Location based content delivery solution using iBeacon

Shaffat Shahriar

Shaffat Shahriar
Spring 2015
Bachelor’s thesis, 15 hp
Supervisor: Lars-Erik Janlert
Abstract

There is a growing trend of using mobile devices in retail stores. Access to instant product related information is desirable by consumers in making a purchase decision. This information may include offers or sales discount associated with a product, comparing prices, ingredients or materials composition and many more. Retail business owners have seen this as a big opportunity to increase their in-store sales and they are adapting various technologies to deliver helpful contents and information to consumer.

This thesis project aims to design and develop a location based dynamic content delivery system for the consumers of retail stores and supermarkets. A system architecture is proposed here which makes use of iBeacon to capture customer’s location based contextual information and then deliver relevant contents back to the customer.

In order to design and develop the system, iBeacon technology and in-store environment were studied in depth. Choice of techniques and technologies that can be used to develop such a system is demonstrated here along with the important factors to consider while putting the system in use.

It is concluded that in the right circumstance this design can be adapted to implement a context aware mobile application for stores and supermarkets.
Acknowledgement

I would first like to thank Mr. Dipak Surie for giving me the chance to work with this interesting project in Smartmiljöös. I also want to extend my heartfelt gratitude to my supervisor Lars-Erik Janlert whose guidance was very vital in completing this project. A special thanks to Marie Nordström and Pedher Johansson at Umeå University for their academic guidance.

I also thank my parents for their unceasing encouragement, moral support and motivation all along the way. And finally a special note of thanks to Hoa Vo whose support was required in many different ways to accomplish this project.
List of figures and tables

Table 1: Beacon’s UUID, Major, Minor
Figure 1: iBeacon proximity states
Table 2: potential sources of Bluetooth interference
Figure 2: iBeacon Advertising packet
Table 3: Description of iBeacon data fields
Table 4: List of some popular Bluetooth beacons
Figure 3: Initial design of the system
Figure 4: Entity Relationship Diagram (ERD)
Figure 5: Final design consideration
Figure 6: Distance readings over time in open space
Figure 7: Distance readings over time in the intended environment
Figure 8: Distance readings over time at 0.5m
Figure 9: Distance readings over time at 2m
Figure 10: Notification screen
Figure 11: Discount and Offer Screen
### List of abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROI</td>
<td>Return On Investment</td>
</tr>
<tr>
<td>UUID</td>
<td>Universally Unique Identifier</td>
</tr>
<tr>
<td>RSSI</td>
<td>Received Signal Strength Indication</td>
</tr>
<tr>
<td>JSON</td>
<td>JavaScript Object Notation</td>
</tr>
<tr>
<td>API</td>
<td>Application Program Interface</td>
</tr>
<tr>
<td>REST</td>
<td>Representational State Transfer</td>
</tr>
<tr>
<td>PDU</td>
<td>Protocol Data Unit</td>
</tr>
<tr>
<td>ERD</td>
<td>Entity Relationship Diagram</td>
</tr>
<tr>
<td>URI</td>
<td>Uniform resource identifier</td>
</tr>
<tr>
<td>BI</td>
<td>Business Intelligence</td>
</tr>
</tbody>
</table>
Chapter 1: Introduction

According to a market research conducted by Deloitte Consulting LLP in 2012, the mobile-influenced (Smartphone and Tablets) in-store retail sales are forecasted to be 689 billion U.S. dollar in the US by the year 2016, overtaking e-commerce. Mobile-influenced in-store sales increased from 159 billion US dollar in 2012 to 593 billion dollar in 2013, representing 19% of all in-store sales [1]. What exactly is mobile-influenced sales? Deloitte defined mobile influenced sales as “use of mobile device before or during in-store shopping trips influenced in a sale” [2]. This may include looking up product information, checking for a receipt, making product comparison and many other activities. From these statistics it is clear that mobile devices are playing a very important role in in-store sales. The business owners have seen this as a big opportunity and are focusing in finding innovative ways to reach potential consumers.

From the consumers' point of view, they expect to get instantly any information they need. A consumer will expect to get the right content and service at the right moment. If the business owner can ensure that, the consumer will be satisfied and most likely decide to purchase. According to another survey conducted by JiWire, 53% of consumers are willing to share their current location to receive more relevant advertising [3]. So it can be said that consumers have individual needs and this need can be addressed only by understanding a consumer one-to-one basis. This would mean understanding the consumer’s preference, location, time of an activity and other contextual information. The more contextual information a business has on its consumers, the better position it is going to be for providing contents and information that are relevant to the consumers individually. The mobile trend indicates that mobile devices will play the role of in-store assistant providing interactive personal shopping experience [4].

1.1 Problem Description

Smartmiljö is an IT company located in Umeå that is researching and developing innovative products that will connect customers to retail businesses. Smartmiljö seeks to enhance the in-store shopping experience of customers through its shopping cart device. The device will run on a content and analytics platform to serve personalized, intelligent content and information that will enhance the customers shopping experience. The same application can also be downloaded on customers’ smartphones or tablets.

My role at Smartmiljö is to develop a system that can present context-based information to the users, more particularly to the customers of retail stores and supermarkets. The task involves developing a mobile application that will be able to sense user’s context through iBeacon technology and deliver relevant contents based on the user's context. The idea in general is, when the user takes the shopping cart or the mobile device to a particular location in the supermarket and comes close to the proximity of pre-installed beacon, the application will be able to deliver some contents to the user about a product or products located in that region. A detailed analysis was performed on the project requirements, tasks and underlying subtasks.

