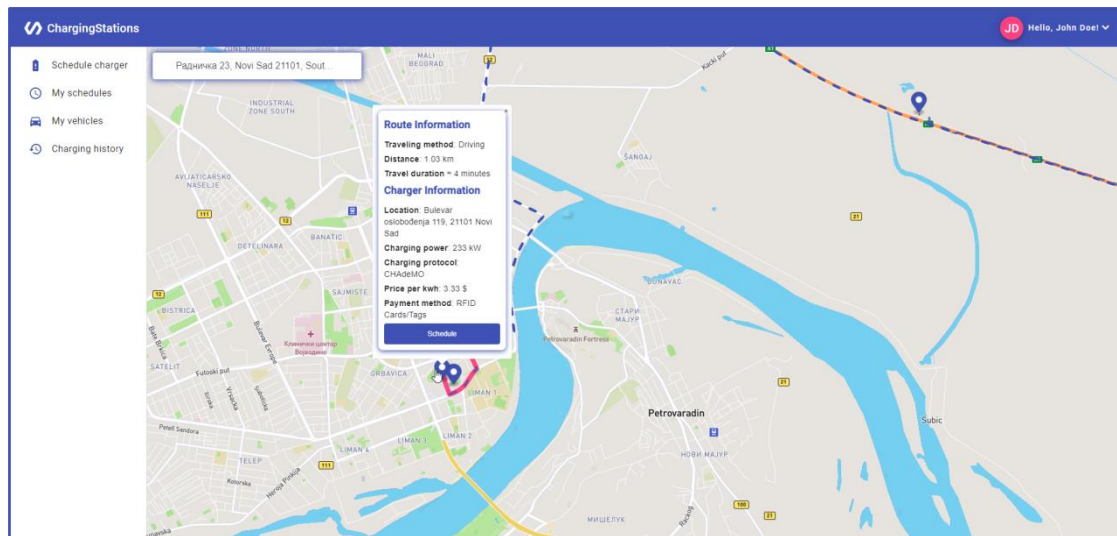
6.2.2. Prikaz mogućnosti otkrivanja opcija sa strane



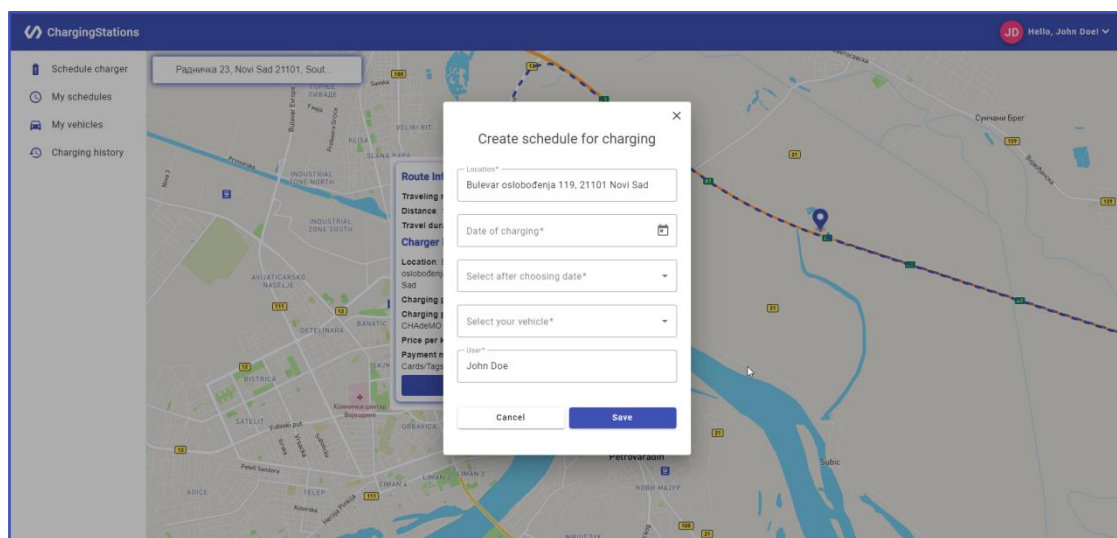
6.2.3. Prikaz ponuđenih opcija na osnovu unošenja adrese



