Indoor Navigation for a Supermarket Using Bluetooth low energy (BLE) Beacons and Analysis of Consumer Behavior

by
D.N.S Perera
14222P

Faculty of Information Technology
University of Moratuwa

May 2017

Indoor Navigation for a Supermarket Using Bluetooth low energy (BLE) Beacons and Analysis of Consumer Behavior

Mobile application and a Web admin panel

D.N.S Perera

149222P

(MSCIT/14/056)

Dissertation submitted to the Faculty of Information Technology, University of Moratuwa, Sri Lanka for the partial fulfillment of the requirements of the Degree of MSc in Information Technology.

Faculty of Information Technology

University of Moratuwa

May 2017

Declaration

I declare that this thesis is my own work and has not been submitted in any form for another degree or diploma at any university or other institution of tertiary education. Information derived from the published or unpublished work of others has been acknowledged in the text and a list of references is given.

Name of Student | Signature of Student

D.N.S Perera

………………………

Date:

Supervised by | Signature of the supervisor

Mr. B.H Sudantha

………………………

Date:

Dedication

This Dissertation is dedicated to my loving parents for their support and encouragement.

Acknowledgements

First I would like to express my heartfelt appreciation and gratitude to my supervisor Mr.B.H Sudantha for his most valued guidance, commitment and kind support to make this research success. He consistently allowed this paper to be my own work, but steered me in the right the direction whenever he thought I needed it.

I would also like to thank Prof. Asoka S Karunanda who taught us Research Methodology and Literature Review and Thesis Writing subjects which helped me to ease this process.

It is my great pleasure to thank all the other Lecturers and all my batch mates of the M.Sc in Information Technology batch 8 in faculty of Information Technology for their various help and support.

Abstract

Retailers typically make a variety of strategic level decisions including the type of store, the market to be served, the optimal product assortment, customer service, supporting services and the store's overall market positioning. Once the strategic retail plan is in place, retailers devise the retail mix which includes product, price, place, promotion, personnel and presentation.
In this digital age, there are many ways to analyze buying patterns. Yet retailers would like to understand consumer behavior inside stores so they can organize and place their products based on that and gather more information based on gender, age, profession etc.

Localization and navigation have been important topics in research. There are many impossibilities when trying to perform positioning within indoor environments, with the use of GPS technology. In order to overcome these limitations, we look into Bluetooth Low Energy technology based localization model.

In real time applications such as object tracking and distance estimations, continuous receptions of RSSI measurements are needed in order to estimate accurately the position of the object. In adjacent to those considerations, there are some additional constraints to be inspected such as signal attenuation, signal loss, multipath effects, temperature, reflection, a human body and other communication signals. Hence, this research work has examined the RSSI smoothing approaches in order to obtain preferable results. Although there are so many solutions, no RSSI smoothing method has been recognized as a standard method.

During experiment, we found that the fluctuation of the RSSI values are hard to handle and many techniques were used to overcome this. Kalman filter algorithm was used to smoothing the RSSI values. Many techniques were tried to get the exact position of the user and trilateration algorithms are used to estimate the position of the user.

Table of contents

List of Figures

Chapter 1

1. Introduction

1.1 Prolegomena

Location Based Services (LBS) are mobile applications which rely on a user's location to deliver context aware functionality. Industry forecasts for this area predict huge market g rowth and revenue. The Global Positioning System (GPS) is the most popular positi oning system. However, it is not suitable for indoor positioning. Real-time indoor positioning is still a challenge using existing techniques. There are multiple tech nologies for indoor positioning, e.g., WiFi, ZigBee, Bluetooth,
inertial, Magnetic positioning. This application use beacons which are small, often inexpensive devices that enable more accurate location within a narrow range than GPS , cell tower triangulation and Wi-Fi
proximity. Beacons transmit small amounts of data via Bluetooth Low Energy (BLE)

1.2 Background & Motivation

Consumers face many difficulties when searching for a particular item in a supermarket/d epartment store, searching for a book in a large library, finding stalls in an exhibition cent er, walking inside an airport, etc. Retailers are always seeking for ways to interact wi th consumers, analyzetheir behaviors and buying patterns, notify them about new offers, t argeted advertising. Locationbased services are growing enabled by mobile devices with GPS. Yet people spend most of the time indoors.
No wide-spread indoor positioning systems and services are available yet in countries like Sri Lanka.
GPS operation is very limited indoors. Personal communication devices can enable indoo r positioning through local wireless networks.
This research focus on where retail needs to go, and how technology can facilitate their n eeds as well as making shopping much easier and quicker.

1.3 Problem Statement

While navigation systems for outdoor environments are readily available, navigation with in buildings still poses a challenge. Though there are maps drawn at super markets, shopp ing malls, airports and etc getting to where you want as quickly as possible and getting more information about things around you is a difficult task.

Increasing sales and monitoring user behavior while providing a much interactive and pleasurable shopping experience is the challenge.

1.4 Hypothesis

By using Bluetooth Low Energy(BLE) beacons we can solve the addressing problem by collecting and analyzing user behavior and feedback.

1.5 Aim and Objectives

1. Identifying a smoothing approach and applying it.
2. Study techniques used to find the position of the object and implementation
3. Provide an interactive map with a product inventory search.
4. Keep user posted about special promotions and discounts.
5. Allow administrators to easily update the inventory.
6. Analyze consumer behavior and provide valuable information for marketing and management.

1.6 Base approach

This entire process has 3 major parts.

1. Mapping out the area
2. Positioning beacons and measuring signal strengths.
3. Real time updating of the database with location information.
4. Web application

1.7 Structure of Thesis

Following chapters of the thesis are organized as follows. Chapter 2 critically reviews the literature of indoor navigation researches and projects and identify the research problems and possible solutions. Chapter 3 is about the Bluetooth Low Energy Technology, BLE beacons and mobile development which are the base technologies of this research. Chapter 4 present our novel base approach to use better navigation and interactive shopping. Chapter 5 and Chapter 6 describe the design and implementation respectively. Chapter 7 describe how this research use in practices. Chapter 8 is on evaluation of the new solution. Chapter 9 concludes the research with a note on further work.

1.8 Summary

A brief introduction about the research problem and the solution was covered in this chapter. The next chapter covers a critical discussion, showing insight and an awareness of differing arguments, theories and approaches of published information in indoor navigation.

Chapter 2

2. Literature Review

2.1 Introduction

We will now point out some important concepts those are closely related with BLE indoor navigation. This chapter consists of 2 major sections. First, existing positioning technologies enabling outdoor and indoor navigation are briefly explained by using several comprehensive studies from the literature; the challenges of these systems are given as well. In the second subsection, BLE technology, its operating modes, and other BLE related issues are described for a clear understanding. The brief research background will be helpful for understanding the significance of the contributions of BLE-based navigation systems compared to existing ones.

2.2 Indoor Positioning Technologies

In recent years, positioning systems have become a popular field in both academic and industry research and there already exist several research and commercial products in this area. Indoor navigation systems have become an especially hot research area recently, due to the unavailability of GPS in indoor environments, as explained in the previous section. With the aim of circumventing this lack of performance, an assortment of technologies has been tested and new designs generated for indoor navigation in the literature.

Positioning systems enable the appropriate device to determine its position, and make the location information available for position-based services [6]. Positioning system topology is an important issue for understanding and developing a positioning system. According to [8,9], four different system topologies are defined for wireless positioning systems: (a) remote positioning; (b) self-positioning; (c) indirect remote positioning and (d) indirect self-positioning. In case of a *remote positioning system*, there is a signal transmitter and several fixed measuring units. The signal transmitter is mobile, and fixed measuring units receive the transmitter's signal. The location of the transmitter is computed in a master station after collection of all data from measuring units. In a *self-positioning system*, the measuring unit is mobile and it receives the signals of several transmitters placed in known places, and with the help of these signals, it can compute its location. In the case of *indirect remote positioning*, a self-positioning measuring unit sends the measurement results to the remote side via a wireless data link. A remote positioning unit can also send measurement results to a mobile unit, which is then named *indirect self-positioning*.

According to another study [5,10], indoor positioning techniques can be divided into two categories: *network dependent* systems and *device dependent (network independent)* systems. Network dependent navigation systems are based on networking technologies such as IR, sensors, ultrasound, WLANs, UWB, Bluetooth, RFID technologies, whereas independent navigation systems provide autonomous user positions such as A-GPS as an indoor GPS system.

The study in [11] defines two approaches for indoor positioning: locating in relative coordinates and locating at choke points. Locating in relative coordinates requires usage of active devices (*i.e.*, devices that use their own power sources) for positioning, however passive devices can also be used for locating at choke points. Figure 1 illustrates a simple classification of indoor positioning technologies.

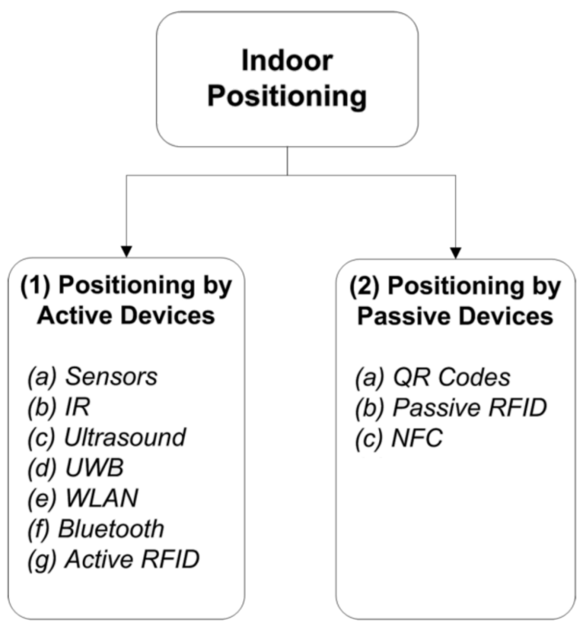


Figure 1

In case of *locating in relative coordinates*, reference transmitters determine the position of the user and the accuracy of positioning depends on the range and coverage of reference transmitters [11]. These systems usually calculate the position of the mobile object using the signal strength at reception time, which is expressed by the Received Signal Strength Indication (RSSI) technique. The technologies used in this method are IR, ultrasound technologies (e.g., Active Bat, Crickets, Dolphin [7]), UWB (e.g., Ubisense [6]), WLAN (e.g., RADAR, Ekahau, COMPAS [7,8]), Bluetooth (e.g., Topaz [6]), active RFID (e.g., LANDMARC [12,13]).

