

**SVEUČILIŠTE U ZAGREBU**  
**FAKULTET ORGANIZACIJE I INFORMATIKE**  
**V A R A Ž D I N**

**Tim: AIR1613**

**Paula Kokić/ Acerinth**

**Ivan Pokec/ ivanpokec**

**Žana Zekić/ zanzekic**

**GitHub repozitorij: <https://github.com/ivanpokec/Indoor-Tracking>**

# **INDOOR TRACKING**

**Projekt iz kolegija Analiza i razvoj programa**

## **TEHNIČKA DOKUMENTACIJA**

**Mentor:**

doc.dr.sc. Zlatko Stapić

**Varaždin, siječanj 2017.**

# Sadržaj

1.	Korištene tehnologije i alati .....	1
2.	Arhitektura aplikacije i sustava.....	2
2.1.	Arhitektura aplikacije Indoor Tracking .....	2
2.2.	Arhitektura sustava .....	3
3.	Podatkovni model .....	4
4.	IndoorTracking WEB API .....	5
5.	Dijagram klasa .....	6
7.	Testiranje aplikacije .....	13

# 1. Korištene tehnologije i alati

U sklopu ovog projekta, odlučili smo koristiti sljedeće tehnologije i alate:

- Android Studio (v 2.2) – razvojno okruženje za Android OS
- VivifyScrum (<https://www.vivifyscrum.com>) – online alat za agilno upravljanje projektom i kolaboracijom koristeći Scrum metodu
- GitHub (<https://github.com/>) – online sustav za verzioniranje programskog koda
- SourceTree – alat za lakše upravljanje kodom u skladu s GitHub sustavom
- VisualParadigm – alat za izradu UML dijagrama
- VisualStudio – razvojno okruženje za .NET i C# aplikacije
- MS SQL Management Studio – alat za upravljanje MS SQL bazom podataka

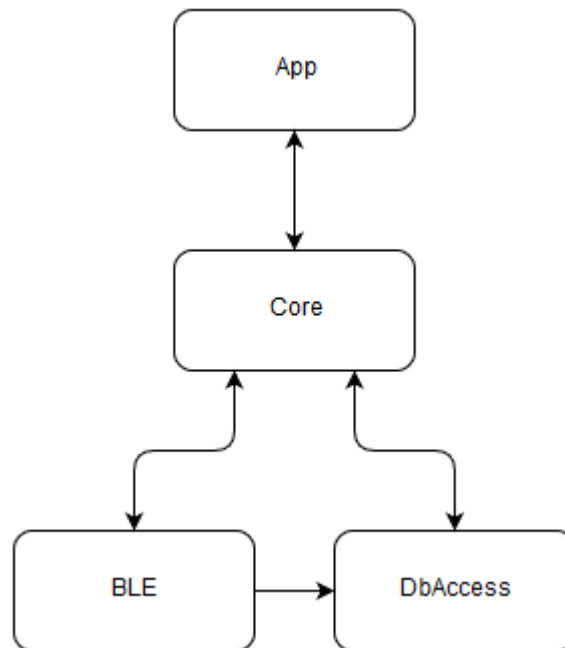
Svi korišteni alati su besplatni za preuzimanje i korištenje. VivifyScrum, VisualStudio i MS SQL Management Studio dobiveni su besplatno putem studentske licence.

Projekt je impementiran za Android 4.4 KitKat (API level 19).

## 2. Arhitektura aplikacije i sustava

### 2.1. Arhitektura aplikacije Indoor Tracking

Aplikaciju čine 4 modula: App modul, DbAccess modul, BLE modul (eng. *Bluetooth Low Energy*) te Core modul. Slika 1. prikazuje navedenu arhitekturu te veze između modula.