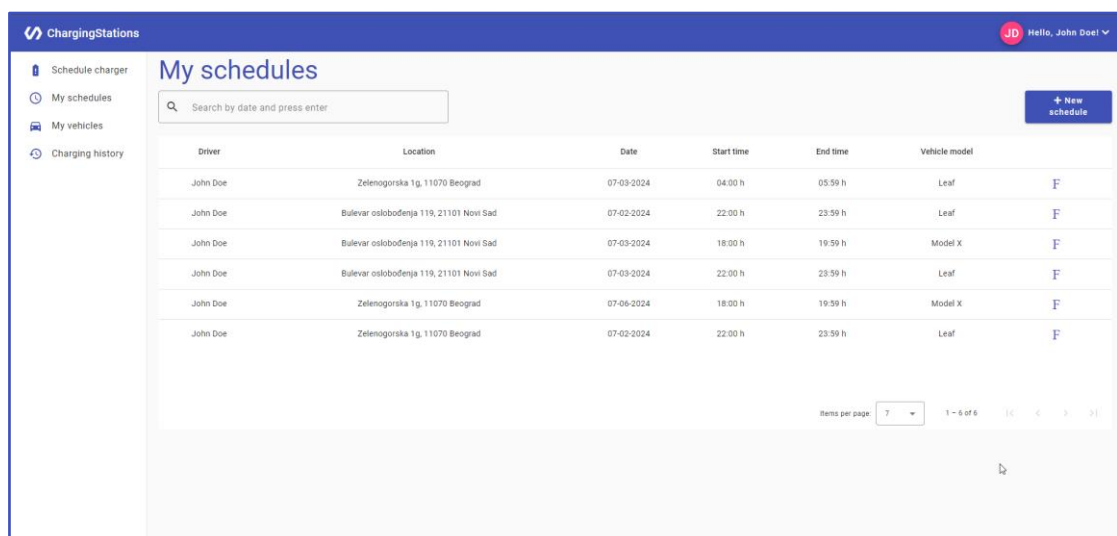
6.2.4. Prikaz rezultata izbora adrese



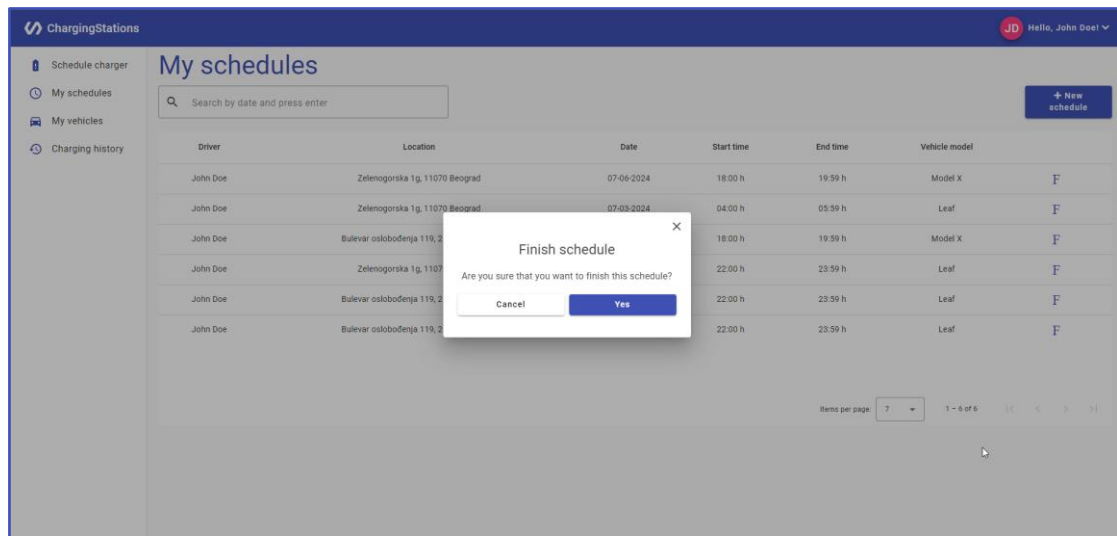
6.2.5. Prikaz modala sa osnovnim informacijama o punjaču



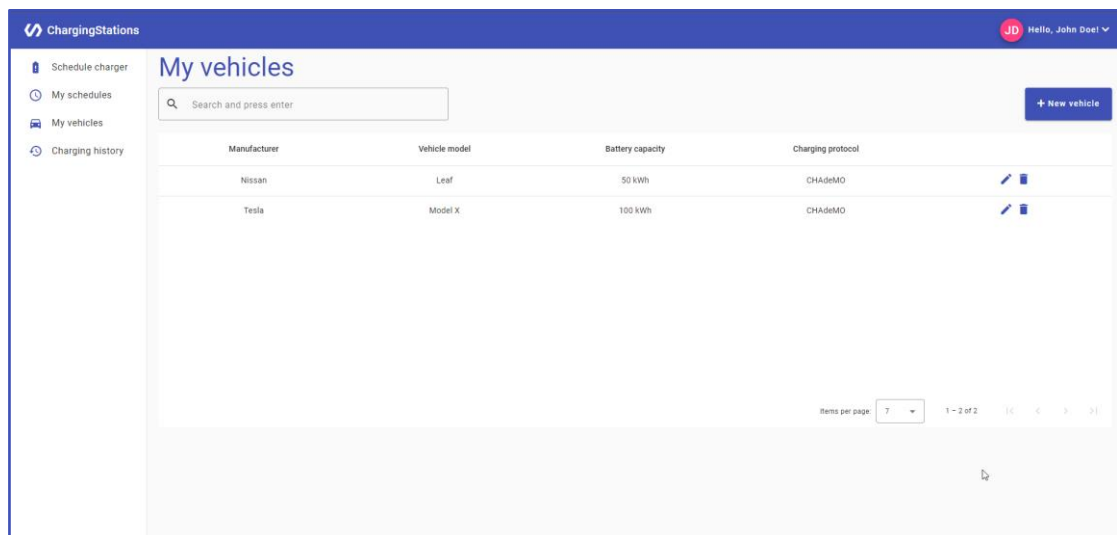
6.2.6. Prikaz modala za kreiranje rezervacije punjača



