

**Univerzitet u Beogradu
Fakultet organizacionih nauka**

**WEB APLIKACIJA ZA UPRAVLJANJE
PARKIRALIŠTIMA**

**Seminarski rad
iz predmeta Internet tehnologije**

Student:

Marko Milošević
1057/2017

Beograd, novembar 2021.

Sadržaj

1. Korisnički zahtev	1
2. Opis slučajeve korišćenja.....	2
3. Opis procesa – dijagrami sekvenci	5
4. Konceptualni model	7
5. Specifikacija REST api-ja	8
6. Opis tehnologija korišćenih u izradi aplikacije	14
6.1 React	14
6.2 Express	14
6.3 Aaxis	15
6.4 Mysql.....	15
6.5 TypeORM	15
7. Uputstvo za korisnika	16
8. Link ka udaljenom repozitorijumu.....	21

1. Korisnički zahtev

Potrebno je napraviti aplikaciju za automatizaciju procesa upravljanja parkirališta. Osnovna funkcionalnost aplikacije bi trebala biti praćenje ulaska i izlaska vozila sa parkirališta kao i kreiranje računa za boravak vozila. Pored toga, potrebno je omogućiti kreiranje i brisanje parkirališta i kreiranje vozila u sistemu kako bi mogli da se sakupe određeni statistički podaci o boravku na parkingu kao što su:

- Prosečno vreme zadržavanja vozila na parkingu
- Prosečan broj vozila na parkingu
- Prosečan broj dolazaka jednog vozila u određenom vremenskom periodu...

Ove statistike je moguće računati na celom skupu podataka, a mogu se i grupisati po raznim kriterijumima (npr. dan u nedelji).

Kao poslednja funkcionalnost javlja se upravljanje korisnicima. Podrazumevano je da postoje dve vrste korisnika: običan korisnik i administrator. Običnom korisniku je omogućeno izdavanje računa, beleženje vremena dolaska i odlaska vozila, odjava sa sistema i promena sopstvenih podataka. Ukoliko korisnik nije prijavljen na sistem ili želi da se registruje on to može učiniti preko odgovarajućih formi. Da bi se korisnik uspešno registrovao potrebno je da zahtev prethodno bude odobren od strane administratora. Administrator, takođe, može vršiti kreiranje, izmenu i brisanje parkirališta, brisanje korisnika, praćenje statistika kao i sve funkcionalnosti koje ima običan korisnik.

2. Opis slučajeva korišćenja

Prepoznali smo sledeće slučajeve korišćenja:

1. Prijava korisnika na sistem
2. Registrovanje korisnika
3. Odjava korisnika sa sistema
4. Odobravanje novog korisnika
5. Pregled parkirališta
6. Kreiranje parkirališta
7. Izmena parkirališta
8. Brisanje parkirališta
9. Pregled računa
10. Kreiranje računa
11. Pretraga boravaka
12. Kreiranje boravka
13. Beleženje napuštanja vozila sa parkirališta
14. Pregled popularnosti parkirališta
15. Pregled korisnika

U narednom tekstu biće opisana dva važnija slučaja korišćenja, a to su pretraga boravaka i kreiranje računa. Svi ostali slučajevi korišćenja mogu biti prepoznati odeljku [7. Uputstvo za korisnika](#).

SK11: Slučaj korišćenja – Pretraga boravka**Aktori SK**

Korisnik

Učesnici SK

Korisnik i sistem

Preduslov: Sistem je uključen i korisnik je ulogovan pod svojom šifrom. Sistem prikazuje korisniku formu za rad sa boravkom.

Osnovni scenario SK

1. Korisnik unosi vrednost po kojoj pretražuje boravak. (APUSO)
2. Korisnik poziva sistem da nađe boravak po zadatoj vrednosti. (APSO)
3. Sistem traži boravak po zadatoj vrednosti. (SO)
4. Sistem prikazuje korisniku boravke. (IA)

Alternativna scenarija

- 4.1 Ukoliko sistem ne može da nađe boravak on prikazuje korisniku objavu: „Sistem ne može da nađe boravak“. (IA)

SK10: Slučaj korišćenja – Kreiranje računa**Aktori SK**

Korisnik

Učesnici SK

Korisnik i sistem

Preduslov: Sistem je uključen i korisnik je ulogovan pod svojom šifrom. Sistem prikazuje korisniku formu za rad sa računom.

