

SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 680

**Utvrđivanje mikrolokacije  
mobilnog uređaja u zatvorenom  
prostoru**

Ivan Kraljević

Zagreb, lipanj 2014.



# SADRŽAJ

<b>1. Uvod</b>	<b>1</b>
<b>2. Tehnologija Bluetooth Low Energy</b>	<b>2</b>
2.1. Tehničke značajke . . . . .	3
<b>3. Odašiljači iBeacon</b>	<b>6</b>
3.1. Odašiljač Kontakt.io Beacon . . . . .	9
3.1.1. Tehnička specifikacija . . . . .	10
3.1.2. Struktura paketa . . . . .	11
<b>4. Utvrđivanje relativne i absolutne lokacije u zatvorenom prostoru</b>	<b>13</b>
4.1. Utvrđivanje relativne udaljenosti od odašiljača . . . . .	15
4.2. Utvrđivanje apsolutne lokacije . . . . .	26
<b>5. Radno okruženje Apache Cordova</b>	<b>28</b>
5.1. Biblioteka Cordova BLE . . . . .	30
5.2. Radno okruženje Ionic . . . . .	32
<b>6. Praktični rad</b>	<b>34</b>
6.1. Poslužiteljska aplikacija - Spree Commerce . . . . .	34
6.1.1. Prilagođavanje i nadogradnja . . . . .	37
6.1.2. Instalacija i pokretanje poslužiteljske aplikacije . . . . .	39
6.2. Klijentska aplikacija . . . . .	40
6.2.1. Eksperimentalna aplikacija . . . . .	41
6.2.2. Mobilna aplikacija Spree . . . . .	43
<b>7. Zaključak</b>	<b>48</b>
<b>Literatura</b>	<b>49</b>

# 1. Uvod

Danas najkorišteniji sustav za pozicioniranje, navigaciju i vremenske usluge je GPS (engl. *Global Positioning System*). U idealnim uvjetima preciznost određivanja lokacije GPS-a je oko deset metara, ali zbog toga što signali do uređaja dolaze od jako udaljenih satelita na nju utječe ogroman broj parametara, od atmosferskih uvjeta (ionosferska i troposferska refrakcija) do metala u okolini uređaja. Dodatno, u zatvorenim prostorima signal satelita remete i zidovi kroz koje oni prolaze te ostale prepreke u okolini. Posljedično, korištenje GPS-a za pozicioniranje i navigaciju u zatvorenim prostorima je izrazito nepouzdano i nepraktično [3].

Iz gore navedenih razloga očita je potreba za sustavom koji će moći pouzdano odrediti lokaciju korisnika u zatvorenom prostoru. Ovaj problem se našao u središtu velikog broja znanstvenih istraživanja pri čemu se većina fokusira na određivanje lokacije na temelju signala bežične lokalne računalne veze (engl. *wireless local area network*, *WLAN*). Uvođenjem Bluetooth 4.0 specifikacije i tehnologije *Bluetooth Low Energy* dolazi do razvoja niza jeftinih uređaja male potrošnje koji se potencijalno mogu iskoristiti za rješavanje problema pozicioniranja i navigacije u zatvorenom prostoru.

U ovom diplomskom radu govori se o tehnologiji Bluetooth, problemu određivanja lokacije u zatvorenom prostoru, rješenju temeljenom na Bluetooth Smart odašiljačima, odnosno tehnologiji iBeacon, te na kraju su ponuđene smjernice za daljnji razvoj. U drugom dijelu rada prikazati će se kako se Bluetooth Smart odašiljači mogu primijeniti u mobilnim aplikacijama.

## 2. Tehnologija Bluetooth Low Energy

Tehnologija Bluetooth je standard bežične komunikacije koji se koristi za razmjenu podataka na maloj udaljenosti. Bluetooth je razvijen 1994. godine u Ericssonu, a 1998. godine Ericsson, IBM, Intel, Nokia i Toshiba osnivaju posebno nadležno tijelo, Bluetooth Special Interest Group (SIG). Uloga nadležnog tijela je unaprijeđenje standarda, ispravna implementacija i licenciranje tehnologije Bluetooth.

Glavne odlike tehnologije Bluetooth su niska cijena Bluetooth uređaja, niska potrošnja energije, mali domet, robusnost te korištenje na globalnoj razini. Bluetooth omogućava brzinu prijenosa reda veličine 1 Mbit/s te koristi nelicencirani frekvencijski pojas od 2.4 do 2.485 GHz, odnosno koristi ISM područje (engl. *industrial, scientific and medical*) koje je frekvencijski usklađeno na globalnoj razini. Uz to, Bluetooth nudi radijsku vezu prema drugim sustavima, uređaji različitih proizvođača su međusobno kompatibilni i dopuštena je komutacija paketa i kanala.

Sredinom 2010. godine Bluetooth SIG objavljuje Bluetooth 4.0 specifikaciju koja uključuje *Classic Bluetooth*, *Bluetooth high speed* i *Bluetooth low energy* protokole. *Bluetooth low energy* (u daljem tekstu BLE), poznat i pod nazivom *Bluetooth Smart*, je tehnologija koja je optimizirana tako da ima veoma nisku potrošnju energije. Uz to, glavne odlike tehnologije su mogućnost višegodišnjeg rada na malom izvoru energije (poput *button-cell* ili AAA baterije), mala veličina i niska cijena te kompatibilnost sa mobilnim uređajima, tabletima i računalima.

Za ugradnju tehnologije BLE u uređaje Bluetooth 4.0 specifikacija uvodi dva načina rada: *single-mode* i *dual-mode*. *Single-mode* način rada obuhvaća integraciju samo BLE funkcionalnosti u kontroler, dok *dual-mode* načina rada omogućava integraciju BLE funkcionalnost u standardni Bluetooth kontroler. Proizvođači uređaja imaju na raspolaganju te dvije opcije i pri tome je važno napomenuti da uređaji sa *single-mode* načinom rada ne mogu komunicirati sa uređajima koji koriste klasični Bluetooth protokol.

Danas se velika većina mobilnih uređaja proizvodi sa podrškom i za standardni Bluetooth i za BLE, tj. u uređaju se ugrađuje Bluetooth mikrokontroler sa *dual-mode* načinom rada. Mobilni operacijski sustavi koji trenutno podržavaju BLE su:

- Android 4.3 i noviji
- iOS 5 i noviji
- Windows Phone 8.1
- Blackberry 10

Bitno je napomenuti da BLE nije i ne pokušava biti optimizirana verzija klasične Bluetooth tehnologije, već cilja na sasvim nove načine primjene. Predviđene primjene su u sportu, zdravstvu, trgovini, turizmu, mjerenu udaljenosti i druge.

## 2.1. Tehničke značajke

Bluetooth Low Energy tehnologija temelji se na profilima i *Generic Attribute Profile* (u dalnjem tekstu GATT) specifikaciji.

BLE profili su posebne specifikacije koje definiraju servise koji se koriste u određenim scenarijima. Od proizvođača konkretnih uređaja se očekuje da zadovolje određen profil kako bi se osigurala međusobna kompatibilnost između raznih uređaja (npr. ako uređaj koji prati krvni tlak osobe zadovoljava HRP profil svi uređaji mogu uniformno čitati njegove servise). Popularniji standardizirani profili su:

- HTP (*Health Thermometer Profile*) - definira način spajanja i interakcije sa termometrom
- GLP (*Glucose Profile*) - definira način spajanja i interakcije sa senzorom koji mjeri razinu glukoze u krvi osobe
- BLP (*Blood Pressure Profile*) - definira način spajanja i interakcije sa senzorom koji mjeri krvni tlak osobe
- HRP (*Heart Rate Profile*) - definira način spajanja i interakcije sa senzorom rada srca
- FMP (*Find Me Profile*) - definira ponašanje gdje pritisak gumba na jednom uređaju šalje obavijest drugom uređaju
- PXP (*Proximity Profile*) - omogućava praćenje udaljenosti između dva uređaja

- LNP (*Location and Navigation Profile*) - definira način spajanja i interakcije sa senzorom navigacije

Ostali profili mogu se naći na službenim Bluetooth stranica<sup>1</sup>. Svi BLE profili su definirani na temelju GATT specifikacije.

Specifikacija GATT definira način na koji se šalju i primaju podaci kratke duljine (često zvani atributi) preko BLE veze. GATT koristi *Attribute* protokol (u dalnjem tekstu ATT) koji ima sličnu funkcionalnost kao protokol SDP (engl. *service discovery protocol*) kod standardnog Bluetootha, samo što je optimiziran i pojednostavljen za korištenje u uređajima male potrošnje. SDP (klasični Bluetooth) i ATT (Bluetooth Low Energy) protokoli omogućavaju uređajima da međusobno saznaju koje servise podržavaju i koje parametre trebaju koristiti pri stvaranju podatkovne veze. GATT definira na koji način su ATT atributi grupirani tako da čine konkretne servise. Bitni koncepti kod GATT specifikacije su:

### **Klijent**

Uređaj koji pokreće komunikaciju te šalje zahtjeve i prima odgovore (npr. mobilni uređaj ili računalo).

### **Poslužitelj (server)**

Uređaj koji prima zahtjeve i šalje odgovore (npr. senzor temperature ili iBeacon odašiljač).

### **Servis**

Kolekcija povezanih karakteristika koje zajedno čine nekakvu funkciju (npr. *Health Thermometer* servis uključuje karakteristike za vrijednost temperature, interval čitanja i mjernu jedinicu temperature). Svaki servis može imati proizvoljan broj karakteristika.

### **Karakteristika<sup>2</sup>**

Vrijednost koja se izmjenjuje između klijenta i poslužitelja (npr. krvni tlak osobe ili trenutno stanje baterije). Svaka karakteristika može imati proizvoljan broj opisnika.

### **Opisnik<sup>3</sup>**

Vrijednost koja pobliže opisuje neku karakteristiku (npr. minimalna i maksi-

---

<sup>1</sup><https://developer.bluetooth.org/gatt/profiles/>

<sup>2</sup>engl. *Characteristic*

<sup>3</sup>engl. *Descriptor*

malna vrijednost karakteristike ili mjerna jedinica karakteristike).

### 3. Odašiljači iBeacon

Odašiljači iBeacon su jeftini uređaji, niske potrošnje energije koji korištenjem tehnologije BLE obavještavaju obližnje uređaje o svojoj prisutnosti. Obližnji uređaji (poput mobilnih uređaja i tableta) mogu se preplatiti na obavijesti (notifikacije, engl. *notifications*) odašiljača te mogu primati razne sadržaje (poput teksta, slike ili URL adresa) od njih. Krajem 2013. godine tehnologiju iBeacon patentirala je američka multinacionalna korporacija Apple Inc.

Neki od zanimljivih načina primjene tehnologije iBeacon su u muzejima, trgovinama, bolnicama i ostalim ustanovama gdje se sadržaj mijenja ovisno o položaju u prostoriji. Posjetitelj muzeja može na mobilni uređaj ili tablet primiti sadržaj o objektu kojega trenutno promatra (npr. informacije o skulpturi ili slici). Uz to, osoblje muzeja može pratiti koji su objekti najgledaniji i slično. U bolnici se može primijeniti tako da liječnik na prijenosni tablet dobije sve podatke o pacijentu, od povijesti bolesti do trenutne dijagnoze, kada se približi njegovoj sobi ili krevetu. U trgovini se može iskoristiti tako da kupca obavijesti o predmetima na popustu u blizini i sličnim scenarijima. Također, odašiljači iBeacon mogu se iskoristiti i kao sustav beskontaktnog plaćanja na sličan način kako se i NFC<sup>1</sup> tehnologija koristi. Ovo su samo najjednostavniji i najopćenitiji slučajevi gdje se tehnologija iBeacon može ukomponirati, broj načina korištenja tehnologije je ogroman.

Uređaj iBeacon se može konfigurirati tako da se sadržaj šalje samo kad se uređaj približi odašiljaču na određenu udaljenost. Pri tome su definirana tri parametra udaljenosti: neposredna (*immediate*), mala (*near*) i velika (*far*) udaljenost. Neposredna udaljenost je do nekoliko centimetara, mala do nekoliko metara, dok je velika udaljenost iznad deset metara. Ove vrijednosti su aproksimativne jer ovisno o stvarnim uvjetima u kojima su odašiljači postavljeni signal može dosta varirati pa precizne vri-

---

<sup>1</sup>*Near field communication*, bežična tehnologija jako kratkog dometa korištena za komunikaciju između dva krajnja uređaja

jednosti nisu upotrebljive.

Budući da je tehnologija iBeacon relativno nova, detaljna specifikacija nije javno dostupna stoga su gotovo svi proizvođači odašiljača iBeacon reverznim inženjeringom otkrili značajan dio iBeacon Bluetooth profila.

Kada je Apple predstavio tehnologiju iBeacon objavili su aplikaciju AirLocate pomoću koje se iPhone ili iPad koji podržava BLE može ponašati kao odašiljač iBeacon. Također, pomoću iste aplikacije moguće je podešiti sve parametre odašiljača. Kombinacijom te aplikacije pokrenute na nekom iPhone ili iPad uređaju i uređaja koji može snimati Bluetooth Low Energy pakete (poput računala sa ugrađenim Texas Instruments CC2540 čipom) istraživači uključujući i [4] su došli do strukture *advertising* paketa koja je prikazana u tablici 3.1.