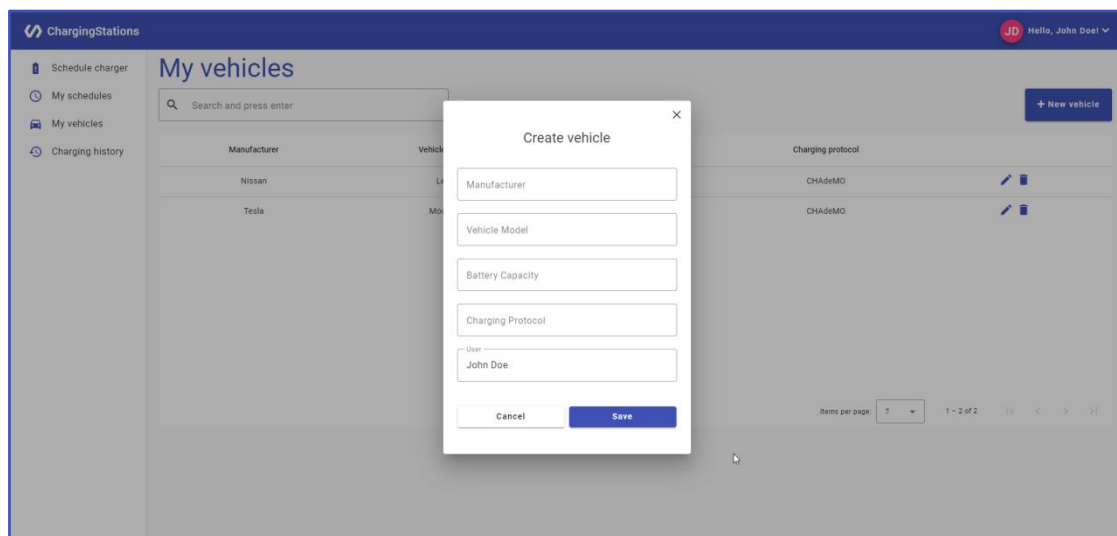
6.2.7. Prikaz stranice vozačevih termina za punjenje



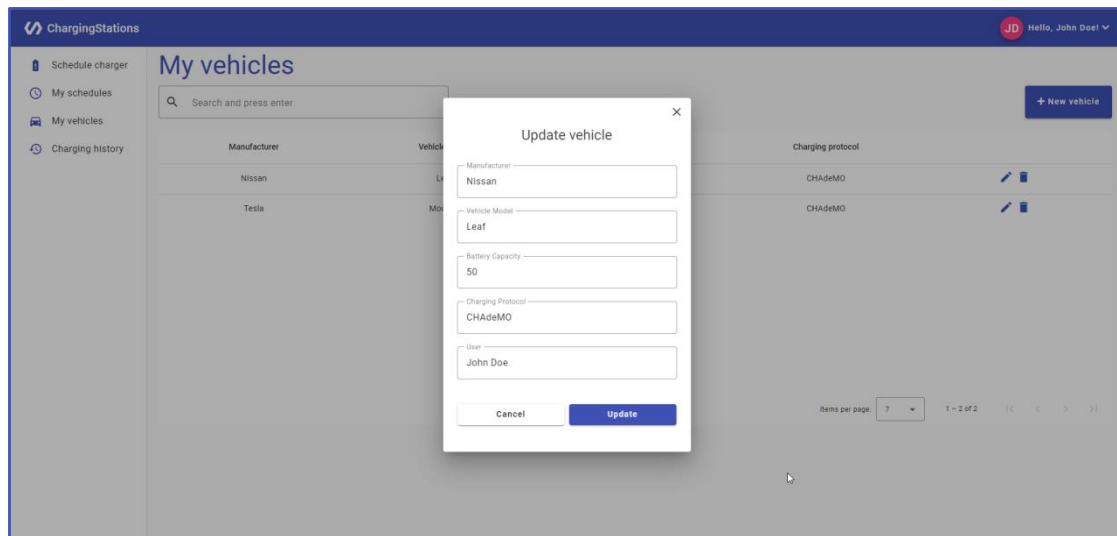
6.2.8. Prikaz modala za potvrdu završetka termina punjenja