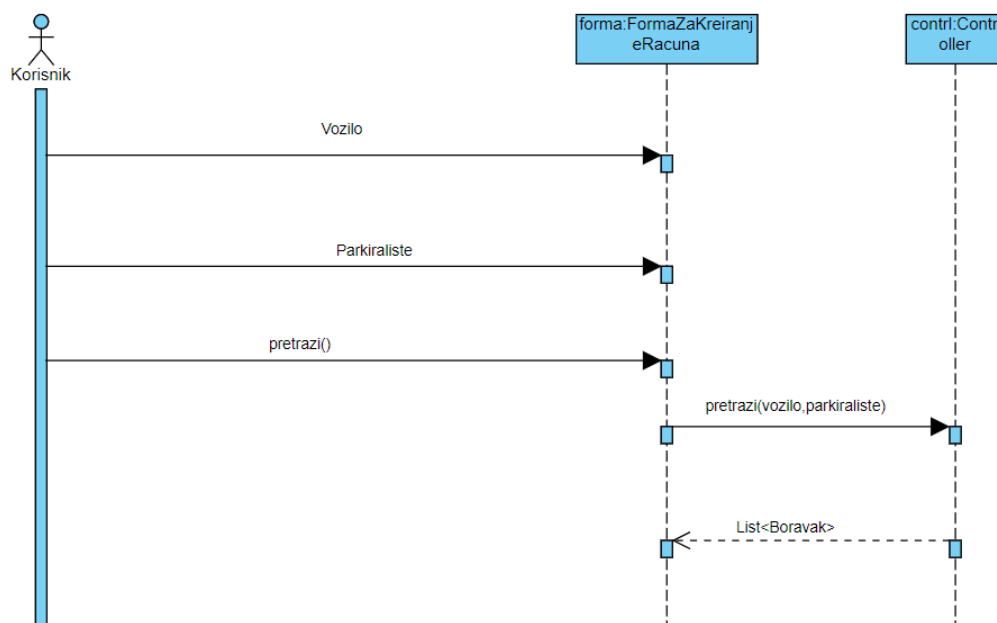
Osnovni scenario SK

1. Korisnik bira boravke koje želi da budu uključene u račun.(APUSO)
2. Korisnik poziva sistem da kreira račun.(APSO)
3. Sistem kreira račun.(SO)
4. Sistem prikazuje korisniku novi račun.(IA)

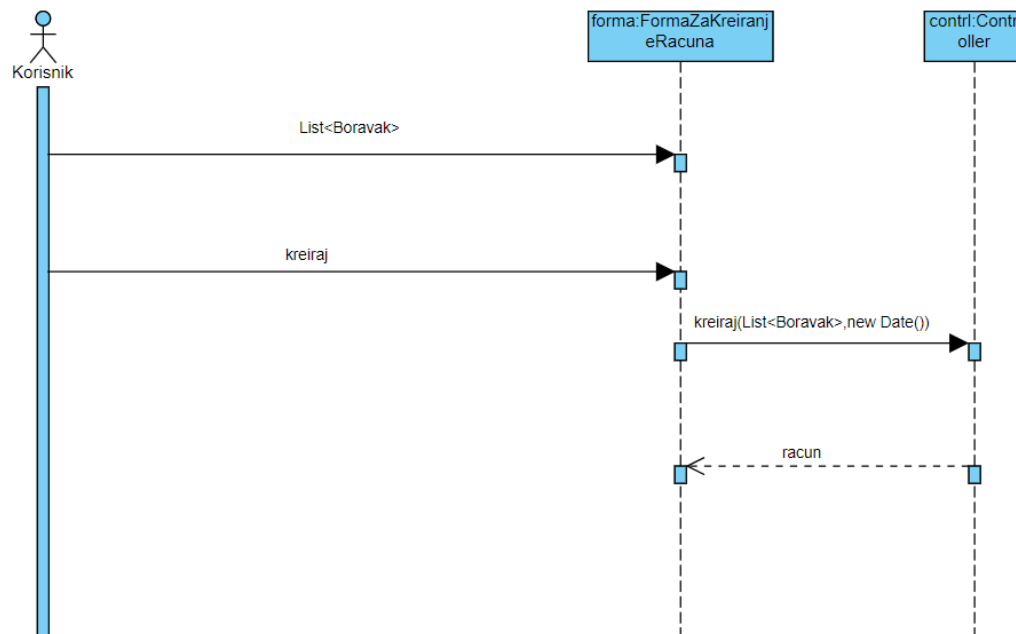
Alternativna scenarija

- 4.1 Ukoliko sistem ne može da kreira račun on prikazuje korisniku poruku: „Sistem ne može ka kreira račun“.