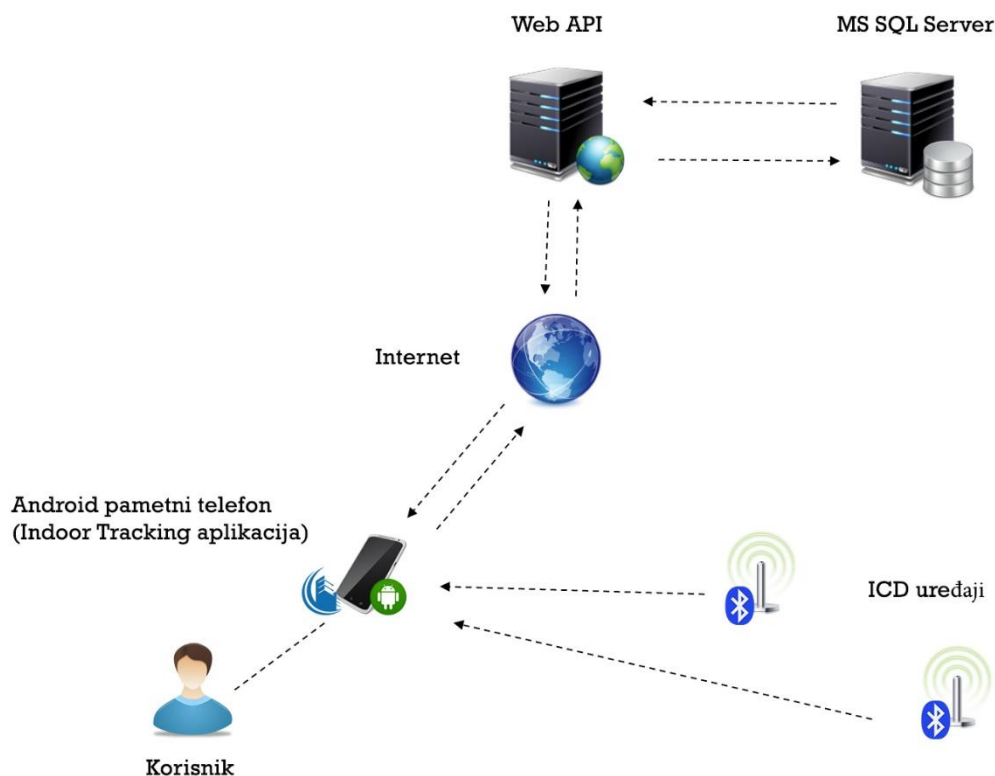
Slika 1. Arhitektura aplikacije - moduli

App modul implementirat će korisničko sučelje te će služiti za prikaz podataka, koje dobiva od Core modula. Core modul implementira sva sučelja i klase potrebne za komuniciranje svih modula, tako da je on u vezi za svim modulima. BLE modul sadrži klase koje implementiraju algoritme za učitavanje ICD uređaja te algoritme za određivanje najbližeg ICD uređaja, odnosno određivanje korisničke lokacije. DbAccess modul sadrži web API za pristupanje serverskoj MS SQL bazi podataka, budući da baza nije javna te se nalazi iza sigurnosne stijene (eng. *firewall*) pa joj se ne može direktno pristupiti.

Navedeni opis odnosi se na aplikaciju koja se instalira na svaki Android pametni telefon, pojedinačno za svakog korisnika. U nastavku bit će opisana arhitektura cijelog sustava.

## 2.2. Arhitektura sustava

Na Slici 2. prikazana je arhitektura cijelog sustava.