In case of *locating at choke points*, the sensors are located at fixed points, which provide location values in the network. In this approach, tagged objects including location coordinates determine the location [11]; some technologies for this approach are Quick Response (QR) codes, Passive RFID and NFC.

UWB positioning systems use Radio Frequency (RF) signals for indoor positioning and provide high accuracy [4,6]. UWB positioning systems has their own limitations as well; an efficient indoor navigation system cannot be provided because of several technology problems such as low power emission, antenna mismatch, and possible external interference from other systems [10].

Position accuracy depends on the type of the tags in the RFID case, which may be either passive or active; the positional density of these tags affects the accuracy as well. Passive tags are capable for small communication distances, whereas batteries are integrated into active tags to increase the communication distance. Available RFID-based indoor navigation models are mostly based on RFID tag usage and hence require vast amounts of tag usage for accurate positioning [14]. The major drawback of an active RFID-based indoor positioning system is the high cost of the active RFID tags, which does not provide a cost-efficient solution [10].

Another popular existing technology for both outdoors and indoors is the A-GPS system, which is network independent. It expands the working area of the GPS technology to indoor environments by processing indoor GPS signals. However, the signal strength in indoor environments is sometimes too low [10].

2.3 Bluetooth Low Energy

Bluetooth low energy (**Bluetooth LE**, **BLE**, marketed as **Bluetooth Smart**) is a wireless personal area network technology designed and marketed by the Bluetooth Special Interest Group aimed at novel applications in various areas.

Bluetooth Smart was originally introduced under the name **Wibree** by Nokia in 2006. It was merged into the main Bluetooth standard in 2010 with the adoption of the Bluetooth Core Specification Version 4.0.

Researchers at Nokia determined various scenarios contemporary wireless technologies did not address.[11] The company began developing a wireless technology adapted from the Bluetooth standard which would provide lower power usage and cost while minimizing its differences from Bluetooth technology[1].

Cell-ID and WiFi network based positioning is already widely available but accuracy is limited to building level. Turn-by-turn indoor navigation requires systems specifically designed for positioning purposes. Nokia HAIP technology is built on BLE and offers sub-meter location accuracy indoors. Standardization work currently taking off in BT SIG. Pre-commercial pilots are being planned by Nokia and Navteq to start the ecosystem creation for indoor positioning systems[1].

2.3.1 BLE history

2009 - The SIG announces the adoption of Bluetooth low energy wireless technology, the hallmark feature in Bluetooth Core Specification Version 4.0[20].

2010 - The Bluetooth SIG announces the formal adoption of Bluetooth Core Specification Version 4.0 with low energy technology[20].

Profile Tuning Suite (PTS) v4.1 is launched, including Bluetooth low energy technology test suites[20].

2011 - Apple announces that the new iPhone 4S will support Bluetooth v4.0 and become the first Bluetooth with low energy phone[20].

2012 - The first Bluetooth with low energy tablets and music players hit the market[20].

2013 - Google announced native support for Bluetooth with low energy in Android[20].

2015 - SIG and FIDO Alliance Deliver Two-factor Authentication via Bluetooth with low energy[20].

2.3.2 Different approaches for accurate indoor localization/navigation

Proximity approach:

The nearest network node is determined using some kind of signal emitted by the node. The smartphone can sense this signal and determine its own position related to the known network node position [15]. This approach is simple and low cost but provides low-level accuracy.

RSSI (Received Signal Strength Indication) fingerprint:

The smartphone measures the RSS values in the environment and compares them with previously measured values, which are saved in a database (the fingerprint), to estimate the position of the user and to perform indoor navigation. This approach, often used with wi-fi signals which are available for free in the buildings, provides a good level of accuracy but needs experienced personnel to create the fingerprint for each new area [16], which bring to high cost and non-scalable solutions that need frequent reconfigurations. – Dead Reckoning: the indoor navigation is performed by using the inertial sensors embedded on the device. Usually, due to the unacceptable drift error of the inertial sensors, a step-counter system is used to estimate the distance covered by the user. This approach is easy to deploy and low cost but needs a periodic position recalibration to reset the error which quickly becomes too high for indoor environments [17].

Visual tag approach:

A visual-tags system is deployed inside the building at known points. The user, using the embedded camera on the smartphone, can navigate inside the building [18]. This approach provides a variable level of accuracy (depending on density of tags), but has high latency due to tag detection and is uncomfortable for the user who has to find the tag, and focus the camera on it.

Hybrid approach:

Usually multiple techniques are used together to improve the efficiency of the localization/navigation [19].

2.4 Problem definition

Discussion about indoor navigation and Bluetooth low energy shows that various approaches has been taken and what are the advantages and drawbacks of them. The literature review has identified various unsolved problem such as security, accuracy and reliability.

Therefore, we should define the research problem as inadequate attention given to the indoor navigation and analysis of user behavior and find the solution for more accurate, user-friendly and adequate technique for indoor navigation and monitoring.

2.5 Summary

This chapter presented a comprehensive literature review on indoor navigation and identified the research problem. We have realized that a hybrid technique of BLE and visual tagging could solve the above problem. Next chapter will discuss the technologies to be used for our solution.

Chapter 3

3. Technologies

3.1 Introduction

We discussed about existing technologies and techniques used to achieve indoor navigation in the previous chapter. During that study, strengths and weaknesses of the technologies used by others were identified. This chapter describes the technologies identified to conduct the research.

Software

- Web Development Technologies
    - Html/ CSS / Ajax / java Script
    - PHP
    - Laravel framework

- Database Management Systems
    - MySQL
    - SQLite

- Mobile Technology
    - Android, Apache Cordova, Ionic

- Web Server
    - Apache

- Version control tool
    - GIT

Hardware

- Beacons

- Android mobile phone

3.2 Web Programming

3.2.1 PHP

PHP is a server scripting language, and a powerful tool for making dynamic and interactive Web pages. PHP is a widely-used, free, and efficient alternative to competitors such as Microsoft's ASP.

Reasons for using php are,

1. Fast Load Time – PHP results in faster site loading speeds. PHP codes runs much faster than many others because it runs in its own memory.

2. Less Expensive Software – In working with PHP, most tools associated with the program are open source software, so you need not pay for them.

3. Less Expensive Hosting – PHP would only require running on a Linux server, which is available through a hosting provider at no additional cost.

4. Database Flexibility– PHP is flexible for database connectivity. It can connect to several databases the most commonly used is the MySQL.

3.2.2 Laravel

Laravel is a free, open-source PHP web framework, intended for the development of web applications following the model–view–controller (MVC) architectural pattern.

2 Routing System: With Laravel, we can easily approach to routing. The route can be triggered in the application with good flexibility and control.

3 Database Query Builder: Laravel provides fluent interface to running database queries

4 Artisan Console: Artisan is the command line interface in the laravel. Provides number of command while developing a web application.

5 HTTP Middleware: Provide a convenient mechanism for filtering HTTP requests entering your application.

6 A lightweight Blade Templating Engine

3.2.3 AWS EC2

Amazon Elastic Compute Cloud (Amazon EC2) [22] is a web service that provides secure, resizable compute capacity in the cloud. It is designed to make web-scale cloud computing easier for developers. Amazon EC2's simple web service interface allows the developer to obtain and configure capacity with minimal friction.

3.3 Database Management systems

3.3.1 MySQL

MySQL is an open-source relational database management system (RDBMS). The database is free and open source with a commercial license available.
MySQL has one major advantage, since it is free, it is usually available on shared hosting packages and can be easily set up in a Linux, Unix or Windows environment. If a web application requires more than database, requires load balancing, it is easy to set up maybe instances of the database requiring only the hardware costs, as opposed to commercial databases that would require a single license for each instance.

3.4 Mobile Technologies

3.4.1 Cordova

Apache **Cordova** (formerly PhoneGap) is a mobile application development framework. Apache Cordova enables software programmers to build applications for mobile devices using CSS3, HTML5, and JavaScript instead of relying on platform-specific APIs like those in Android, iOS, or Windows Phone.[4] It enables wrapping up of CSS, HTML, and JavaScript code depending upon the platform of the device. It extends the features of HTML and JavaScript to work with the device.

3.5 Bluetooth Beacons

Bluetooth beacons are hardware transmitters - a class of Bluetooth low energy (BLE) devices that broadcast their identifier to nearby portable electronic devices. The technology enables smartphones, tablets and other devices to perform actions when in close proximity to a beacon.

There are 4 major protocols of beacons, Eddystone and IBeacon are the 2 most popular of them.

In here we use Eddystone protocol as it is compatible with android platforms.

3.5.1 Eddystone

Eddystone is an open beacon format developed by Google and designed with transparency and robustness in mind.

Eddystone can be detected by both Android and iOS devices. It supports 3 types of packets,

Eddystone-UID: A unique, static ID with a 10-byte Namespace component and a 6-byte Instance component.

Eddystone-URL: A compressed URL that, once parsed and decompressed, is directly usable by the client.

Eddystone-TLM: Beacon status data that is useful for beacon fleet maintenance, and powers Google Proximity Beacon API's diagnostics endpoint. -TLM should be interleaved with an identifying frame such as Eddystone-UID or Eddystone-EID (for which the encrypted eTLM version preserves security).