3. Opis procesa – dijagrami sekvenci

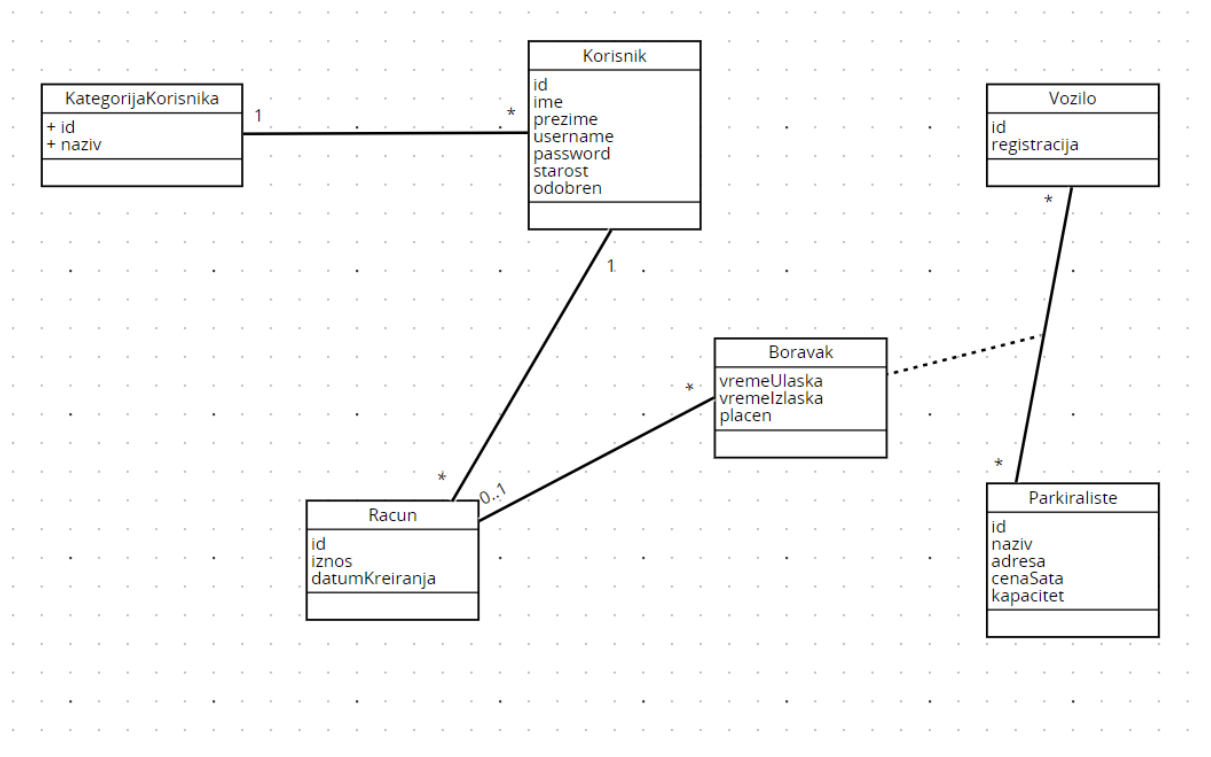


Slika 1: Pretraga boravaka



Slika 2: Kreiranje racuna

4. Konceptualni model



Slika 3: Konceptualni model

5. Specifikacija REST api-ja

Tabela 1: Vрати sve korisnike

Opis funkcije	Vрати sve korisnike
HTTP metoda	GET
URL	/user
URL parametri	(nema)
HTTP body parametri	(nema)
Format HTTP body parametara	(nema)
Izlazni parametri	<p>Niz JSON objekata. Svaki element niza ima attribute: category, id, firstName, lastName, age, username, odobren</p> <p>Primer:</p> <pre>[{ "category": "admin", "id": 1, "firstName": "Marko", "lastName": "Milosevic", "age": 34, "username": "mareko87", "odobren": true }]</pre>
Format izlaznih parametara	application/json

Tabela 2: Prijavi se na sistem

Opis funkcije	Prijavi se na sistem
HTTP metoda	POST
URL	/login
URL parametri	(nema)
HTTP body parametri	<pre>{ "username": "mareko87", "password": "mareko87" }</pre>
Format HTTP body parametara	(nema)
Izlazni parametri	<p>JSON objekat koji ima attribute: category, id, firstName, lastName, age, username, odobren</p> <p>Primer:</p> <pre>{ "category": "admin", "id": 1, "firstName": "Marko", "lastName": "Milosevic", "age": 33, "username": "mareko87", "odobren": true }</pre>
Format izlaznih parametara	application/json

Tabela 3: Vрати sva parkirališta

Opis funkcije	Vрати sva parkirališta
HTTP metoda	GET
URL	/parkiraliste
URL parametri	(nema)
HTTP body parametri	(nema)
Format HTTP body parametara	(nema)
Izlazni parametri	<p>Niz JSON objekata. Svaki objekat ima atribute: adresa, id, kapacitet, cenaSata</p> <p>Primer:</p> <pre>[{ "id":1, "naziv":"vracar", "adresa":"Borislava Pekica 3", "cenaSata":100, "kapacitet":500 }]</pre>
Format izlaznih parametara	application/json

Tabela 4: Kreiraj parkiralište

Opis funkcije	Kreiraj parkiralište
HTTP metoda	POST
URL	/parkiraliste
URL parametri	(nema)
HTTP body parametri	<pre>{ "naziv": "vracar", "adresa": "Borislava Pekica 3", "cenaSata": 100, "kapacitet": 500 }</pre>
Format HTTP body parametara	(nema)
Izlazni parametri	<p>JSON objekat koji predstavlja novo parkiralište.</p> <p>Primer:</p> <pre>[{ "id": 1, "naziv": "vracar", "adresa": "Borislava Pekica 3", "cenaSata": 100, "kapacitet": 500 }]</pre>
Format izlaznih parametara	application/json

Tabela 5: Izmeni parkiralište

Opis funkcije	Izmeni parkiralište
HTTP metoda	PATCH
URL	/parkiraliste/:id
URL parametri	Id parkiralista
HTTP body parametri	<pre>{ "naziv": "vracar", "adresa": "Borislava Pekica 3", "cenaSata": 100, }</pre>
Format HTTP body parametara	(nema)
Izlazni parametri	<p>JSON objekat koji predstavlja izmenjeno parkiralište.</p> <p>Primer:</p> <pre>[{ "id": 1, "naziv": "vracar", "adresa": "Borislava Pekica 3", "cenaSata": 100, "kapacitet": 500 }]</pre>
Format izlaznih parametara	application/json

Tabela 6: Obriši parkiralište

Opis funkcije	Obrisi parkiralište
HTTP metoda	DELETE
URL	/parkiraliste/:id
URL parametri	(nema)
HTTP body parametri	(nema)
Format HTTP body parametara	(nema)
Izlazni parametri	Putanja ne vraca nikakve podatke
Format izlaznih parametara	application/json

6. Opis tehnologija korišćenih u izradi aplikacije

Prilikom izrade aplikacije korišćene su sledeće tehnologije:

- React – kreiranje poslovne logike na strani klijenta
- Express – kreiranje servera
- Axios – biblioteka za pravljenje AJAX zahteva
- Typeorm – objektno – relacioni mapper
- Mysql – baza podataka

6.1 React

React predstavlja biblioteku za kreiranje korisničkog interfejsa i poslovne logike na klijentskoj strani. React služi za kreiranje SPA web aplikacija odnosno aplikacija koje imaju samo jednu html stranu i kojima glavnu ulogu ima javascript.

Glavni koncept u React-u je komponenta.

Komponenta predstavlja nešto što može da se iscrta kao jedna celina.

Celokupni prikaz aplikacije dobija se kompozicijom raznih elemenata u okviru jedne glavne komponente koja se, najčešće, naziva App.

Komponente mogu definisane kao funkcija i kao klasa.