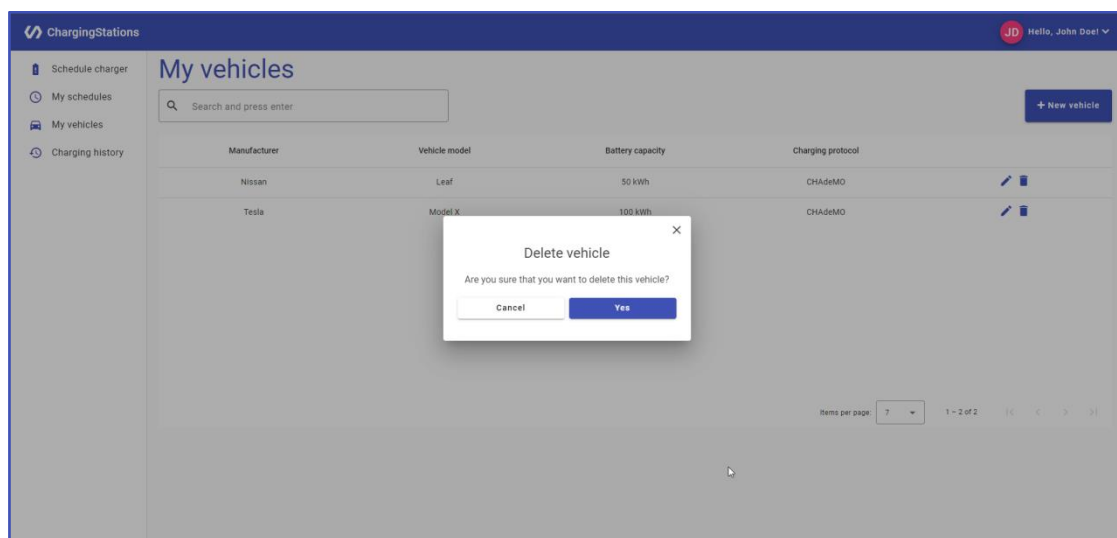
6.2.9. Prikaz modala za potvrdu nepotvrđenog korisnika



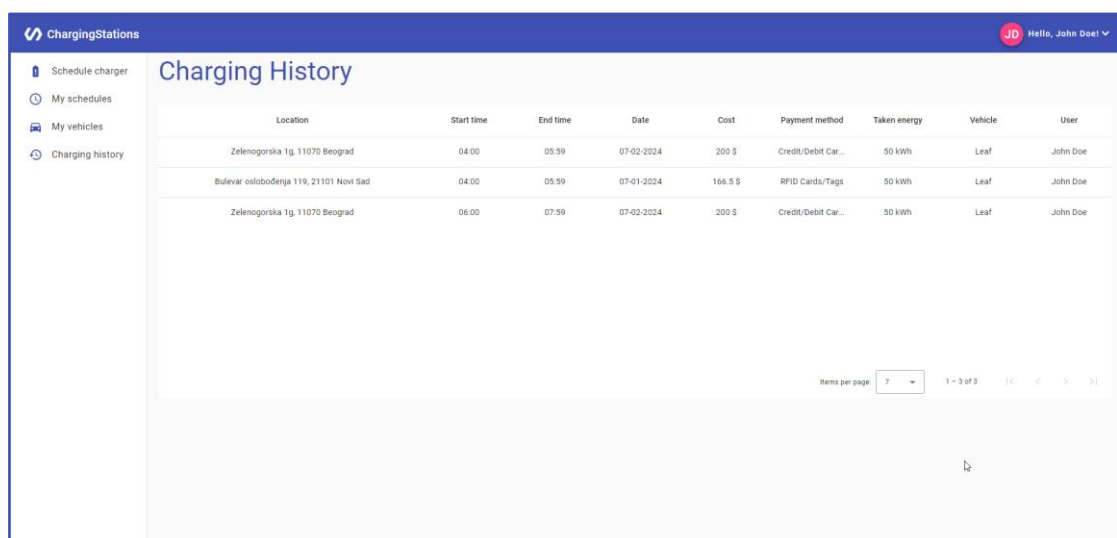
6.2.10. Prikaz modala za kreiranje vozila



6.2.11. Prikaz modala za ažuriranje vozila



6.2.12. Prikaz modala za brisanje vozila



6.2.13. Prikaz stranice istorije punjenja

7. Funkcionalnosti

7.1 Funkcionalnosti administratora

Ispod su prikazane, po stavkama, mogućnosti korisnika koji je u ulozi administratora.

Potvrđeni korisnici:

- 1) Pregled svih potvrđenih korisnika.
- 2) Dodavanje novog korisnika (mora da potvrdi na stranici za nepotvrđene korisnike).
- 3) Ažuriranje korisnika.
- 4) Brisanje korisnika.
- 5) Blokiranje korisnika.
- 6) Odblokiranje korisnika.
- 7) Pretraga svih korisnika.

Punjači:

- 1) Pregled svih punjača.
- 2) Pregled slobodnih termina za svaki punjač pojedinačno.
- 3) Dodavanje novog punjača.
- 4) Ažuriranje punjača.
- 5) Brisanje punjača.
- 6) Pretraga svih punjača.

Vozila:

- 1) Pregled svih vozila.
- 2) Dodavanje novog vozila.
- 3) Ažuriranje vozila.
- 4) Brisanje vozila.
- 5) Pretraga svih vozila.

Nepotvrđeni korisnici (oni koji čekaju na potvrdu registracije)

- 1) Pregled svih nepotvrđenih korisnika.
- 2) Detalji nepotvrđenog korisnika.
- 3) Potvrda nepotvrđenog korisnika.
- 4) Odbijanje nepotvrđenog korisnika.

U aplikaciji postoji samo jedan administrator i on je unapred registrovan i postavljen, bez potrebe za dodatnom registracijom.

7.2 Funkcionalnosti vozača

Ispod su prikazane, po stavkama, mogućnosti korisnika koji je u ulozi vozača.