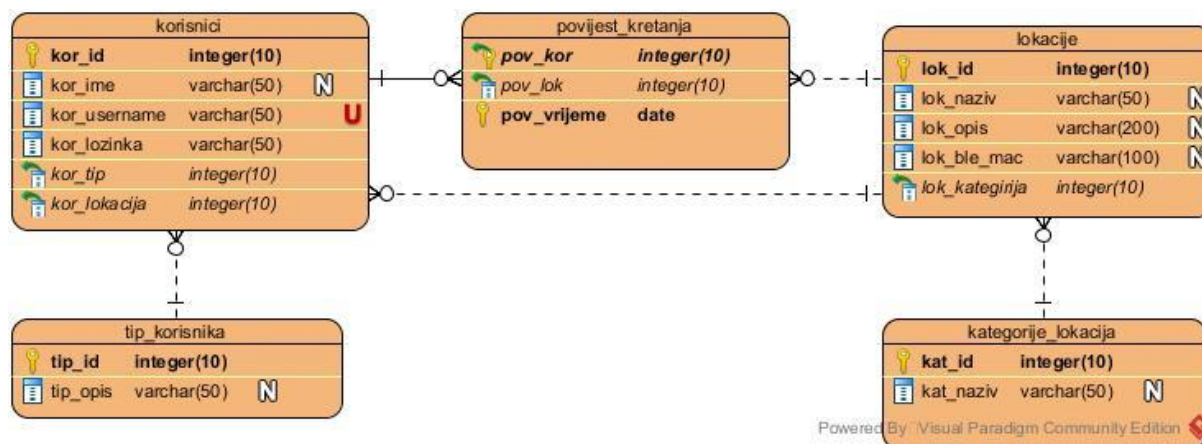


Slika 2. Dijagram arhitekture sustava

Aplikacija Indoor Tracking instalira se na Android pametni telefon svakog korisnika. Aplikacija zahtijeva korištenje Bluetooth i Wi-Fi tehnologija. Nakon pokretanja aplikacije, uređaj može putem Bluetooth-a učitavati obližnje ICD uređaje, koji su postavljeni po jedan u svakoj prostoriji zgrade. Nakon određivanja najbližeg ICD uređaja i dohvaćanja njegove MAC adrese, aplikacija putem Interneta, preko Web API sučelja šalje zahtjev sustavu za upravljanje bazom podataka (MS SQL Server). Sustav preko istog API-ja vraća odgovor aplikaciji, odnosno naziv lokacije na kojoj se nalazi korisnik. Kada se promijeni lokacija, odnosno najbliži ICD uređaj, postupak se ponavlja na isti način. Kada korisnik putem aplikacije zatraži bilo kakve podatke (npr. o korisnicima, postojećim lokacijama i sl.), podaci se na isti način dohvaćaju preko API-ja iz baze podataka te šalju korisniku na pametni telefon.

### 3. Podatkovni model

Na Slici 3. prikazan je podatkovni model (ERA dijagram) aplikacije.



Slika 3. ERA model

Tablica *korisnici* sadrži sve potrebne podatke o korisnicima aplikacije. Kao primarni ključ koristit će se atribut *kor\_id*, korisničko ime sprema se u atribut *kor\_username*, a ostali atributi služe za općeniti opis pojedinog korisnika. *Kor\_ime* predstavlja ime i prezime korisnika, *kor\_lozinka* njegovu lozinku, *kor\_tip* predstavlja tip korisnika, *kor\_lokacija* predstavlja korisnikovu primarnu lokaciju (njegovu radnu sobu).

Tablica *tip\_korisnika* sadrži kategorije korisnika, za potrebe kategoriziranja korisnika (obični korisnik, šef odjela, direktor i sl.).

Tablica *lokacije* sadrži podatke o lokacijama (prostorijama) u zgradi. Primarni ključ je *lok\_id*, a svaku lokaciju dodatno opisuju sljedeći atributi: *lok\_naziv* predstavlja naziv prostorije, *lok\_opis* predstavlja njezin opis, *lok\_ble\_mac* predstavlja MAC adresu BLE uređaja koji je postavljen u toj prostoriji (i prema kojemu se zapravo određuje položaj korisnika) i *lok\_kategorija* predstavlja vanjski ključ na kategoriju lokacije.

Tablica *kategorija\_prostorija* sadrži nazive kategorija prema kojima su svrstane prostorije, radi lakše pretrage u aplikaciji.

Navedeni podatkovni model implemantiran je kao zasebna baza podataka unutar MS SQL Servera tvrtke Mobilisis. Podacima se pristupa preko Web API servisa koji će biti opisan u sljedećem poglavlju.