Figure 3.1: Beacon Hardware

3.6 Summary

Technologies used for this project are been briefly described in this chapter. More about the technologies will be discussed in the Design and Implementation chapters.

Chapter 4

4. Approach to Implement Indoor navigation

4.1 Introduction

Here we describe our approach to Indoor navigation and user behavior analysis. This section present novel solution to address our research problem. The approach is described under the headings hypothesis, inputs to the system, outputs of the system, process, users of system and features of the system

4.2 Hypothesis

Our hypothesis is addressing problem can be solve with the use of Bluetooth beacons.

4.3 Inputs to the system

Signal strengths of the beacons around the mobile device and their transmission power along with the beacon name and nid.

Searching item(s) under different categories are entered to the system by admins.

Supermarket admins update the map with items and add special promotions to trigger when user gets closer to relevant beacons.

4.4 Outputs of the system

Position of the beacons.

Current Location.

Special promotions (Notifications).

Selected Items on the map.

Analytical graphs of user behavior on the web panel.

4.5 Process

The entire process could be divided into 4 main processes.

1. Mapping
2. Estimating current location
3. Data collection
4. Analysis

4.5.1 Mapping

Super market should be mapped and measured at first before placing beacons. Then beacons should be placed accordingly with minimum interference. Continuous testing an d data
collection should be carried out for signal strength measurement before final implementations.
Beacon labeling should be done in this process.
Coordinates of the beacons are entered to the system.

4.5.2 Estimating Current Location

When the mobile device detects the beacons, it receives the RSSI values of each beacon every 350ms. They are then used to calculate the distance to the device and every 10 set of values are sent into a filter where the distortions are minimized and an estimated distance value is returned.
By using trilateration algorithm, the current location of the device is found and displayed.

4.5.3 Data Collection

User movements are recorded and synced to the server as the position is determined every 5 seconds. User id and location coordinates are sent to the server.

4.5.4 Analysis

Collected data are analyzed based on different criterias to generate reports for Store administrators.

4.6 Users of system

Many users will benefit from this system. Mainly the consumers and supermarket administrators.
More importantly suppliers, advertising agencies, social research organizations and research students could also benefit from this system.

4.7 Features

1. Provide an interactive map with a product inventory search.
2. Items can be quickly being added to shopping lists and allow easy and quick navi gation
   to consumers.
3. Keep user posted about special promotions and discounts.
4. Allow administrators to easily update the inventory.
5. Analyze consumer behavior and provide valuable information for marketing and management.

4.8 Summary

This chapter we have described overall solution of our research. We have mentioned problem definition and assumption of the solution to that problem. We have described clearly inputs to the system, outputs of the system and how convert input to output, who has beneficially of this system, and incredible features of solution. Next chapter we will describe in detail, extended design of our process and what system does.

Chapter 5

5. Design

5.1 Introduction

This chapter describes the design of the solution for the process presented in the approach. Languages, tools, and technologies used during development are key factors in making applications different from one another. This chapter describes mobile development approaches and the technologies and tools used during implementation. Once decisions are made on which tools and technologies to use, the overall system architecture, API and backend models are created.

5.2 Frontend

This section discusses decisions regarding the tools and technologies related to the presentation layer of the application. The frontend is considered as an interface between the user and the backend system.

5.2.1 Mobile Development Approaches

Mobile application development can be a very complex and challenging task due to the variation of mobile platforms (iOS, Android, Windows Phone). Thus, different development approaches have emerged and we must make an informed decision about which one to use.

5.2.2 Mobile Development Approaches

Native

Native applications are written in the native language of the platform (Objective-C for iOS, Java for Android, C#/.NET for Windows Phone). They have direct access to native services, such as cameras, through the API provided by the platform's SDK. The platforms SDKs also provide user interface components such as animations, dialogs, gestures, tabs, or menus, and, as a result, each platform has its own unique look and feel. The main advantage of native applications is their performance. Native code is compiled and runs at a low level, allowing developers to build computing intensive applications or applications with complex animations. A disadvantage of native applications is their portability – they can only run on the platform they were developed for. From the user interface perspective, no two platforms have the same or even similar paradigm, therefore most of the code will have to be rewritten with little able to be shared. In a nutshell, native code development can be a very complex task that requires developers skilled in each of the platforms. On the other hand, it offers the best in terms of performance.

Hybrid

The hybrid approach combines the advantages of the native and web approaches. Hybrid applications are installed on the device in the same way as native applications. The only difference is that they are built with a combination of web technology, such as HTML, CSS, and JavaScript, and are hosted inside the application's WebView (one can think of the WebView as a browser window that runs in fullscreen within the native application). This enables the application to access device services even though it is implemented as a web application.

Hybrid applications have the advantage of portability to different platforms. However, compared with web applications, they can still access device services which are restricted to access from inside mobile browsers. Hybrid applications consists of two parts – the web applications written in HTML, CSS, and JavaScript and hosted in WebView, and a wrapper that opens the WebView and provides JavaScript APIs for platform specific services such as an accelerometer or bluetooth. As web technologies are becoming well standardized and familiar to many developers, it makes the development of hybrid applications more rapid and easier to maintain. Hybrid frameworks like Apache Cordova let one build the application for more than one platform just by adding a line of code. Developers do not like to get locked into proprietary platforms and hybrid applications allow them to reuse their existing web development skills. The main problem with hybrid applications is their performance. As mentioned in the previous section, the problem is becoming less significant as devices are becoming more powerful and JavaScript interpretation engines are becoming smarter and more optimized.

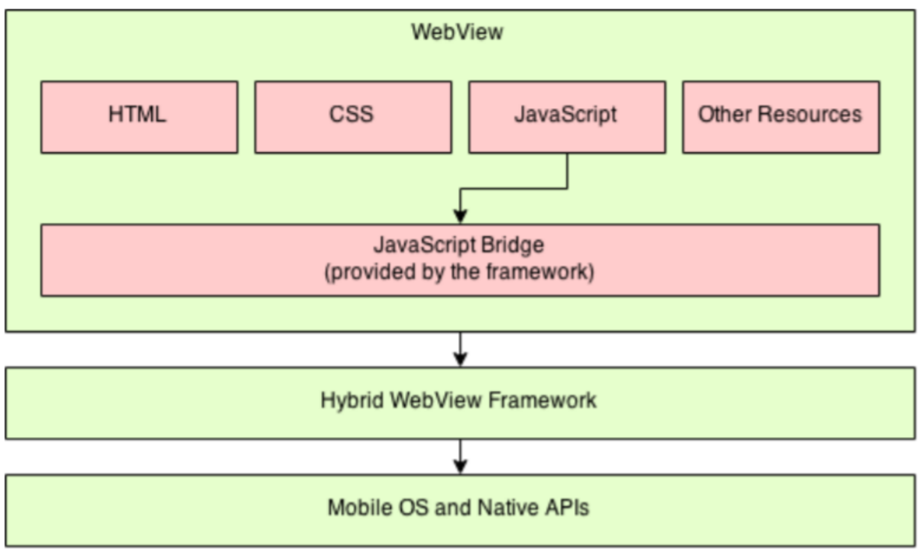


Figure 5.1 Architecture of a hybrid application

5.2.3 Hybrid WebView Frameworks

Apache Cordova / PhoneGap

Apache Cordova2 is an open source hybrid framework for building native mobile applications using HTML, CSS, and JavaScript. In a nutshell, it is a set of APIs that allow JavaScript functions that map to native code or plugins to be called, creating a bridge from web into native code.

Cordova provides powerful low-level API allowing developers to do almost anything a native application can do. It comes with a set of pre-made easy-to-use plugins to access, for example, the camera or compass. The provided APIs are consistent across multiple device platforms, supporting iOS, Android, Blackberry, Windows Phone, Palm WebOS, Bada, and Symbian.

Cordova has a large active community and provides more than 900 third-party plugins. Following is a list of the out-of-the-box plugins:
• Battery Status – Monitor the status of the device's battery.
• Camera – Capture a photo using the device's camera.
• Console – Add additional capability to console.log().
• Contacts – Work with the devices contact database.
• Device – Gather device-specific information.
• Device Motion (Accelerometer) – Tap into the device's motion sensor.
• Device Orientation (Compass) – Obtain the direction that the device is pointing.
• Dialogs – Visual device notifications.
• FileSystem – Hook into native file system through JavaScript.
• File Transfer – Hook into native file system through JavaScript.
• Geolocation – Make the application location-aware.
• Globalization – Enable representation of objects specific to a locale.
• InAppBrowser – Launch URLs in another in-app browser instance.
• Media – Record and play audio files.
• Media Capture – Capture media files using device's media capture applications.
• Network Information (Connection) – Quickly check the network state and cellular network information.
• Splashscreen – Show and hide the application's splash screen. • Vibration – An API to vibrate the device.
• StatusBar – An API for showing, hiding and configuring the status bar background.

It is necessary to mention the difference between Apache Cordova and PhoneGap as these two names are sometimes confused. PhoneGap is a former product of Nitobi startup that was acquired by Adobe in 2011. Shortly after the acquisition, Adobe decided to donate PhoneGap's source code to the Apache Software Foundation under the name Cordova and continued building proprietary services around its PhoneGap ecosystem. Today, PhoneGap is a distribution of Apache Cordova where Cordova plays the role of an engine that powers Phonegap. In other words, PhoneGap is Cordova plus extra proprietary Adobe services.

5.2.4 Responsive Web Design and CSS Preprocessors

As we are planning to build a mobile application using web technologies, we need to build a solid foundation of tools that would help to create a design that looks great at any size. We want to build a responsive design that adapts to any screen size and delivers on great user experience across a wide range of devices (from desktop monitors to mobile phones).

It takes the following core ingredients to create a responsive design:

- A flexible, grid-based layout
- flexible images and media
- media queries, a module from the CSS3 specification.

