**Funkcijski zahtjevi**

1. **Autentifikacija radnika**:
   * Radnik se mora prijaviti u aplikaciju kako bi pristupio nalozima za očitavanje brojila.
   * **Login**
     + POST api/users/login
     + **Request**: { id\_radnik: string, password: string }
     + **Response**: { token }
   * **Register**
     + POST api/users/register
     + **Request**: { username: string, email: string, password: string }
     + **Response**: { token }
2. **Generiranje naloga za očitanje**:
   * Generiranje naloga za očitanje brojila (potrebno potvrditi s mentorom).
   * POST api/generiraj
   * **Request**: { id\_kupac: int, datum: datetime, id\_radnik: int, id\_brojilo: char, id\_ocitanje: int }
   * **Response**: { id\_nalog: int, success }
3. **Preuzimanje naloga za očitanje**:
   * Radnik može preuzeti naloge za očitavanje.
   * GET api/nalozi:{nalog\_id}
   * **Request**: { nalog\_id }
   * **Response**: { id\_nalog\_nalog.pdf } (integrirati API za mijenjanje PDF-ova)
4. **Slanje očitanih podataka na poslužitelj**:
   * Slanje podataka nakon očitanja brojila.
   * POST api/ocitanje
   * **Request**: { id\_nalog, komentar, datum\_ocitavanja, tarifa\_visoka, tarifa\_niska }
   * **Response**: { id\_ocitanje, success }
5. **Pohrana podataka u lokalnu bazu**:
   * Svi podaci o očitanju, uključujući stanje brojila i komentare, pohranjuju se u lokalnu SQLite bazu podataka.
6. **Unos očitanja**:
   * Forme za unos očitanja s poljima za nisku tarifu, visoku tarifu, ID naloga i komentar, te gumbom za spremanje unosa kao JSON.
7. **Upravljanje lokalnim podacima**:
   * Radnik može obrisati obavljene naloge iz baze podataka.
   * POST api/obrisiNalog
   * **Request**: { nalog\_id, radnik\_id }
   * **Response**: { success: string }
8. **Pregled očitanja**:
   * Radnik može pregledati sva svoja prijašnja očitanja.
   * GET api/ocitanja
   * **Request**: { radnik\_id }
   * **Response**: { cijela tablica očitanja }
9. **Pregled naloga**:
   * Radnik može pregledati sve svoje naloge.
   * GET api/nalozi
   * **Request**: { radnik\_id }
   * **Response**: { cijela tablica naloga }
10. **Pregled brojila**:

* Radnik može pregledati svoja brojila sa svim pripadajućim atributima, adresama i kupcima.

1. **CRUD operacije za admina**:

* Admin može pregledavati i manipulirati svim podacima u bazi podataka (kupci, brojila, radnici, nalozi za očitavanje) koristeći DELETE, PUT, itd.

**Dodatni funkcionalni zahtjevi**

1. **Validacija i rukovanje greškama**:

* Definirati jasne odgovore za neuspješne zahtjeve (npr., pogrešan unos za login ili registraciju, greške u prijenosu podataka zbog problema s mrežom).

1. **Povratne informacije za radnike**:

* Dodati obavijesti ili poruke koje potvrđuju uspješnost radnji, poput prijenosa podataka, brisanja i uređivanja.

1. **Pretraživanje i filtriranje podataka**:

* Omogućiti radnicima i adminima pretraživanje i filtriranje popisa kupaca, očitanja i naloga prema datumu, statusu, kupcu i drugim parametrima.

1. **Automatska sinkronizacija podataka**:

* Opcija automatske ili ručne sinkronizacije lokalnih podataka s poslužiteljem kako bi se osigurala konzistentnost između SQLite baze na mobilnom uređaju i centralne baze.

1. **Odabir vremenskog raspona za pregled očitanja i naloga**:

* Omogućiti radnicima filtriranje očitanja i naloga prema vremenskom rasponu kako bi lakše upravljali podacima.

1. **Upravljanje korisničkim ulogama i pravima pristupa**:

* Osigurati administraciju korisničkih uloga i prava pristupa u aplikaciji radi kontrole funkcionalnosti koje su dostupne različitim korisnicima (npr., radnicima i adminima).

1. **Generiranje izvješća i izvoz podataka**:

* Omogućiti izvoz očitanja ili naloga u PDF ili CSV formatu za administrativne potrebe ili dijeljenje s vanjskim sustavima.

1. **Evidentiranje promjena podataka (audit log)**:

* Kreirati zapis svih radnji koje admin obavlja na podacima, kao što su dodavanje, uređivanje i brisanje, za povijest izmjena.

### Nefunkcionalni zahtjevi

1. \*Performanse\*:

- Aplikacija mora omogućiti brzo preuzimanje i slanje podataka između klijenta i poslužitelja.

- PostgreSQl baza podataka

2. \*Sigurnost\*:

- Autentifikacija radnika mora biti sigurna, uz enkripciju osjetljivih podataka pri prijenosu (npr., korištenje HTTPS protokola).

4. \*Upotrebljivost\*:

- Korisničko sučelje treba biti jednostavno za korištenje, posebno prilagođeno za rad na terenu.

5. \*Kompatibilnost\*:

- Mobilna aplikacija treba raditi na web-browserima, poslužitelj koristi PostgreSQL i Spring Boot za kompatibilnost s REST servisima.

6. \*Održavanje i proširivost\*:

- Kod treba biti strukturiran i dokumentiran kako bi omogućio proširenje funkcionalnosti.

7. \*Sigurnost podataka\*:

- Podaci moraju biti sigurni tijekom prijenosa i pohrane, uz implementaciju osnovnih integritetskih ograničenja (primarni i strani ključevi, NOT NULL) u bazi podataka.