**Tablica 3.1:** Struktura *advertising* paketa iBeacon odašiljača

Bajt	Vrijednost	Opis
1.	02	Duljina podataka - 2 bajta
2.	01	Tip podataka - zastavice
3.	X	LE i BR/EDR zastavice
4.	1A	Duljina podataka - 26 bajtova
5.	FF	Podaci o proizvođaču
6.	4C	Podaci o proizvođaču - Apple
7.	00	Podaci o proizvođaču - Apple
8.	02	Tip podataka
9.	15	Duljina podataka - 15 bajta
10.	X	<i>Proximity UUID</i> 1. bajt
11.	X	<i>Proximity UUID</i> 2. bajt
12.	X	<i>Proximity UUID</i> 3. bajt
13.	X	<i>Proximity UUID</i> 4. bajt
14.	X	<i>Proximity UUID</i> 5. bajt
15.	X	<i>Proximity UUID</i> 6. bajt
16.	X	<i>Proximity UUID</i> 7. bajt
17.	X	<i>Proximity UUID</i> 8. bajt
18.	X	<i>Proximity UUID</i> 9. bajt
19.	X	<i>Proximity UUID</i> 10. bajt
20.	X	<i>Proximity UUID</i> 11. bajt
21.	X	<i>Proximity UUID</i> 12. bajt
22.	X	<i>Proximity UUID</i> 13. bajt
23.	X	<i>Proximity UUID</i> 14. bajt
24.	X	<i>Proximity UUID</i> 15. bajt
25.	X	<i>Proximity UUID</i> 16. bajt
26.	X	<i>Major value</i> 1. bajt
27.	X	<i>Major value</i> 1. bajt
28.	X	<i>Minor value</i> 1. bajt
29.	X	<i>Minor value</i> 2. bajt
30.	X	RSSI na udaljenosti od 1m

X - vrijednosti ovise o postavkama konkretnog odašiljača

Izuvez trećeg bajta sa zastavicama LE<sup>2</sup> i BR/EDR<sup>3</sup>, prvih devet bajtova svakog pa-

<sup>2</sup>Low Energy

<sup>3</sup>Bluetooth Radio/Enhanced Data Rate

keta su jednaki kod svakog odašiljača. Nakon toga slijedi šesnaest bajtova koji čine *Proximity UUID*<sup>4</sup> parametar, dva bajta koji čine *Major value* parametar, dva bajta koja čine *Minor value* parametar te posljednji, trideseti, bajt koji sadrži RSSI (mjera jakosti primljenog signala, engl. *received signal strength indicator*) izmjerena na udaljenosti od jednog metra.

Konfigurabilni parametri *Proximity UUID*, *Major value* i *Minor value* koriste se za identifikaciju pojedinog odašiljača. Preporuča se korištenje tih vrijednosti na način da *Proximity UUID* bude nekakav globalni identifikator, a *Major value* i *Minor value* specifičniji identifikatori. Konkretno, *Proximity UUID* može biti oznaka konkretnog trgovačkog lanca, *Major value* oznaka konkretnе trgovine, dok *Minor value* može biti oznaka konkretnе kategorije u trgovini.

Posljednji bajt *advertising* paketa sadrži RSSI izmjerena na udaljenosti od jednog metra od odašiljača. On se često koristi kod određivanja udaljenosti od odašiljača. Pošto jakost signala na nekoj udaljenosti ovisi o okolini u kojoj je uređaj postavljen, radi pouzdanijeg određivanja relativne udaljenosti od odašiljača preporuča se prije početka korištenja odašiljača izmjeriti RSSI na udaljenosti od jednog metra te pohraniti tu vrijednost u sam odašiljač.

Za određivanje lokacije korišteni su Kontakt.io<sup>5</sup> i Bluecats<sup>6</sup> iBeacon odašiljači. U nastavku je navedena specifikacija odašiljača Kontakt.io, dok za vrijeme pisanja rada detaljna specifikacija odašiljača Bluecats nije uopće objavljena. Iz tog razloga su se kod većeg dijela istraživanja koristili odašiljači Kontakt.io jer su vrijednosti nekih parametara odašiljača Bluecats nepoznate. Npr. jakost odašiljanja signala (engl. *transmission power level*) se može postaviti na vrijednosti poput *loud* i *silent*, ali numeričke vrijednosti koju ti izrazi opisuju su nepoznati stoga je taj parametar neprimjeniv.

### 3.1. Odašiljač Kontakt.io Beacon

Glavne značajke uređaja su odašiljanje podatkovnih paketa korištenjem tehnologije BLE te kompatibilnost sa svim uređajima koji podražavaju Bluetooth 4.0. Uz to, uređaji se mogu jednostavno klonirati i nadograditi, imaju visoku razinu sigurnosti te malu energetsku potrošnju.

Pojedini uređaj ima nekoliko konfigurabilnih parametara: *Proximity UUID*, *major* i

---

<sup>4</sup>*Universally unique identifier*

<sup>5</sup><http://kontakt.io>

<sup>6</sup><http://www.bluecats.com>

*minor* vrijednosti, ime uređaja, jakost odašiljanja signala te interval odašiljanja poruka. Uređaj napaja jedna CR2477 baterija s kojom uređaj može konstantno raditi više od 24 mjeseca [6]. Pri standardnoj (tvornički podešenoj) snazi odašiljanja iznosa -4 dBm, mobilni uređaj (ili bilo koji drugi uređaj sa podruškom za BLE) može prepoznati odašiljač do oko šest metara udaljenosti, dok na najjačoj snazi odašiljanja iznosa 4 dBm, odašiljač se može prepoznati i na udaljenosti većoj od dvadeset metara.

### 3.1.1. Tehnička specifikacija

Dimenzije odašiljača su  $55\text{ mm} \times 55\text{ mm} \times 15\text{ mm}$  te on teži svega 23 grama. Vanjski okvir je od ABS<sup>7</sup> plastike, a uređaj napaja zamjenjiva CR2477 baterija s kojom odašiljač može raditi više od dvije godine. Prema [7] vrijeme rada može se dodatno povećati na čak šest godina ukoliko se smanje jakost odašiljanja i interval odašiljanja poruka.

Ostale karakteristike odašiljača su:

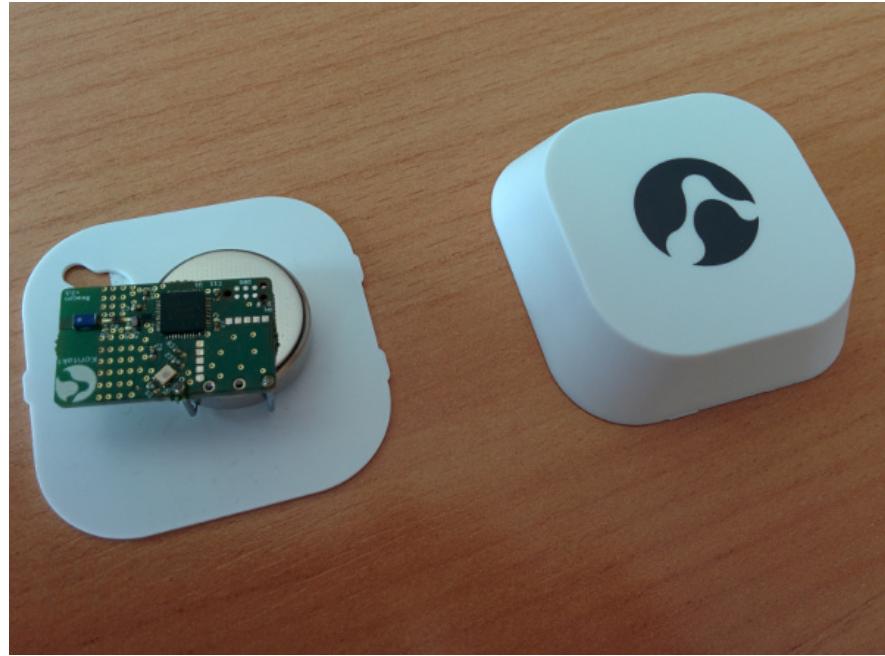
- Nordic Semiconductors Bluetooth Smart multiprotokol SOC<sup>8</sup> IC<sup>9</sup>
- 32-bitni ARM Cortex M0 procesor
- 256 kB flash memorije
- 16 kB RAM memorije
- Jakost odašiljanja u rasponu od -30 dBm do +4 dBm
- Dozvoljene brzine prijenosa: 250 kB/s, 1 M/s i 2 M/s

---

<sup>7</sup>Akrylonitril butadien stiren, izdržljiva plastika koja je pogodna za recikliranje

<sup>8</sup>sustav na čipu (engl. *System on a chip*)

<sup>9</sup>integrirani krug (engl. *integrated circuit*)



**Slika 3.1:** Kontakt.io iBeacon odašiljač

Dozvoljene vrijednosti jakosti odašiljanja su: -30, -20, -16, -12, -8, -4, 0 i 4 dBm.

### 3.1.2. Struktura paketa

Uređaj odašilje dvije vrste paketa podataka: *advertising* i *scan response*.

Tijekom rada uređaj kontinuirano odašilje *advertising* pakete i na taj obavještava okolne uređaje o svojoj prisutnosti. Drugi tip paketa, *scan response* paket, šalje se odmah nakon *advertising* paketa i sadrži dodatne informacije o odašiljaču, poput imena odašiljača, stanja baterije i slično.

Struktura *advertising* paketa odgovara strukturi iBeacon *advertising* paketa navedenog u tablici 3.1, dok je struktura *scan response* paketa navedena u tablici 3.2.

**Tablica 3.2:** Struktura *scan response* paketa

Bajt	Vrijednost	Opis
1.	08*	Duljina podataka - 8 bajta
2.	09	Tip podatka - Lokalno ime uređaja
3.	4B*	Lokalno ime uređaja
4.	6F*	Lokalno ime uređaja
5.	6E*	Lokalno ime uređaja
6.	74*	Lokalno ime uređaja
7.	61*	Lokalno ime uređaja
8.	6B*	Lokalno ime uređaja
9.	74*	Lokalno ime uređaja
10.	02	Duljina podataka - 2 bajta
11.	0A	Tip podatka - Jakost odašiljanja signala
12.	F4	Jakost odašiljanja signala
13.	0A	Duljina podataka - 10 bajta
14.	16	Tip podatka - service data
15.	0D	<i>Service UUID</i> 1. bajt
16.	D0	<i>Service UUID</i> 2. bajt
17.	X	Identifikator odašiljača 1. bajt
18.	X	Identifikator odašiljača 2. bajt
19.	X	Identifikator odašiljača 3. bajt
20.	X	Identifikator odašiljača 4. bajt
21.	X	<i>Firmware version</i> 1. bajt
22.	X	<i>Firmware version</i> 2. bajt
23.	X	Razina baterije

X - vrijednosti ovise o postavkama konkretnog odašiljača

\* - varijabilne duljine, ovisi o duljini postavljenog imena odašiljača (max. 15 bajtova)

## 4. Utvrđivanje relativne i absolutne lokacije u zatvorenom prostoru

Tehnologije navigacije i pozicioniranja koje se oslanjaju na udaljene satelite (poput tehnologija GPS i GNSS) nisu pogodne za korištenje u zatvorenim prostorima iz razloga što njihove signale apsorbiraju i reflektiraju krovovi, zidovi i ostali objekti u okolini. Iz istih razloga određivanje lokacije preko mobilnih signala, odnosno preko radio tornjeva nije moguće.

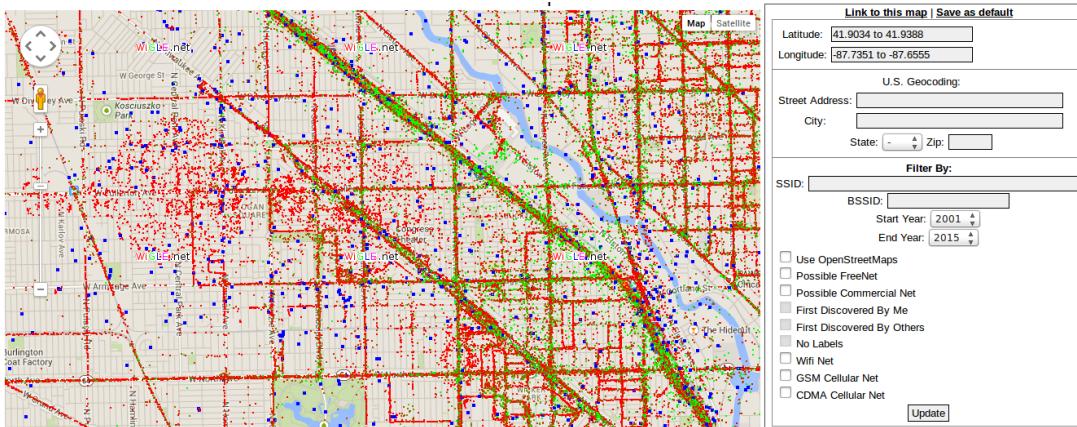
Stoga su za utvrđivanje lokacije u zatvorenom prostoru potrebni sasvim novi i drugačiji pristupi problemu te shodno tome, sasvim nove metode određivanja lokacije. Zbog velikog porasta pristupačnosti i popularnosti pametnih telefona, posljednjih nekoliko godina sve je veća potražnja za pouzdanim rješenjem problema. Kako ne postoji nikakav *de facto* standard, gotovo sva ponuđena rješenja su međusobno različita i koriste cijeli niz različitih tehnologija, od optičkih (npr. kamera uređaja) i radio (npr. signali obližnje bežične mreže) skroz do akustičnih (npr. akustični sljednici koji se često koriste sustavima virtualne stvarnosti) tehnologija.

Vjerojatno najznačajniji uspjeh je postignut sa praćenjem signala kojega odašilje obližnja bežična Wi-fi mreža. Velika prednost ove metode je značajan porast bežičnih pristupnih točaka na koje se mobilni i drugi uređaji mogu spojiti. Da bi se ovom metodom odredila lokacija uređaja potrebno je na neki način mapirati dotičnu pristupnu točku te zatim na temelju jakosti primljenog signala utvrditi poziciju mobilnog uređaja u odnosu na nju. Parametri mapiranja uključuju absolutnu poziciju, SSID (engl. *service set identification*) i MAC adresu (engl. *media access control address*) pristupne točke (tj. WLAN uređaja). Neke od poznatijih web aplikacija poput WeFi<sup>1</sup> i WiGLE<sup>2</sup> sadrže više od sto milijuna mapiranih bežičnih pristupnih točaka.