Svaka komponenta ima svoje atribute koje je dobila od komponente roditelja i svoje stanje koje je ona sama kreirala. Komponenta može da menja svoje stanje ali ne može da menja svoje atribute. Prilikom promene stanja poziva se posebna funkcija koja to radi nakon čega se prikaz na ekranu automatski menja na svim mestima koja su se iscrtavala na osnovu tog stanja.

6.2 Express

Predstavlja „framework“ za izradu serverske strane u Node.js okruženju. Za razliku od većine ostalih servera, Express ne podržava višenitno programiranje. Problem konkurentnosti on rešava pomoću asinhronog programiranja i „event loop“ – a.

Express aplikacije su zamišljene da rade kao REST servisi, a obrada zahteva klijenta vrši se na sledeći način:

- Pristiže zahtev klijenta
- Izvršavaju se funkcije koje su definisane da se izvrše nad svakim zahtevom („middleware“)
- Zahtev se rutira na osnovu svog URL – a

- Izvršava se funkcija pridodata za dati URL
- Šalje se odgovor korisniku

6.3 Axios

Axios je biblioteka koja pojednostavljuje rad sa AJAX zahtevima. Axios omogućava kreiranje svih vrsta http zahteva kao i kreiranje zaglavlja. Kao odgovor sa servera, axios vraća korisniku originalni odgovor sa pridruženim metapodacima koje je axios izgenerisao.

6.4 Mysql

Relaciona baza podataka koja je besplatna za korišćenje.

6.5 TypeORM

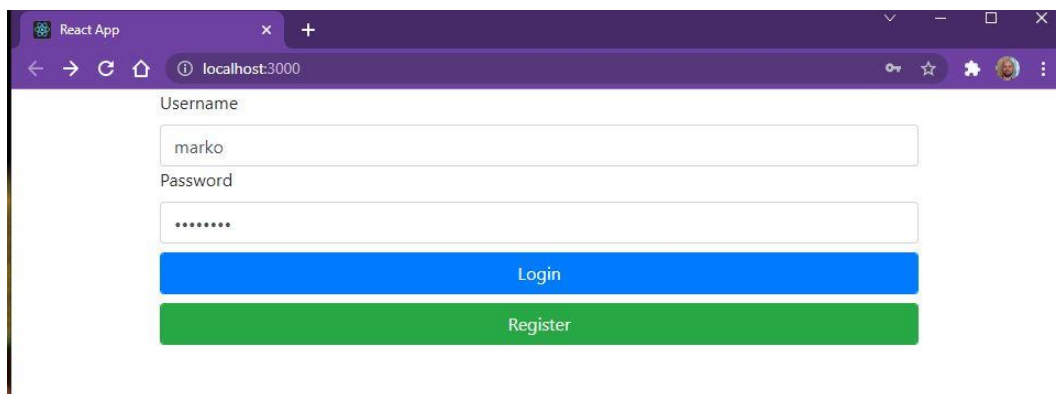
Objektno – relacion mapper u javascript – u. Pravila prevođenja vrše se anotacijom klasa u programskom kodu. Sva konfiguracija nalazi se fajlu ormconfig.json. TypeORM omogućava rad sa bazom na 2 načina:

- Preko repozitorijuma (Repository pattern)
- Preko menadžera

U prvom slučaju olakšan je rad sa bazom ali su mogućnost kontrole toka i prilagođavanje rada ograničeni dok je u drugom omogućena mnogo veća kontrola i kastomizacija na račun komplikovanijeg koda.

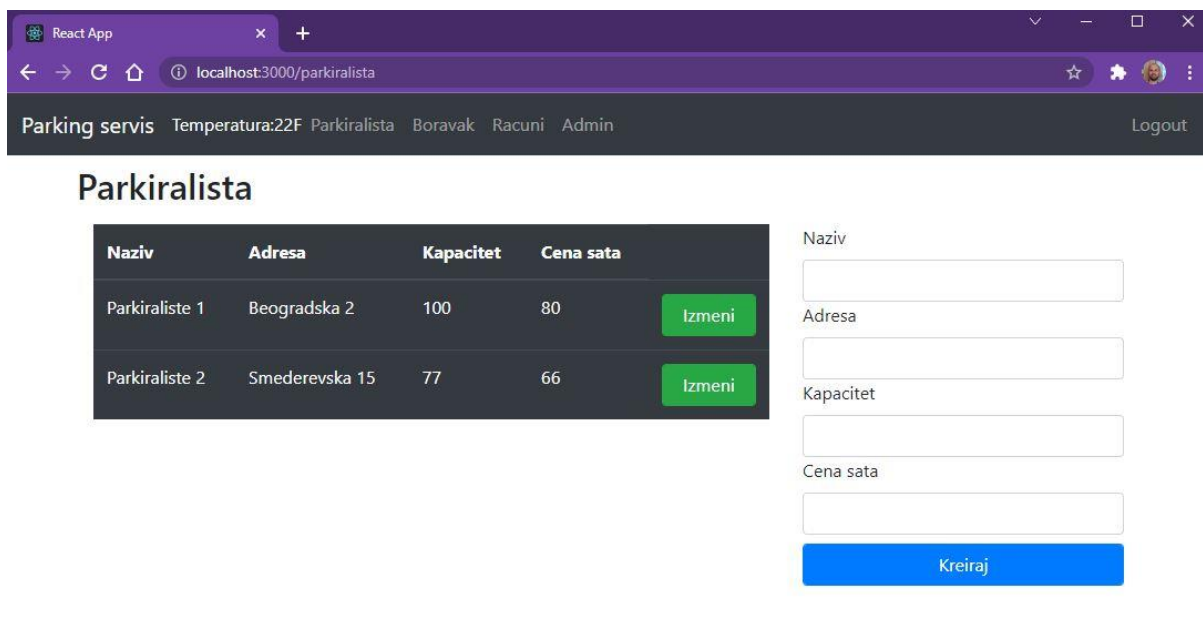
7. Uputstvo za korisnika

Pokretanjem aplikacije dolazi se na stranu za logovanje. Potrebno je uneti kredencijale i klikom na Login dugme vrsi se validacija korisnika (Slika 4).