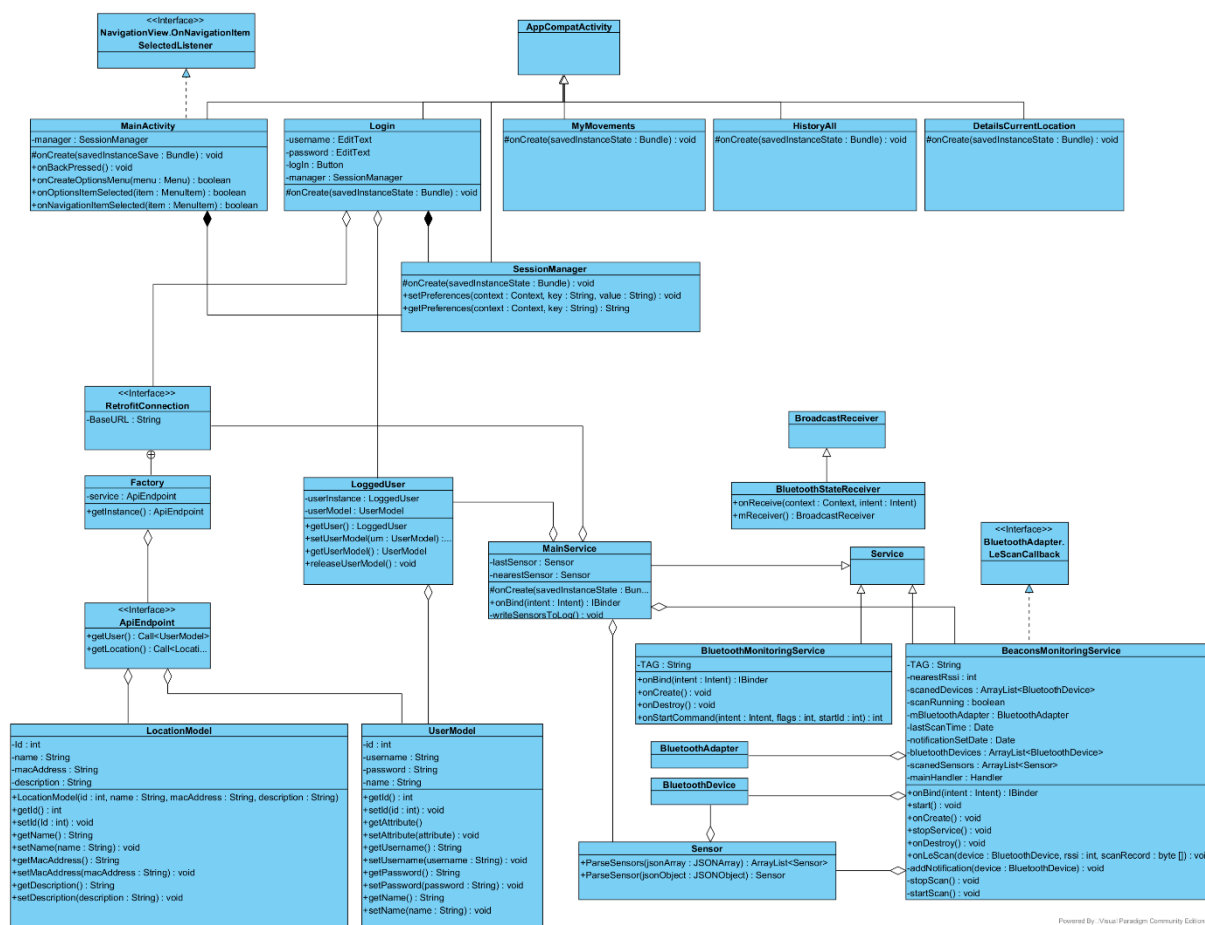
## 4. IndoorTracking WEB API

IndoorTracking Web API je RESTful online servis preko kojeg se dohvaćaju podaci iz baze koja se nalazi iza sigurnosne stijene (eng. *firewall*) te nema javni pristup. REST (eng. *Representational State Transfer*) ili RESTful web servisi su jedan od načina za pružanje interoperabilnosti računalnih sustava. Ovaj servis omogućuje dohvaćanje i manipulaciju podacima preko predefiniranih setova operacija. Ovom servisu se pristupa preko URI adrese <http://development.mobilisis.hr/IndoorTracking/api> te ovisno o poslanim paramtrima se vraća odgovor u JSON obliku ili u obliku zadanom u poslanom zaglavlju HTTP zahtjeva. Unutar servisa su implementirane GET i POST metode. Kako bismo dobili željene podatke, kreiraju se klase kontrolori koje služe za upravljanje zahtjevima i klase modeli koje predstavljaju oblik odgovora koji se vraća na zahtjev. Unutar klase kontrolera je implementirana komunikacija s bazom i vraćanje podataka korisniku. Kako bismo dohvatili željene podatke o lokaciji moramo preko POST zahtjeva na <http://development.mobilisis.hr/IndoorTracking/api/location> poslati u JSON obliku MAC adresu najbližeg uređaja npr: { 'mac': 'A0:E6:F8:4F:AF:BC' } te ćemo ako je tražena MAC adresa nađena u bazi, kao odgovor dobiti polje koje sadrži jedan element čija struktura je definirana u klasi Location unutar servisa. Prema tome, za našu MAC adresu će biti vraćen odgovor:

```
[{  