I do not intend to get deeper into responsive web design, but I believe it is important to keep in mind the preceding ingredients when deciding on the technologies and tools we want to use. Based on my previous experience, creating a responsive web design for a large application can be a very challenging task resulting in complex CSS styles that grow in size and are hard to maintain. Therefore, we plan to use a CSS preprocessor and frameworks that ease the process of implementing web design in CSS.

A CSS preprocessor is a layer between the stylesheet that one authors and the .css file that is provided to the server. It extends the CSS language by adding features that make the development process easier and the CSS code maintainable.

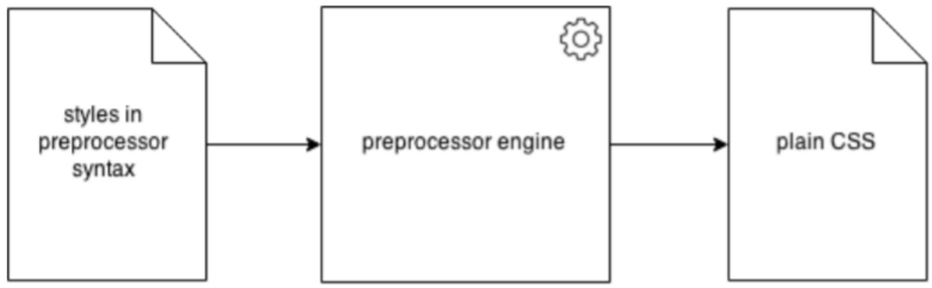


Figure 5.2 Compilation of styles from preprocessor syntax to plain CSS

Sass

Sass (Syntactically Awesome Style Sheets) is a meta-language on top of CSS that offers more power than is allowed by flat CSS. It comes with a simple and elegant syntax and features that are useful for creating manageable stylesheets. Moreover, Sass' syntax is a superset of CSS3, meaning any valid CSS3 document is also a valid Sass document. The Sass preprocessor runs on Ruby and uses a command-line program to compile Sass files, that usually use the .scss file extension, into CSS. Following is the list of features with example code snippets:

• variables – Variables are used to store any CSS value that could be reused throughout the stylesheets.

```
$primary-color: #333;
body { color: $primary-color; }
```

• nesting – CSS does not support nesting, Sass lets one nest CSS selectors in a way that follows the same visual hierarchy of HTML.

```
nav ul {
	margin: 0;
	padding: 0;
	list-style: none;
	li {
		display: inline-block;
		a {
			display: block;
			&:hover {
			color: #666;
			}
		}
```

```
    }
}
```

5.2.5 UI Frameworks

Ionic

Ionic is an HTML5 mobile application development framework that comes with very native-styled mobile UI elements. Although Ionic is a young framework with its first alpha being released in late November 2013, its becoming a very popular framework with a fast-growing community of developers.

Ionic's goal is to close the gap between what native SDK on iOS or Android provides and is not available on the web. It offers a set of beautifully designed UI elements, layouts, and animations that make the development of hybrid applications easy and fast. Besides this, Ionic is built on top of AngularJS, which allows for minimal DOM manipulations and hardware accelerated transitions, making Ionic applications fast. Ionic comes with Sass and Apache Cordova under the hood, allowing for easy UI customizations and integration with device native services.

Figure 5.3 Multilayer architecture of Ionic applications

5.3 Backend

In the first part of the Design chapter I described several frameworks and toolsets that can be used in the development of mobile hybrid applications. Those frameworks provide functionalities that can be used in the development of the frontend part of the application. In this part of the Design chapter, we decide on server-side technologies such as the computing platform, runtime environment, databases, and source code management.

5.3.1 Laravel

Laravel is a free, open-source PHP web framework, intended for the development of web applications following the model–view–controller (MVC) architectural pattern. Some of the features of Laravel are a modular packaging system with a dedicated dependency manager, different ways for accessing relational databases, utilities that aid in application deployment and maintenance, and its orientation toward syntactic sugar.

Few key features of laravel are:

- Restful controllers provide an optional way for separating the logic behind serving HTTP GET and POST requests.
- Blade templating engine combines one or more templates with a data model to produce resulting views, doing that by transpiling the templates into cached PHP code for improved performance.
- Migrations provide a version control system for database schemas.

5.3.2 MySQL

MySQL is the most widely used open-source relational database management system, which enables the delivery of reliable, high-performance, and scalable applications.

There are several reasons why we decided to use MySQL over other relational databases:

• MySQL is a reliable and proven technology that is definitely going to meet the demand for our application (we do not expect more than a few hundreds of concurrent users, nor a big amount of user generated content).

• Object-relational mapping (ORM) package.

We choose a relational database as the primary data source over a non-relational one because the relational database enforces a schema (imposes integrity constraints on a database) that ensures that the data persists in a declared format.

5.4 Beacon-Based Point Positioning

Here are three values that describe the power of a Beacon's signal: Broadcasting Power, RSSI and Measured Power. Broadcasting Power is the power with which the Beacon broadcasts its signal, i.e., the power with which the signal leaves the Beacon's antenna. The owner of the Beacon can change this setting. The value ranges between −30 dBm and +4 dBm, the lowest to the highest power settings, respectively. The higher the power, the bigger the Beacon's range and the more stable the signal, but it also shortens the battery life of the Beacon. RSSI is the strength of the Beacon's signal as seen on the receiving device, e.g., a smart phone. In general, the greater the distance between the device and the Beacon is, the lesser the strength of the received signal. This inverse relation between the distance and RSSI is used to estimate the approximate distance between the device and the Beacon using another value defined by the Beacon standard: Measured Power. Measured Power is a calibrated value which indicates the expected RSSI at a distance of one meter to the Beacon, called txPower. Combined with RSSI, this allows estimating the actual distance between the device and the Beacon. For example, we can measure a bunch of RSSI measurements at known distances, do a best fit curve to match the data points and convert the best fit curve into an algorithm. In an ideal environment, this method is able to guarantee accurate positioning. However, since RSSI has been affected by multiple factors including signal reflection, scattering and diffraction, large errors will arise in practice. Therefore, it will be very hard to secure a 100% accurate distance measurement based on this principle. Meanwhile, when we are using Bluetooth for the positioning, we just want to define an approximate position range.

Hexadecimal value	TX Power level	Decimal value	RSSI @ 1 metre	Range (metres)[1]
e2	0	-30 dBm	-115 dBm	2
ec	1	-20 dBm	-84 dBm	4
f0	2	-16 dBm	-81 dBm	10
f4[2]	3	-12 dBm	-77 dBm	20
f8	4	-8 dBm	-72 dBm	30
fc	5	-4 dBm	-69 dBm	40
00	6	0 dBm	-65 dBm	60
04	7	4 dBm	-59 dBm	70

Figure 5.4 Available transmission power values

5.5 Summary

This chapter covered many design considerations. In the first part I wrote about different mobile development approaches and decided to pursue the hybrid approach. Then we decided to use Apache Cordova.

Laravel(PHP) and MySQL is used as the backend.

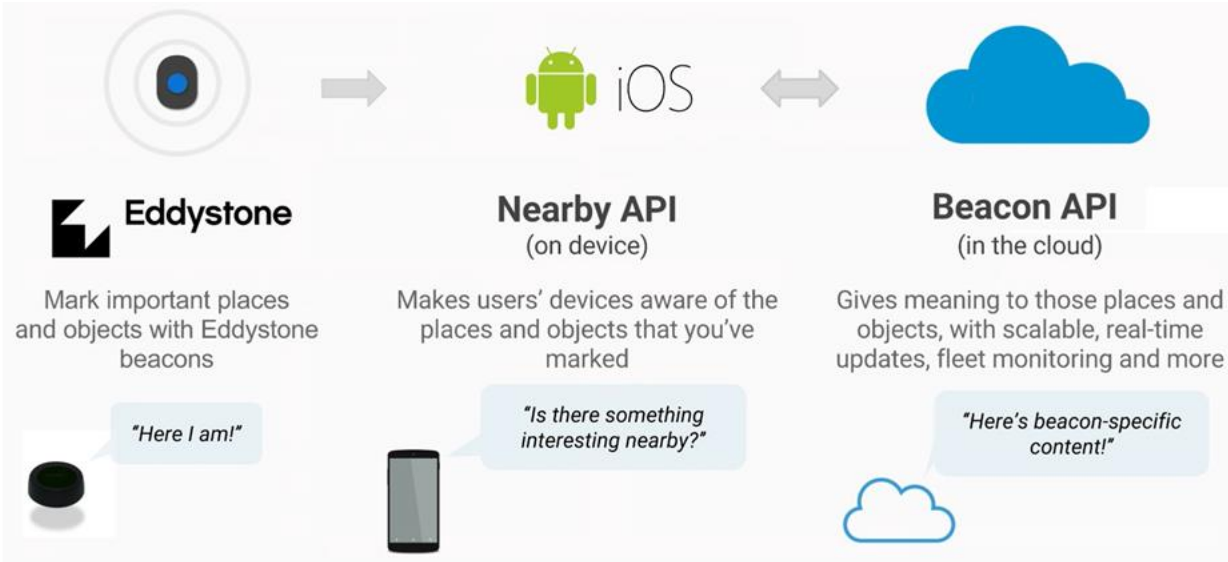


Figure 5.5 Simple design summary of the whole system

Chapter 6

6. Implementation

6.1 Introduction

This chapter covers the implementation phase of the application development lifecycle. All technology used during implementation was discussed in the Design chapter. The following subsections demonstrate how the most challenging parts of the application were developed.

6.2 Web Panel

Web panel is developed using Laravel framework. This is for the administrators. Retailers get to view and analyze customer behavior from this.

6.2.1 Features of the web admin panel

6.2.2

- **Manage beacons**

  Add new beacons to the system with their coordinates, floor area, beacon name, nid and tx power. Once it’s synced with the app the app knows where to place the beacons and identify them when it’s on

- **Manage floor**

  A map of the floor can be uploaded with the length and the width of the floor. Every time the app is opened it’ll load the map from the server.