Slika 4: Logovanje korisnika

Nakon uspešnog logovanja, vrsi se prikaz strane za rad za parkiralištima. Kako je ovo *single-page* aplikacija, nova strana koja se vidi zapravo i nije nova strana, već prikaz uparen sa url putanjom, ali za korisnika to i nije od važnosti. Na slici 5 vidimo formu za kreiranje novog parkirališta (desno) i podatke o postojećim parkiralištima (levo).



Naziv	Adresa	Kapacitet	Cena sata	
Parkiraliste 1	Beogradska 2	100	80	<button>Izmeni</button>
Parkiraliste 2	Smederevska 15	77	66	<button>Izmeni</button>

Naziv

Adresa

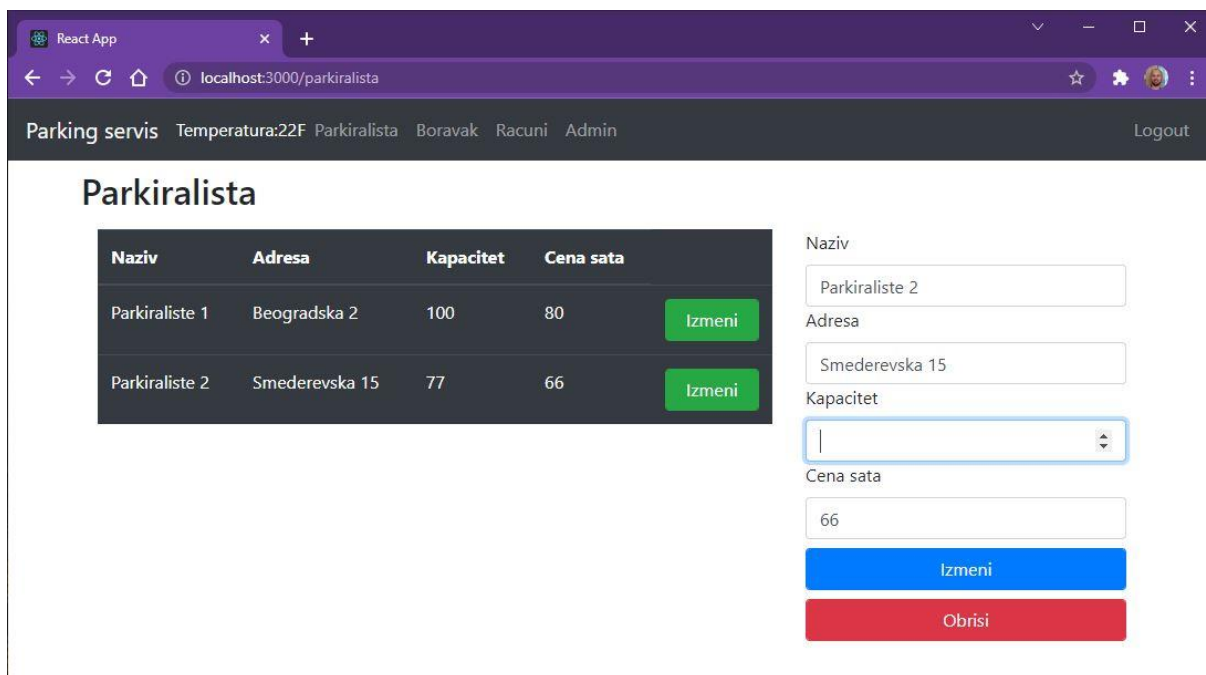
Kapacitet

Cena sata

Kreiraj

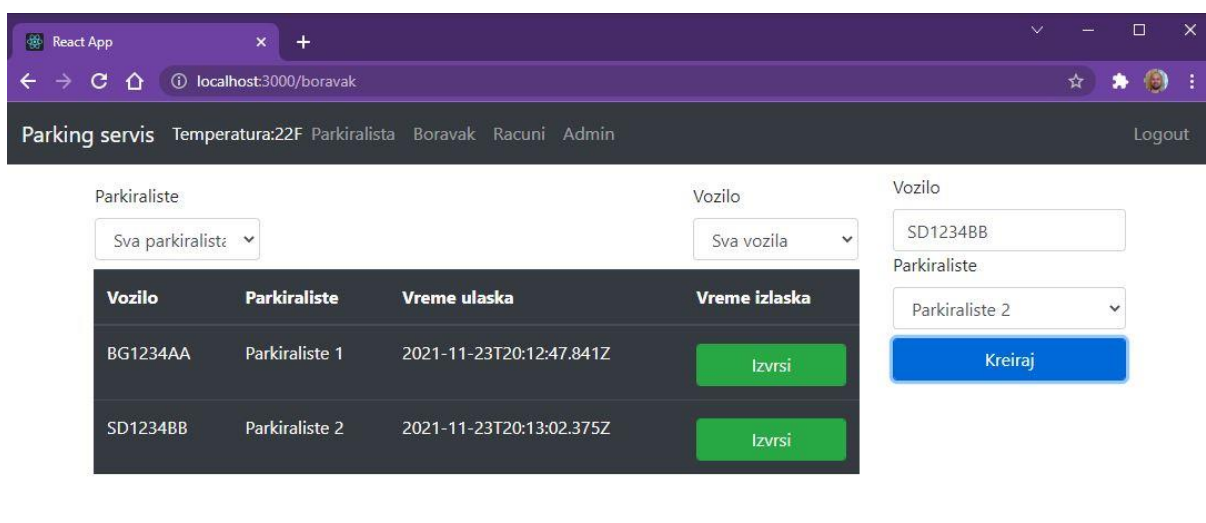
Slika 5: Forma za rad sa parkiralištima

Kada se klikne na dugme Izmeni, podaci izabranog parkirališta se setuju u formu (desno) i spremni su za izmenu (Slika 6). Potvrda izmenjenih podataka vrši se klikom na dugme Izmeni. Ukoliko se izabere dugme Obriši, parkiralište se briše iz sistema.