---

<sup>1</sup><http://www.wifi.com/maps/>

<sup>2</sup><https://wigle.net>



**Slika 4.1:** Prikaz mapiranih bežičnih pristupnih točaka preko sučelja WiGLE web aplikacije

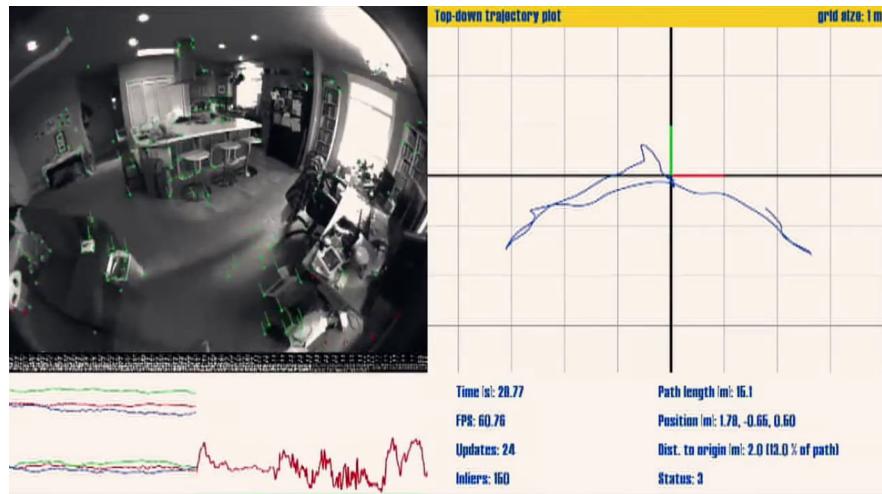
Uz navigaciju i pozicioniranje pomoću Wi-fi mreža, dosta su popularne metode koje se služe tehnikama proširene i virtualne stvarnosti te tehnikama strojnog učenja, raspoznavanja uzorka i računalnog vida.

Jedna od popularnijih tehnika proširene (engl. *augmented*) i virtualne stvarnosti je korištenje posebnih oznaka (markera) kojega kamera mobilnog uređaja može prepoznati u prostoru.

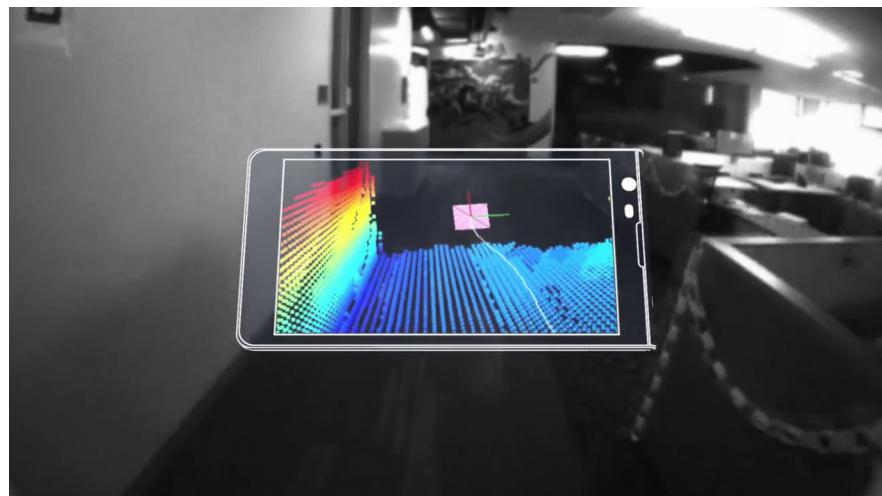
Iz područja strojnog učenja, raspoznavanja uzorka i računalnog vida vjerojatno najistaknutiji i najambiciozni projekt je projekt Tango<sup>3</sup> kojega razvija tehnološki gigant Google. Cilj projekta je ponuditi rješenje koje će korištenjem kamere i ostalih senzora mobilnog uređaja nuditi pomoć pri navigaciji, pozicioniranju i prepoznavanju okoline, odnosno 3D mapiranje, u zatvorenom i otvorenom prostoru. Unatoč tome što je projekt još u ranoj fazi razvoja, razvojni tim je ostvario vidljiv napredak te inicijalni rezultati upućuju na preciznost od jednog centimetra. Uz to, prvi prototipovi mobilnih uređaja i tableta već su dostupni partnerima koji sudjeluju u razvoju i na internetu se polako objavljaju nove i zanimljive informacije o projektu.

---

<sup>3</sup><https://www.google.com/atap/projecttango/>



**Slika 4.2:** Google Project Tango - prepoznavanje predmeta u okolini [8]



**Slika 4.3:** Google Project Tango - prikaz mapirane okoline [8]

U nastavku su opisane relativno nove metode za određivanje lokacije u zatvorenom prostoru koje se temelje na tehnologiji BLE i odašiljačima iBeacon. Za rješavanje problema prvo je potrebno sa što većom preciznošću odrediti relativnu poziciju uređaja u odnosu na odašiljač/odašiljače stoga je glavni fokus istraživanja usmjeren na taj dio. Nakon toga uz poznavanje relativne pozicije uređaja i apsolutne pozicije odašiljača možemo odrediti apsolutni položaj uređaja.

## 4.1. Utvrđivanje relativne udaljenosti od odašiljača

Funkcija pomoću koje se može odrediti jakost primljenog signala navedena je u (4.1).

$$P_r = P_t + 20 \log \frac{\lambda}{4\pi} + 10n \log \frac{1}{d} \quad (4.1)$$

$P_t$  predstavlja jakost odašiljanja signala,  $P_r$  jakost primljenog signala,  $d$  udaljenost između prijamnika i odašiljača,  $\lambda$  je valna duljina elektromagnetskog vala, a  $n$  indeks refrakcije (u slobodnom prostoru iznosi 2).

Ukoliko se ista jednadžba zapiše po varijabli  $d$  dobiva se izraz (4.2). Uz poznate vrijednosti svih ostalih parametara taj izraz se može iskoristiti za procjenu udaljenosti između prijamnika i odašiljača. Uz to, zbog relativno niske složenosti dotični izraz je idealan kandidat za ugradnju u mobilnu aplikaciju.

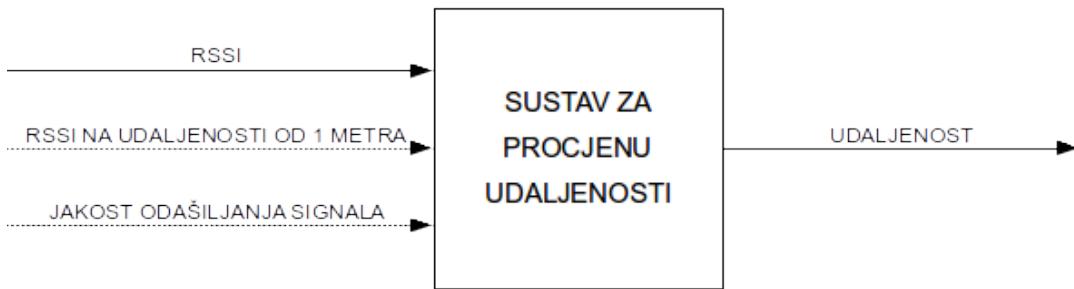
$$d = 10^{-\frac{P_r - P_t - 20 \log \frac{\lambda}{4\pi}}{10n}} \quad (4.2)$$

Problem izraza (4.2) je što ona zanemaruje bilo kakva ometanja signala, odnosno on je primjenjiv samo u idealnim uvjetima, a kao što je prethodno već rečeno, na signal u zatvorenom prostoru utječe (tj. ometa ga) velik broj parametara. Na stabilnost signala koji putuje zatvorenim prostorom najvećim dijelom djeluju apsorpcija i refleksija signala, dok razasiljanje i difrakcija djeluju manjom mjerom.

Zbog refleksije signala dio elektromagnetskog vala se odbija (reflektira) od površinu s kojom se sudario, a zbog apsorpcije signala dio elektromagnetskog vala se apsorbira te se val nastavlja gibati sa prigušenjem. Smetnje uzrokovane refleksijom mogu se smanjiti povećanjem frekvencije, dok apsorpcija ovisi o dubini prodiranja i udaljenosti granice između dva sredstva [2].

U zatvorenom prostoru smetnje koje uzrokuju prethodno opisani faktori su često prevelike i u velikoj većini slučajeva na to se ne može utjecati. Zaključak je da procjena udaljenosti temeljena na (4.2) nije precizna na tako nestabilnom signalu stoga je za rješavanje problema potreban drugačiji pristup.

Sustav za procjenu udaljenosti od odašiljača iBeacon može se opisati kao što je prikazano na slici 4.4.



**Slika 4.4:** Sustav za procjenu udaljenosti

Ulaz u sustav procjene udaljenosti je RSSI, odnosno jakost primljenog signala, optionalni ulazni parametri su jakost odašiljanja signala te RSSI izmjerena na udaljenosti od jednog metra od odašiljača, a izlaz sustava je procjena udaljenosti izražena u metrima.

Sustav procjene udaljenosti može biti funkcija (4.2) ili bilo koja druga funkcija.

Kako na signal djeluje velik broj vanjskih smetnji, predloženi sustav će morati biti otporan na šum u podacima, a također će morati biti i relativno brz te imati nisku složenost zbog toga što mu je primarna namjena korištenje u mobilnim uređajima i tabletima ograničene procesorske snage.

Kao što je navedeno u poglavlju o odašiljačima iBeacon, Apple je za procjenu udaljenosti od odašiljača namijenio tri izraza: neposredna blizina (*immediate*), mala (*near*) i velika *far* udaljenost. Zajedno s tim u radno okruženje Apple Core Location dodana je funkcija koja na temelju jakosti primljenog signala (RSSI) i jakosti signala na udaljenosti jednog metra od odašiljača (trideseti bajt *advertising* paketa iBeacon odašiljača) određuje relativnu udaljenost od odašiljača. Sam izvorni tekst funkcije nije javno dostupan i nije poznata metoda koju su Appleovi inženjeri upotrijebili, no **ne preporučuju** da se ona koristi u situacijama gdje je potreban precizan rezultat, odnosno precizna procjena udaljenosti. Stoga se nameće zaključak kako ni Appleovi inženjeri nisu uspjeli zaobići prethodno spomenute smetnje koje djeluju na signal.

Inženjeri američke kompanije Radius Networks su za vrijeme izrade svoje neslužbene slobodno dostupne Android iBeacon biblioteke<sup>4</sup> pristupili problemu procjene udaljenosti tako da su napravili veliki broj RSSI mjerenja na poznatim udaljenostima i rezultate opisali funkcijom najbolje sličnosti [5]. Dotična funkcija opisana je izrazom (4.3).

<sup>4</sup><https://github.com/RadiusNetworks/android-ibeacon-service>

$$f(x, y) = \begin{cases} 0.89976 \frac{x^{7.7095}}{y} + 0.111 & \text{ako } \frac{x}{y} \geq 1 \\ \frac{x^{10}}{y} & \text{ako } 0 < \frac{x}{y} < 1 \\ -1 & \text{ako } x = 0 \end{cases} \quad (4.3)$$

Parametar  $x$  je RSSI primljenog signala, dok  $y$  predstavlja RSSI na udaljenosti od jednog metra od odašiljača.

Njihovo rješenje može se isprobati tako da se funkciju (4.3) ugradi u vlastitu aplikaciju, a može se iskoristiti i njihova besplatna Android aplikacija koja je dostupna na Google Play Store<sup>5</sup>.

Empirijskim ispitivanjem aplikacije primjetio sam kako i za male pomake u iznosu od nekoliko desetaka centimetara detektirani pomak aplikacije nerijetko bio u iznosu većem od deset metara. Stoga se može zaključiti kako ni rješenje koje su ponudili inženjeri Radius Networksa nije pouzdano, a ni primjenjivo na stvarnim problemima. Bez obzira na to, njihova Android iBeacon biblioteka, objašnjenja na službenim stranicama te rasprave njihovog inženjera David Younga na stranicama Stackoverflowa su izuzetno pomogle oko shvaćanja cjelokupne problematike određivanja lokacije korištenjem iBeacon odašiljača.

Za modeliranje sustava može se iskoristiti umjetna neuronska mreža (engl. *artificial neural network*). Smatram da su neuronske mreže dobar model za rješavanje problema iz razloga što one mogu uspješno rješavati zahtjevne probleme gdje je tražena funkcija nepoznata, a uz to otporne su otporne na šum u podacima.

Neuronska mreža će se trenirati na prethodno prikupljenim podacima. Prve tri skupine podataka uzorokovane su na na udaljenostima od 50 cm do 8 m s korakom od 50 cm, po dvije stotine uzorka za svaku udaljenost. Važnija svojstva sve tri skupine uzorka prikazane su u tablicama 4.1, 4.2 i 4.3.

---

<sup>5</sup><https://play.google.com/store/apps/details?id=com.radiusnetworks.ibeaconlocate>

**Tablica 4.1:** Prva skupina uzoraka

Udaljenost	$\mu_{RSSI}$	$\sigma_{RSSI}$	$min_{RSSI}$	$max_{RSSI}$
0.5	-81.075	2.426	-88	-74
1.0	-80.320	2.799	-86	-76
1.5	-75.575	3.298	-80	-70
2.0	-73.990	2.558	-79	-68
2.5	-78.725	0.907	-81	-74
3.0	-82.470	4.053	-91	-74
3.5	-81.685	2.892	-87	-77
4.0	-83.115	1.903	-88	-77
4.5	-91.125	3.057	-98	-85
5.0	-88.705	2.198	-93	-83
5.5	-88.935	3.487	-97	-83
6.0	-92.450	2.399	-99	-87
6.5	-93.610	1.691	-99	-90
7.0	-	-	-	-
7.5	-	-	-	-
8.0	-	-	-	-

Podatke prve skupine uzoraka poslao je kontakt.io odašiljač sa jakosti odašiljanja od  $-4$  dBm. Na udaljenosti većoj od 6.5 metara mobilni uređaj nije uspio detektirati dotični odašiljač, stoga su podaci za te udaljenosti nedostupne. U tablici se može uočiti kako je standardna devijacija za gotovo svaku grupu uzoraka poprilično velika (devijacija grupe na udaljenosti od tri metra iznosi čak 4.053). Uzrok tako velikoj standardnoj devijaciji su prethodno opisane smetnje koje utječu na signal u zatvorenom prostoru.