1.2 Project Goal

After analyzing the requirements and discussing with the CEO of Smartmiljö, two primary goals for this project were identified. They are,

1. Investigating the advantages and limitations of using iBeacon to provide location-based services.
2. Proposing a design and developing a prototype application that will use iBeacon to deliver location-based content.

1.3 Related Work

Over the past couple of years many companies understood the advantages of indoor location-based marketing and its impact on sales and ROI and came forward with their own innovative solution. Some of the popular commercial systems in the market today are listed below

**Swirl**
Swirl offers solution for creating, managing and analyzing proximity-based mobile campaigns for retailers and advertisers with operations in North America. Swirl claims to be an advertising technology innovator that will power the next generation of mobile advertising. One of their product, similar to Smartmiljös, is Swirl Ad Exchange providing beacon powered mobile advertisement solution. The company has also developed a platform for the retailers and brands to engage with in-store shoppers [5].

**inMarket**
inMarket has already deployed its beacon technology to around 200 stores in US in the beginning of 2014. Its service includes, personalized promotional push notifications to the shoppers, shopping list and delivering digital coupons and deals to the shoppers. [6].

**SmartBlink**
SmartBlink is a German company which created a beacon platform and coupon toolkit for mobile app which allows a customer to receive offers and promotion directly to their app when they are in the proximity of a store equipped with SmartBlink’s beacon [7].

A research on using smartphone to interact with smart objects in the stores was done in Linköping University, Sweden. In his report ‘Physical Mobile Interaction in Omni-Channel Retailing’ [8] the author examined some of the techniques that can be used to interact with objects in a store using technologies like Barcode, QR Code, NFC and iBeacon. Although no design solution was proposed, the author gave a good overview of omni-channel customer experience and how a customer can start their purchase in one channel and complete it in another.

1.4 Methodology

The choice of method in doing this project work was highly influenced by the paper ‘The experimental paradigm of Software Engineering’ by Victor R Basili [9]. In his paper the author described a couple of methods that can be followed to perform software development research. Among them ‘The Engineering method’ is one approach where, the existing solutions are observed, new solution is developed, tested and evaluated and at last the whole process is repeated for further improvements. This method is interpreted and applied to this project in the following way,

**Literature Study**
An extensive literature study have been done to accomplish this project. The main purpose of the literature study is to get a clear idea on context aware system and to learn about the technologies that can be used to design such a system. Part of the literatures are scientific articles from IEEE and ACM Digital library. Existing commercial systems are also tested and reviewed as much as possible. The proposed system is based on iBeacon technology and therefore it is necessary to understand this technology in depth. I have access to 4 beacons purchased by Smartmiljös to develop the system. The book ‘Building Applications with iBeacon’ [10] is therefore a very valuable resource. After understanding the theoretical concepts, few experiments are done in order to practically evaluate the beacons’ performances.
**Design and Prototyping**
Potential techniques and technologies are identified from the theoretical studies and practical observations. Later these techniques and technologies are combined to come up with a solution design and a prototype is developed.

**Testing and Evaluating**
After the prototype of the system is developed then it is necessary to test the system and evaluate its performance. The test results are analyzed and necessary modifications are taken into consideration to further improve the system.
Chapter 2: Literature Review

2.1 Context Awareness

The definition of context aware computing is provided by many researchers in many different context aware computing literatures. Among them, one of the early pioneer to propose Context aware computing is Bill Schilit. According to Schilit,

“Context aware computing is the notion of having computer system being aware of the context. To be aware of the context means that the system can recognize and perceive the real world using sensors and can act upon those stimulate that it gets from the real world” [11].

One of the important aspect of context aware computing is to detect location specific information [12]. With the rapid growth of technology, a user is no longer confined within the four walls of home or office operating a desktop computer, rather a typical user is widely exposed to many devices in many different environments and states, mobile or fixed. Thus with time context aware computing evolved into context aware mobile computing and later context aware application and so on. With the evolution of smart devices along with the exponential growth of computing power, a developer can create and deploy a context aware application almost in any location or situation.

Another definition of a context aware system is given by the author Anind K. Dye [13]. According to the author,

“A system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task”.

This definition of context was explained in details in his paper and it resulted in categorization for features of a context aware application. The main features a context aware system may provide are as follows:

1. **Presentation of information and services to a user** - providing information to the user based on the user's contextual information or current context. For example, an application to find pizza restaurants nearby a user's location.

2. **Automatic execution of a service** - starting a service automatically on detecting the context.

3. **Tagging of context to information for later retrieval** - associating digital data with user's context.

In respect to the above definitions we can see that, an important property of a context aware system is to sense the context and capture contextual data from the sensors. Sensing and capturing context is a broad term and can be expanded into a research topic. This project is concerned with location based contextual information and the particular area of interest is indoor location or micro location more specifically.