Rezervisanje termina punjača:

- 1) Pregled svih lokacija punjača na mapi.
- 2) Unos trenutne lokacije od vozača.
- 3) Prikaz putanja od vozača do svih punjača.
- 4) Detalji putanje na klik lokacije punjača.
- 5) Kreiranje rezervacije.

Rezervisani termini:

- 1) Pregled svih rezervisanih termina.
- 2) Potvrda o završetku rezervisanog termina.
- 3) Pretraga svih rezervisanih termina.

Vozila:

- 1) Pregled svih korisnikovih vozila.
- 2) Dodavanje novog korisnikovog vozila.
- 3) Ažuriranje korisnikovog vozila.
- 4) Brisanje korisnikovog vozila.
- 5) Pretraga svih korisnikovih vozila.

Istorija punjenja:

- 1) Pregled celokupne istorije punjenja.

Ukoliko vozač nema nalog u aplikaciji, mora se registrovati. Takođe, mora sačekati da admin prihvati registraciju kako bi mogao pristupiti aplikaciji.

7.3 Algoritam za nalaženje najpogodnijeg punjača

Aplikacija treba omogućiti putanje za pristup od **tačke A**(vozač) do ostalih **tački B1, B2, B3, ...** (punjači), pri čemu će se na svakoj putanji imati informacije o:

- 1) Distanca do punjača.
- 2) Vremenu trajanja vožnje do punjača.
- 3) Naziv lokacije punjača.
- 4) Snaga punjača.
- 5) Protokol punjača.
- 6) Cena po kWh.
- 7) Metoda plaćanja koja je podržana.

Pored toga, prilikom rezervisanja punjača, aplikacija treba omogućiti vozaču uvid u najbliži punjač na osnovu parametara koji se posmatraju.

Parametri koji se uzimaju u obzir tokom izbora punjača:

- 1) Snaga punjača
- 2) Cena po kWh
- 3) Dostupnost punjača
- 4) Udaljenost punjača

7.4 Slučajevi korišćenja

[US-1] Ja kao **korisnik** želim da registrujem svoj nalog.

Preduslov: Korisnik nema registrovan nalog na platformi.

Opis koraka:

1. Korisnik otvara stranicu za registraciju.
2. Korisnik popunjava polja **ime, prezime, datum rođenja, korisničko ime, elektronska adresa, šifra, pol, biografija**.
3. Korisnik pritiska dugme "Registruj se".

Posledica: Korisnik je registrovao svoj nalog na platformi i čeka potvrdu naloga.

[US-2] Ja kao **korisnik** želim da se prijavim na platformu.

Preduslov: Korisnik mora imati registrovan nalog.

Opis koraka:

1. Korisnik otvara stranicu za prijavu.
2. Korisnik popunjava polja **korisničko ime i lozinka**.
3. Korisnik pritiska dugme "Prijavi se".

Posledica: Korisnik je prijavljen na platformu.

[US-3] Ja kao **administrator** želim da pregledam sve potvrđene korisnike.

Preduslov: Korisnik mora biti prijavljen u ulozi administratora.

Opis koraka:

1. Korisnik otvara stranicu za prikaz svih potvrđenih korisnika.

Posledica: Korisniku se prikazuju svi potvrđeni korisnici.

[US-4] Ja kao **administrator** želim da dodam novom korisnika.

Preduslov: Korisnik mora biti prijavljen u ulozi administratora.

Opis koraka:

1. Korisnik otvara stranicu za prikaz svih potvrđenih korisnika.
2. Pritiska na dugme "Dodaj novog korisnika".
3. Otvara se forma za dodavanje novog korisnika.
4. Na formi se popunjavaju potrebne informacije za korisnika.
5. Korisnik pritiska dugme "Potvrdi".
6. Korisnik prelazi na stranicu "Nepotvrđeni korisnici".
7. Korisnik potvrđuje korisnika kojeg je kreirao.

Posledica: Korisnik je dodao novog korisnika.

[US-5] Ja kao **administrator** želim da ažuriram korisnika.

Preduslov: Korisnik mora biti prijavljen u ulozi administratora.

Opis koraka:

1. Korisnik otvara stranicu za prikaz svih potvrđenih korisnika.
2. Pritiska na ikonicu "Ažuriraj korisnika".
3. Otvara se forma za ažuriranje korisnika.
4. Na formi se ažuriraju potrebne informacije za korisnika.
5. Korisnik pritiska dugme "Potvrdi".

Posledica: Korisnik je ažurirao korisnika.

[US-6] Ja kao **administrator** želim da obrišem korisnika.

Preduslov: Korisnik mora biti prijavljen u ulozi administratora.

Opis koraka:

1. Korisnik otvara stranicu za prikaz svih potvrđenih korisnika.
2. Pritiska na ikonicu "Obriši korisnika".
3. Otvara se forma za potvrdu o brisanju korisnika.
4. Korisnik pritiska dugme "Potvrdi".

Posledica: Korisnik je obrisao korisnika.

[US-7] Ja kao **administrator** želim da blokiram korisnika.

Preduslov: Korisnik mora biti prijavljen u ulozi administratora.