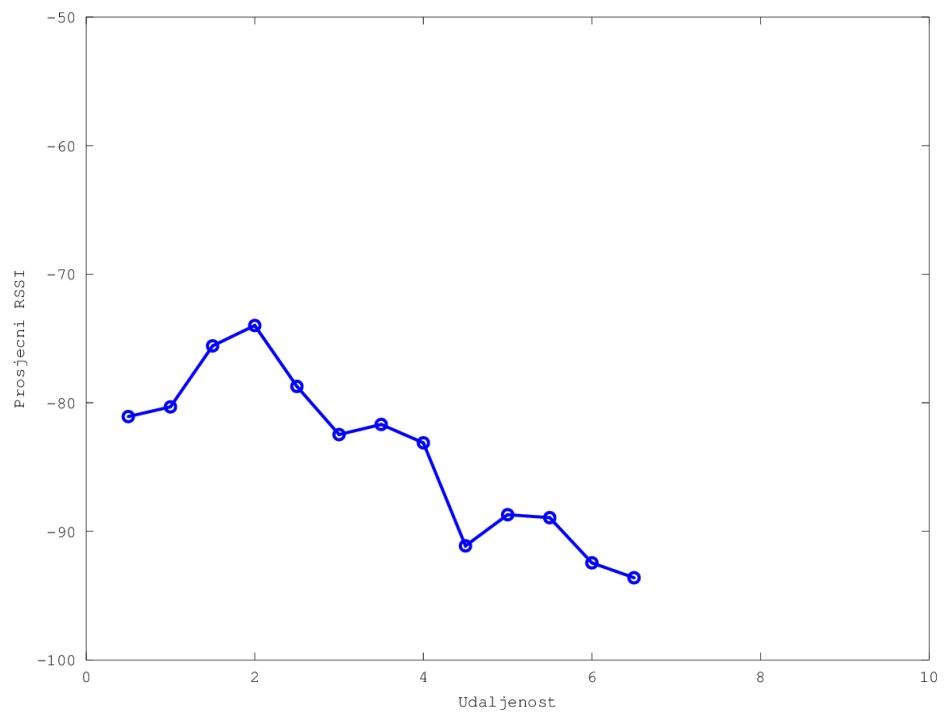
**Tablica 4.2:** Druga skupina uzoraka

Udaljenost	$\mu_{RSSI}$	$\sigma_{RSSI}$	$min_{RSSI}$	$max_{RSSI}$
0.5	-65.165	2.546	-76	-56
1.0	-66.250	2.316	-73	-59
1.5	-62.695	4.3238	-69	-57
2.0	-58.750	1.982	-62	-53
2.5	-63.595	0.875	-66	-59
3.0	-67.810	2.619	-73	-62
3.5	-71.620	3.537	-80	-65
4.0	-68.72	2.000	-73	-65
4.5	-74.975	3.065	-81	-68
5.0	-72.445	3.485	-78	-65
5.5	-79.920	6.883	-93	-68
6.0	-82.175	5.896	-98	-71
6.5	-74.085	2.977	-80	-68
7.0	-71.700	2.858	-77	-65
7.5	-76.065	2.241	-80	-71
8.0	-78.295	4.926	-91	-71

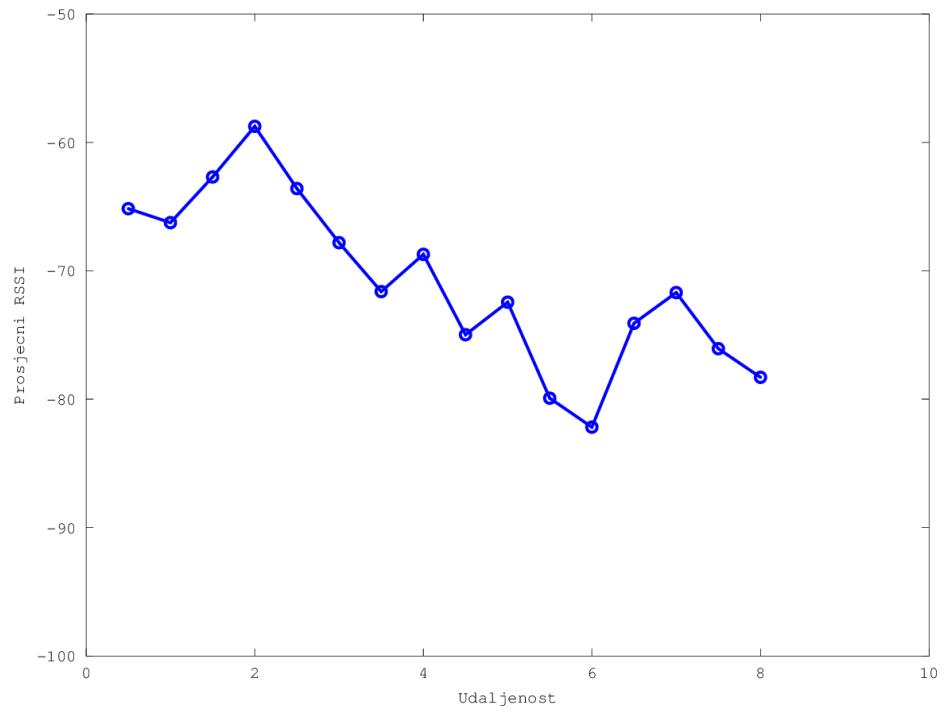
**Tablica 4.3:** Treća skupina uzoraka

Udaljenost	$\mu_{RSSI}$	$\sigma_{RSSI}$	$min_{RSSI}$	$max_{RSSI}$
0.5	-89.550	3.526	-98	-83
1.0	-90.895	2.101	-96	-86
1.5	-91.350	3.162	-101	-86
2.0	-91.695	2.134	-99	-86
2.5	-89.630	1.583	-99	-86
3.0	-93.385	2.156	-100	-88
3.5	-	-	-	-
4.0	-	-	-	-
4.5	-	-	-	-
5.0	-	-	-	-
5.5	-	-	-	-
6.0	-	-	-	-
6.5	-	-	-	-
7.0	-	-	-	-
7.5	-	-	-	-
8.0	-	-	-	-

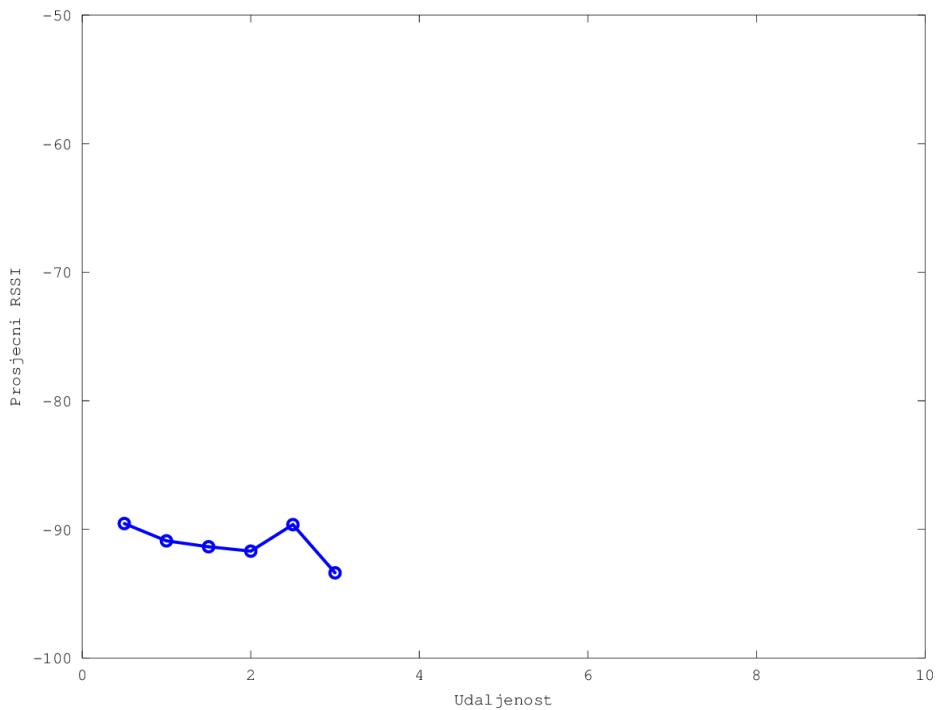
Podatke druge skupine uzoraka posao je odašiljač Kontakt.io sa jakosti odašiljanja od 4 dBm (najveća jakost odašiljanja odašiljača). Podatke treće skupine uzoraka posao je Bluecats odašiljač sa jakosti odašiljanja postavljenom na parametar *Talk* (numerička vrijednost koju dotični parametar predstavlja nije poznata). Na udaljenosti većoj od tri metra mobilni uređaj nije uspio detektirati treći odašiljač, stoga su podaci za te udaljenosti nedostupne. Promatranjem tablica 4.2 i 4.3 možemo primijetiti kako je standardna devijacija visoka i kod gotovo svake grupe druge i treće skupine uzoraka. U na slikama 4.5, 4.6 i 4.7 prikazan je kako se prosječni RRSI mijenja sa povećanjem udaljenosti za svaku skupinu uzoraka.



**Slika 4.5:** Grafički prikaz srednje vrijednosti primljenog signala prve skupine uzoraka



**Slika 4.6:** Grafički prikaz srednje vrijednosti primljenog signala druge skupine uzoraka



**Slika 4.7:** Grafički prikaz srednje vrijednosti primljenog signala treće skupine uzoraka

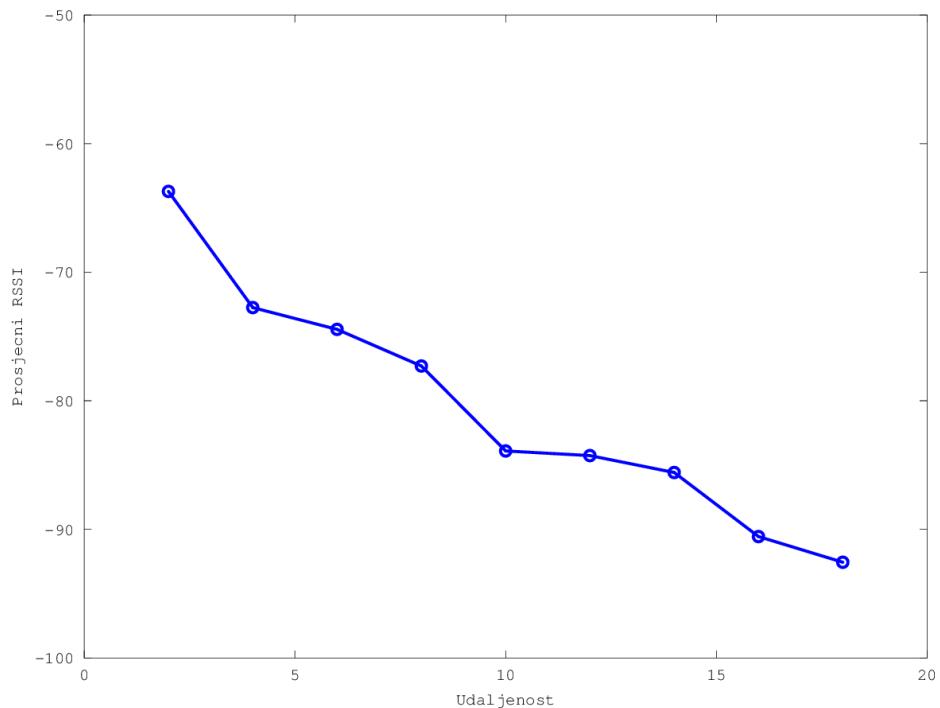
U teoriji RSSI bi trebao monotono opadati sa povećanjem udaljenosti, no na grafovima se može lako uočiti kako to ovdje nije slučaj. Uzrok tomu su već spomenuta ometanja signala.

Posljedično, određivanje precizne udaljenosti na temelju RSSI-a nije moguće jer za primljeni RSSI sustav procjene udaljenosti ne može jednoznačno odrediti na kojoj se udaljenosti odašiljač nalazi. Na primjer, ako pogledamo drugu skupinu uzoraka (tablica 4.2 i slika 4.6) uočava se kako je srednja vrijednost signala podjednaka na udaljenostima od 3.5, 5 i 7 metara, stoga ako je ulazni RSSI jednak -72 dBm, sustav neće moći jednoznačno procijeniti udaljenost (tj. radi li se o udaljenosti od 3.5, 5 ili 7 metara).

Unatoč tome, u nastavku će se pokušati procijeniti udaljenost sa preciznošću od dva metra. Za to je prikupljena nova skupina podataka koji su uzorkovani na udaljenostima od 2 m do 18 m sa korakom od 2 m, po dvije stotine uzoraka za svaku udaljenost. Prikupljene uzorke (tj. RSSI) poslao je odašiljač Kontakt.io sa jakosti odašiljanja od 4 dBm (najveća jakost odašiljanja odašiljača). U tablici 4.4 prikazana su važnija svojstva nove, četvrte, skupine uzoraka, a na slici 4.8 je prikazano kretanje srednje vrijednosti primljenog signala za svaku udaljenost.