  You can define areas of the map by giving there coordinates, for example if it’s the vegetables area we get the coordinates like it’s shown in the below diagram.

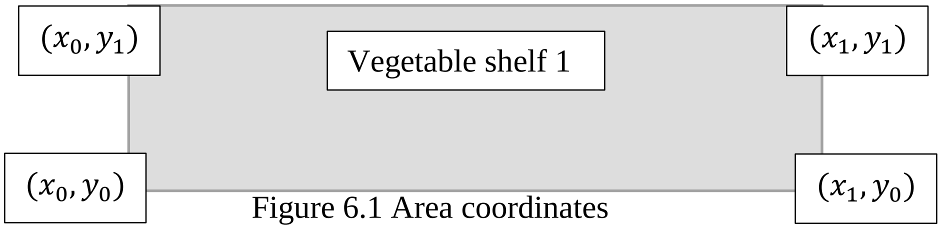


Figure 6.1 Area coordinates

When you know the coordinates of each area and the current position of the customer then you can easily direct customer towards the area he wants.

Figure 6.2 Example of a complex floor map

- ❖ **Inventory management**

  Products in store can be added to the inventory by specifying the area.

  Item name, category, price and area are saved in the database

- ❖ **User management**

  There are 2 main types of users, Web admins and customers.

  Under web admins there's a super admin and department admins who are responsible for updating the inventory and areas.

Customer's register through our app and their profiles can be viewed from the web admin panel.

❖ **Analysis**

Consumer behavior is analyzed and could be filtered based on the gender, age and profession.

We save the location of the user every 5 seconds.

Our charts indicate how many consumers visit each area monthly and daily, what areas they spent most time at.

Retail store administrators will be able to make decisions by looking at our analytics like whether to move some of their items to a certain area and what changes they must make to get more customers spend more time.

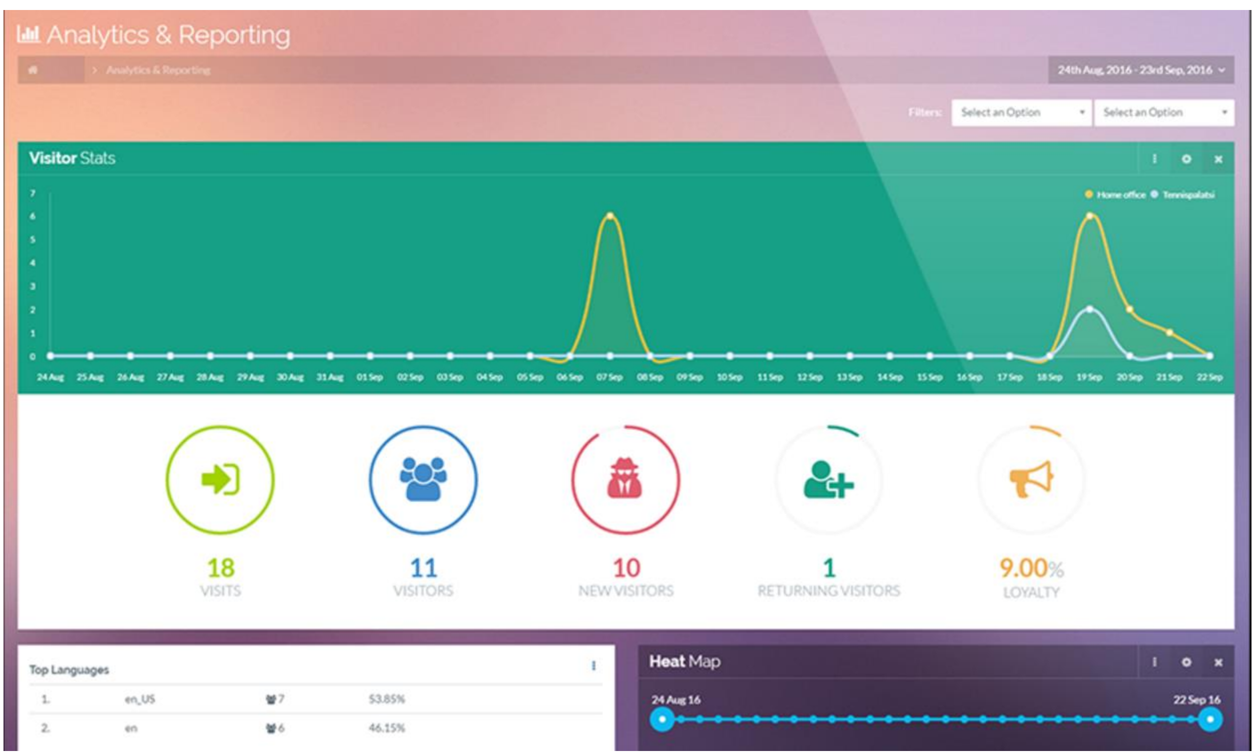


Figure 6.3 Consumer behavior analysis – visitor chart

6.3 Mobile App

Mobile app is implemented using apache cordova we discussed much about it in the Design chapter. We can build both android and iphone app from this framework but in this research, we only build and test the android app since we are using an eddystone beacon.

6.3.1 Mobile app features

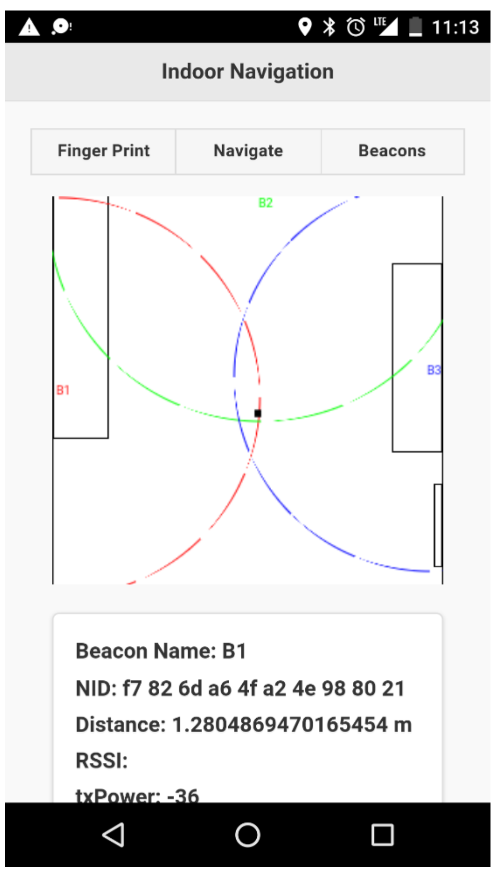
- **Identifying the current position**

  App will receive the signals from beacons. Each UID frame consists of the RSSI, Tx power, beacon name and the nid.

  Once the app retrieves frames from the 3 beacons those values are stored in 3 arrays until 10 rssi values for each beacon are retrieved, then those values are sent to a filtering function which uses Kalman filtering technique.

  We use trilateration to calculate distance and identify the position of the device.

  More about Kalman filter and Trilateration will be discussed in the later part of this chapter.



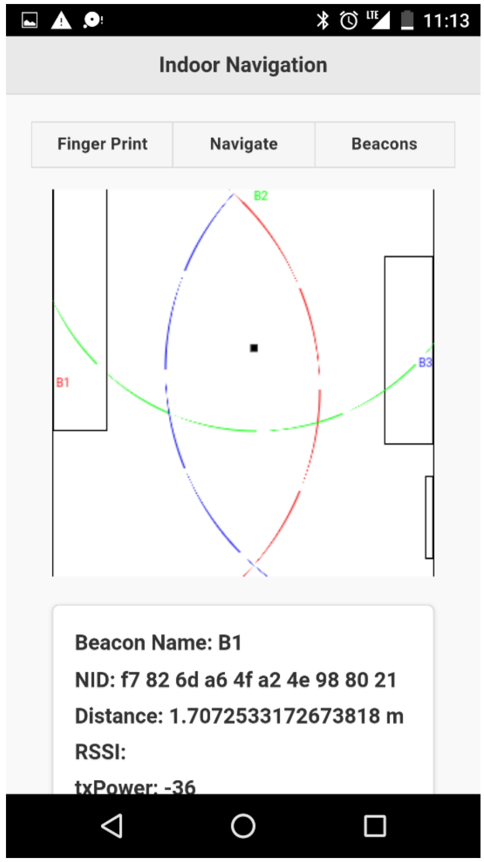


Figure 6.4 User location on mobile research app

- **Inventory search**

  Users can search items from the app and add to their list and they will be displayed on the map of the app.

  Items are synced with the server through the api.

  Searched items will also be stored for analysis.

- **Promotions and notifications**

  Notifications about promotions and discounts will be displayed on the app.

  Users can go to the promotions section of the app and checkout what's new.

  Retail store administrators can also set notifications to pop up when the user enters to a particular area. This is achieved by checking the distance to the device from a designated location. Store admin specifies a distance and if the user comes within that radius he will receive the notification.

The core Eddystone frames: -UID, -TLM, -URL

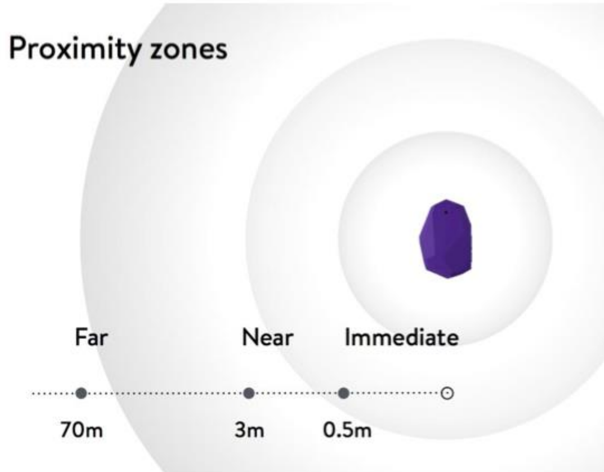


Figure 6.5 Proximity Zones

6.4 Identifying user position

User position is decided by the broadcasting values received from the 3 beacons.