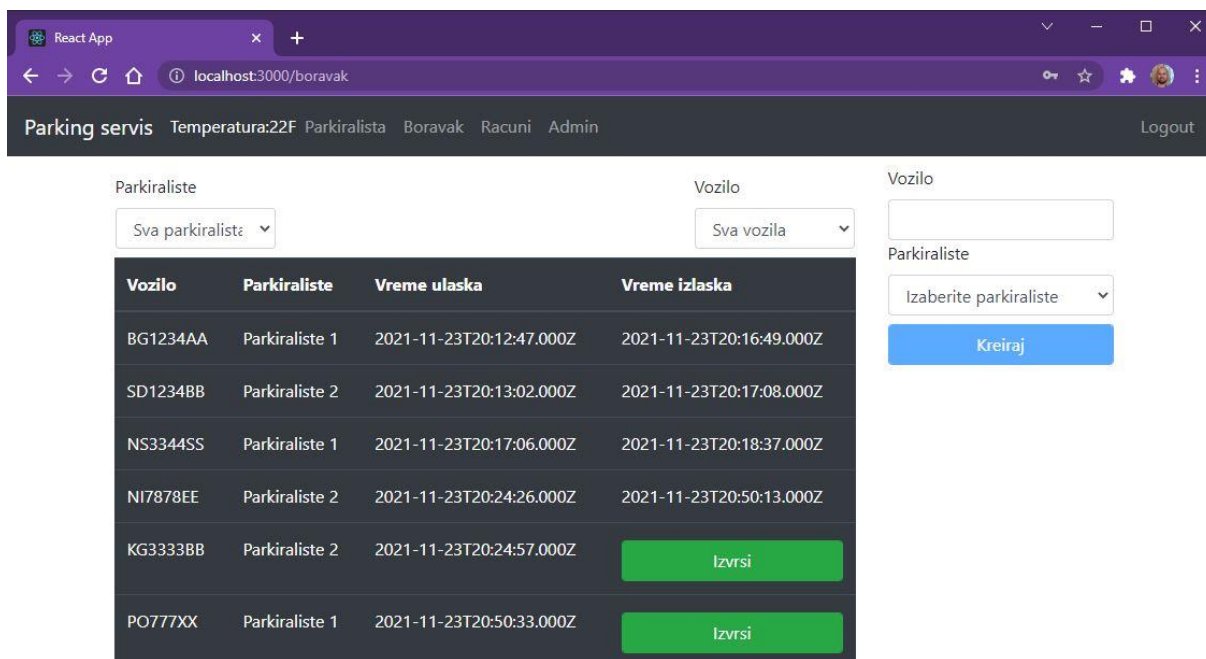
Slika 6: Izmena parkirališta

Kada se u navigacionom meniju izabere kartica Boravak, prelazi se na prikaz svih boravaka vozila na parkiralištima. Moguće je filtriranje po parkiralištu ili vozilu. Desno se nalazi forma za kreiranje novog boravka. Tada je vozilo ušlo na parkiralište i beleži se vreme ulaska.



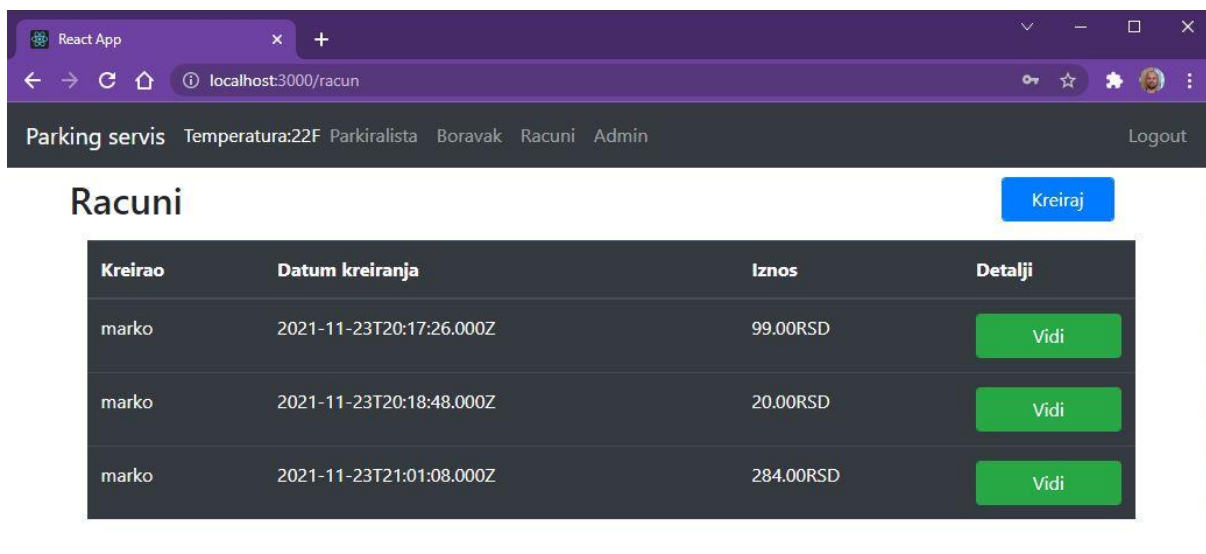
Slika 7: Boravci

Sve dok je vozilo na parkiralištu, dostupno je dugme izvrši (Slika 8). Kada vozilo napusti parkiralište, može se odabrati dugme izvrši, nakon čega se beleži vreme izlaska i boravak je takoreći zaokružen. Za ovaj boravak se može kreirati račun.