**Tablica 4.4:** Četvrta skupina uzoraka

Udaljenost	$\mu_{RSSI}$	$\sigma_{RSSI}$	$min_{RSSI}$	$max_{RSSI}$
2	-63.720	1.371	-66	-59
4	-72.755	4.294	-80	-67
6	-74.450	6.894	-89	-65
8	-77.300	2.112	-83	-71
10	-83.905	4.691	-92	-77
12	-84.265	3.489	-95	-77
14	-85.580	2.430	-90	-81
16	-90.560	2.846	-97	-85
18	-92.555	1.787	-98	-88



**Slika 4.8:** Četvrta skupina uzoraka

Na slici je očito kako prosječni RSSI monotono opada sa povećanjem udaljenosti, no iako je to dobar znak, ono ne garantira da će se udaljenost moći jednoznačno procijeniti jer je standardna devijacija pri svakoj udaljenosti poprilično velika.

Neuronska mreža će se trenirati nad četvrtom skupinom uzoraka. Neuronska mreža se sastoji od tri sloja:

- ulazni sloj - ne obavlja nikakvu funkciju već preslikava dovedene ulazne podatke i čini ih dostupnima ostatku mreže
- skriveni sloj - sastoji se od devet neurona (broj neurona je jednak broju grupa podataka), svaki neuron na svoj ulaz dobiva isključivo vrijednosti od neurona iz ulaznog sloja; prijenosna funkcija svakog neurona je sigmoidalna funkcija
- izlazni sloj - sastoji se od jednog neurona koji na ulaze dobiva vrijednosti izlaza neurona skrivenog sloja

Pojedini neuron skrivenog sloja ima dva parametra (težina), dok je broj parametara izlaznog neurona jednak broju neurona skrivenog sloja i ima dodatni parametar koji čini prag okidanja(engl. *threshold*). Neuronska mreža ima ukupno 28 parametara.

Neuronsku mrežu će trenirati genetski algoritam iz razloga što su evolucijski algoritmi kao univerzalni optimizacijski algoritmi primjenjivi na evoluciju parametara raznih arhitektura neuronskih mreža stoga nije potrebno izvoditi algoritme koji su prilagođeni pojedinim arhitekturama [11].

Parametri neuronske mreže su pohranjeni u vektor realnih brojeva stoga će se vektor realnih brojeva koristiti i za prikaz rješenja genetskog algoritma. Korišten je genetski genetski algoritam, a od genetskih operatora korišteno je  $BLX-\alpha$  križanje, turnirska selekcija te normalna mutacija. Funkcija za evaluaciju rješenja je:

$$fitness = \frac{1}{N} \sum_{i=1}^N (o_i - y_i)^2 \quad (4.4)$$

gdje je  $N$  ukupni broj uzoraka,  $o_i$  je izlaz neuronske mreže za uzorak  $i$ , a  $y_i$  označava ispravni rezultat. Cilj genetskog algoritma je pronaći parametre tako da *fitness* bude što manji, odnosno radi se o minimizacijskom problemu.

Nakon treniranja neuronske mreže i provedenog ispitivanja zaključujem kako rezultat neuronske mreže nije zadovoljavajući. Unatoč tome što se tijekom treniranja, odnosno evolucije parametara, greška postupno smanjivala, testiranjem neuronske mreže pokazalo se da nije naučila podatke kao što se očekivalo. Uzrok tomu je prevelika standardna devijacija stoga neuronska mreža ne može ispravno procjenjivati udaljenost. Smatram da bi se sustav kombinacije neuronske mreže i sustava neizrazitog zaključivanja možda mogao pokazati kao precizniji sustav procjene, no to nadilazi temu ovog rada.

Iz svega ovoga zaključujem kako je zbog djelovanja raznih smetnji koje djeluju na

signal koji putuje zatvorenim prostorom, određivanje lokacije u zatvorenom prostoru pomoću odašiljača temeljenog na tehnologiji BLE poprilično nepouzdano.

## 4.2. Utvrđivanje absolutne lokacije

Unatoč tome što u prethodnom potpoglavlju nije pronađeno zadovoljavajuće rješenje za određivanje relativne udaljenosti od odašiljača, u nastavku je opisan postupak trilateracije pomoću kojega se može odrediti absolutna lokacija u prostoru.

Trilateracija je postupak određivanja relativne ili absolutne lokacije objekta u prostoru pomoću mjerena udaljenosti od objekata čija je lokacija poznata. Za određivanje lokacije nekog objekta u 2D prostoru potrebne su koordinate tri poznata objekta i njihova udaljenost od objekta čija se lokacija pokušava odrediti. Iz tih podataka mogu postaviti sljedeće tri jednadžbe:

$$A : (x - x_0)^2 + (y - y_0)^2 = r_0^2 \quad (4.5)$$

$$B : (x - x_1)^2 + (y - y_1)^2 = r_1^2 \quad (4.6)$$

$$C : (x - x_2)^2 + (y - y_2)^2 = r_2^2 \quad (4.7)$$

gdje su  $(x, y)$  koordinate objekta čija je lokacija nepoznata,  $(x_0, y_0)$ ,  $(x_1, y_1)$  i  $(x_2, y_2)$  koordinate tri objekta čija je lokacija poznata, a  $r_0$ ,  $r_1$  i  $r_2$  udaljenost pojedinog poznatog objekta od nepoznatog. Proširenjem dotičnih jednadžbi dobivaju se sljedeće tri jednadžbe:

$$A : x^2 - 2xx_0 + x_0^2 = r_0^2 \quad (4.8)$$

$$B : x^2 - 2xx_1 + x_1^2 = r_1^2 \quad (4.9)$$

$$C : x^2 - 2xx_2 + x_2^2 = r_2^2 \quad (4.10)$$

Ako se izračuna  $A - B$  dobiva se sljedeće:

$$-2x(x_0 - x_1) + (x_0^2 + x_1^2) - 2y(y_0 - y_1) + (y_0^2 - y_1^2) = (r_0^2 - r_1^2) \quad (4.11)$$

Izlučivanjem varijable  $y$  dobiva se:

$$y = \frac{-2x(x_0 - x_1) + (x_0^2 + x_1^2) + (y_0^2 - y_1^2) - (r_0^2 - r_1^2)}{-2(y_0 - y_1)} \quad (4.12)$$

Ako se izračuna  $B - C$  dobiva se sljedeće:

$$-2x(x_1 - x_2) + (x_1^2 + x_2^2) - 2y(y_1 - y_2) + (y_1^2 - y_2^2) = (r_1^2 - r_2^2) \quad (4.13)$$

Izlučivanjem varijable  $y$  dobiva se:

$$y = \frac{-2x(x_1 - x_2) + (x_1^2 + x_2^2) + (y_1^2 - y_2^2) - (r_1^2 - r_2^2)}{-2(y_1 - y_2)} \quad (4.14)$$

Izjednačavanjem  $y$  varijabli iz (4.12) i (4.14) dobiva se jednadžba samo po  $x$  varijabli iz koje se zatim dotična varijabla može izračunati. Nakon toga se uvrštavanjem  $x$  varijable u (4.12) ili (4.14) jednadžbu može izračunati nepoznata  $y$  koordinata.

## 5. Radno okruženje Apache Cordova

Radno okruženje Apache Cordova je skup aplikacijsko-programskih sučelja (engl. *application programming interface, API*) koji omogućavaju da razvijatelj mobilnih aplikacija pristupa osnovnim funkcijama mobilnoga uređaja, poput kamere, sustava za pohranu podataka i telefonskog imenika preko JavaScript jezika. U kombinaciji sa radnim okruženjima poput Sencha Touch, Dojo Mobile i Ionic aplikacije za pametne telefone mogu se razvijati korištenjem samo HTML, CSS i JavaScript programskog jezika.

Korištenjem Apache Cordove programer je oslobođen pisanja aplikacija u nativnim jezicima uređaja (npr. Java za Android, Objective-C za iOS), već se koriste isključivo prethodno spomenute web tehnologije. Bez obzira na to što aplikacije nisu napisane u nativnim jezicima, Apache Cordova aplikacije se prevode i pakiraju pomoću biblioteke za razvoj programske potpore (engl. *software development kit, SDK*) željene platforme stoga se aplikacije mogu postaviti na trgovine aplikacija (engl. *app store*) dotične platforme.

Cordova nudi skup uniformnih JavaScript biblioteka čije funkcije programer može pozivati. One imaju podršku za povezivanje sa specifičnim platformama. Cordova je trenutno dostupna za sljedeće platforme: Android, iOS, Blackberry, Windows Phone, Palm WebOS, Bada i Symbian.

Za pristup Bluetooth Low Energy funkcijama mobilnog uređaja korištena je **Cordova BLE** biblioteka<sup>1</sup>, dok je oko dvije stotine dodatnih i besplatnih biblioteka dostupno na službenom repozitoriju<sup>2</sup>.

Službene Cordova JavaScript biblioteke koje održava Cordova tim su:

---

<sup>1</sup><https://github.com/evothings/cordova-ble>

<sup>2</sup><http://plugins.cordova.io>

## **Battery Status**

Sadrži funkcije za nadgledanje stanja baterije uređaja.

## **Camera**

Sadrži funkcije za pristup kamери uređaja.

## **Contacts**

Sadrži funkcije za pristup telefonskom imeniku uređaja.

## **Device**

Sadrži funkcije za pristup specifičnim informacijama uređaja (npr. ime uređaja, operacijski sustav).

## **Device Motion (Accelerometer)**

Sadrži funkcije za pristup senzoru ubrzanja (akcelerometar).

## **Device Orientation (Compass)**

Sadrži funkcije za pristup kompasu uređaja.

## **Dialogs**

Sadrži funkcije za korištenje sustava obavijesti uređaja.

## **FileSystem**

Sadrži funkcije za korištenje datotečnog sustava uređaja.

## **FileTransfer**

Sadrži funkcije za pristup sustavu za prijenos datoteka.

## **Geolocation**

Sadrži funkcije za pristup geolokacijskom sustavu.

## **Globalization**

Sadrži funkcije koje omogućavaju različite reprezentacije objekata ovisno o postavkama lokacije uređaja.

## **InAppBrowser**

Omogućava otvaranje URL-ova u novoj instanci web-preglednika uređaja.

## **Media**

Sadrži funkcije za snimanje i reprodukciju audio datoteka.

## **Media Capture**

Sadrži funkcije za snimanje audio i video datoteka.

### **Network Information (Connection)**

Omogućava pristup informacijama o stanju mreže uređaja.

### **Splashscreen**

Sadrži funkcije za manipuliranje početnog zaslona aplikacije.

### **Vibration**

Omogućava korištenje mehanizma za vibriranje uređaja.

## **5.1. Biblioteka Cordova BLE**

Neslužbena biblioteka Cordova BLE omogućava hibridnim mobilnim aplikacijama koje se temelje na Cordovi korištenje BLE funkcionalnosti mobilne platforme bez potrebe za pisanjem nativnog programskog kôda. Preko biblioteke moguće je korištenjem JavaScript programskog jezika skenirati obližnje odašiljače temeljene na tehnologiji BLE, spajati se na njih, čitati servise, karakteristike i opisnike te pisati nove vrijednosti karakteristika i opisnika (ukoliko odašiljač to dopušta). Također, biblioteka omogućava da se aplikacija pretplati na obavijesti odašiljača te može ponovno pokrenuti (resetirati) Bluetooth sustav mobilnog uređaja. Ponovno pokretanje Bluetooth sustava osobito je korisno aplikacijama koje se pokreću na Android operacijskom sustavu iz razloga što raniji Android sustavi imaju popriličan broj pogrešaka u svojoj Bluetooth implementaciji. Dobar dio grešaka ispravlja Android 4.4.3 koji je objavljen početkom lipnja 2014. godine, no dotični operacijski sustav imaju imaju samo noviji moderniji uređaji stoga je korištenje funkcije resetiranja nužno kako bi se osigurao ispravan rad mobilne aplikacije i u nešto starijim Android uređajima.

Biblioteka funkcioniра na način da se razvijatelju aplikacije nudi skup uniformnih funkcija za pristup BLE funkcionalnosti mobilnog uređaja preko programskog jezika JavaScript. U pozadini dotične JavaScript funkcije pozivaju funkcije koje su napisane u nativnom jeziku platforme (tj. operacijskog sustava) mobilnog uređaja na kojem je aplikacija pokrenuta. Nadalje, dotične nativne funkcije pristupaju samoj BLE funkcionalnosti uređaja koje u konačnici obave zadani posao. Na kraju, nativne funkcije šalju odgovore JavaScript sloju preko prethodno predanih asinkronih povratnih (engl. *callback*) funkcija. Konkretno, kompletan tok poziva određene funkcije prikazan je na sljedećem primjeru gdje se pokreće traženje obližnjih odašiljača BLE.

Iz našeg JavaScript programa pozivamo funkciju za skeniranje obližnjih uređaja:

```

// povratna funkcija koja se poziva kada se pronade odasiljac
var win = function(device) {
    //ucini nesto sa pronadenim uredajem
};

// povratna funkcija koja se poziva ukoliko dode do pogreske
var fail = function(error) {
    // obrada pogreske
};

// pokretanje skeniranja
evothings.ble.startScan(win, fail);

```

Pozvana funkcija poziva nativnu funkciju i prosljeđuje joj predane povratne funkcije:

```

// ovo je zapravo prethodno pozvana evothings.ble.startScan funkcija
exports.startScan = function(win, fail) {
    exec(win, fail, 'BLE', 'startScan', []);
};

```

Na Android platformi gore navedena `exec` funkcija je zapravo `execute` metoda razreda `BLE` koji nasljeđuje `CordovaPlugin` razred te implementira funkcionalnost `LeScanCallback` sučelja.

```

public class BLE extends CordovaPlugin implements LeScanCallback {
    @Override
    public boolean execute(String action, CordovaArgs args,
        final CallbackContext callbackContext) throws JSONException {
        if ("startScan".equals(action)) {
            // poziv startScan funckije napisane u nativnom programskom
            // jeziku Android platforme
            startScan(args, callbackContext);
            return true;
        } else if (...) {
            ...
        }
    ...
}
...
}

```

Svaka Cordova biblioteka mora naslijediti razred `CordovaPlugin` i nadjačati(engl. *override*) neku od izvršnih metoda, odnosno u ovom slučaju nasljeđuje se `execute` metoda. Sučelje `LeScanCallback` je nužno zbog pristupa BLE funkcionalnosti Android platforme.