They need to be filtered and the distance to each beacon should be calculated and then we use trilateration to get the position of the device.

Estimating proximity/distance

```
eddystone.calculateAccuracy = function(txPower, rssi)
{
        if (!rssi || rssi >= 0 || !txPower)
        {
                return null
        }

        // The beacon distance formula uses txPower at 1 meters, but the
Eddystone
        // protocol reports the value at 0 meters. 41dBm is the signal loss
that
        // occurs over 1 meter, so we subtract that from the reported txPower.
        var ratio = rssi * 1.0 / (txPower - 41)
        if (ratio < 1.0)
        {
                return Math.pow(ratio, 10)
```

```
        }
        else
        {
                var accuracy = (0.89976) * Math.pow(ratio, 7.7095) + 0.111
                return accuracy
        }
}
```

6.4.1 Kalman Filter-based Smoothing

Kalman filter is an estimator for a linear-quadratic problem. It is a set of mathematical equations which provides an efficient computational (recursive) way of estimating the state of a process in a way that minimizes the mean of the squared error. The filter estimates the instantaneous state of a linear dynamic system perturbed by white noise. It can estimate past, present, and future states even when the precise behavior of the modeled system is unknown. The typical uses of Kalman filters include smoothing noisy data and providing estimates of parameters of interest.

It uses very small memory resources since it only saves the previous state. And it is also suitable for solving real time problems since it's very fast and the calculation is recursive, so new values can be processed as they arrive.

Code segment from the kalment filter functions used in the app

```
/**
* Create 1-dimensional kalman filter
* @param  {Number} options.R Process noise
* @param  {Number} options.Q Measurement noise
* @param  {Number} options.A State vector
* @param  {Number} options.B Control vector
* @param  {Number} options.C Measurement vector
* @return {KalmanFilter}
*/
var R = 1; // noise power desirable
var Q = 1; // noise power estimated

var A = 1;
var C = 1;
var B = 0;
var cov = NaN;
var x = NaN; // estimated signal without noise

/**
* Filter a new value
* @param  {Number} z Measurement
* @param  {Number} u Control
```

```
* @return {Number}
*/
function filter(z, u = 0) {

    if (isNaN(x)) {
        x = (1 / C) * z;
        cov = (1 / C) * Q * (1 / C);
    } else {

        // Compute prediction
        const predX = (A * x) + (B * u);
        const predCov = ((A * cov) * A) + R;

        // Kalman gain
        const K = predCov * C * (1 / ((C * predCov * C) + Q));

        // Correction
        x = predX + K * (z - (C * predX));
        cov = predCov - (K * C * predCov);
    }

    return x;
}

/**
* Return the last filtered measurement
* @return {Number}
*/
function lastMeasurement() {
    return x;
}

/**
* Set measurement noise Q
* @param {Number} noise
*/
function setMeasurementNoise(noise) {
    Q = noise;
}

/**
* Set the process noise R
* @param {Number} noise
*/
function setProcessNoise(noise) {
    R = noise;
}
```

6.4.2 Trilateration

In geometry, trilateration is the process of determining absolute or relative locations of points by measurement of distances, using the geometry of circles, spheres or triangles. In addition to its interest as a geometric problem, trilateration does have practical applications in surveying and navigation, including global positioning systems (GPS). In contrast to triangulation, it does not involve the measurement of angles.

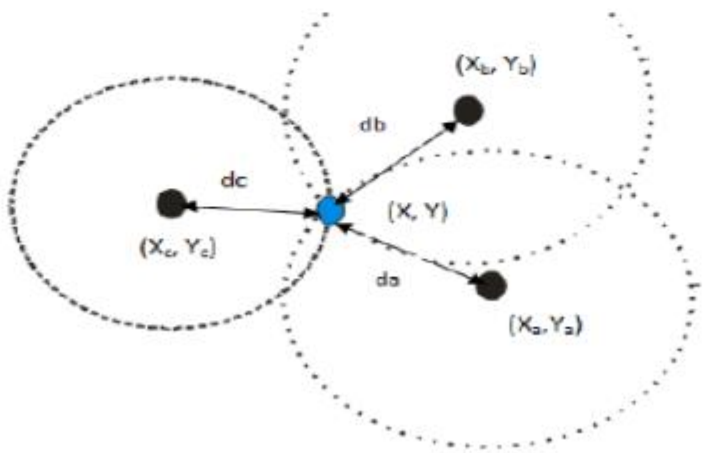


Figure 6.6 Trilateration Algorithm

Code segment of the trilateration function

```
/**
  * Calculates the coordinates of a point in 3D space from three known points
  * and the distances between those points and the point in question.
  *
  * If no solution found then null will be returned.
  *
  * If two solutions found then both will be returned, unless the fourth
  * parameter (return_middle) is set to true when the middle of the two
solution
  * will be returned.
  *
  * @param {Object} p1 Point and distance: { x, y, z, r }
  * @param {Object} p2 Point and distance: { x, y, z, r }
  * @param {Object} p3 Point and distance: { x, y, z, r }
```

```
  * @param {bool} return_middle If two solution found then return the center
of them
  * @return {Object|Array|null} { x, y, z } or [ { x, y, z }, { x, y, z } ] or
null
  */
function trilaterate(p1, p2, p3, return_middle)
{
        // based on: https://en.wikipedia.org/wiki/Trilateration

        // some additional local functions declared here for
        // scalar and vector operations

        function sqr(a)
        {
                return a * a;
        }

        function norm(a)
        {
                return Math.sqrt(sqr(a.x) + sqr(a.y) + sqr(a.z));
        }

        function dot(a, b)
        {
                return a.x * b.x + a.y * b.y + a.z * b.z;
        }

        function vector_subtract(a, b)
        {
                return {
                        x: a.x - b.x,
                        y: a.y - b.y,
                        z: a.z - b.z
                };
        }

        function vector_add(a, b)
        {
                return {
                        x: a.x + b.x,
                        y: a.y + b.y,
                        z: a.z + b.z
                };
        }

        function vector_divide(a, b)
        {
                return {
                        x: a.x / b,
                        y: a.y / b,
                        z: a.z / b
                };
        }

        function vector_multiply(a, b)
```

```
        {
                return {
                        x: a.x * b,
                        y: a.y * b,
                        z: a.z * b
                };
        }

        function vector_cross(a, b)
        {
                return {
                        x: a.y * b.z - a.z * b.y,
                        y: a.z * b.x - a.x * b.z,
                        z: a.x * b.y - a.y * b.x
                };
        }

        var ex, ey, ez, i, j, d, a, x, y, z, b, p4;

        ex = vector_divide(vector_subtract(p2, p1), norm(vector_subtract(p2,
p1)));

        i = dot(ex, vector_subtract(p3, p1));
        a = vector_subtract(vector_subtract(p3, p1), vector_multiply(ex, i));
        ey = vector_divide(a, norm(a));
        ez =  vector_cross(ex, ey);
        d = norm(vector_subtract(p2, p1));
        j = dot(ey, vector_subtract(p3, p1));

        x = (sqr(p1.r) - sqr(p2.r) + sqr(d)) / (2 * d);
        y = (sqr(p1.r) - sqr(p3.r) + sqr(i) + sqr(j)) / (2 * j) - (i / j) * x;

        b = sqr(p1.r) - sqr(x) - sqr(y);

        // floating point math flaw in IEEE 754 standard
        // see https://github.com/gheja/trilateration.js/issues/2
        if (Math.abs(b) < 0.0000000001)
        {
                b = 0;
        }

        z = Math.sqrt(b);

        // no solution found
        if (isNaN(z))
        {
                return null;
        }

        a = vector_add(p1, vector_add(vector_multiply(ex, x),
vector_multiply(ey, y)))
        p4a = vector_add(a, vector_multiply(ez, z));
        p4b = vector_subtract(a, vector_multiply(ez, z));

        if (z == 0 || return_middle)
```

```
        {
                return a;
        }
        else
        {
                return [ p4a, p4b ];
        }
}
```

6.5 Summary

This chapter we have described in detail, implementation of our Indoor Navigation process and how system works. With a clear Idea about what we have been doing during this study, we move to next chapter which describe how system works in practice.

Chapter 7

7. Evaluation

7.1 Introduction

In previous chapter the implementation details of the research is described. In this chapter, we will be discussing about how the implemented solution was evaluated. We will discuss how the filtering techniques helped in improving accuracy and how trilateration works.

7.2 Interval and signal ability

In our own analysis, we have determined that higher interval settings (over 700ms) cause major issues with signal stability.
We've determined that 350MS offers the perfect balance between signal stability and battery life for the Kontakt.io Beacon preset – which is why we use this setting as our default option.

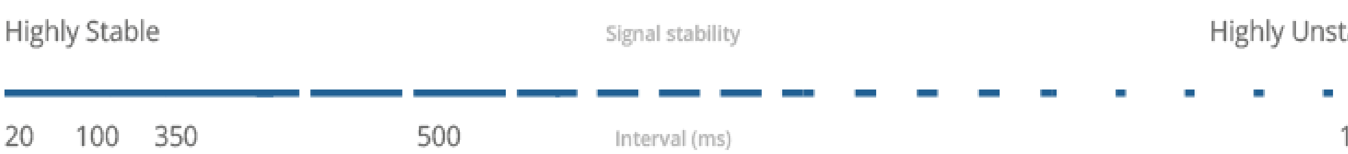


Figure 7.1 Transmission interval vs stability

7.3 Kalman filter smoothing

7.3.1 Raw values

The received RSSI are unstable even in a well-controlled indoor scenario due to multipath fading. Figure 7.1 shows the received RSSI that beacon is 1m away and facing directly to the mobile device. As shown in the figure, RSSI varies vigorously from -61dB to -80dB. Thus outliers of RSSI needs to be removed before any further process.