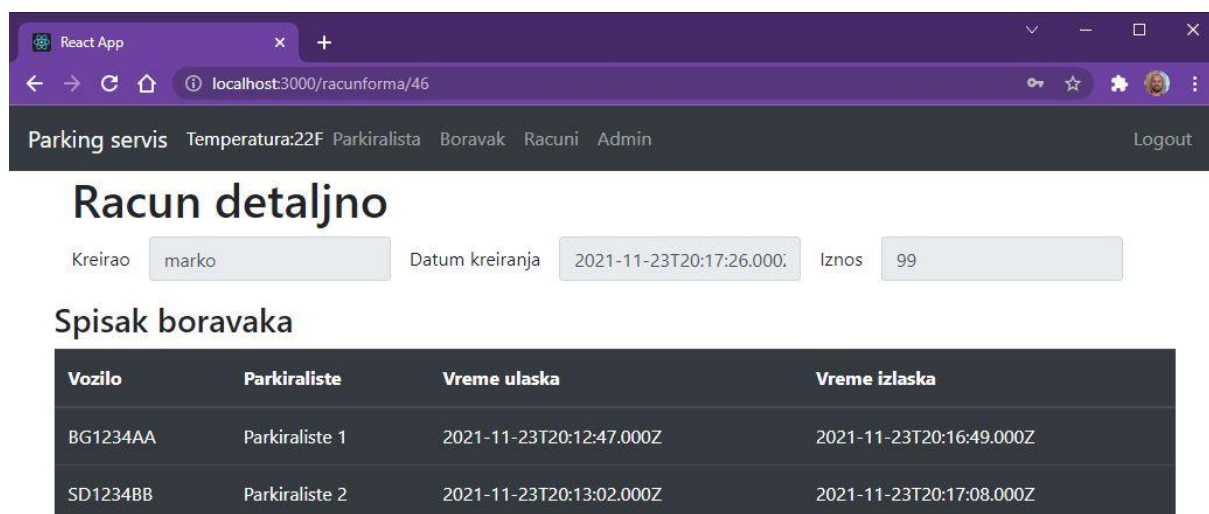
Slika 8: Izlazak vozila

Odabirom opcije računi bira se prikaz svih kreiranih računa što se vidi na sledećoj slici.



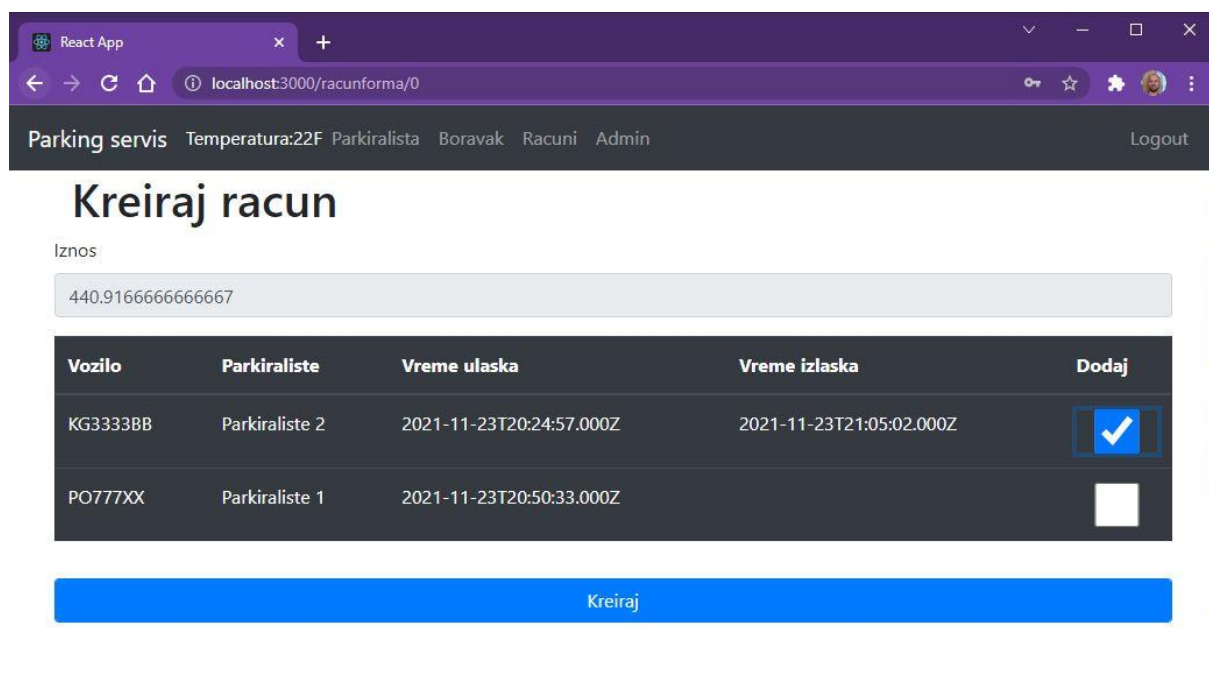
Slika 9: Računi

Kada se klikne na dugme Vidi za određeni račun, prikazuju se detalji odabranog računa (Slika 10). Računi se ne mogu brisati iz sistema.