Ako funkcija uspješno obradi zadatok pozvat će se povratna funkcija `win`, a ako je došlo do pogreške pozvati će se povratna funkcija `fail`. Obe funkcije su pohranjene u instanci `CallbackContext` razreda.

U vrijeme pisanja rada Cordova BLE biblioteka nudi podršku samo prema iOS i Android operacijskim sustavima.

## 5.2. Radno okruženje Ionic

Ionic je radno okruženje napisano sa HTML, CSS i JavaScript programskim jezikom čiji je cilj olakšati razvoj hibridnih mobilnih aplikacija stoga je ono jako dobar izbor prilikom izrade Cordova aplikacija. Hibridne mobilne aplikacije su aplikacije napravljene korištenjem web tehnologija. Pokreću se unutar posebnog spremnika uređaja te koriste funkcije web preglednika za prikaz HTML sadržaja i izvođenje JavaScript kôda. Ionic je primarno okrenut prema olakšanju izrade korisničkog sučelja, odnosno nudi cijeli niz funkcija koje razvijatelju olakšavaju izradu velikih i složenih mobilnih aplikacija. U pozadini, Ionic koristi danas sve popularnije JavaScript *frontend* radno okruženje AngularJS koje je namjenjeno izradi *single-page* web aplikacija. Korištenjem AngularJS-a u Ionic je dodan cijeli niz direktiva<sup>3</sup>, filtera<sup>4</sup>, servisa<sup>5</sup> i drugih funkcija koje programeru rješavaju cijeli niz problema poput osjetljivog dizajna (engl. *responsive design*), hvatanja raznih korisnikovih interakcija (dodiri s jedim ili više prstiju, povlačenje (engl. *swipe*) stavki, stezanju i širenju stavki (engl. *pinch*) i cijeli niz drugih).

Ionic se sastoji od dva temeljna dijela. Prvi dio čine CSS i Sass datoteke čija je svrha da razvijateljima olakšaju izradu vizualnoga dizajna aplikacije. Drugi dio čine JavaScript datoteke i HTML predlošci koji pojednostavljaju izradu aplikacija složene arhitekture te programerima nude cijeli niz pomoćnih funkcija.

Ionic je relativno novo radno okruženje i u vrijeme pisanja ovog rada je u beta verziji. Unatoč tome, veliki broj funkcija već je implementiran i samo njihovim ko-

---

<sup>3</sup>Posebne oznake DOM elemenata koje obavještavaju AngularJS HTML prevoditelja da transformira elemente ili im doda novo ponašanje.

<sup>4</sup>Funkcije za obradu ili transformaciju podataka koji se prikazuju korisniku. Npr. mogu se koristiti za abecedni prikaz elemenata liste, prikaz teksta isključivo velikim slovom itd.

<sup>5</sup>Jedinstveni objekti (engl. *singleton*) koji se koriste za dijeljenje funkcija i resursa unutar aplikacije.

rištenjem mogu se napraviti velike, složene i vizualno privlačne mobilne aplikacije. Uz to, Ionic je izdan pod MIT licencom i njegov izvorni kod je javno dostupan na službenom Github reponitoriju<sup>6</sup>.

---

<sup>6</sup><https://github.com/driftyco/ionic>

# 6. Praktični rad

U poglavlju o odašiljačima iBeacon spomenuto je kako se tehnologija iBeacon može iskoristiti u gotovo svim situacijama gdje se sadržaj mijenja ovisno o kontekstu prostora, npr. trgovinama, muzejima, bolnicama i sličnim objektima.

U nastavku poglavlja opisano je kako se tehnologija iBeacon može integrirati u trgovine.

Za integraciju je potrebno imati postavljene odašiljače iBeacon u prostoru, mobilni uređaj sa podrškom za BLE i instaliranom aplikacijom koja može prepoznati BLE, odnosno iBeacon, odašiljače te poslužiteljsku aplikaciju s koje će mobilni uređaj dohvaćati relevantne podatke.

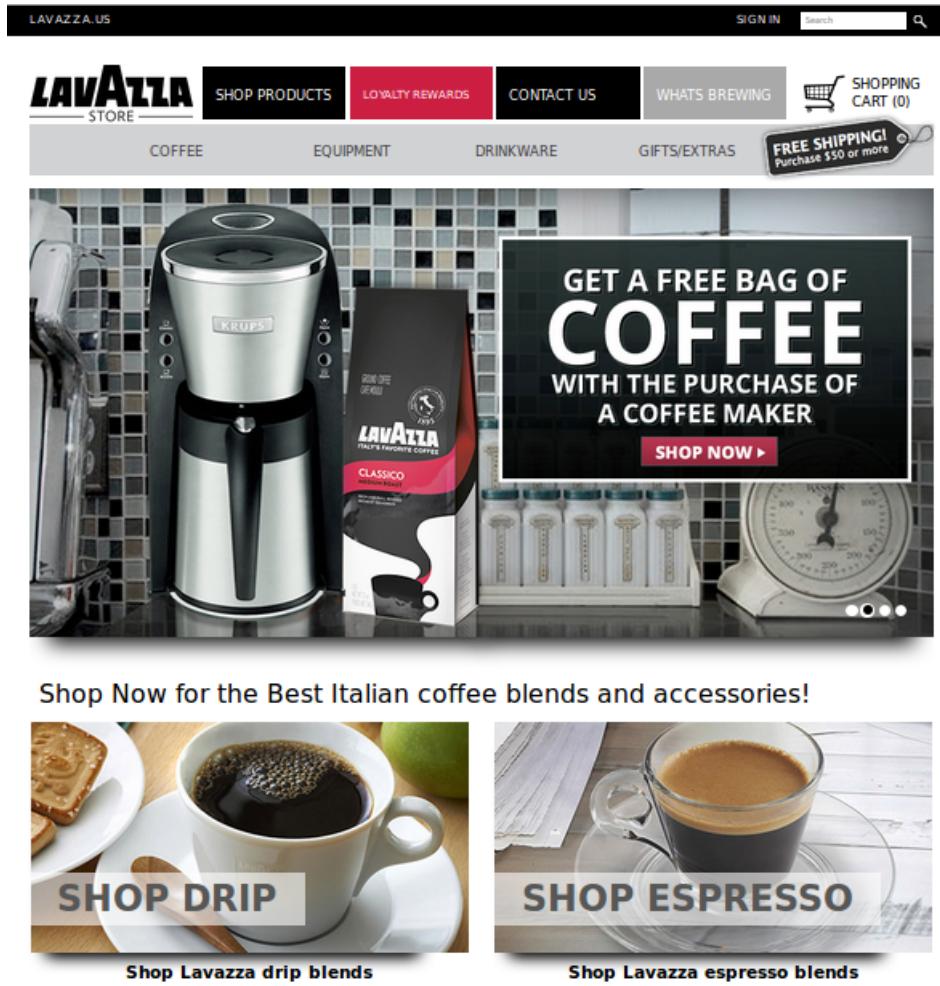
## 6.1. Poslužiteljska aplikacija - Spree Commerce

Spree Commerce (u dalnjem tekstu Spree) je elektronička trgovina (engl. *e-commerce*) otvorenog koda napisana u Ruby on Rails radnom okruženju. Rad na Spreeu je počeo 2007. godine, a danas je jedna od najpopularnijih platformi za elektroničku trgovinu. Trenutno više od 45 tisuća trgovina koristi Spree [10].

Odlike Spree rješenja su ogroman broj već implementiranih funkcija koje imaju sve moderne elektroničke trgovine, modularan izvorni tekst programa, lakoća dodavanja novih funkcionalnosti te izmjena i konfiguracija postojećih te brojne druge. Uz to, za gotovo svaki aspekt sustava dostupan je skup aplikacijsko programskih sučelja što pojednostavljuje integraciju sustava sa drugim tehnologijama (poput integracije sa mobilnim platformama).

U 2013. godini objavljen je Spree Hub čija je funkcija rješavanje dodatnih poslova vezanih za elektroničku trgovinu. Neke odlike Spree Huba uključuju rješavanje logističkih problema (prijenos informacija o narudžbi iz trgovine u skladište), olakšana integracija sustava za računovodstvo, olakšana integraciju sustava za korisničku službu i brojne druge.

Spree Hub se može integrirati skupa sa postojećim električkim trgovinama i pri tome je bitno napomenuti da one ne moraju biti temeljene na Spree Commerce rješenju, već mogu biti temeljene na bilo kojoj drugoj platformi električke trgovine (npr. Magento, Shopify i slične).



Slika 6.1: Lavazza Store - jedna od mnogobrojnih električkih trgovina koje koriste Spree

Spree je podijeljen na nekoliko ključnih komponenti (tj. Ruby gemova):

#### **spree\_api**

Skup aplikacijsko programskih sučelja koji zadovoljavaju REST (engl. *Representational State Transfer*) principe.

#### **spree\_frontend**

Komponente korisničkog sučelja.

#### **spree\_backend**

Komponente korisničkog sučelja administratora aplikacije.

## **spree\_cmd**

Komponente sučelja naredbenog retka (engl. *command-line interface, CLI*).

## **spree\_core**

Ključne komponente, sadrži veliku većinu poslovne logike.

## **spree\_sample**

Primjeri podataka.

Ako programer želi dodati neku od Spree komponenti u svoju postojeću aplikaciju, sve što treba napraviti je dodati željene komponente u Gemfile<sup>1</sup> te pokrenuti instalaciju. Najkraći način za instalaciju Spree gema, stvaranje novoga Ruby on Rails projekta i dodavanja Spreea u novostvoreni projekt prikazan je u nastavku (pretpostavlja se da su Ruby i Ruby on Rails prethodno instalirani i ispravno konfigurirani).

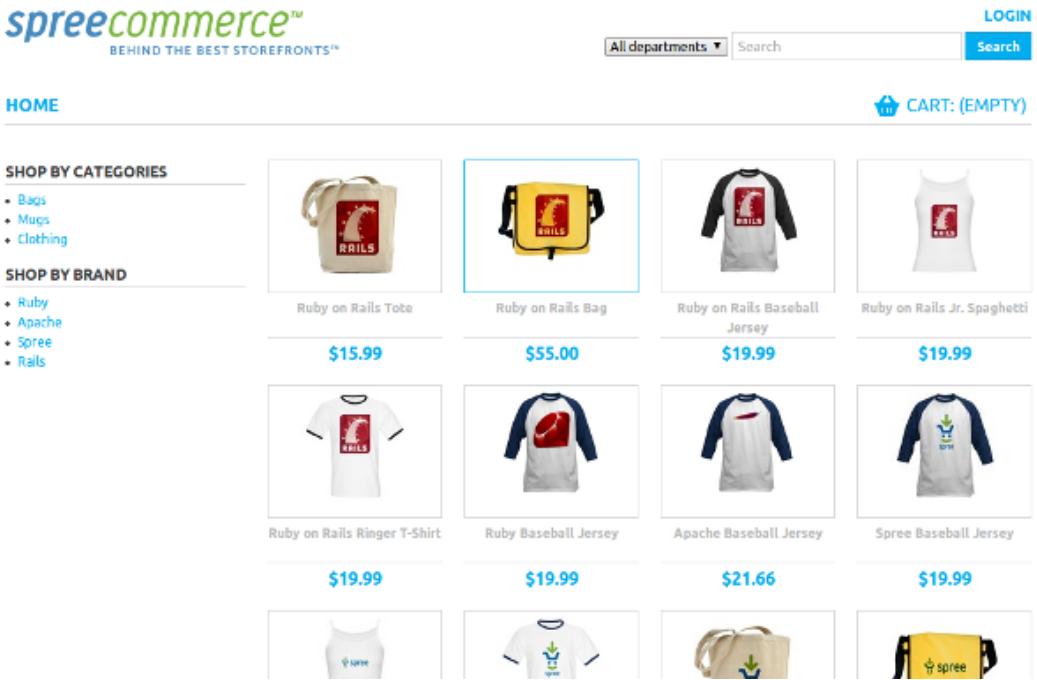
```
$ gem install spree
$ rails _4.0.5_ new moj_projekt
$ spree install moj_projekt
```

Spree projekt se pokreće tako da se pozicioniramo u direktorij projekta i tamo pokrenemo server:

```
$ cd moj_projekt
$ rails server
```

---

<sup>1</sup>datoteka koja sadrži popis svih Ruby gemova koji se koriste u projektu



Slika 6.2: Početni ekran Spree korisničkog sučelja

The screenshot shows the Spree Commerce administrator interface. At the top, there's a navigation bar with the Spree Commerce logo, account information (logged in as: spree@example.com), and links for Account, Logout, Go Back To Store. Below the navigation, there are five main menu items: ORDERS, PRODUCTS, REPORTS, CONFIGURATION (highlighted in green), PROMOTIONS, and USERS. The main content area is titled 'Listing Orders' and features a 'NEW ORDER' button. Below this, there's a search bar and several filtering options: DATE RANGE (Start, Stop), ORDER (text input), FIRST NAME BEGINS WITH (text input), ONLY SHOW COMPLETE ORDERS (checkbox checked), STATUS (dropdown), EMAIL (text input), LAST NAME BEGINS WITH (text input), PROMOTION (dropdown), and a 'FILTER RESULTS' button. At the bottom, there's a table listing two orders:

COMPLETED AT	NUMBER	STATE	PAYMENT STATE	SHIPMENT STATE	CUSTOMER E MAIL	TOTAL
2014-03-09	R987654321	COMPLETE	PAID	SHIPPED	spree@example.com	\$25.35
2014-03-09	R123456789	COMPLETE	PAID	READY	spree@example.com	\$17.63

Slika 6.3: Početni ekran Spree administratorskog sučelja

### 6.1.1. Prilagođavanje i nadogradnja

Modularna arhitektura Spree platforme omogućava jednostavno izmjenu, uklanjanje ili dodavanje novih funkcija u sustav, a kako bi korisničko sučelje bilo optimalno pri-

kazano krajnjim korisnicima Spree nudi i osjetljivi dizajn. Postojeći dizajn je također modularan i željeno korisničko sučelje se može jednostavno ostvariti.