Bluetooth sensors are used in various ways in order to sense the environment and capture the context data related to proximity. Bluetooth beacon is an example of such a sensor.
2.2 iBeacon

2.2.1 Beacon and iBeacon

Beacons are basically any transmitting devices that transmit radio signal at some interval. It is often compared with a lighthouse, where the job of the lighthouse is to provide navigational aid to ships and boats that are within its vicinity. Bluetooth beacons typically use low energy Bluetooth also known as BLE (Bluetooth Low Energy). BLE is a short range radio technology which is optimized for ultra-low power consumption. Bluetooth beacons broadcasting these BLE signal can be discovered by smart devices such as smartphones or tablets residing within the signal’s range. These beacons are powered by batteries having a typical battery life from 6 months to 2 years or by a fixed power source [14].

iBeacon is Apple’s trademark, introduced in mid-2013, that refers to protocols devices and uses of BLE to create user experience. Apple has to certify each vendor that wants to carry the iBeacon name, which means, the vendor has access to Apple’s trademark and their beacons are configured to work well with the Apple devices. This however does not pose any restriction for the iBeacon to be used by Android or other devices [15].

Along with the signal transmitted from iBeacon, a packet of data is sent. This packet, often referred as advertisement packet broadcasted periodically by the iBeacon, typically containing information about who they are and where they are. Basic information provided by beacon are UUID, Major and Minor.

Major and Minor values can be used to deploy several beacons which suits the layout of a store or supermarket. These values are provided by the manufacturer of the beacons and can easily be edited using the manufacturer’s SDK. Table 1 shows how UUID, Major and Minor values can be used [19].

<table>
<thead>
<tr>
<th>UUID</th>
<th>Major</th>
<th>Minor</th>
</tr>
</thead>
<tbody>
<tr>
<td>a unique ID to distinguish one (or a group of) beacon from another (or another group)</td>
<td>an optional arbitrary numeric value, used to specify subgroup within a group of beacons</td>
<td>another optional numeric value used to identify specific beacon within a group or subgroup</td>
</tr>
</tbody>
</table>

*Table 1: Beacon’s UUID, Major, Minor*

The two major functions provided by iBeacon are geofencing and proximity. A geofence is a virtual perimeter around a geographic area [16]. A network of beacons can be used to achieve this geofencing property. A mobile application can be developed to detect if the device is inside this geofence perimeter or not and execute some particular function based on that condition. Proximity refers to the quality of nearness. Beacons can be used to identify how far or how near a receiving device (a smartphone) is. For example, a mobile application can perform certain function when the device is within the proximity of 1.5 meter of a beacon.

2.2.2 iBeacon and Proximity

Actual location is not provided by iBeacon, instead a relative distance is calculated based on RSSI. RSSI is the strength of a beacon’s signal as seen on a receiving mote (a smartphone for example) expressed as negative number in unit of decibel (dB) [17].

In general, the greater the distance between the device and the beacon, 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 receiving device and the beacon, using another value defined by the iBeacon standard known as Measured Power. Measured Power (also known as TxPower) is a factory-calibrated, read-only constant which indicates what the expected RSSI is at a distance of 1 meter to the beacon. Combined with RSSI, this allows to estimate the actual distance between the device and the beacon [18].
The literature study shows that distance estimate changes when the condition or environment changes. Even in the very ideal condition the RSSI value does not remain constant due to the factors involving signal path loss as described in the Section 2.2.3. The radio signal from the beacon is subjected to reflect, absorbed and vulnerable to interference from other radio signal of similar frequency. Physical objects and materials can also block signals, reducing the RSSI. The human body itself is an excellent attenuator of Bluetooth signals. Simply having ones back to a beacon will affect the signal strength and thereby lower the accuracy [19]. Due to these factors RSSI could potentially be used as an indicator rather than an accurate measurement of the distance at which the sending mote, a beacon for example, is located from the receiving mote, such as smartphone [20].

Apple already considered these issues and came up with a unique distancing idea, determining the estimated proximity based on three states [19] as shown in Figure 1.

![Figure 1: iBeacon proximity states](image)

**Immediate** - This represents a high level of confidence that the device is physically very close to the beacon, usually few centimeters.

**Near** - With a clear line of sight from the device to the beacon, this would indicate a proximity of approximately 1-3 meters

**Far** - This state indicates that a beacon device can be detected but the confidence in the accuracy is low generally depicting a distance of 10m or more.

### 2.2.3 Indoor Path Loss

Practically a beacon's range is affected by obstacles such as walls, furniture or people around it. All of these can weaken a beacon's signal, which is why the measured distance is a rough estimate rather than an accurate value. These obstacles results in signal *path loss*.

Path loss (or path attenuation) is the reduction in power density (attenuation) of an electromagnetic wave as it propagates through space. In BLE Beacons this path loss happens due to several factors such as increasing distance between the transmitter and the receiver. Beside this, another reason for signal loss is *multipath propagation* which comprises the factors like reflection, diffraction and scattering of signal [20]. Signal interference is also another major reason for uneven RSSI values and is likely caused by other transmitters (Wi-Fi and Bluetooth) and barriers [21]. Apple has done its own research and identified some materials that can cause interference [22] as shown in Table 2.
<table>
<thead>
<tr>
<th>Type of Barriers</th>
<th>Interference Potential</th>
</tr>
</thead>
<tbody>
<tr>
<td>Metal</td>
<td>Very High</td>
</tr>
<tr>
<td>Concrete</td>
<td>High</td>
</tr>
<tr>
<td>Wood</td>
<td>Low</td>
</tr>
</tbody>
</table>

*Table 2: potential sources of Bluetooth interference*

### 2.2.4 iBeacon Advertising Packet

iBeacon uses the BLE advertising packet to transmit data about its identifier. The data is embedded in the BLE’s PDU [23] as shown in Figure 2.