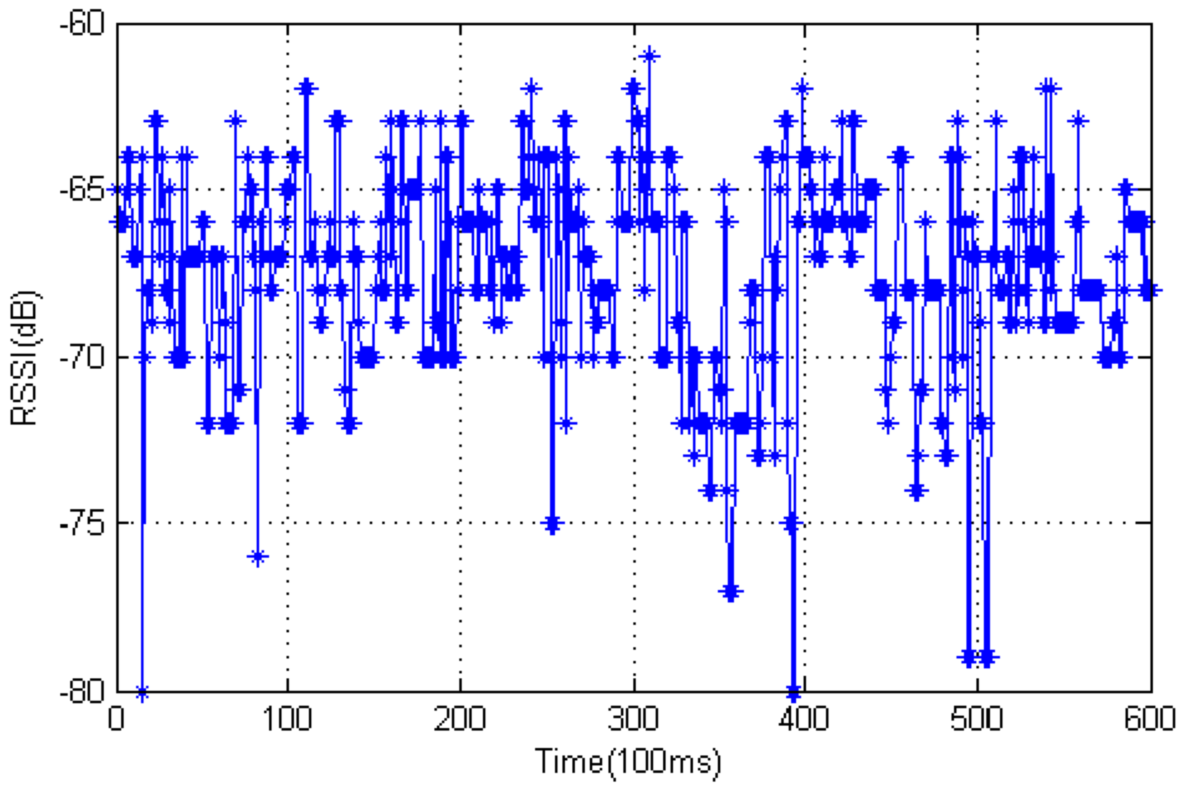


Figure 7.2 RSSI values in 1m distance

7.3.2 Proposed solution

Ten recent RSSI values are stored, 2 lowest values and 2 highest values are removed. Then the rest are sent to the Kalman filter function and the distance between beacon and cellphone is calculated.

Figure 7.3 shows the result of pre-processed RSSI, where received RSSI is measured by facing mobile device directly to the beacon in 1m distance. The received RSSI has variance of 9.86dB, where pre-processed RSSI only has variance of 3.35dB. As observed in the figure, filtered RSSI has less spikes in the data curve.

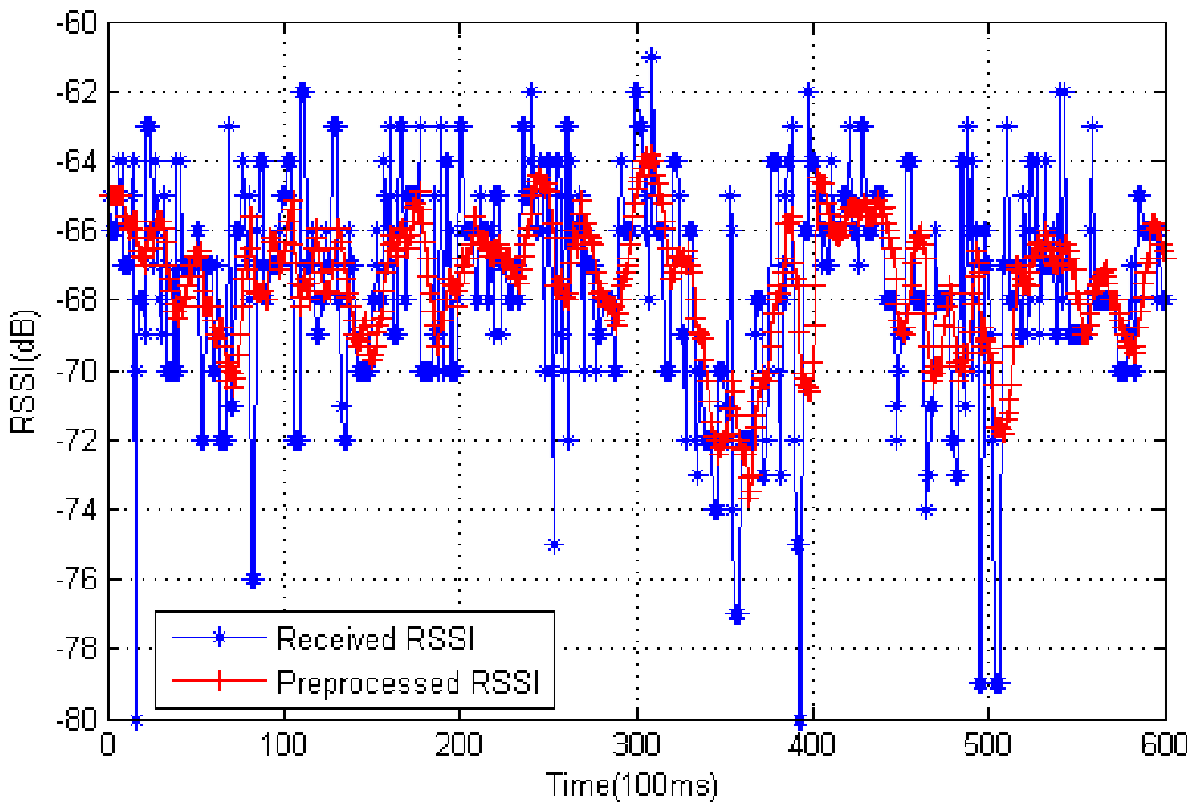


Figure 7.3 Filtered RSSI values in 1m distance

7.4 Application of Trilateration

Once mobile device knows distance from three known beacon, trilateration is performed to determine its coordinates. Three circles, centered at each beacon with radius equals to the distance between each beacon and mobile device are drawn in Figure 7.4. The trilateration location is the centroid of the triangle ABC, which is consisted of cords of the intersection part of three circles.

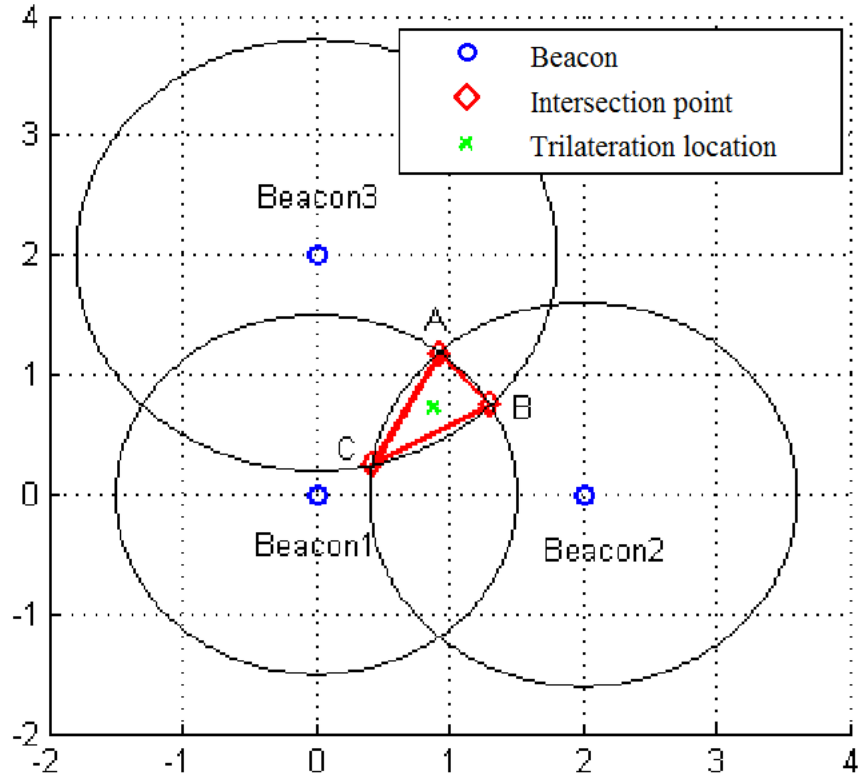


Figure 7.4 Trilateration

7.5 Experiments and results

Our application was tested in a 4.5m x 5m room and our app achieves an average positioning accuracy of 0.2~0.4m. Notice that in Figure 7.5 (b), calculated locations are grouped into two clusters. The upper left cluster to the actual location is caused by unstable transmitting power of beacons.