  "Id": 1,  
  "name": "soba1 T1",  
  "macAddress": "A0:E6:F8:4F:AF:BC",  
  "description": "T1 ICD"  
}]
```

## 5. Dijagram klasa

Na Slici 4. prikazan je dijagram klasa za aplikaciju Indoor Tracking.



*Slika 4. Dijagram klasa*

Na dijagramu su navedene klase koje su trenutno implementirane u projektu, a navedene su i neke općenite Android klase i sučelja koja se koriste (AppCompatActivity, Service, BluetoothAdapter, BluetoothDevice i sl.) te njihovi atributi i metode nisu navedene.

Klasa MainActivity prikazuje početni zaslon aplikacije (u budućnosti Trenutnu lokaciju), te su u njoj implementirane metode za upravljanje glavnim izbornikom aplikacije.

Klasa Login prikazuje zaslon za prijavu (prije zaslona MainActivity klase) i zadužena je za prijavu korisnika i kreiranje njegove sesije nakon uspješne prijave. Zato ona koristi klasu RetrofitConnection koja se spaja na IndoorTracking Web API servis i provjerava podatke za prijavu, odnosno vraća sve ostale podatke o korisniku koje nam trebaju.

Klasa `SessionManager` zadužena je za pohranjivanje korisničke sesije, jer izlazak iz aplikacije ne znači odjavu korisnika. Prijavljeni korisnik se sprema u sesiju, i prilikom ponovnog pokretanja aplikacije ne traži se prijava, već se odmah otvara zaslon `MainActivity`.



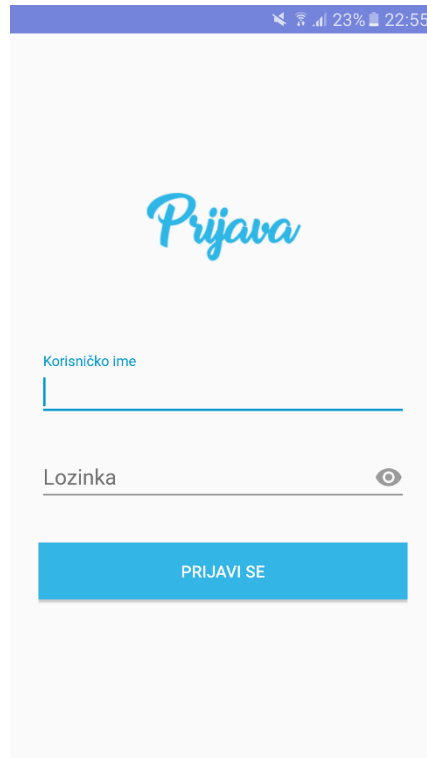
Ipak, kada se klikne na gumb „Odjava“, korisnik se odjavljuje, sesija se briše i prilikom ponovnog pokretanja aplikacije traži se korisnička prijava.