In the BLE advertising packet, there are several fields that are important for understanding the data being transmitted:

- **Preamble**: Contains advertisement Length, advertisement type, company identifier and beacon code. Example: `0x02011061AFF4C000215`

- **UUID**: Contains the Proximity UUID (Universal Unique identifier), identical for all the beacons owned by the same organization. Example: `f7826da6-4fa2-4e98-8024-bc5b71e0893e`

- **Major**: Used to group the beacons into some kind of logical area. Example: `10`

- **Minor**: Further grouping and sequential ordering of the beacons. Example: `349`

- **TxPower**: Each iBeacon must be calibrated with this TxPower value to allow accurate distance estimates. Example: `-40`

*Figure 2: iBeacon Advertising packet*

The description of the fields in iBeacon advertisement data is summarized in Table 3.

<table>
<thead>
<tr>
<th>Field</th>
<th>Size</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preamble</td>
<td>9 bytes</td>
<td>Contains advertisement Length, advertisement type, company identifier and beacon code.</td>
<td><code>0x02011061AFF4C000215</code></td>
</tr>
<tr>
<td>UUID</td>
<td>16 bytes</td>
<td>Contains the Proximity UUID(Universal Unique identifier), identical for all the beacons owned by the same organization</td>
<td><code>f7826da6-4fa2-4e98-8024-bc5b71e0893e</code></td>
</tr>
<tr>
<td>Major</td>
<td>2 bytes</td>
<td>Use to group the beacons into some kind of logical area.</td>
<td><code>10</code></td>
</tr>
<tr>
<td>Minor</td>
<td>2 bytes</td>
<td>Further grouping and sequential ordering of the beacons.</td>
<td><code>349</code></td>
</tr>
<tr>
<td>TxPower</td>
<td>1 byte</td>
<td>Each iBeacon must be calibrated with this TxPower value to allow accurate distance estimates.</td>
<td><code>-40</code></td>
</tr>
</tbody>
</table>

*Table 3: Description of iBeacon data fields*
2.2.5 Popular Bluetooth beacons

<table>
<thead>
<tr>
<th>Brand</th>
<th>Description</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>BlueSense</td>
<td>It is powered by a single coin-cell battery and can last up to 2 years while operating continuously 24 hours/7 days a week. web - <a href="http://bluesensenetworks.com/">http://bluesensenetworks.com/</a></td>
<td>300 SEK</td>
</tr>
<tr>
<td>Estimote</td>
<td>Each beacon consists of a Bluetooth Low Energy module that broadcasts information within a range of 70 meters (approx. 200 feet). Estimated battery life is up to 24 months. web - <a href="http://estimote.com/">http://estimote.com/</a></td>
<td>250 SEK</td>
</tr>
<tr>
<td>Gimbal</td>
<td>Gimbal’s secure beacons transmit information to mobile devices from 50 meters down to inches. Typical battery life of 3 months in its default configuration. Size 40mm x 28mm x 5.5mm web - <a href="https://www.gimbal.com/">https://www.gimbal.com/</a></td>
<td>45 SEK</td>
</tr>
<tr>
<td>Kontakt</td>
<td>Up to 2 years of battery life with default settings and 24-hours daily usage web - <a href="http://kontakt.io/">http://kontakt.io/</a></td>
<td>250 SEK</td>
</tr>
</tbody>
</table>

Table 4: List of some popular Bluetooth beacons

The following areas are worth considering while choosing a beacon

1. **Battery Life** - In order to get the optimal performance from a beacon a steady power source is needed. A steady power source would mean a continuous steady broadcast and thus a great experience for the customers. From the above list of beacons it can be seen that typical battery life spans from 3 months to 2 years depending on the brand of beacon.

2. **Size** - Beacons come in all shapes and sizes. If it is required to hide the beacon from public’s view, a smaller beacon would be ideal in that case. The actual chip inside a beacon is quite small in size, it is often the battery and casing that take up more space thus making the size little bigger.

3. **Security** - Security is a major concern for beacons as the signal is often public and open to other receivers. Special security features (as stated in the Section 5.4) should be of high priority while choosing a beacon for commercial use.
4. **Software support** - Most beacons are equipped with vendor's SDK, depending on the nature of use, this SDK can be very useful and handy to develop the software that uses the beacon. Having a robust management platform can ease deployments and assist in managing the fleet of beacons remotely.

Considering the above factors Smartmiljös bought 6 beacons from Kontakt.io to develop and test the system.

### 2.3 Alternative Solution to iBeacon

IndoorAtlas is a Helsinki based indoor position company whose core technology is based on magnetic positioning. It figures out the unique signature of a building by measuring the unique signals produced by its steel girders. Once a single space is mapped, it can be shared with the whole community of users. IndoorAtlas uses the built in compass in smartphones and tablet and sense the distortion of the earth’s magnetic field to pinpointing user’s location. The primary advantage of IndoorAtlas platform is that, it is purely software based and does not require any additional sensors or infrastructure to operate, offering developers unprecedented scale and freedom when building location-based apps. [24]
Chapter 3: System Design

3.1 Overview

The system design is highly inspired from an existing research titled as 'Fundamental Design for a Beacon Device Based Unconscious Participatory Sensing System' [25]. The research paper proposes a beacon based environment sensing system. The proposed system requires participants to install an application on their mobile phone to scan neighboring beacons and store distance measurement information to a management server. The system architecture that I propose will also use a management server that will communicate a database server and the client application.