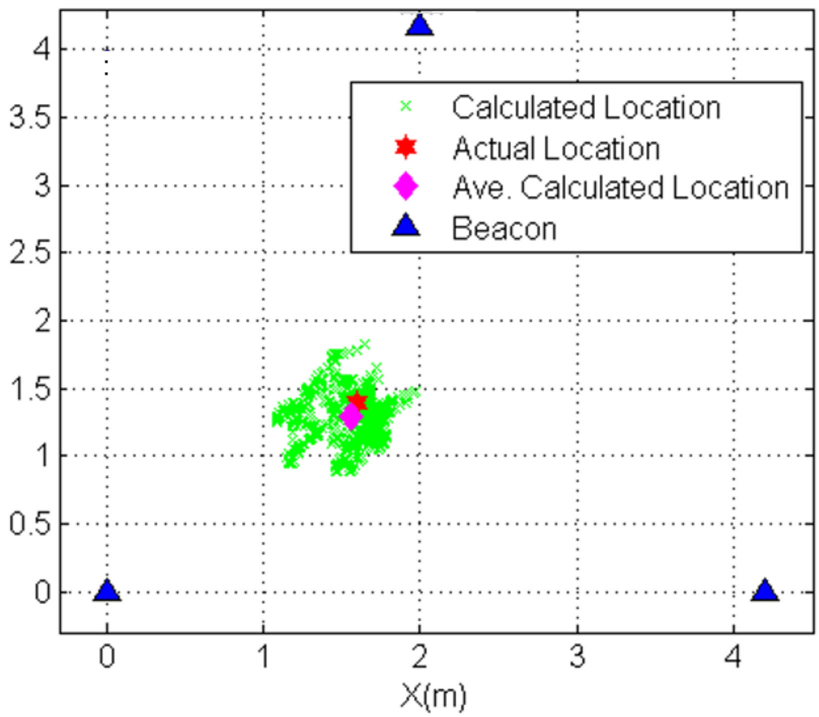


Figure 7.5 (a) Mobile device located at (1.6, 1.4)

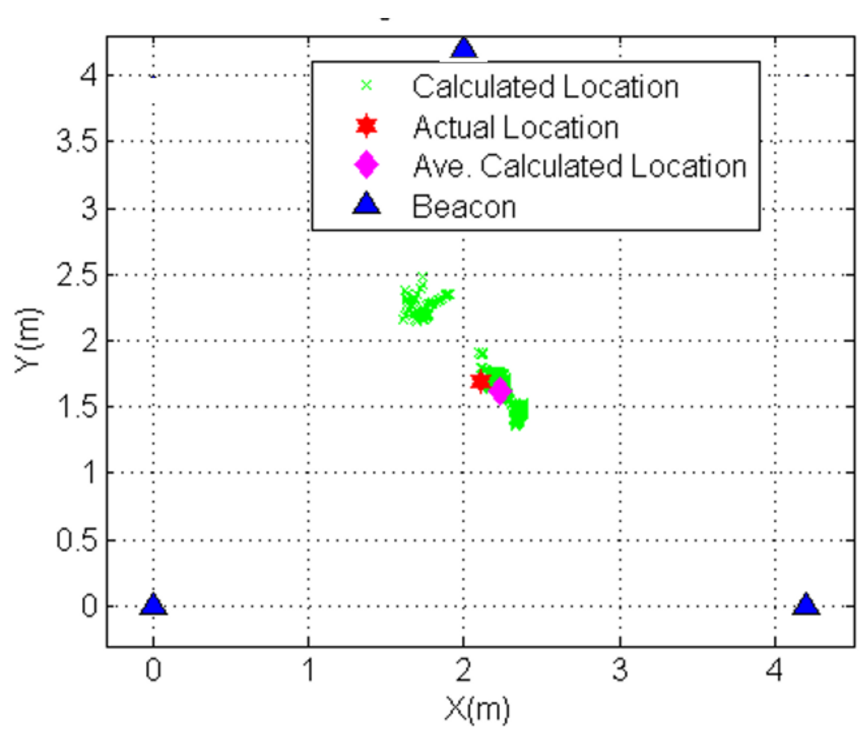


Figure 7.5 (b) Mobile device located at (2.3, 1.7)

7.6 Summary

In this chapter, we fully discussed about the evaluation and the testing of the system. The purpose of the evaluation was to ensure that the system was developed according to the aim and objectives. Conclusion and further work is discussed in the next chapter.

Chapter 8

8. Conclusion & Further work

8.1 Introduction

This chapter contains a discussion on the results obtained from the research and how the objectives are met. The future works can be done to continue this research for make the broader environment is also discussed.

8.2 Accuracy and performance

We discussed about the accuracy based on our experiment in the previous chapter. However, the findings are that the trilateration could not be performed, because of the lack of accuracy in the measure of the distance from receiver to beacon. There seem to be four main reasons to that lack of accuracy.

1. The signal strength of each beacon fluctuates a lot even when the beacon, the receiver and the testing environment are stationary. The noise of the signal data from the beacons was so random that I couldn't smooth the data to capture important patterns in the data set.

2. Depending on the way that the user is handling the receiver device, the readings can change a lot as well.

3. The state and configuration of the beacons such as battery life, advertising interval and broadcasting power impact the readings.

4. The readings also change a bit with the receiving device.

To achieve more accuracy we need to install more beacons. Minimum distance between 2 beacons should at least be 5m.

8.3 Future work

Even though the results are somewhat accurate more and more testing needs to be carried out and a hybrid solution should be given to make the navigation smoother.
In order to make this app and the webpanel more user-friendly and interactive we have plans to implement below ideas.

- Show previously searched results.
- Display records of previous shopping lists.
- Allow multiple people to do shopping together at the same time so that they can save time.
- An interactive drawing tool for admins to draw maps and add locations.

8.4 Summary

At the end of the research we realize that presence detection is good with this system, but precise positioning needs to be more advanced. This chapter contained a conclusion and future works of this research and consists of a summary of this report. It provided explanations on problems encountered during implementation and further work to be done.

References

Kimmo Kalliola April, 2011. High Accuracy Indoor Positioning Based on BLE.

Tushar Gothivarekar, Ajay Motwani, Chaitanya Pathak, Sunil Yadav, Ajay Perupalli, Pramod Gauda. Indoor Navigation System.

Marco Altini , Davide Brunelli , Elisabetta Farella , Luca Benini .
Bluetooth indoor localization with multiple neural networks. Published in
Wireless Pervasive Computing (ISWPC), 2010

Koyuncu H., Yang S.H. A Survey of Indoor Positioning and Object Locating Systems. Int. J.
Comput. Sci. Netw. Secur. 2010;10:121–128.

Giaglis G.M., Pateli A., Fouskas K., Kourouthanassis P., Tsamakos A. On the Potential Use
of Mobile Positioning Technologies in Indoor Environments; Proceedings of the 15th Bled
Electronic Commerce Conference eReality: Constructing the eEconomy; Bled, Slovenia. 17–19 June 2002; pp. 419–423.

Gu Y., Lo A., Niemegeers I. A Survey of Indoor Positioning Systems for Wireless Personal
Networks. IEEE Commun. Surv. Tutor. 2009;11:13–32. doi: 10.1109/SURV.2009.090103.

Mautz R. Overview of Current Indoor Positioning Systems. Geodesy Cartogr. 2009;35: 18–22. doi: 10.3846/1392-1541.2009.35.18-22.

Liu H., Darabi H., Banerjee P., Jing L. Survey of Wireless Indoor Positioning Techniques and Systems. IEEE Trans. Syst. Man Cybern. Part. C Appl. Rev. 2007;37:1067–1079. doi: 10.1109/TSMCC.2007.905750.

Drane C., Scott C. Positioning GSM Telephones. IEEE Commun. Mag. 1998;36:46–59. doi: 10.1109/35.667413.

Renaudin V., Yalak O., Tomé P., Merminod B. Indoor Navigation of Emergency Agents. Eur. J. Navig.2007;5:36–45.

AlZuhair M.S., Najjar A.B., Kanjo E. NFC Based Applications for Visually Impaired People—A Review; Proceedings of the IEEE International Conference on Multimedia and Expo Workshops; Chengdu, China. 14–18 July 2014; pp. 1–6.

Jin G., Lu X., Park M. An Indoor Localization Mechanism Using Active RFID Tag; Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and

Trustworthy Computing; Taichung, Taiwan. 5–7 June 2006.

Ni L.M., Liu Y., Lau Y.C., Patil A.P. LANDMARC: Indoor Location Sensing Using Active RFID; Proceedings of the First IEEE International Conference on Pervasive Computing and Communications; Fort Worth, TX, USA. 23–26 March 2003; pp. 407–415.

Ivanov R. Indoor Navigation System for Visually Impaired; Proceedings of the International Conference on Computer Systems and Technologies; Sofia, Bulgaria. 17–18 June 2010; pp. 143–149.

Liu, H., Darabi, H., Banerjee, P., Liu, J.: Survey of wireless indoor positioning techniques and systems. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 37(6), 1067–1080 (2007)

Liu, Y., Wang, Q., Liu, J., Wark, T.: Mcmc-based indoor localization with a smart phone and sparse wifi access points. Pervasive Computing and Communications Workshops, IEEE International Conference on 0, 247–252 (2012)

Liu, Y., Dashti, M., Zhang, J.: Indoor localization on mobile phone platforms using embedded inertial sensors. In: WPNC. pp. 1–5. IEEE (2013)

Bellot Arias, S.: Visual tag recognition for indoor positioning (2011)

Wang, H., Sen, S., Elgohary, A., Farid, M., Youssef, M., Choudhury, R.R.: No need to war-drive: Unsupervised indoor localization. In: Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services. pp. 197–210. MobiSys ’12, ACM, New York, NY, USA (2012), http: //doi.acm.org/10.1145/2307636.2307655

https://www.bluetooth.com