Razvijatelji Spree sustava ne preporučuju prethodno spomenuti način prilagođavanja Spreea svojim potrebama, odnosno ne preporuča se izravna izmjena postojećeg izvornog teksta Spree programa [9]. Preporučeni način prilagođavanja i nadogradnje Spree sustava je korištenjem takozvanih Spree ekstenzija.

Spree ekstenzija nije ništa drugo nego zasebni Ruby gem kojega se dodaje u Gemfile postojećeg projekta. Primjer stvaranja i instaliranja Spree ekstenzije naveden je u nastavku.

Stvaranje Spree ekstenzije:

```
$ spree extension moja_ekstenzija
```

Za dodavanje ekstenzije u postojeći Spree projekt potrebno je dodati ime i putanju ekstenzije u Gemfile projekta:

```
gem 'moja_ekstenzija', path: '../moja_ekstenzija'
```

I nakon toga instalirati ekstenziju:

```
$ bundle install  
$ bundle exec rails g moja_ekstenzija:install
```

Nadogradnja modela i kontrolera je krajnje jednostavna i ne zahtjeva ponovno pišanje postojećih metoda pojedinih razreda, već se pišu samo nove metode ili metode čiji se sadržaj mijenja.

Za nadogradnju postojećih pogleda Spree razvojni tim je razvio Deface. Deface je biblioteka za jednostavno nadograđivanje, izmjenu i uklanjanje komponenti iz postojećih predložaka. Instaliranjem Spree gema, automatski se instalira i Deface. Korištenjem Deface biblioteke programer ne mora kompletno mijenjati sve predloške koje njegova ekstenzija nadograđuje, već na elegantan način može dohvaćati pojedine komponente predloška i mijenjati njihov sadržaj.

Prednost ekstenzija nad izravnom izmjenom izvornog teksta programa je ta što se ostvaruje bolja modularnost, olakšano je ponovno korištenje ekstenzija u drugim projektima te održavanje i nadogradnja sustava sa novim verzijama Spreea je puno jednostavnija. Uz to, Spree tim održava i službeni repozitorij<sup>2</sup> gdje se može pronaći veliki broj besplatnih ekstenzija.

---

<sup>2</sup><http://spreecommerce.com/extensions>

Za integraciju iBeacon funkcionalnosti u Spree projekt razvijena je *Spree Taxon Webservice* ekstenzija koja svakoj kategoriji proizvoda dodaje *reference id* atribut. Na taj način će se ovisno o vrijednostima identifikatora obližnjeg odašiljača iBeacon prikazivati određena kategorija proizvoda.

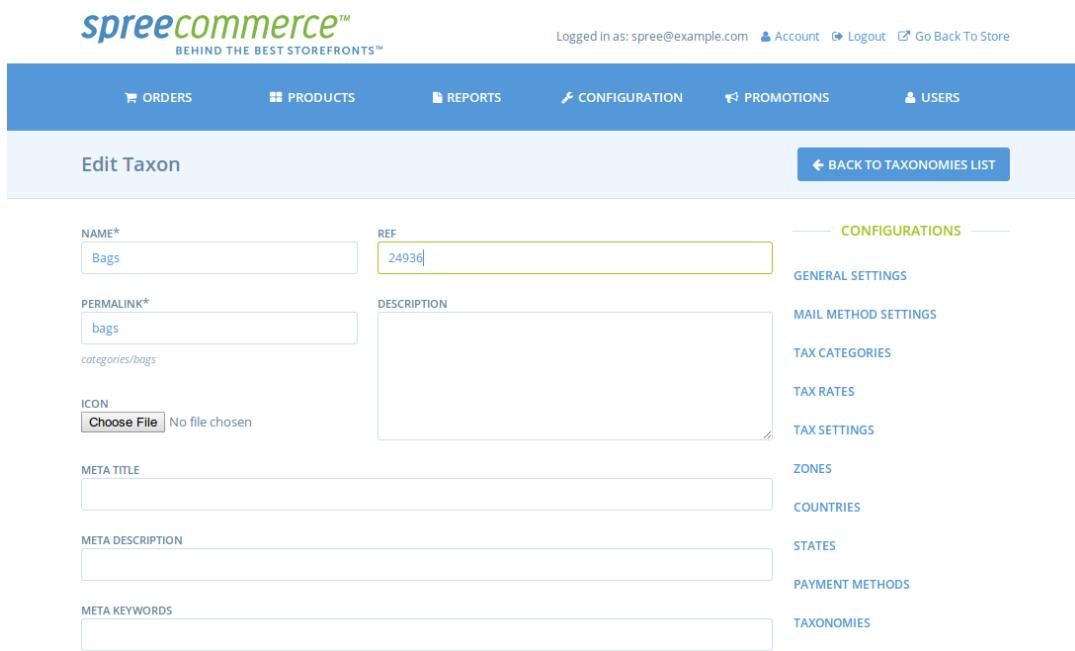
Sada mobilni uređaj nakon identifikacije obližnjeg odašiljača može poslati poslužitelju HTTP zahtjev sa pripadnim sadržajem:

```
GET /api/products?q[taxons_ref_id_eq]=beaconId
```

U slučaju da je detektirano više odašiljača sadržaj zahtjva je:

```
GET /api/products?q[taxons_ref_id_in][]=beaconId1  
&q[taxons_ref_id_in][]=beaconId2
```

Poslužitelj će kao odgovor poslati JSON objekt sa proizvodima koje pripadni odašiljač (ili odašiljači) referencira.



**Slika 6.4:** Sučelje za izmjenu *reference id* atributa kategorije proizvoda

### 6.1.2. Instalacija i pokretanje poslužiteljske aplikacije

Prije instaliranja i podešavanja poslužiteljske aplikacije potrebno je imati ispravno instaliran i konfiguriran Ruby te instalirane Ruby on Rails, Spree i Bundler Ruby move. Nakon toga se može stvoriti novi Ruby on Rails projekt i u njega ugraditi funkcionalnost Spree sustava:

```
$ rails new spree_server_app  
$ spree install spree_server_app  
$ cd spree_server_app
```

Nakon toga je potrebno dodati ekstenziju Spree Taxon Webservice u Gemfile projekta:

```
gem 'spree_taxon_webservice', path: '../spree_taxon_webservice'
```

I instalirati dotičnu ekstenziju:

```
$ bundle install  
$ bundle exec rails g spree_taxon_webservice:install
```

Nakon toga, potrebno je dodati mehanizam CORS (engl. *cross-origin resource sharing*) u aplikaciju, kako bi se sadržaj poslužiteljske aplikacije mogao slati vanjskim aplikacijama (tj. aplikacijama iz druge domene). Najlakši način za rješavanje dotičnog problema je instalacija *Rack CORS Middleware*<sup>3</sup> gema:

```
$ gem install rack-cors
```

Dodavanje dotičnog gema u Gemfile projekta:

```
gem 'rack-cors', :require => 'rack/cors'
```

I konačno, omogućiti *cross-origin* zahtjeve u projektu dodavanjem sljedećeg od-sječka u Application razred koji se nalazi u app/config/application.rb datoteci:

```
config.middleware.use Rack::Cors do  
  allow do  
    origins '*'  
    resource '*', headers: :any, methods: [:get, :post, :options]  
  end  
end
```

Ukoliko su svi prethodni koraci uspješno izvedeni, aplikacija se može pokrenuti sa sljedećom naredbom:

```
$ rails server
```

## 6.2. Klijentska aplikacija

Za prevođenje neke od klijentskih aplikacija potrebno je imati instaliranu i konfiguriranu biblioteku za razvoj programske podrške željene mobilne platforme te Cordova

<sup>3</sup><https://github.com/cyu/rack-cors>

sučelje naredbenog retka.

Za instalaciju biblioteke za razvoj programske podrške dostupne su službene Cordova upute [1], a mogu se koristiti i službene upute proizvođača mobilne platforme.

Cordova sučelje naredbenog retka može se instalirati sljedećom naredbom (prepostavlja se da je npm instaliran i ispravno konfiguriran):

```
| $ sudo npm install -g cordova
```

Pojedina aplikacija se pokreće tako da se pozicioniramo u direktorij aplikacije (npr. /dev/cordova/experimental-app) i upišemo:

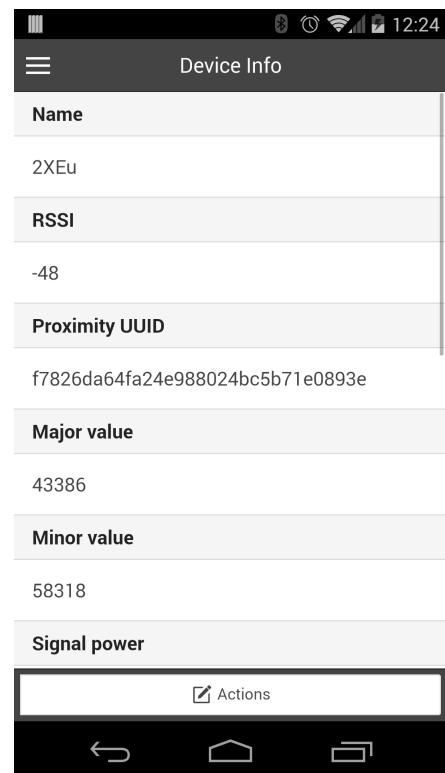
```
| $ cordova run android # za pokretanje na android ureduju
```

### 6.2.1. Eksperimentalna aplikacija

Za istraživanje i ispitivanje odašiljača iBeacon razvijena je posebna aplikacija. Aplikacija detektira obližnje odašiljače iBeacon, obrađuje primljene *advertising* pakete i ispisuje osnovne informacije o odašiljaču, a može ispisati sve usluge (servise), karakteristike i opisnike odašiljača. Uz to, pomoću aplikacije mogu se prikupljati podaci od određenog odašiljača i ta funkcionalnost se može iskoristiti za kalibriranje određenog odašiljača, a i za izračun određenih informacija (poput srednje vrijednosti, standardne devijacije te minimalne i maksimalne jakosti primljenog signala). Također, određeni odašiljači mogu se blokirati tako da se informacije o njima ne prikazuju korisniku te u aplikaciju su ugrađene funkcije (4.2) i (4.3) koje se mogu iskoristiti za (nepouzdanu) procjenu udaljenosti.

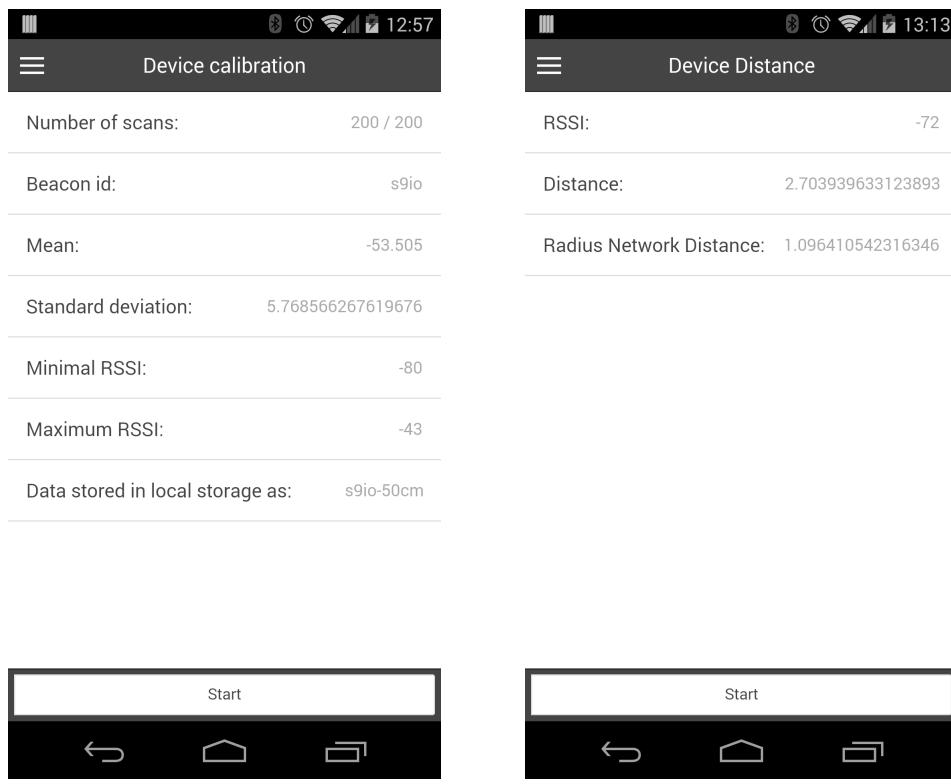


(a) Skeniranje obližnjih odašiljača



(b) Prikaz informacija o odašiljaču

**Slika 6.5:** Dio ostvarenih pogleda eksperimentalne aplikacije



(a) Prikupljanje podataka od odašiljača

(b) Procjena udaljenosti

**Slika 6.6:** Dio ostvarenih pogleda eksperimentalne aplikacije

### 6.2.2. Mobilna aplikacija Spree

U mobilnu aplikaciju Spree ugrađene su osnovne funkcionalnosti bilo koje elektroničke trgovine, poput pregledavanja proizvoda i kategorija proizvoda, košarica za kupnju te sam proces naplate. Svi elementi trgovine (popis proizvoda, kategorija i slično) dohvaćaju se sa poslužitelja na kojem je pokrenuta Spree aplikacija obrađena u poglavljju 6.1.