Opis koraka:

1. Korisnik otvara stranicu za prikaz svih potvrđenih korisnika.
2. Pritiska na ikonicu "Blokiraj korisnika".
3. Otvara se forma za potvrdu o blokiranju korisnika.
4. Korisnik pritiska dugme "Potvrdi".

Posledica: Korisnik je blokirao korisnika.

[US-8] Ja kao **administrator** želim da odblokiram korisnika.

Preduslov: Korisnik mora biti prijavljen u ulozi administratora.

Opis koraka:

1. Korisnik otvara stranicu za prikaz svih potvrđenih korisnika.
2. Pritiska na ikonicu "Odblokiraj korisnika".
3. Otvara se forma za potvrdu o odblokiranju korisnika.
4. Korisnik pritiska dugme "Potvrdi".

Posledica: Korisnik je odblokirao korisnika.

[US-9] Ja kao **administrator** želim da pretražim korisnike.

Preduslov: Korisnik mora biti prijavljen u ulozi administratora.

Opis koraka:

1. Korisnik otvara stranicu za prikaz svih potvrđenih korisnika.
2. Korisnik unosi kriterijum pretrage u polje.
3. Korisnik pritiska dugme enter.

Posledica: Korisnik dobija rezultat pretrage.

[US-10] Ja kao **administrator** želim da pregledam sve punjače.

Preduslov: Korisnik mora biti prijavljen u ulozi administratora.

Opis koraka:

1. Korisnik otvara stranicu za prikaz svih punjača.

Posledica: Korisniku se prikazuju svi punjači.

[US-11] Ja kao **administrator** želim da dodam novi punjač.

Preduslov: Korisnik mora biti prijavljen u ulozi administratora.

Opis koraka:

1. Korisnik otvara stranicu za prikaz svih punjača.
2. Pritiska na dugme "Dodaj novi punjač".
3. Otvara se forma za dodavanje novog punjača.
4. Na formi se popunjavaju potrebne informacije za punjač.
5. Korisnik pritiska dugme "Potvrdi".

Posledica: Korisniku je dodao novi punjač.

[US-12] Ja kao **administrator** želim da ažuriram punjač.

Preduslov: Korisnik mora biti prijavljen u ulozi administratora.

Opis koraka:

1. Korisnik otvara stranicu za prikaz svih punjača.
2. Pritiska na ikonicu "Ažuriraj punjač".
3. Otvara se forma za ažuriranje punjača.
4. Na formi se ažuriraju potrebne informacije za punjač.
5. Korisnik pritiska dugme "Potvrdi".

Posledica: Korisnik je ažurirao punjač.

[US-13] Ja kao **administrator** želim da obrišem punjač.

Preduslov: Korisnik mora biti prijavljen u ulozi administratora.

Opis koraka:

1. Korisnik otvara stranicu za prikaz svih punjača.
2. Pritiska na ikonicu "Obriši punjač".
3. Otvara se forma za potvrdu o brisanju punjača.
4. Korisnik pritiska dugme "Potvrdi".

Posledica: Korisnik je obrisao punjač.

[US-14] Ja kao **administrator** želim da pretražim punjače.

Preduslov: Korisnik mora biti prijavljen u ulozi administratora.

Opis koraka:

1. Korisnik otvara stranicu za prikaz svih punjača.
2. Korisnik unosi kriterijum pretrage u polje.
3. Korisnik pritiska dugme enter.

Posledica: Korisnik dobija rezultat pretrage.

[US-15] Ja kao **administrator** želim da vidim termine punjača.

Preduslov: Korisnik mora biti prijavljen u ulozi administratora.

Opis koraka:

1. Korisnik otvara stranicu za prikaz svih punjača.
2. Pritiska na ikonicu "Termini".
3. Otvara se nova stranica sa pregledom svih termina za selektovani punjač.

Posledica: Korisnik vidi sve termine za punjač.

[US-16] Ja kao **administrator** želim da pregledam sva vozila.

Preduslov: Korisnik mora biti prijavljen u ulozi administratora.

Opis koraka:

1. Korisnik otvara stranicu za prikaz svih vozila.

Posledica: Korisniku se prikazuju sva vozila.

[US-17] Ja kao **administrator** želim da dodam novo vozilo.

Preduslov: Korisnik mora biti prijavljen u ulozi administratora.

Opis koraka:

1. Korisnik otvara stranicu za prikaz svih vozila.
2. Pritiska na dugme "Dodaj novo vozilo".
3. Otvara se forma za dodavanje novog vozila.
4. Na formi se popunjavaju potrebne informacije za vozilo.
5. Korisnik pritiska dugme "Potvrdi".

Posledica: Korisniku je dodao novo vozilo.

[US-18] Ja kao **administrator** želim da ažuriram vozilo.

Preduslov: Korisnik mora biti prijavljen u ulozi administratora.

Opis koraka:

1. Korisnik otvara stranicu za prikaz svih vozila.
2. Pritiska na ikonicu "Ažuriraj vozilo".
3. Otvara se forma za ažuriranje vozila.
4. Na formi se ažuriraju potrebne informacije za vozilo.

5. Korisnik pritiska dugme "Potvrdi".

Posledica: Korisnik je ažurirao vozilo.

