

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, studeni 2016.

Sadržaj

1.	Korištene tehnologije i alati	1
2.	Arhitektura aplikacije.....	2
3.	Podatkovni model	3
4.	IndoorTracking WEB API	4
5.	Dijagram klasa	5

1. Korištene tehnologije i alati

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

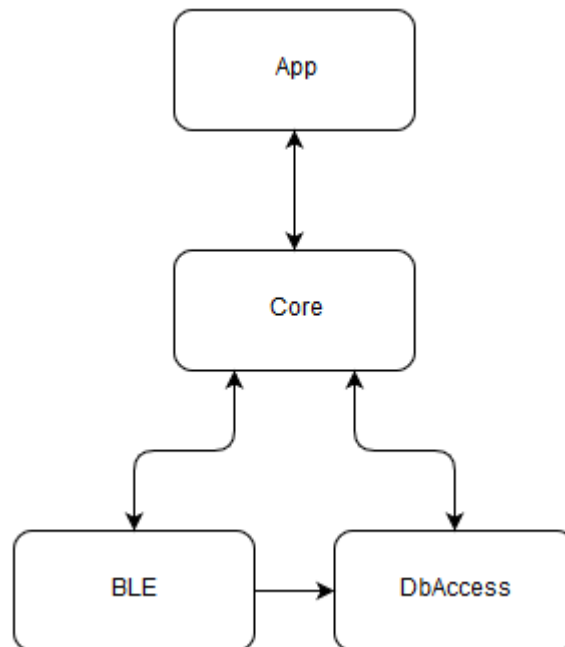
- Android Studio (v 2.2) – razvojno okruženje za Android OS
- VivifyScrum (<https://www.vivifyscrum.com>) – online alat za agilno upravljanje projektom i kolaboracijom koristeći Scrum metodu
- GitHub (<https://github.com/>) – online sustav za verzioniranje programskog koda
- SourceTree – alat za lakše 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

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.

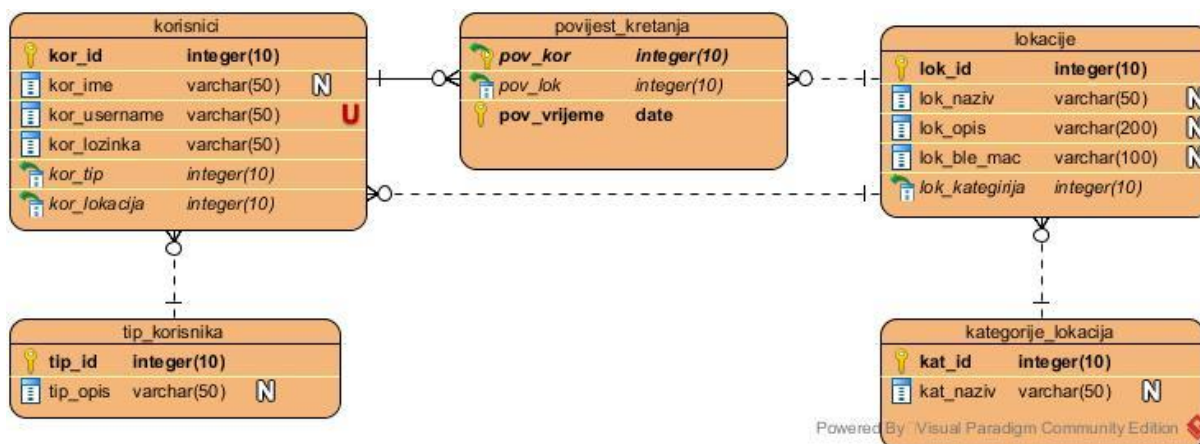


Slika 1. Arhitektura aplikacije - moduli

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 servis 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.

3. Podatkovni model

Na Slici 2. prikazan je podatkovni model (ERA dijagram) aplikacije.



Slika 2. 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.