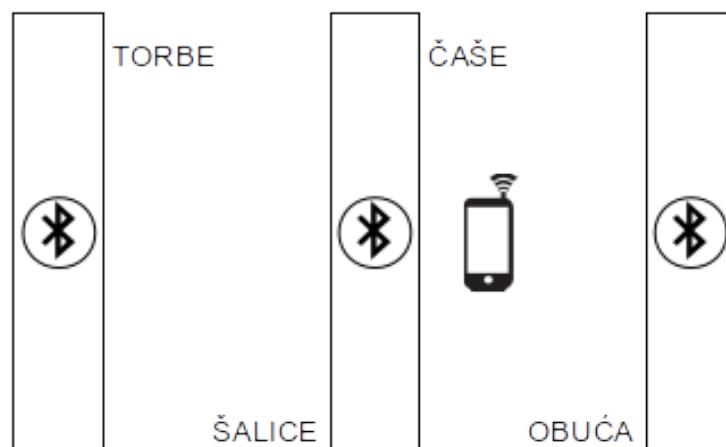
Uz to u aplikaciju je integrirana i Cordova BLE biblioteka te BLE funkcionalnost. Ugrađene su dvije BLE funkcije:

1. Uređaj nekoliko sekundi skenira obližnje odašiljače iBeacon te od poslužitelja dohvaća sadržaj kojega referencira najbolji odašiljač.
2. Uređaj nekoliko sekundi skenira obližnje odašiljače iBeacon te na temelju dva najbliža odašiljača od poslužitelja dohvaća referencirani sadržaj.

Prva funkcionalnost može se iskoristiti u slučajevima gdje je velika trgovina raspoređena u manje odjele, a jedan odašiljač pokriva jedan odjel. Na primjer, trgovina

može biti podijeljena na odjel sa bijelom tehnikom, odjel sa prehrambenim proizvoda i ostale odjele.

Druga funkcionalnost može se iskoristiti kod svakodnevnih trgovina gdje se proizvodi nalaze na policama, po jedna kategorija proizvoda sa svake strane police.



**Slika 6.7:** Primjer korištenja druge funkcionalnosti

Na slici 6.7 pravokutnici predstavljaju police gdje se na desnoj strani lijeve police nalaze torbe, lijevoj strani srednje police šalice, desnoj strani srednje police čaše, a na lijevoj strani desne police obuća. Na svakoj polici se nalazi jedan odašiljač iBeacon. Kupac sa mobilnim uređajem se nalazi između desne i srednje police, s tim da mu je bliža srednja polica, stoga će mu aplikacija prikazati kategoriju sa čašama.

U slučaju da koristimo prvu funkcionalnost, nakon skeniranja odašiljača mobilna aplikacija će poslužitelju poslati sljedeći zahtjev:

```
GET /api/products?q[taxons_ref_id_eq]=bestBeaconMinorValue
```

gdje parametar *bestBeaconMinorValue* označava *minor value* identifikator najboljeg (tj. najbližeg) odašiljača.

U slučaju da koristimo drugu funkcionalnost, nakon skeniranja odašiljača identifikatori dva najbolja odašiljača se na neki način moraju mapirati u broj koji će poslati poslužitelju:

```
GET /api/products?q[taxons_ref_id_eq]=referenceId
```

Parametar *referenceId* označava mapiranu vrijednost.

Kada se pokrene BLE funkcionalnost aplikacije, aplikacija skenira sve obližnje odašiljače te za svaki novopranođeni odašiljač stvara dinamičko polje u koju se pohrani jakost primljenog signala odašiljača (tj. RSSI odašiljača). Kada se opet primi signal od istoga odašiljača, nova jakost signala se dodaje u istu dinamičku listu. Pošto pojedini odašiljač šalje signal više puta u sekundi, nakon nekoliko sekundi svaka će lista imati veći broj elemenata. Tada se pokreće sljedeći algoritam:

- iz svih dinamičkih polja izbací 10% marginalnih vrijednosti
- sortiraj odašiljače tako da se na prvim mjestima nalaze odašiljači čija dinamička polja imaju najvišu srednju vrijednost

Izvorni tekst programa je prikazan u nastavku.

```
// pomocna funkcija za izracun srednje vrijednosti
MathService.mean = function (data) {
    sum = 0;
    angular.forEach(data, function (val) {
        sum += val;
    });
    return sum / data.length;
};

// pomocna funkcija za uklanjanje marginalnih vrijednosti
MathService.removeMarginals = function (data, percentage) {
    percentage = angular.isDefined(percentage) ? percentage : 0.9;
    var average = this.mean(data);

    // sortiranje tako da se na pocetku nalaze vrijednosti
    // najblize srednjoj vrijednosti
    data.sort(function (a, b) {
        var distance = Math.abs(a - average) - Math.abs(b - average);
        if (distance < 0) {
            return -1;
        } else if (distance > 0) {
            return 1;
        }
        return 0;
    });
    // broj elemenata koji se uklanja
    var numOfElToRm = Math.floor((1 - percentage) * data.length);
    // uklanjanje elemenata
    data.splice(data.length - numOfElToRm, numOfElToRm);
```

```

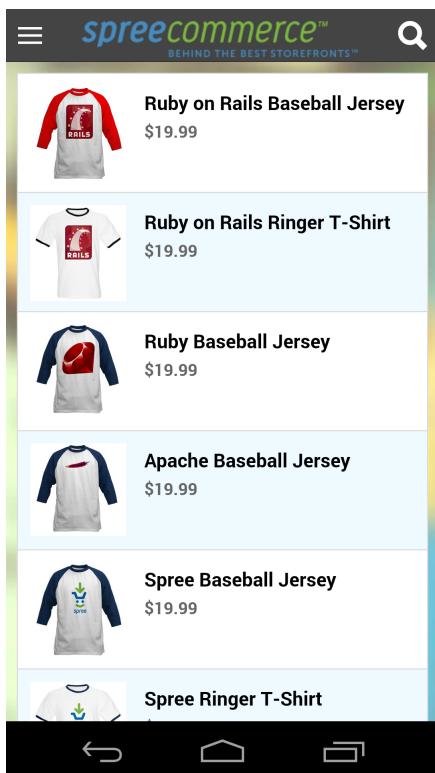
};

// uklanjanje marginalnih vrijednosti svih odasiljaca
angular.forEach(devices, function (device) {
    MathUtilities.removeMarginals(device.data, 0.9);
    device.averageRSSI = MathUtilities.mean(device.data);
    device.data = [];
});

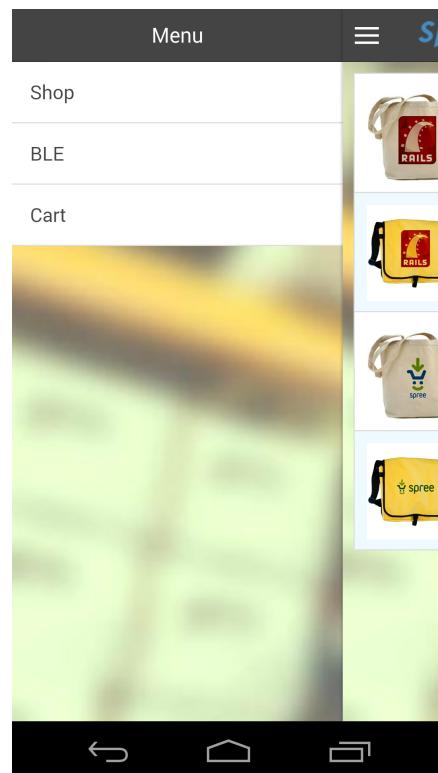
// sortiranje odasiljaca
devices.sort(function (a, b) {
    if (a.averageRSSI < b.averageRSSI) {
        return 1;
    } else if (a.averageRSSI > b.averageRSSI) {
        return -1;
    }
    return 0;
}
);

```

Ukoliko se koristi prva funkcionalnost, poslužitelju se šalje zahtev za podacima i u parametrima se navodi *minor value* identifikator prvog (najboljega) odašiljača. Ukoliko se koristi druga funkcionalnost, *minor value* identifikatori dva najbolja odašiljača se mapiraju u broj koji se šalje poslužitelju prilikom dohvata podataka. Nakon toga se prazne liste svih odašiljača te se ciklus skeniranja obližnjih uređaja nastavlja.



(a) Prikaz jedne kategorije proizvoda



(b) Prikaz glavnog menija

**Slika 6.8:** Mobilna aplikacija Spree

## 7. Zaključak

Unatoč tome što je razvijen cijeli niz novih metoda za rješavanje problema navigacije i pozicioniranja u zatvorenom prostoru, još nema općeprihvaćenih rješenja.

Korištenje tehnologije Bluetooth Low Energy i odašiljača iBeacon za rješavanje problema procjene udaljenosti u zatvorenom prostoru nije precizno ni pouzdano iz razloga što na signal koji putuje zatvorenim prostorom djeluje veliki broj smetnji koje se često ne mogu smanjiti ni otkloniti. Bez obzira na to, odašiljači iBeacon mogu se primijeniti u gotovo svim situacijama gdje se sadržaj mijenja ovisno o kontekstu prostora. Iz tog razloga, tehnologija iBeacon se sve više integrira u mobilne aplikacije.

U praktičnom dijelu rada prikazano je kako zbog smetnji koje djeluju na signal nije moguće precizno procijeniti udaljenost mobilnog uređaja od odašiljača. Ponađeno rješenje procjene temelji se na modelu neuronske mreže koju je trenirao genetski algoritam. Također, u praktičnom dijelu je prikazano kako se tehnologija iBeacon može integrirati sa hibridnom mobilnom aplikacijom tako da prikazuje različite podatke ovisno o lokaciji mobilnog uređaja.

# LITERATURA

- [1] Apache Cordova. Platform guides, 2014. [http://cordova.apache.org/docs/en/edge/guide\\_platforms\\_index.md.html#Platform%20Guides](http://cordova.apache.org/docs/en/edge/guide_platforms_index.md.html#Platform%20Guides), datum pristupa 3.6.2014.
- [2] Gupta D. Wlan signal characteristics in an indoor environment—an analytic model and experiments. magistarski rad, Faculty of the Graduate School of the University of Maryland, 2005.
- [3] Schneider D. You are here. *IEEE Spectrum*, stranice 30–35, prosinac 2013.
- [4] Young David. Reverse engineering the ibeacon profile, 2013. <http://developer.radiusnetworks.com/2013/10/01/reverse-engineering-the-ibeacon-profile.html>, datum pristupa 6.5.2014.
- [5] Young David. Understanding ibeacon distancing, 2013. <http://stackoverflow.com/a/20434019>, datum pristupa 11.5.2014.
- [6] Kontakt.io. Kontakt.io beacon datasheet v2.0, 2014. <http://docs.kontakt.io/beacon/kontakt-beacon-v2.pdf>, datum pristupa 14.4.2014.
- [7] Kontakt.io. Tehnical specifications, 2014. <http://kontakt.io/technology/technical-specification>, datum pristupa 8.4.2014.
- [8] Google Advanced Technology and Projects. Say hello to project tango, 2014. <https://www.youtube.com/watch?v=Qe10ExwzCqk>, datum pristupa 22.5.2014.
- [9] Spree. View customization-developer guide, 2014.. <http://guides.spreecommerce.com/developer/view.html>, datum pristupa 1.6.2014.

- [10] Spree. View customization-developer guide, 2014.. <http://spreecommerce.com/storefront> datum pristupa 28.5.2014.
- [11] Čupić M., Dalbelo Bašić B., i Golub M. Neizrazito, evolucijsko i neuroračunarstvo, 2012–2013.

# **Utvrđivanje mikrolokacije mobilnog uređaja u zatvorenom prostoru**

## **Sažetak**

Tehnologije navigacije i pozicioniranja koje se oslanjaju na udaljene satelite (npr. tehnologija GPS) nisu pogodne za korištenje u zatvorenim prostorima. Stoga su za utvrđivanje lokacije u zatvorenom prostoru potrebne nove i drugačije metode. Kako ne postoji nikakav *de facto* standard, danas postoji niz različitih rješenja. Uvođenjem Bluetooth 4.0 specifikacije i tehnologije Bluetooth Low Energy dolazi do razvoja niza jeftinih uređaja koji se mogu iskoristiti za rješavanje problema određivanja lokacije. Kako na signal u zatvorenom prostoru djeluje veliki broj smetnji određivanje lokacije pomoću tehnologije BLE nije precizno i preporuča se korištenje samo za određivanje okvirne lokacije. Unatoč tome, ono se može primijeniti u gotovo svim situacijama gdje se sadržaj mijenja ovisno o kontekstu prostora stoga se ono danas sve više integrira u mobilne aplikacije.

**Ključne riječi:** Bluetooth, BLE, mikrolokacija, iBeacon, Android, iOS, Apache Cordova, Ruby on Rails, Ruby, JavaScript, razvoj mobilnih aplikacija

## **Determining a micro-location of a mobile device**

### **Abstract**

Positioning technologies that receive signals from distant satellites (such as GPS) are not suitable indoors. Because of that, the indoor navigation and positioning problem requires a new and different approach. With the adoption of the Bluetooth 4.0 specification and the Bluetooth Low Energy technology, a large number of cheap Bluetooth Smart Ready devices that could be used for solving the indoor positioning problem has been manufactured. The radio signal that propagates in a indoor environment is affected by a large number of factors and because of that, using BLE technology for precise indoor positioning isn't appropriate. Despite of that, it can be used in almost all situations where the content changes depending on the context of the environment so today BLE is being increasingly integrated into mobile applications.

**Keywords:** Bluetooth, BLE, micro-location, iBeacon, Android, iOS, Apache Cordova, Ruby on Rails, Ruby, JavaScript, mobile development