Klasa `BluetoothMonitoringService` provjerava da li je došlo do kakvih promjena vezanih uz sam bluetooth na uređaju, te bilježi te promjene. Unutar klase `Sensor` se nalazi struktura podataka vezana uz BLE beacons te upravljanje njihovim promjenama, dok se u klasi `BeaconsMonitoringService` nalaze metode koje se cijelo vrijeme izvršavaju u pozadini i detektiraju BLE beacons, čitaju njihove podatke te ih zapisuju u liste iz kojih se zatim određuje koji je beacon najbliži i koja je njegova MAC adresa. Važno je napomenuti da smo ovu klasu, kao i sve ostale klase koje sudjeluju u algoritmu detekcije BLE uređaja te određivanja najbližeg uređaja, dobili gotove na slobodno korištenje od tvrtke Mobilisis. Bilo je potrebno samo uskladiti neke metode, odnosno izvući iz tih klasa ono što je nama bilo potrebno za aplikaciju Indoor Tracking te integrirati s ostatkom koda.

Sučelja `RetrofitConnection` i `ApiEndpoint` služe za pristupu bazi podataka. `LocationModel` i `UserModel` su klase koje odgovaraju entitetima lokacija i korisnik iz baze podataka te sadrže odgovarajuće attribute koje ih opisuju. Klasa `LoggedUser` je klasa namijenjena pristupu klasi `UserModel`, koja sadrži podatke o logiranom korisniku. Klasa `MainService` je klasa koja povezuje sve algoritme za nadzor i određivanje najbližeg senzora, te javlja ostalim klasama kada dođe do promjene lokacije. Klase `MyMovements`, `DetailsCurrentLocation` i `HistoryAll` još su uvijek u razvoju, a služit će realizaciji prikaza pojedinih funkcionalnosti.

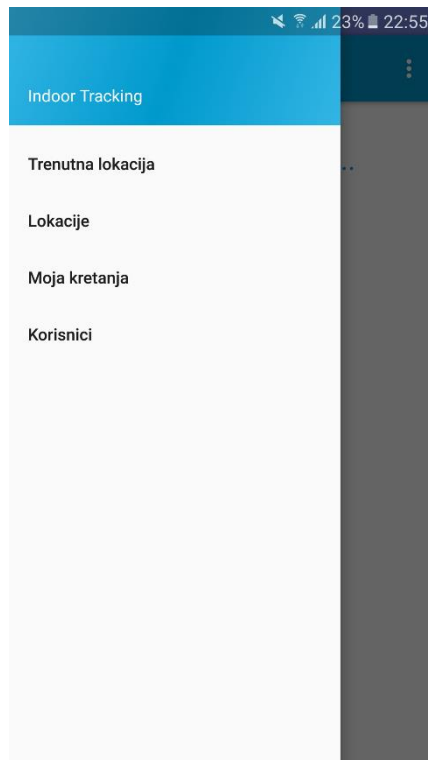
## 6. Implementirane funkcionalnosti

U nastavku će biti prikazani ekrani dosada implementiranih funkcionalnosti aplikacije.



*Slika 5. Prijava*

Kada korisnik pokrene aplikaciju otvori mu se početni ekran s prijavom koju možemo vidjeti na Slici 5.



Slika 6. Izbornik 1



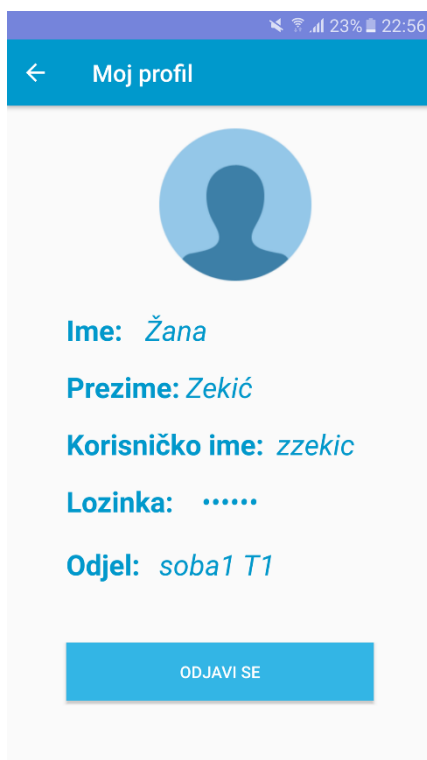
Slika 7. Izbornik 2

Na slikama 6. i 7. prikazani su izbornici. Iz prvog izbornika korisnik može izabrati neku od glavnih funkcionalnosti koje nudi aplikacija. Iz drugog izbornika korisnik se može odjaviti ili pogledati postavke ili svoj profil.