The first part of the system consists of a database to store the beacons and products information. These entities (beacons and products) will have certain relation among themselves which will be described using the database schema. From the analysis of iBeacon we have already seen that beacon’s UUID, Major and Minor values are the three most important attributes required to identify a beacon individually. Therefore multiple tables will be created to store data for all these attributes along with some additional attributes required to manage the beacons and the products. Section 3.5 shows in details the analysis on the choice of database and ERD.

The proposed architecture consists of a management server as stated before. The primary purpose of the management server is to manage the collection of beacons and store product related information. The beacon and product data will actually be stored in a database and the management server will provide an interface to communicate with the database and make queries. Using a management server will ease the process of adding and updating beacon and product information. This server will also query the database as requested by the mobile application and create a response along with the data in a format that is understandable by the mobile application. The primary advantage of using such a server is that the response, usually in XML or JSON format, can be processed by most of the major programming languages, thus giving the opportunity to create client applications for different platforms. If the company, for example, wants to develop a mobile application for IOS devices, it can simply connect to the management server, send a request for the resources and process the response in Objective C language. The same method is used here to develop the prototype for Android devices. More details on the management server can be found in Section 3.6.

Finally the system consists of a client application that the shoppers can download and use on their mobile phones or tablets. The mobile application for Android devices which will scan the environment for beacons using Bluetooth. Upon detection and identifying a beacon successfully, the application will send a request to the management server. This request will contain the beacon’s identifiers (described by UUID, Major and Minor sent from the beacon) and the type of data or resources required by the application. The management server will process this request, query the database for that data, format it and send a response back to the application. The application will then process this response and execute the required function. The implementation of the prototype is described in Chapter 4.
Figure 3 depicts the initial design:

![Initial design of the system](image)

### 3.2 The Environment

In order to gain an in-depth knowledge on retail store environment, two retail stores were studied for the purpose of this project.

**Environment 1: ICA Maxi**

ICA Maxi supermarket located in Strömpilen, Umeå was studied by visiting and observing it in person. ICA is one of the most popular supermarket in Sweden having almost 1400 retail stores of different sizes scattered all over the country [26]. Among them ICA Maxi type is considered as the biggest promising to simplify life by gathering everything under one roof. [28]. Summary of my observation crucial to the system design is as follows:

1. Upon entry, the current offers and sales discount are displayed on a LCD screen. Comparing to the dimensions and spaces, these screens are quite small and easy to miss by the customers entering the supermarket.

2. There are too many posters and banners hanging here and there giving some sort of information, for example, products’ prices or sales. If someone is not carefully reading those, it is easy to miss what it says.

3. There are individual sections where each category of the product is found. There are separate sections for vegetable, meat, chocolate, chips, tools etc. Each section contains similar types of products. There are numbers of shelves or containers where the products are divided into sub categories and labeled.

**Environment 2: JC Umeå**

The second store that was studied is a fashion and clothing store called JC also located in Umeå. It is a small store housing different fashion accessories such as men’s and women’s clothes, sunglasses, undergarments etc. It is less organized in terms of categorizing the products, as I found men’s jeans in two different locations and shirts and t-shirts are hanging on the same holder. There are sales discount on certain items, but the customer has to locate the shelf or holder where it says ‘Sales’ and then check the price tag of each item one by one to see if there is a sale on that item.
3.3 Relevant Contents

After studying both the environments in depth it was found that one of the most relevant content for the customers of the above mentioned stores is information on sales discount and offers. Customers tend to show interest and gather in the places where there are discount or offers on products and items. Therefore it is concluded that the system will be developed to deliver contents to the user regarding these sales discount or offers that a store has.

3.4 Use case scenario

Given the time and resources, Environment 2 was chosen to develop this prototype for. A simple use case scenario was generated as described here,

1. Multiple beacons will be installed on the entry to the store.
2. Whenever a customer crosses the alley and comes to the proximity of one beacon a notification will be shown on the customer’s mobile phone or tablet that contains the client application.
3. The customer can then take out the phone or tablet and click on the notification to see the sales discounts and offers provided.

3.5 Database Design

A database is required to store all the data that the system will use. Thus designing the database played an important role in developing the entire system. MySQL database server was chosen for this purpose because of its speed and reliability. The information that the database will handle are the beacon information and the sales discount or offer information associated with the products. Further analysis was done to identify the entities, attributes and their relation and an ERD was generated as shown in Figure 4.

![Figure 4: Entity Relationship Diagram (ERD)](image)

3.6 Management Server

The role of the management server is to create the API and communicate with the database server. API stands for Application Programming Interface and it is basically something that allows one application to communicate with another. A REST API stands for Representational State Transfer which works pretty much like a website, we make a Request to the server and we get a response back over HTTP
The beauty of the REST API lies in the response message, which is also called *Representation* of a resource. The client is actually making an API request to access the resource that is available on the server. This resource can be any data or functionality identified by a specific *URI (Universal Resource Identifier)*. The resource in the server must be accessed using this URI by all the clients. This response message or resource representation can be presented to the client in different formats, usually in XML, JSON, HTML etc., that is readable by the client application. The main features of a RESTful API are [27]

1. REST API is platform and language independent.
2. Client directly access a resource using its URI.
3. The response message which the client gets is self-descriptive in nature containing both the data and the format.
4. The server do not contain any client state and each request is processed as a new one.
5. Restful API uses four main HTTP methods:
   a. **GET** - to retrieve a resource
   b. **POST** - to create a resource
   c. **PUT** - to update a resource
   d. **DELETE** - to delete a resource

For this application JSON was chosen as the response format as it is easily consumed by java and also in the human readable form. Due to the time constraint an existing API platform called *API Spark* [30] was used rather than creating an API platform from scratch. A choice had to be made between using APISpark and *Espresso Logic* to accomplish this. APISpark was finally chosen as it is free and based on a simple and easy to understand interface.