[US-19] Ja kao **administrator** želim da obrišem vozilo.

Preduslov: Korisnik mora biti prijavljen u ulozi administratora.

Opis koraka:

1. Korisnik otvara stranicu za prikaz svih vozila.
2. Pritiska na ikonicu "Obriši vozilo".
3. Otvara se forma za potvrdu o brisanju vozila.
4. Korisnik pritiska dugme "Potvrdi".

Posledica: Korisnik je obrisao vozilo.

[US-20] Ja kao **administrator** želim da pretražim vozila.

Preduslov: Korisnik mora biti prijavljen u ulozi administratora.

Opis koraka:

1. Korisnik otvara stranicu za prikaz svih vozila.
2. Korisnik unosi kriterijum pretrage u polje.
3. Korisnik pritiska dugme enter.

Posledica: Korisnik dobija rezultat pretrage.

[US-21] Ja kao **administrator** želim da pregledam sve nepotvrđene korisnike.

Preduslov: Korisnik mora biti prijavljen u ulozi administratora.

Opis koraka:

1. Korisnik otvara stranicu za prikaz svih nepotvrđenih korisnika.

Posledica: Korisniku se prikazuju svi nepotvrđeni korisnici.

[US-22] Ja kao **administrator** želim da pregledam detalje nepotvrđenog korisnika.

Preduslov: Korisnik mora biti prijavljen u ulozi administratora.

Opis koraka:

1. Korisnik otvara stranicu za prikaz svih nepotvrđenih korisnika.
2. Pritiska na ikonicu "Detalji korisnika".

3. Otvara se forma sa detaljima o korisnika.

Posledica: Korisnik vidi detalje nepotvrđenog korisnika.

[US-23] Ja kao **administrator** želim da potvrdim nepotvrđenog korisnika.

Preduslov: Korisnik mora biti prijavljen u ulozi administratora.

Opis koraka:

1. Korisnik otvara stranicu za prikaz svih nepotvrđenih korisnika.
2. Pritiska na ikonicu "Potvrdi korisnika".
3. Otvara se forma za potvrdu o potvrdi korisnika.
4. Korisnik pritiska dugme "Potvrdi".

Posledica: Korisnik je potvrdio korisnika.

[US-24] Ja kao **administrator** želim da odbijem nepotvrđenog korisnika.

Preduslov: Korisnik mora biti prijavljen u ulozi administratora.

Opis koraka:

1. Korisnik otvara stranicu za prikaz svih nepotvrđenih korisnika.
2. Pritiska na ikonicu "Odbij korisnika".
3. Otvara se forma za potvrdu o odbijanju korisnika.
4. Korisnik pritiska dugme "Potvrdi".

Posledica: Korisnik je odbio korisnika.

[US-25] Ja kao **vozač** želim da pregledam sva svoja vozila.

Preduslov: Korisnik mora biti prijavljen u ulozi vozača.

Opis koraka:

1. Korisnik otvara stranicu za prikaz svih svojih vozila.

Posledica: Korisniku se prikazuju sva njegova vozila.

[US-26] Ja kao **vozač** želim da dodam svoje novo vozilo.

Preduslov: Korisnik mora biti prijavljen u ulozi vozača.

Opis koraka:

1. Korisnik otvara stranicu za prikaz svih svojih vozila.

2. Pritiska na dugme "Dodaj novo vozilo".
3. Otvara se forma za dodavanje novog vozila.
4. Na formi se popunjavaju potrebne informacije za vozilo.
5. Korisnik pritiska dugme "Potvrdi".

Posledica: Korisniku je dodao novo vozilo.

[US-27] Ja kao **vozač** želim da ažuriram svoje vozilo.

Preduslov: Korisnik mora biti prijavljen u ulozi administratora.

Opis koraka:

1. Korisnik otvara stranicu za prikaz svih svojih vozila.
2. Pritiska na ikonicu "Ažuriraj vozilo".
3. Otvara se forma za ažuriranje vozila.
4. Na formi se ažuriraju potrebne informacije za vozilo.
5. Korisnik pritiska dugme "Potvrdi".

Posledica: Korisnik je ažurirao vozilo.

[US-28] Ja kao **vozač** želim da obrišem svoje vozilo.

Preduslov: Korisnik mora biti prijavljen u ulozi vozača.

Opis koraka:

1. Korisnik otvara stranicu za prikaz svih svojih vozila.
2. Pritiska na ikonicu "Obriši vozilo".
3. Otvara se forma za potvrdu o brisanju vozila.
4. Korisnik pritiska dugme "Potvrdi".

Posledica: Korisnik je obrisao vozilo.

[US-29] Ja kao **vozač** želim da pretražim svoja vozila.

Preduslov: Korisnik mora biti prijavljen u ulozi vozača.

Opis koraka:

1. Korisnik otvara stranicu za prikaz svih svojih vozila.
2. Korisnik unosi kriterijum pretrage u polje.
3. Korisnik pritiska dugme enter.

Posledica: Korisnik dobija rezultat pretrage.

[US-30] Ja kao **vozač** želim da pregledam sve dostupne punjače na mapi.

Preduslov: Korisnik mora biti prijavljen u ulozi vozača.