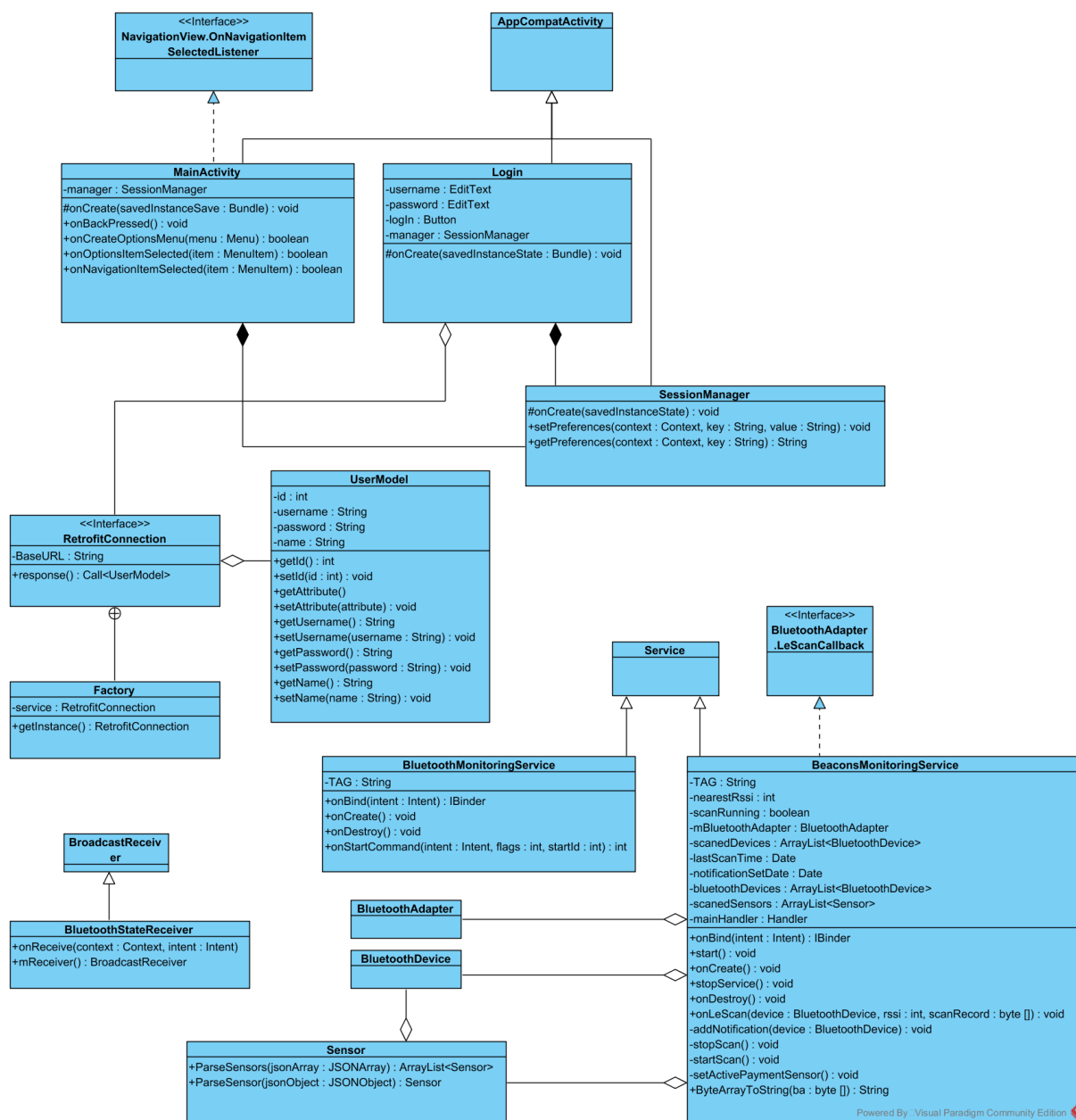
4. IndoorTracking WEB API

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 3. prikazan je dijagram klasa za aplikaciju Indoor Tracking.



Slika 3. 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.