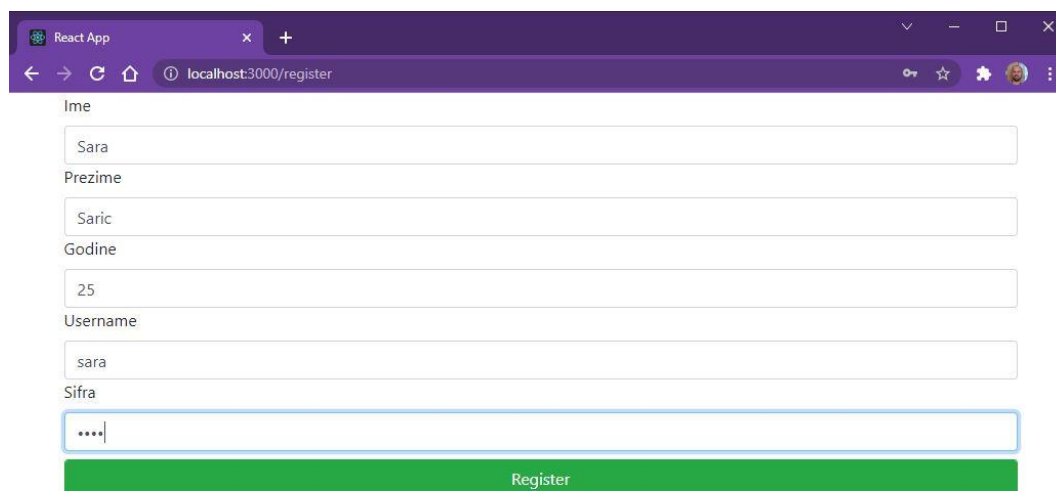
Slika 10: Detalji računa

Da se pak odabrano dugme Kreiraj, otvorio bi se prikaz za kreiranje računa. Odaberu se boravci kojima je registrovano vreme izlaska, i za njih se računa ukupna cena na računu koja zavisi od dužine boravka.



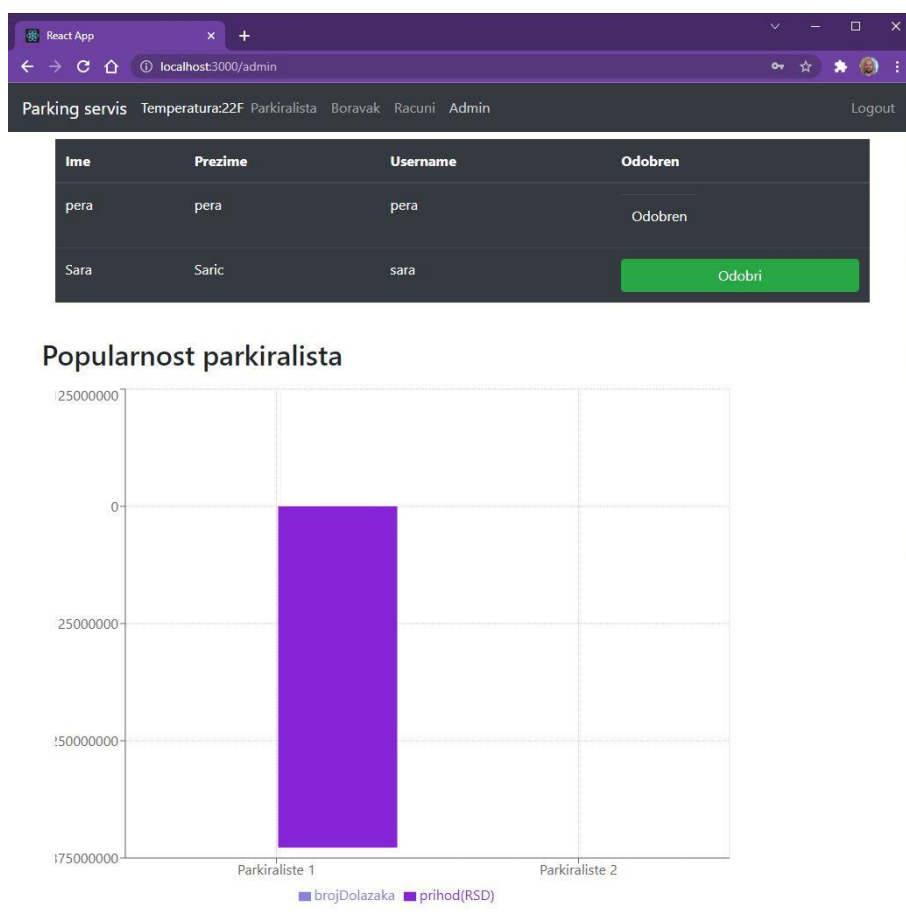
Slika 11: Kreiranje novog računa

Što se tiče korisnika, imamo klasične korisnike i administratore. Kada se neko prvi put registruje, popunjava formu za registraciju (Slika 12).



Slika 12: Forma za registraciju

Međutim, novi korisnik se može ulogovati tek kad ga verifikuje, odnosno odobri administrator. Administrator to može učiniti u Admin delu koji se prikazuje samo administratorima (Slika 13).



Ime	Prezime	Username	Odobren
pera	pera	pera	Odobren
Sara	Saric	sara	<button>Odobri</button>

Popularnost parkiralista

Bar chart showing 'brojDolazaka' (blue bars) and 'prihod(RSD)' (purple bars) for 'Parkiraliste 1' and 'Parkiraliste 2'.

Slika 13: Admin odeljak

U admin odeljku se prikazuje i statistika korišćenja parkirališta. Ovo su neke osnovne funkcionalnosti aplikacije.

8. Link ka udaljenom repozitorijumu

Kod aplikacije se nalazi na repozitorijumu GitHub naloga autora ovog dokumenta, a na sledećoj adresi:

<https://github.com/mareko87/parking>