The stages involved in setting up this API were configuring the authentication (making the GET method public), importing the database to APISpark, configuring the relation between beacons and products and finally deploying the API.

A GET Request typically looks like this,

```
https://beekoon.apispark.net/v1/beekinfos/1
```

The response, if successful, will be,

```json
{
  "record": [
    {
      "beaconId": 1,
      "buuid": "f7826da6-4fa2-4e98-8024-bc5b71e0893e",
      "bmaj": 35788,
      "bmin": 47206,
      "storeName": "JC Umeå",
      "notifyMsg": "Sale discount at JC, check it out.",
      "products_by_beacon": [
        {
          "productId": 101,
          "beaconId": 1,
          "productName": "Denim Jeans",
          "salesImage": "http://webserver.com/client/1/files/101_sale.png",
          "active": 1
        },
        {
          "productId": 102,
          "beaconId": 1,
```
3.7 System Design

The final design of the overall system is shown in Figure 5.

![Diagram](image-url)
Chapter 4: Prototype Implementation

4.1 Android Application

The application prototype was built using Android Studio. Android studio is based on IntelliJ IDEA IDE from the company JetBrains[33]. Android studio is becoming popular amongst the Android developer community as well as the Open Source community. Google already announced it as their official IDE to develop android application and is recommended over using Android SDK plugin in Eclipse [34]. One of the best features of using Android Studio is its layout preview option. Its preview rendering is very fast and it becomes very easy to incorporate changes in the layout and view those changes instantly. Debugging and testing is also much easier and faster in Android Studio comparing to Eclipse, speeding up the overall workflow.

Smartmiljöö aims to develop a shopping cart that will have a device attached to it where this application will run. The next challenge was to select an android device to run the prototype application. It is important to mention that the device selected at this stage will be actually used in a real store. After comparing and testing several Android devices, LG Pad 7.0 was finally chosen to test the application prototype on.

4.2 Android Beacon Library

The format of iBeacon signal is Apple proprietary and it poses a challenge to write apps that will provide iBeacon functionality for Android devices. My study on related work reveals that the companies like Swirl and inMarket have their own versions of apps that runs on Android but those apps use the vendor’s API, often locked with the API key, and typically will work with the vendor’s beacons only. That means, even if we succeed in writing an app for android devices using official Android Bluetooth API, it will discard any beacon installed by vendors like Swirl or inMarket. On the other hand iOS does not have this issue as most of the beacon vendors have adapted or will adapt iBeacon protocol to support their beacons. As a result, just by using Apple’s CoreLocation library one can easily build a custom app for iOS devices to detect iBeacon. Android’s Bluetooth API has methods to detect beacons but it is very hard to implement for iBeacon. The solution to this problem is to use a third party library called Android Beacon Library.

Android beacon library is an open source third party android library providing APIs to interact with beacons. It allows Android devices to use beacons much like iOS devices do. An app can request to get notifications when one or more beacons appear or disappear. The library can be configured to work with different kind of beacons by using the library’s BeaconParser module. A BeaconParser may be used to tell the library how to decode a beacon’s fields from a Bluetooth LE advertisement by specifying what byte offsets match what fields, and what byte sequence signifies the beacon. [31]. For example, in order to detect all iBeacons, the BeaconParser can be configured to detect beacons with the layout, m:2-3=0215,i:4-19,i:20-21,i:22-23,p:24-24, which is the layout of iBeacon.

The code snippet below shows how Android Beacon Library works.

Monitoring for beacons

```java
public class MonitoringActivity extends Activity implements BeaconConsumer {
    private BeaconManager beaconManager;

    protected void onCreate(Bundle savedInstanceState) {
        beaconManager = BeaconManager.getInstanceForApplication(this);
        beaconManager.bind(this);
    }
```
public void onBeaconServiceConnect() {
    beaconManager.setMonitorNotifier(new MonitorNotifier() {
        public void didEnterRegion(Region region) {
            // send a notification that a beacon is nearby
        }

        public void didExitRegion(Region region) {
            // do something when the device exits a beacon range
        }

        public void didDetermineStateForRegion(int state, Region region) {
            // do something when the state changes from entry to exit
        }
    });

    // start monitoring a specific region
    beaconManager.startMonitoringBeaconsInRegion(new Region());
}

//start ranging a specific region
beaconManager.startRangingBeaconsInRegion(new Region());

Proximity / Ranging

public class RangingActivity extends Activity implements BeaconConsumer {

    private BeaconManager beaconManager;

    protected void onCreate(Bundle savedInstanceState) {
        beaconManager = BeaconManager.getInstanceForApplication(this);
        beaconManager.bind(this);
    }

    public void onBeaconServiceConnect() {
        beaconManager.setRangeNotifier(new RangeNotifier() {
            public void didRangeBeaconsInRegion(Collection<Beacon> beacons, Region region) {
                if (beacons.size() > 0) {
                    // get the beacon’s distance, connect to the server to see
                    // what resource is associated with that distance,
                    // fetch that data and show it
                }
            }
        });

        // start ranging a specific region
        beaconManager.startRangingBeaconsInRegion(new Region());
    }

This prototype application will always perform a background scan for the beacons. This scan will continue even if the user does not launch the application and bring it to foreground. This is one of the very useful features of android beacon library [32]. On detecting a beacon installed in the stores the application will send a notification to the user. Clicking on the notification the user will be able to launch the activity (screen) that shows the sale discount.
Chapter 5: Result and Discussion

Several tests were performed in order to validate the system. Here is a summary of the result obtained from the tests.