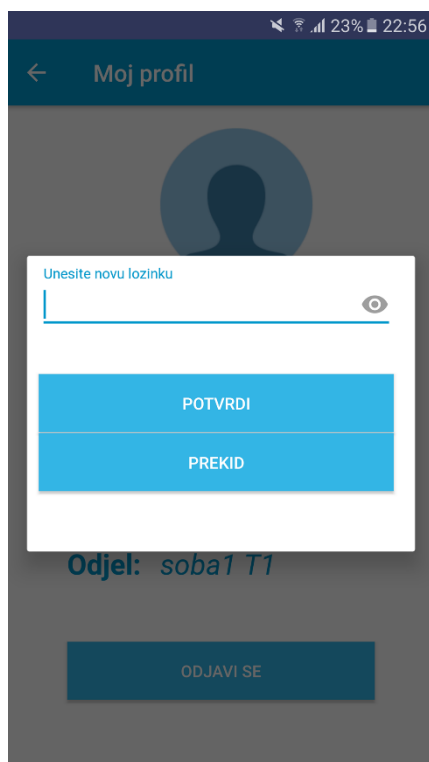
*Slika 8. Moj profil*

Na slici 8. možemo vidjeti prikaz ekrana koji se otvori nakon što se korisnik prijavi u aplikaciju. Prikaže mu se naziv lokacije na kojoj se trenutno nalazi te kategorija kojoj ta lokacija pripada. Također, prikaže se i kratki opis te lokacije. Na početnom ekranu također se prikazuje i gumb detalji te kada korisnik klikne na njega prikazat će se svi korisnici na istoj lokaciji.

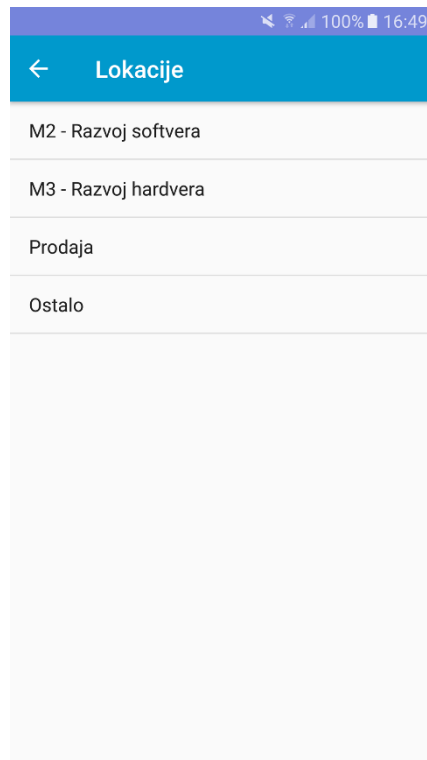


*Slika 9. Moj profil*

Na slici 9. možemo vidjeti profil prijavljenog korisnika. On može pregledavati svoje vlastite podatke, a pritiskom na lozinku može ju promijeniti što je vidljivo na slici 10.



*Slika 10. Promjena lozinke*



*Slika 11. Prikaz kategorija lokacija*

Kada korisnik izabere prikaz lokacija iz izbornika najprije mu se prikažu sve kategorije lokacija.

## 7. Testiranje aplikacije

Aplikaciju je potrebno testirati na odgovarajućim Android uređajima, s instaliranim API-jem 19 ili više. Prije pokretanja aplikacije, potrebno je uključiti Bluetooth, osigurati valjanu internetsku vezu te dozvoliti upotrebu lokacije.

Pri prvom pokretanju, aplikacija će zahtijevati autentikaciju. Podaci za prijavu koji se mogu koristiti jesu:

`korisničko ime: zzekic`

`lozinka: zanaz`

Kako bi mogli u potpunosti koristiti sve funkcionalnosti aplikacije, potrebno je nalaziti se u blizini barem jednog ICD uređaja koji je registriran u bazu podataka aplikacije. Za naše potrebe razvoja i testiranja aplikacije, tvrtka Mobilisis nam je posudila na slobodno korištenje svoje vlastite ICD uređaje.