Opis koraka:

1. Korisnik otvara stranicu za prikaz mape.

Posledica: Korisnik dobija sve punjače prikazane na mapi.

[US-31] Ja kao **vozač** želim da unesem svoju trenutnu lokaciju.

Preduslov: Korisnik mora biti prijavljen u ulozi vozača.

Opis koraka:

1. Korisnik otvara stranicu za prikaz mape.
2. Korisnik unosi svoju lokaciju u polje "Unesi svoju lokaciju".
3. Korisnik pritiska dugme enter.

Posledica: Korisnik dobija putanje od svoje lokacije do svih punjača.

[US-32] Ja kao **vozač** želim da vidim informacije o putanji do punjača.

Preduslov: Korisnik mora biti prijavljen u ulozi vozača i morao je uneti svoju trenutnu lokaciju.

Opis koraka:

1. Korisnik pritiskom na marker od lokacije punjača dobija informacije o putanji.

Posledica: Korisnik dobija informacije o putanji do željenog punjača.

[US-33] Ja kao **vozač** želim da rezervišem punjač.

Preduslov: Korisnik mora biti prijavljen u ulozi vozača i morao je otići na informacije o punjaču.

Opis koraka:

1. Korisnik pritiska dugme "Rezerviši".

Posledica: Korisnik je rezervisao punjač.

[US-34] Ja kao **vozač** želim da potvrdim završetak termina.

Preduslov: Korisnik mora biti prijavljen u ulozi vozača.

Opis koraka:

1. Korisnik otvara stranicu za prikaz svih njegovih termina.
2. Pritiska na ikonicu "Završi".

Posledica: Korisnik je završio termin punjenja.

[US-35] Ja kao **vozač** želim da pretražim svoje rezervisane termine.

Preduslov: Korisnik mora biti prijavljen u ulozi vozača.

Opis koraka:

1. Korisnik otvara stranicu za prikaz svih svojih rezervisanih termina.
2. Korisnik unosi kriterijum pretrage u polje.
3. Korisnik pritiska dugme enter.

Posledica: Korisnik dobija rezultat pretrage.

[US-36] Ja kao **vozač** želim da pregledam celokupnu istoriju punjenja.

Preduslov: Korisnik mora biti prijavljen u ulozi vozača.

Opis koraka:

1. Korisnik otvara stranicu za prikaz istorije punjenja.

Posledica: Korisniku se prikazuje celokupna istorija punjenja.

7.5 Podržani HTTP zahtevi

HTTP zahtevi koje podržava aplikacija su prikazani u tabeli ispod.

HTTP method	URL	Parameters	Role
GET	accounts/getcurrentuser/id	id: string	Admin/Driver
GET	accounts/checkcredentials	credentials: string	Admin/Driver
POST	appointments/save	dto: model	Admin
POST	appointments/allow	dto: model	Admin
POST	appointments/unallow	dto: model	Admin
GET	appointments/getbyid/id	id: string	Admin
GET	appointments/getall/chargerId	chargerId: string	Admin/Driver
GET	appointments/all/chargerId/date	id, date: string	Admin/Driver
DELETE	appointments/delete/id	id: string	Admin
POST	auth/register	dto: model	Driver
POST	auth/login	dto: model	Admin/Driver
POST	auth/logout/id	id: string	Admin/Driver
POST	auth/refresh/id	id: string	Admin/Driver
POST	chargers/save	dto: model	Admin
GET	chargers/getbyid/id	id: string	Admin
GET	chargers/getall	/	Admin
GET	chargers/search/query	query: string	Admin
DELETE	chargers/delete/id	id: string	Admin
POST	charging-histories/save	dto: model	Driver
GET	charging-histories/getbyid/id	id: string	Driver
GET	charging-histories/getall/driverId	driverId: string	Driver
POST	schedule-charger/save	dto: model	Driver
GET	schedule-charger/getall/driverId	driverId: string	Driver
POST	schedule-charger/search/query	query: string	Driver
GET	schedule-charger/finish/id	id: string	Driver
PUT	users/update	dto: model	Admin
GET	users/getbyid/id	id: string	Admin
GET	users/getall	/	Admin
GET	users/all	/	Admin
GET	users/getallconfirmed	/	Admin
GET	users/searchConfirmed/query	query: string	Admin
GET	users/getallunconfirmed	/	Admin
GET	users/searchUnconfirmed/query	query: string	Admin
GET	users/getallblocked	/	Admin
GET	users/getallunblocked	/	Admin
GET	users/block/id	id: string	Admin
GET	users/unblock/id	id: string	Admin
GET	users/confirm/id	id: string	Admin
DELETE	users/delete/id	id: string	Admin

POST	vehicles/save		Admin/Driver
GET	vehicles/getbyid/id	id: string	Admin/Driver
GET	vehicles/getall/userId	userId: string	Driver
GET	vehicles/getall	/	Admin
GET	vehicles/search/query	query: string	Admin
POST	vehicles/search-by-driver/query/id	query, id: string	Driver
DELETE	vehicles/delete/id	id: string	Admin/Driver

7.5.1. Prikaz svih HTTP zahteva koje podržava aplikacija