5.1 Testing the beacons

Using the Android Beacon Library, a test application was written to test the beacons in different environments. From the literature study it was evident that iBeacon signal behaves differently in different situation. In order to get an overview of the signal’s behavior the followings tests were performed.

Test 1 - Testing beacon in an open space
This test was performed in an outdoor open space, where there were apparently no obstacle and interference medium. A beacon was placed exactly 1 meter away from the Android tablet and distance reading on the tablet was recorded for 1 minute. Interval between each scan was set to 1 second. Graph in Figure 6 shows the actual readings of the distances over time.

![Figure 6: Distance readings over time in open space](image)

Test 2 - Testing beacon in the intended environment
Same test was then performed in the clothing store and the following result was obtained as shown in Figure 7.
Figure 7: Distance readings over time in the intended environment

The beacon was placed exactly 1 meter away from the device in both Test 1 and Test 2. Comparing the readings and the graphs in figure 6 and 7, we see that the signal deviation is more evident in the cloth store comparing to the open space. This is due to the obstacles posed by the materials and objects around the store and also partly because of multipath factors as stated in Section 2.2.3. We also see a sudden deviation in the signal at the initial phases, from 1sec to 7 sec. Thus it can be concluded that it takes a little while before the signal stabilizes.

Test 3 - Testing beacon at 0.5 m in intended environment
A beacon was placed at the entry to the store and the tablet was kept exactly 0.5 meter away from the beacon. The reading were recorded by the test app and the graph below was generated.

Figure 8: Distance readings over time at 0.5 meter
**Test 4 - Testing beacon at 2 m in intended environment**

Similarly to the previous test, the device was moved 2 meter away from the beacon that was placed at the entry to the store and the readings were recorded.

![Distance readings over time at 2 meter](image)

*Figure 9: Distance readings over time at 2 meter*

Comparing the graphs in Figure 8 and 9 we can see that the signal is more stable and accurate at closer distances. In Test 4 that actual distance was 2 meter between the android device and the beacon, but the readings hardly showed that. Whereas in Test 3 at 0.5 meter, even the signal was fluctuating initially, after 30 second, the reading was more accurate.

### 5.2 Testing the system as a whole

An overall system test was performed in the cloth store. Two beacons were placed at the entry of the store. When the Android tablet came in the proximity of one of the beacon, a notification was received. Clicking the notification started another activity which has the title of the store and details on the sales discount the store has to offer. The screen shots are shown in Figure 10 and 11.
5.3 Limitation of the system

There are some limitations noteworthy to the proposed system.

1. No Direction information - The beacon usually shows an estimation of the distance but does not specify in which direction the beacon is located. As a result, if it estimates that there is a beacon within 1 meter relative to the device, it could mean that the beacon can be located anywhere within a 1 meter radius.

2. Inaccurate Estimation - Estimation is often not accurate due to the factors leading to path loss as stated in the Section 2.2.3

3. Subsequent beacon detection - No action is implemented on detecting multiple beacons that are on the same distance relative to the device.

4. Devices – A great diversity of specifications are seen in various Android devices. They may have very different screen resolutions, memories, processing power and Bluetooth antennas. The signal strength varies a lot depending on the hardware and its manufacturer.

5. Minimum Android API – The requirement to run the client application is to have an android device that must run on Android 4.3 (API level 18) and above. BLE is not supported by the older APIs.

6. Security - The broadcasting of beacon identifier is unencrypted and open to public. The proposed system does not implement any security feature at the beacon level.
5.4 Security

The beacon itself is actually quite dumb, its job is to broadcast the signal which contains information about the beacon’s identifiers (UUID, Major, Minor etc). This identifier is pretty much open to the world and anyone can see it. The two major security threats associated with beacons are as follows [35]:

1. **Beacon Cloning** - Due to the open nature of beacon’s identifier, it is possible to detect a beacon and clone another beacon. Malicious message or contents then can be sent to the users by the cloned beacon located at a different place than intended.

2. **Beacon Hijacking** - It is relatively easy to hijack or piggyback someone else’s beacon network. The beacon broadcasts signal just like a radio does and it is possible to tune in the device (a Bluetooth scanner or Bluetooth discovery app for example) to read that signal. This poses a big risk of falling the beacon’s identity in the hand of intruders who can use it in their app to target customers and send their own contents.

To eliminate the mentioned security threats several strategies can be adapted. One of these strategies is making the beacon’s identifiers dynamic [36]. The identifiers that the beacons send are generally static which is vulnerable to security threats as discussed before. But now beacon manufacturer like Estimote have implemented the dynamic UUID that is, changing the identifiers on runtime, making it tough to clone and hijack beacons.

Finally, a consumer always has a fair amount of control. As the consumer installs the app that uses iBeacon, the consumer usually has prior knowledge on what that app does. The app that uses Bluetooth usually asks for permission to use the device’s Bluetooth and location service. Once that permission is manually granted by the user, the user knows exactly what is happening and why the user is getting a message. This is all about consumer allowing the app to feed them with the information. As a consumer if one does not like to receive these notifications and use the service, there is always an option to switch it off.

5.5 Privacy

Since the release of iBeacon in 2013 there are potential concerns about privacy among many stakeholders. The issue of the privacy surfaced when iBeacon started being used in indoor mapping and location tracking. With the growing use of iBeacon for these purposes there are growing concern on privacy, some of which are legitimate while others are misconceptions.

The concerns are legitimate as people had not been tracked indoor on mass scale before. On the other hand the misconceptions arises due to the technology being new and lack of knowledge of iBeacon technology among the consumers.

It is not a secret now that the store owners are gathering consumers’ location data for analytics and iBeacon is what making it possible to gather indoor location data by tracking consumers’ movement. This rises a big concern among the consumers about privacy. So how aggressively this movement can be tracked?

When a consumer is near a beacon, the app picks up the Bluetooth signal and depending on the context connects to the appropriate service such as displaying offers, discount or providing coupons. At the same time, the app can be configured (and is usually is configured) to register the device’s Bluetooth MAC address and saves user’s indoor location as soon on beacon detection. This location information can contain the proximity to that beacon, that means proximity to certain product or products that the user is in, thus getting an idea of what product or products the consumer is interacting with. When the user leaves the proximity of one beacon and goes to the next, the app can update the location and store that data again. Looking at detection history it is then possible to generate a movement pattern of the consumer. As a result the store owner has now the answer to ‘which’ (customer), ‘where’ (location) and ‘what’ (product).
In many cases, data collected by the store owners can be sold to third parties, who can misuse it in many ways. For example, it is clearly stated in inMarket’s privacy policy that

“Third parties, including advertisers and advertising services providers, may collect information about you. They may use Tracking Technologies in connection with our Service, which may include the collection of information about your online activities over time and across third-party sites or services. Their privacy policies, not ours, govern their practices.” [37]

The above mentioned issues more or less poses some sort of threats to privacy. But gaining a good understanding of iBeacon technology and following certain techniques as stated below reduces these risks to a great extent and can make it a technology worth enjoying.

1. Beacons are very simple devices whose job is only to send their identifiers that can be picked up by the mobile devices. It is then up to the device and the application to use that unique identifier to launch a function. Usually there are no other information that are stored in the beacon. So even if someone hacks the beacons, there is no way to access consumer’s private information from it.

2. Beacons are incapable to tracking a user if the user do not give his or her consent. A user has to perform several tasks in order to give that consent, such as, downloading the store’s app, activating the device’s Bluetooth, signing in to that store’s service, accepting their terms etc. Beacon cannot perform these steps automatically, thus leaving the power in the hand of customers [39].

3. This beacon technology is no more or less secure than signing into an ecommerce site and making a purchase [38]. Therefore it is always a wise decision to go through the privacy policy to see the store’s terms and conditions.

4. There is already a code of conduct to protect consumer’s privacy and many location tracking analytics companies signed the agreement to abide by the rules. The code requires stores to post signs warning for the customers if surreptitious tracking is going on [40].
Chapter 6
Conclusion

In this ever expanding age of information there is a need to reach relevant information in right time at right place. This can be facilitated by developing application using iBeacon technology to provide context-aware services in many different environments. The main goal of this project was to perform an in-depth research on iBeacon and design a system that will deliver contents to the shoppers based on the knowledge of their location. This goal was achieved through proposing a design solution and implementing a prototype that uses iBeacon technology to provide location based services to the customer of a clothing store. The different components for developing such a system was identified, evaluated, integrated and tested. It resulted in concluding that the mentioned technologies and techniques can be adapted to develop a commercial system that will provide context-based services to the consumers.

A research on iBeacon reveals the factors that should be taken into account while developing a location-based mobile application. One of the important conclusion drawn from the research is that iBeacon will provide an estimation of the location rather than the accurate location. This estimation can be improved by adapting certain techniques and methods but there will always be a margin of error while estimating the distance and location. The research also shows that there are certain ethical and security issues that should be taken into consideration while deploying such a system commercially. The results from the iBeacon tests will be a vital input to the version of the system that will be developed for commercial purpose. However further analysis on the factors leading to signal deviation is needed in order to identify the beacon deployment techniques for much better result. Through the successful implementation of the prototype application, it is proved that the mentioned technologies and components can be used for developing a full-fledged commercial system. There might be better options when it comes to the choice of techniques and technologies that are not stated here in this report.

6.1 Future work

A significant amount of effort is being given by Smartmiljös in developing the complete system. Work is in progress to address the issues with iBeacon and to come up with deployment techniques for optimal performance. The first priority for the development team is to solve the problems that occurs when detecting multiple beacons at the same distance relative to the device. A solution to this problem is expected to surface in the very near future. The system will be soon optimized to store user’s location information which will provide BI information and analytics to the retailers and store owners.

The system will be then integrated with two major components in the future.

1. **Shopping list** - A customer will be able to add items and products to the shopping list and the in-store offers, discount and promotion will be filtered according to the list and delivered to the user. In this way the system would able to focus in providing relevant content specific to a customer.

2. **Smart Recommendation system** - Based on the customer’s preference, location and shopping behavior the system will recommend certain offers and promotion to the customer.
Bibliography:


