

# **Multilingual Digital Signage using Computer Vision and Bluetooth Beacons**

A Thesis Presented to  
The Faculty of the Computer Science Department  
California State University Channel Islands

In (Partial) Fulfillment  
of the Requirements for the Degree  
Masters of Science in Computer Science

by  
Suhas Dwarakanath  
Advisor: Dr. Brian Thoms  
March 2019

© 2018  
Suhas Dwarakanath  
ALL RIGHTS RESERVED

**APPROVED FOR MS IN COMPUTER SCIENCE**

---

**Advisor: Dr. Brian Thoms** **Date**

---

**Dr. Michael Soltys** **Date**

---

**Dr. Jason Isaacs** **Date**

**APPROVED FOR THE UNIVERSITY**

---

**Dr. Osman Ozturgut** **Date**

## **Non-Exclusive Distribution License**

In order for California State University Channel Islands (CSUCI) to reproduce, translate and distribute your submission worldwide through the CSUCI Institutional Repository, your agreement to the following terms is necessary. The author(s) retain any copyright currently on the item as well as the ability to submit the item to publishers or other repositories.

By signing and submitting this license, you (the author(s) or copyright owner) grants to CSUCI the nonexclusive right to reproduce, translate (as defined below), and/or distribute your submission (including the abstract) worldwide in print and electronic format and in any medium, including but not limited to audio or video.

You agree that CSUCI may, without changing the content, translate the submission to any medium or format for the purpose of preservation.

You also agree that CSUCI may keep more than one copy of this submission for purposes of security, backup and preservation.

You represent that the submission is your original work, and that you have the right to grant the rights contained in this license. You also represent that your submission does not, to the best of your knowledge, infringe upon anyone's copyright. You also represent and warrant that the submission contains no libelous or other unlawful matter and makes no improper invasion of the privacy of any other person.

If the submission contains material for which you do not hold copyright, you represent that you have obtained the unrestricted permission of the copyright owner to grant CSUCI the rights required by this license, and that such third party owned material is clearly identified and acknowledged within the text or content of the submission. You take full responsibility to obtain permission to use any material that is not your own. This permission must be granted to you before you sign this form.

**IF THE SUBMISSION IS BASED UPON WORK THAT HAS BEEN SPONSORED OR SUPPORTED BY AN AGENCY OR ORGANIZATION OTHER THAN CSUCI, YOU REPRESENT THAT YOU HAVE FULFILLED ANY RIGHT OF REVIEW OR OTHER OBLIGATIONS REQUIRED BY SUCH CONTRACT OR AGREEMENT.**

The CSUCI Institutional Repository will clearly identify your name(s) as the author(s) or owner(s) of the submission, and will not make any alteration, other than as allowed by this license, to your submission.

---

Title of Item

---

3 to 5 keywords or phrases to describe the item

---

Author(s) Name (Print)

---

Author(s) Signature

---

Date

# Multilingual Digital Signage using Computer Vision and Bluetooth Beacons

Suhas Dwarakanath

April 27, 2019

## **Abstract**

Due to rapid globalization and changing lifestyle, more people are now visiting foreign countries for business and travel. However, important signages like traffic signs, safety signs and informational signs are displayed in only one language (usually the native language). There are many languages in the world and it is impractical to install signages in all the languages. In this research, by combining computer vision and bluetooth beacons, multilingual digital information is displayed on the user's smartphone. A mobile application is constructed which enables the user's smartphone to listen to nearby digital signages (with bluetooth beacons) and display the respective signage information in user's preferred language. The application also enables the user to capture an image of a printed signage, extract the information in the signage and translate the information to user's preferred language. Optical Character Recognition (OCR) is used to extract the text from the image and Google Translate is used for the language translation. This system was implemented in the university campus and experiments were conducted by to determine the feasibility of using this system on a larger scale. It was found that the system helps the users to understand their surroundings better in their preferred language.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Motivation . . . . .	1
<b>2</b>	<b>Background</b>	<b>3</b>
2.1	Related Work . . . . .	3
2.2	Outline . . . . .	6
2.2.1	GPS . . . . .	7
2.2.2	Wi-Fi . . . . .	7
2.2.3	Near Field Communication (NFC) . . . . .	7
2.2.4	Bluetooth . . . . .	8
2.2.5	iBeacon . . . . .	8
2.2.6	Comparison of indoor positioning methods . . . . .	8
<b>3</b>	<b>Understanding iBeacon Technology</b>	<b>10</b>
3.1	Existing applications implemented using iBeacon technology .	12
3.2	Operating range of iBeacons . . . . .	12
3.3	Calculating distance between iBeacons, based on RSSI . . . . .	12
<b>4</b>	<b>RESTful web services</b>	<b>14</b>
<b>5</b>	<b>Google Vision</b>	<b>15</b>
5.1	Sample Request . . . . .	17
5.2	Sample Response . . . . .	18
<b>6</b>	<b>Google Cloud Translation</b>	<b>19</b>
6.1	Sample Request . . . . .	20
6.2	Sample Response . . . . .	21
<b>7</b>	<b>Algorithm</b>	<b>22</b>
<b>8</b>	<b>System Design</b>	<b>24</b>
8.1	Environment setup . . . . .	25
8.1.1	Digital Signages . . . . .	26
8.1.2	Printed Signages . . . . .	26
8.2	Mobile application . . . . .	26
8.2.1	Background Monitoring . . . . .	27

8.2.2	Beacon Ranging . . . . .	27
8.2.3	Development stack . . . . .	27
8.2.4	Screenshots . . . . .	28
8.3	Database . . . . .	34
<b>9</b>	<b>User Experience Design (UX Design)</b>	<b>37</b>
9.1	Context Analysis . . . . .	38
9.2	Task Analysis . . . . .	38
9.3	User Analysis . . . . .	39
9.3.1	Example Persona . . . . .	39
9.4	Design Principles . . . . .	40
9.4.1	Affordance . . . . .	40
9.4.2	Visibility . . . . .	41
9.4.3	Constraint . . . . .	42
9.4.4	Consistency . . . . .	42
9.4.5	Feedback . . . . .	43
<b>10</b>	<b>Case study</b>	<b>45</b>
10.1	Testing environment . . . . .	45
10.2	Digital Signages . . . . .	45
10.2.1	Test Case 1 . . . . .	46
10.2.2	Test Case 2 . . . . .	48
10.3	Printed Signages . . . . .	49
10.3.1	Test Case 3 . . . . .	50
10.3.2	Test Case 4 . . . . .	52
<b>11</b>	<b>Conclusion and future work</b>	<b>57</b>
<b>References</b>		<b>60</b>

## List of Figures

1	iBeacon packet format . . . . .	10
2	Process flow diagram for Text Detection using Google Vision .	16
3	Example image for testing Google Vision API . . . . .	17
4	Process flow diagram using Google Translation . . . . .	20
5	System design process flow diagram. The diagram shows all the individual components of the system and how it is integrated together to achieve the objective. . . . .	25
6	Home page . . . . .	29
7	Text extraction . . . . .	29
8	Translated text . . . . .	31
9	Share text feature . . . . .	31
10	Type and translate text feature . . . . .	32
11	Digital Signage . . . . .	33
12	Translate feature . . . . .	33
13	Text translated to Spanish language . . . . .	34
14	Database Schema . . . . .	35
15	Affordance design principle . . . . .	41
16	Feedback design principle . . . . .	44
17	Digital signage 1 . . . . .	47
18	Digital signage 2 . . . . .	49
19	Printed Signage 1 . . . . .	51
20	OCR feature . . . . .	51
21	Text translated to English . . . . .	52
22	Printed Signage 2 . . . . .	54
23	OCR feature . . . . .	54
24	Text translated to English . . . . .	55

## List of Tables

1	Comparison of popular indoor positioning methods . . . . .	9
2	Testing environment . . . . .	45
3	Observations from test case 1 . . . . .	47
4	Observations from test case 2 . . . . .	50
5	Observations from test case 3 . . . . .	53
6	Observations from test case 4 . . . . .	56

# 1 Introduction

With changing lifestyle and globalization, more people visit foreign countries. Every country is working towards becoming tourism oriented to increase their economy. Most of the visitors use their smartphone to access information [1]. However, when visiting foreign countries, most people face difficulties in obtaining information due to difference in language, which makes it an inconvenience to stay in foreign countries for a longer time. [2]. Recently, a lot of refugee and migrant families from all over the world faced a lot of struggles at the U.S immigrations. A lot of these refugees faced problems with the documentation because they could not understand the content of the legal documents. It is natural for people to understand information better in their home language. Therefore, an effective method of providing and accessing information in multiple languages is required.

In this study, in order to solve such problems, a multilingual information service is developed using Computer Vision and Bluetooth beacons. The information can be accessed from the user's smartphone in most of the major languages. We focus on the smartphone's camera to 'see' information in multiple languages. We also use bluetooth beacons to 'push' information to smartphones in proximity. The user can then access the information in multiple languages. By using this method, we expect people visiting foreign countries to access information naturally in the same way they would in their home countries. This paper evaluates various options like GPS, NFC and RFID which could be used to provide information based on user's location. We then describe the required functions and configuration of the prototype system developed.

## 1.1 Motivation

When visiting foreign countries, it is essential to understand critical information around you. When visiting points of interest in a foreign land, many visitors would like to learn more about the place - the history, significance of the place and other critical information. In many cases, this information is displayed on signages, usually in the local language of the country.

In recent years, refugee families arriving in the United States of America are required to deal with a lot of new information including legal documents and signs. Many refugees would not be well versed with English and thus, there is a need for a system which can extract text from the documents and translate it to a language which the individual can understand.

This study is inspired by the above problems people are facing right now and aims to provide a low cost solution using computer vision and bluetooth iBeacon technologies.

## 2 Background

Text extraction and text translation are highly researched areas in the field of computer vision to model a smart self-learning system, which include text associated with banners, highway and roadside sign boards or the information on local transport system. This information in the form of text provide significant clues about that environment [3]. Text detection can also help those who are visually impaired and tourist alike to convey the right information in more understandable way [3].

As a method of accessing information on public signages, there are two ways of using a smart phone to find information in relevant languages; the user needs to search information by himself or herself, then translate it relevant language to access necessary information. Alternatively, when digital signage is used, the user can find and access information by coming in proximity of the digital signage without performing any special action. In this study, we explore ways to improve searching and accessing information by using smartphone camera and the method of information provision using the digital signage using iBeacons. This should make users who visit foreign countries get information without making extra efforts.

### 2.1 Related Work

A study of multilingual digital signage using iBeacon communication was done, for the purpose of evaluating the technology in an university campus at Japan [2]. In the experiment, the authors installed 25 iBeacon devices at sightseeing spots, restaurants, souvenir shops, photo shops, etc. in Shirakawa-go, a popular tourist area of Japan. When visitors look at these heritage spots, it is difficult for them to understand whether it is a restaurant or a souvenir shop. By using this system, tourists were able to obtain necessary information in their native languages when they enter each iBeacon area while walking in Shirakawa-go. The paper confirmed that this system can be used effectively for both Japanese and the foreigners [2]. In their system, it was possible to change the displayed language automatically when the user comes near to the digital signage, by using the mechanism of background scanning of iBeacons. However, their study is dependent on installing and configuring bluetooth iBeacons for the user to be able to understand the sig-

nages. It does not support text translations from printed signages. In our study, we aim to support both digital and printed signages.

American MLB has been using iBeacons in some stadiums. Their mobile application can provide navigation system, so the ticket in user's hand will pop out automatically and their position will be tracked when they get close to ticket entrance. Many techniques such as the method of displaying information written in several languages in user's native language, the method of displaying multilingual information simultaneously, and the method of switching displayed language using touch screens, have been used. [4] When iBeacon is scanned by the nearest mobile application, the proximity of the beacon will be accessed and customers will be provided with service based on the proximity [5] [6].

In India, a study of text detection and extraction based on Stroke Width Transform (SWT) to extract letters was conducted [3]. This application focussed on tourism industry and local transport, to help people to deal with different languages in India which involve text associated with natural scenes in the local public places. Using SWT method, various Indian languages and their combination with English were detected. However, this study only translates the regional languages used in India and does not translate the detected texts into International languages languages. This is still a hindrance for foreigners to understand their environment.

To develop a location-based system such as the digital signage system in our study, accurately determining user's proximity to the signage is important. The problem of proximity detection is found to be difficult in a variety of environments. Existing approaches like Global Positioning System and Wi-Fi triangulations are insufficient to meet the requirements of high accuracy and also involve high cost to setup. A study was made using, Bluetooth which is commonly available in all smartphones to find the proximity over shorter distance and determine the distance based on the Received Signal Strength Indicator (RSSI) values of the Bluetooth beacons [7]. In existing systems, GPS is used to find the location, which won't work as accurately indoors and inside most commercial building areas. So the authors proposed a system to overcome the problem by using Bluetooth to Bluetooth proximity estimation. By using the received signal strength of the Bluetooth device,

RSSI value is used to find out the approximate distance between the devices. This technique helps in determining the location of nearby user. The study proposed the proximity determination model by combining Bluetooth RSSI value. The results demonstrate that Bluetooth offers a feasible mechanism that is accurate and power-efficient for measuring user's proximity indoors [7]. While this study does not focus on text detection or digital signage, this is useful for our study to determine the distance of the user/smartphone to the iBeacons.

Optical Character Recognition (OCR) is the electronic conversion of images into machine encoded text. It provides alpha-numeric recognition of printed and handwritten characters. In the recent past, OCR has been an active topic of research and has wide applications in healthcare, banking, finance and education. One of the applications of OCR is to help with visually impaired persons. According to the World Health Organization (WHO), around 285 million people around the world are estimated to be visually impaired, out of which 90% live in developing countries. Thus there is a definite need to develop a system to provide information to the visually impaired. In the research by S. Sonth and J. S. Kallimani, a camera based framework was built on the Raspberry Pi, integrating with Image processing algorithms, OCR and Text-to-Speech (TTS) module [8]. The smartphone camera is used to capture an image of the printed or handwritten text, and the image was then subject to preprocessing before being fed into the OCR system [8]. This study, however does not focus on language translation or digital signage, we adopt image pre-processing techniques like binarization, de-noising, de-skewing, segmentation and feature extraction in our research.

In Taiwan, to encourage learning English, a study was carried out using the iBeacon's micro-positioning function to set the location in museum, restaurant, store, etc. When user's mobile phone detects the presence of iBeacons, the mobile application facilitates learning by interacting with users' environment [5]. Vocabulary includes the single words and expressions commonly used in daily life. With the narration and use of everyday sentence structure and grammar, users can learn English conveniently and quickly and apply it to real situations. In this way, users can learn English from the surroundings to enhance their English ability, achieve better effects with minimal efforts and in a fun way in the proposed system. With the combination of the

proposed application, English learning materials and iBeacon micro-location feature, learners can receive the surrounding related learning materials. In this manner, learners will have chance to build the connection between the learning materials and the environment which can enforce their memory to remember the learning concept [5]. It was found that the system can raise the learning intent of the learners and may improve their learning effectiveness. However, this study focusses on converting Chinese language to English based on the user's location. It does not support multiple languages and the system is entirely dependent on iBeacons.

With the rapid rise in multimedia and data services, demand for positioning has increased, especially in indoor environment which needs to determine the proximity of the user to provide context based information [9]. There is a lack of accuracy in current positioning systems. A study was made to design and develop a mobile-based indoor location system for mobile applications with the Bluetooth Low Energy technology based on the iBeacon [10]. The authors adopted Gaussian filter and Unscented Kalman filter method to read signals from iBeacon device. With the extracted signals, the authors compared them with a database. They were able to display data based on micro-location on the user's smartphone. It was found that the iBeacon approach has better performance compared with WiFi method [10]. However, this study does not translate the detected texts into the user's preferred languages. This is still a hindrance for foreigners to understand their environment.

## 2.2 Outline

With the recent developments in Internet of Things (IoT), the user and their location is closely interrelated. The outdoor positioning system based on satellite position (GPS) has been getting more accurate. However, GPS positioning satellite signals can only be best used in outdoor environments, it is difficult to meet the needs of indoor positioning [10]. So, there is an urgent demand for indoor positioning technology and providing micro-location-aware data. Micro-location is a term used to define the location, accurate down to a few meters.

In recent years, many types of short-range wireless communication protocol and related products are emerging in the market. Some of the short-range wireless data communication includes WIFI, Bluetooth, ZigBee, RFID, NFC and Ultra Wideband (UWB) among other types of technology [11]. These technologies play an important role in their respective areas and requirements. Let us evaluate some of the options we have to provide data based on micro-location.

### **2.2.1 GPS**

The outdoor positioning system, which is based on global satellite position (GPS) provides users with the needs of positioning, accurate to 10 meters. However, GPS positioning satellite signals is only feasible in outdoor environment. GPS fails to determine the user's location accurately inside a building or within a room [10]. This means, we cannot use GPS in our study as we need to determine users location, with accuracy ranging from 0.1 to 50 meters.

### **2.2.2 Wi-Fi**

The coverage of Wi-Fi is wide. It could reach 100 meters, and achieve coverage with comprehensive in clear surroundings. Also, the speed of network transmission is higher, it's useful for real-time interaction of the application. The Wi-Fi network construction has a low cost, is easy to maintain and very suitable for use in mobile phones. However, the core drawbacks are that data security and performance are slightly less compared to Bluetooth [10]. It is also difficult to accurately triangulate the user's position inside the network.

### **2.2.3 Near Field Communication (NFC)**

NFC is a set of short-range wireless technologies that enable two electronic devices, one of which is usually a smartphone, to establish communication by bringing them in close proximity. However, the range of NFC is less than 10 centimeters and it gets very expensive to install a lot of NFC tags to achieve the required proximity services [12]. Thus, it is not feasible to use NFC in our study.

#### **2.2.4 Bluetooth**

Bluetooth is a radio technology which supports short distance communication. The main advantage of Bluetooth indoor positioning technology is that, Bluetooth chips are small, easy to integrate with mobile phones and even in smaller devices. On the other hand, the shortcomings of Bluetooth technology are; relatively high power consumption and lack of stability when the system is setup in complex space environment [10].

#### **2.2.5 iBeacon**

The communication protocol used in iBeacon is the Bluetooth Low Energy (BLE). Bluetooth version 4.0 devices supports BLE [10]. iBeacon communication frequency works on the 2.4GHz band, which is as fast as the Wi-Fi. iBeacon has proximity sensing technology, it can transfer information such as Uniform Code of unique ID (UUID). Mobile applications can scan and read the information about UUID and RSSI, and it can be transformed into a physical location, which triggers location-aware applications and information. iBeacons are also low-cost devices and typically run for a year on a single coin battery such as CR2032 [1]. This makes it feasible for us to implement iBeacons into our study of providing data based on user's micro-location.

#### **2.2.6 Comparison of indoor positioning methods**

The table below presents the advantages and disadvantages of various indoor positioning systems [13]. It can be seen that iBeacon has advantages of low cost, long battery life, feasible accuracy and range.

iBeacon technology runs on Bluetooth, which is widely supported by most of the smartphones today [1]. Also the fact that Bluetooth technology is supported by most of the modern smartphones [10] makes it the most feasible choice to implement the multilingual signage system for our study.

	Wi-Fi	. NFC/RFID	iBeacon
<b>Cost</b>	Medium	Medium	Low
<b>Accuracy</b>	Low	Low	High
<b>Range</b>	Far	Medium	Far
<b>Advantages</b>	Highly popular, far sensing range	Feasible accuracy and range	Low cost, high accuracy and range.
<b>Disadvantages</b>	Low accuracy, influenced by environment	Low popularity, difficult to maintain	Likely to be influenced by environment.

Table 1: Comparison of popular indoor positioning methods

The important aspects we consider for this study to decide the method is the low cost and high accuracy feature [13]. The range of the positioning method also played a huge role in order for us to consider the positioning method.

### 3 Understanding iBeacon Technology

iBeacon is Apple's implementation of Bluetooth low-energy (BLE) wireless technology to create a way of providing micro-location-based information and services to nearby smartphones supporting BLE. This allows mobile applications on a smartphone to determine when it has entered or left the region, along with an estimation of proximity to the smartphone.

Bluetooth 4.0 is proposed according to the Bluetooth technology standard set by the Bluetooth Special Interest Group (SIG) to achieve two-way transmission. When compared with traditional Bluetooth, Bluetooth 4.0 has the advantages of lower cost and lower energy. A button cell, such as CR2032 battery can make the device supporting Bluetooth Low Energy be operated for 1 to 2 years [5].

iBeacon devices simply broadcasts its configured advertising packets in specific time intervals [2]. Apple has standardized the format for BLE Advertising. In this format, a packet of information from a beacon consists of four main pieces of information.

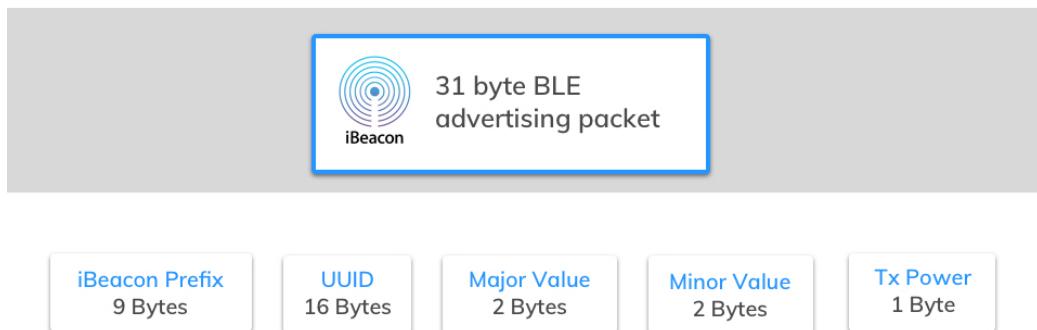


Figure 1: iBeacon packet format

The figure shows how the breakdown of the beacon information. The smartphone needs to read the UUID, Major and Minor to fetch related data to the beacon. Tx power is associated with the strength or range of the beacon

Breaking down the iBeacon information, there are five major parts: fixed iBeacon prefix; the proximity UUID: an identifier used to distinguish the beacons from other systems using similar bluetooth beacons; the major number can group a related set of beacons in an organization, such an exhibition hall in a museum or a floor in a store; the minor number can be used to identify individual beacons in specific areas because each beacon has a different minor number, by which we know where the user is exactly located. TX power, is used to determine how close the user is to a beacon [14]. It can be presented either as an estimate (immediate/near/far/out of range) or it can be used to calculate more precise measurement in meters.

**UUID:** This is a 16 byte string used to differentiate a large group of related beacons. For example, if National Park Services (NPS) maintained a network of beacons among all national parks, all national park beacons would share the same UUID. This allows NPS dedicated smartphone app to know which beacon advertisements come from NPS owned beacons. Example: E6FA79C7-804F-4742-863B-BD4D282ED9BA.

**Major:** This is a 2 byte string used to distinguish a smaller subset of beacons within the larger group. For example, if NPS had fifty beacons in a particular national park, all fifty would have the same Major. This allows NPS to know exactly which park its visitor is in. Example: A beacon with UUID E6FA79C7-804F-4742-863B-BD4D282ED9BA and a major number of 1101.

**Minor:** This is a 2 byte string meant to identify individual beacons. Keeping with the NPS example, a beacon at the front of the park would have its own unique Minor. This allows NPS's dedicated app to know exactly where the visitor is in the park. Example: A beacon with UUID E6FA79C7-804F-4742-863B-BD4D282ED9BA, major number of 1101 and a minor number of 9901.

**Tx Power:** This is used to determine proximity (distance) from the beacon. TX power is defined as the strength of the signal exactly 1 meter from the device. This needs to be calibrated in advance to deployment. Devices can then use this as a baseline to calculate an estimate of distance between the beacon and the smartphone.

### **3.1 Existing applications implemented using iBeacon technology**

- In a store, the retailer can understand shoppers' interests by calculating the time spent by a shopper in front of a specific item in the store, and then, the application in shopper's smartphone will show the other recommendation of related products [14].
- The retailers can track shoppers behavior, for example, the shopper has seen the same goods in the other store of the same chain online and use with iBeacon technology to recommend those when visiting the store [14]. The retailers can push the coupons information as well.
- iBeacon technology brings many options for the retailers to send notifications of the items/events information to user's smartphone [14]; to trace users interests or potential demands; to contextualize shoppers' purchase tendency based on the user's history.
- A study was conducted implementing iBeacons in the campus for the purpose of indoor positioning of the students and send out relevant information [15].

### **3.2 Operating range of iBeacons**

iBeacon devices transmit its information to all devices that use this technology from a range of a few inches to more than 70 meters. The actual distance at which the device operates varies, depending on the environment. Just like all the radio waves, Bluetooth loses its power when it encounters obstacles [16]. All devices capable of supporting the Bluetooth 4.0 can take advantage of this system without the need of having to pair devices previously.

### **3.3 Calculating distance between iBeacons, based on RSSI**

The rate of signal strength will not be stable because of RF signal in actual obstacles in environment. So, there is no longer accurate correspondence

between the RSSI and distance. Also, we need to purify the received data in order to reduce the corresponding positioning errors.

We place iBeacon at a fixed position in the test environment. The mobile terminal detects the response of iBeacon signal strength, and we build out RSSI vector set based on the received signal, this set represents the RSSI vector is  $R=(R_1, R_2, \dots, R_i, \dots, R_P)$ , where  $i$  is the signal strength of  $i$ -node RSSI,  $P$  is the total number of iBeacon. [17]

We then repeatedly measure RSSI and averaging the values, the average value as the iBeacon RSSI value  $r=(r_1, r_2, \dots, r_i, \dots, r_m)$ , the formula is as follows:

$$RSSI = 1/m \sum_{i=1}^m RSSI_i \quad (1)$$

where  $m$  is the total number of collected RSSI vector at the coordinate points. This approach would correct the rather unstable distance of the beacon from the user.

## 4 RESTful web services

The mobile application developed communicates with the web-services such as Google Cloud Vision and Google Translation using Representational State Transfer (REST). REST is an architecture that defines a set of rules to be used for creating web services [18]. Web services that conform to the REST architectural style, or RESTful web services, provide interoperability between computer systems on the Internet.

By using a stateless protocol and standard operations, REST systems aim for fast performance, reliability, and the ability to scale, by re-using components that can be managed and updated without affecting the flow of other functions in the system as a whole, even while it is running in the background. In our system, we interact with Google Vision and Translation web-services through REST method. The response from these web-services is processed in the mobile application and appropriate messages are shown. A sample request and response for each web-services is shown in Sections 5 and 6.

## 5 Google Vision

Google Cloud Vision is an image recognition technology that allows us to remotely compute the content of an image and to retrieve the information present in the media. By using Representational State Transfer (REST), called Google Cloud Vision API, we can use such technology within our own applications [19]. By using Google’s Cloud Platform to compute the content of an image through advanced machine learning processes, this solution allows us to extract relevant information from visual data, including image labeling, face and landmark detection, optical character recognition (OCR). It is possible to interact with Google’s Cloud Vision platform by using specialized REST API, called Google Cloud Vision API [19].

We propose to use such cloud-based software resources in order to achieve a system for people with understand signages and their environment in general. In particular for users who are visiting foreign countries, our solution may help them to interact with the environment and the things around them. In this paper, we focus on the OCR functionality of the Google Cloud Vision. We also discuss the implementation and integration of the Google Vision into our software solution, a mobile application we developed. The smartphone requires an Internet connection to submit the captured images to Google’s cloud platform. The response from platform is processed and displayed on the user’s smartphone.

The algorithm shown in Figure 2 has been implemented in Javascript. In particular, here the request for Google Cloud Vision includes the TEXT DETECTION feature. A sample request and response is shown below:

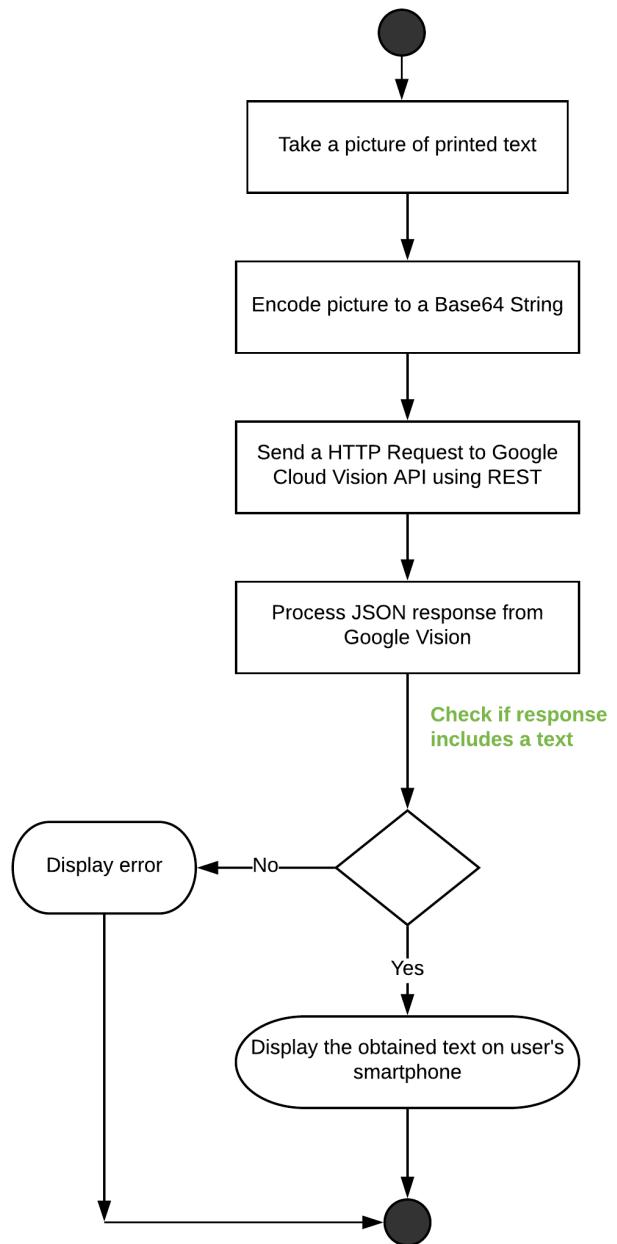


Figure 2: Process flow diagram for Text Detection using Google Vision

## 5.1 Sample Request

Let us consider the image shown in Figure 3 as the source of the image. We use the Algorithm mentioned in Figure 2 to extract the text from the image.



Figure 3: Example image for testing Google Vision API

We convert this image to a base64 encoded string using Javascript and specify in the headers of the REST API to use 'Text Detection' feature of Google's Cloud Vision API. A sample request is as shown below:

```
{  
    'requests': [{  
        'image': {  
            'content': 'base64_encoded_string_of_image'  
        },  
    }]
```

```

        'features': [ { 'type': 'TEXT_DETECTION' ,
        'maxResults': 1 } ] }
}

```

## 5.2 Sample Response

In this section, we discussed the capabilities of Google Cloud Vision API and how we implement it into our system. We notice that the Vision Cloud API has analyzed the image, extracted the text and the same is returned to the device through the RESTful APIs.

The following is a sample response of extracted text from the Figure 3. It is a JSON object containing the image information.

```

{
  'responses': [
    {
      'textAnnotations': [
        {
          'locale': 'en',
          'description': 'Pedestrian access only beyond this point',
          'boundingPoly': {
            'vertices': [
              { 'x': 327, 'y': 412}, { 'x': 1158, 'y': 412},
              { 'x': 1158, 'y': 680}, { 'x': 327, 'y': 680}
            ]
          }
        }
      ]
    }
}

```

## 6 Google Cloud Translation

Google Translation API provides a simple application program interface (API) for translating any string/text into any supported language using the inbuilt Neural Machine Translation. It is highly responsive, so applications can integrate with Translation API for fast, easy translation of source text from the source language to a target language (such as Spanish to English). Language detection feature is also supported in cases where the source language is not known. The underlying technology is updated constantly to include improvements from Google research teams, which results in better translations and new language pairs.

Translation API supports more than one hundred different languages, which makes it an ideal choice for us to use in this study. Translation API can be accessed using REST APIs. We extract text from the printed signages using Google Cloud Vision API, we then send the text to Google Translate API and convert it into the user's desired language.

Figure 4 describes the process of language translation in our system. We obtain the text from the printed media using Google Vision API as mentioned in Section 5. We then pass the obtained text to Google Cloud Translation API with the preferred language in the headers of REST API. The response from the Google Translate is then displayed in the user's smartphone. A sample request and response is shown below:

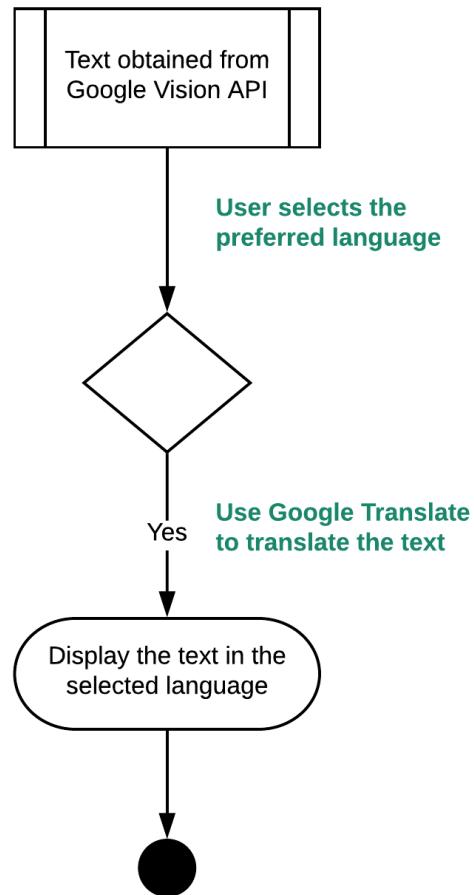


Figure 4: Process flow diagram using Google Translation

## 6.1 Sample Request

The below code segment shows how to use Google Cloud Translate API using REST APIs using Javascript. Developers should replace their own Google API Key at API-KEY-HERE. The target language to be translated is passed under 'target' parameter of the REST API.

We use javascript to interact with Google Translation API as follows:

```

fetch('https://translation.googleapis.com/language/
translate/v2?key=APIKEY', {
  method: 'POST',
  headers: {
    'Accept': 'application/json',
    'Content-Type': 'application/json',
  },
  body: JSON.stringify({
    q: 'text_to_be_translated',
    target: 'target_language'
  })
})
.then(data => data.json())
.then(res => this.cleanText(res))
.catch(err => (err))
}

```

## 6.2 Sample Response

Google Cloud Translation processes the requests, determines the text to be translated and the target language. It then responds with a JSON object containing the translated text. All these is communicated to the application through RESTful APIs. A sample response is shown below:

```
{
  'data': {
    'translations': [
      {
        'translatedText': 'translated_text_here'
      }
    ]
  }
}
```

## 7 Algorithm

The below pseudo-codes show the algorithms involved in our system. The system constructed for the study runs both the below algorithms in the background. For convenience, we have divided the algorithm into 2 sections, one for digital signage and one for printed signages.

Algorithm 1 shows the pseudo-code for scanning iBeacons. The system built in the study scans for beacons in the background. When the beacons are found, the mobile application reads the parameters of each beacon and queries the database in the server with the parameters. If data is found in the server, the mobile application processes the data and displays appropriate signage information. The user can then choose to translate the data into his preferred language using Google Translation.

---

**Algorithm 1** Pseudocode for digital signages

---

```
1: procedure SCAN FOR iBEACONS
2:   if iBeacons found then
3:     loop:
4:       acquire RSSI info of beacon
5:       calculate and display distance of beacon
6:       query beacon parameters with server database
7:       if data exists then
8:         fetch and display data
9:       procedure USER SELECTS PREFERRED LANGUAGE
10:        fetch selected language
11:        send the text and selected language to Google translation
12:        fetch and display text in user's preferred language
```

---

Algorithm 2 shows the pseudo-code for extracting text from printed signages and then translating them to preferred language. The system listens to the event of image capture from smartphone camera or image upload from the camera roll. On this event, the system converts the image into a base64 encoded string and communicates with Google Cloud Vision using REST method. If the server is successful in extracting text from the image, it

responds with a JSON file containing the extracted text. The mobile application then processes the data and displays the extracted text. The user can then choose to translate the data into his preferred language. On selecting a language, the system then sends both the extracted text and selected language to Google Translation using REST method. On successful translation, the system then displays the information in user's selected language on the mobile application.

---

**Algorithm 2** Pseudocode for printed signages

---

```
1: procedure USER CAPTURES AN IMAGE OF SIGNAGE
2:   convert image to Base64 encoded string
3:   send the image to Google cloud vision
4:   if text returned then
5:     parse the response and display text
6:   procedure USER SELECTS PREFERRED LANGUAGE
7:     fetch selected language
8:     send the text and selected language to Google translation
9:     fetch and display text in user's preferred language
```

---

## 8 System Design

In this study, a prototype of multilingual information service system was designed and constructed. 3 beacons were configured and placed next to each of the 3 signage in the university campus. These beacons continuously broadcast their UUID, Major and Minor. A mobile application was constructed to listen to these beacons. The mobile application leverages the smartphone's bluetooth technology to listen to the beacons.

The relevant information of the signage for each beacon is stored in a database using Google Firebase, a cloud storage. Once the mobile application detects a beacon, it makes a call to the cloud, passing the detected beacon configuration to get the appropriate information of the signage using REST APIs. The information returned from the cloud is then translated to required languages using the Google Translate API and displayed on the user's smartphone.

Figure 5 below gives an overview of different processes in our system.

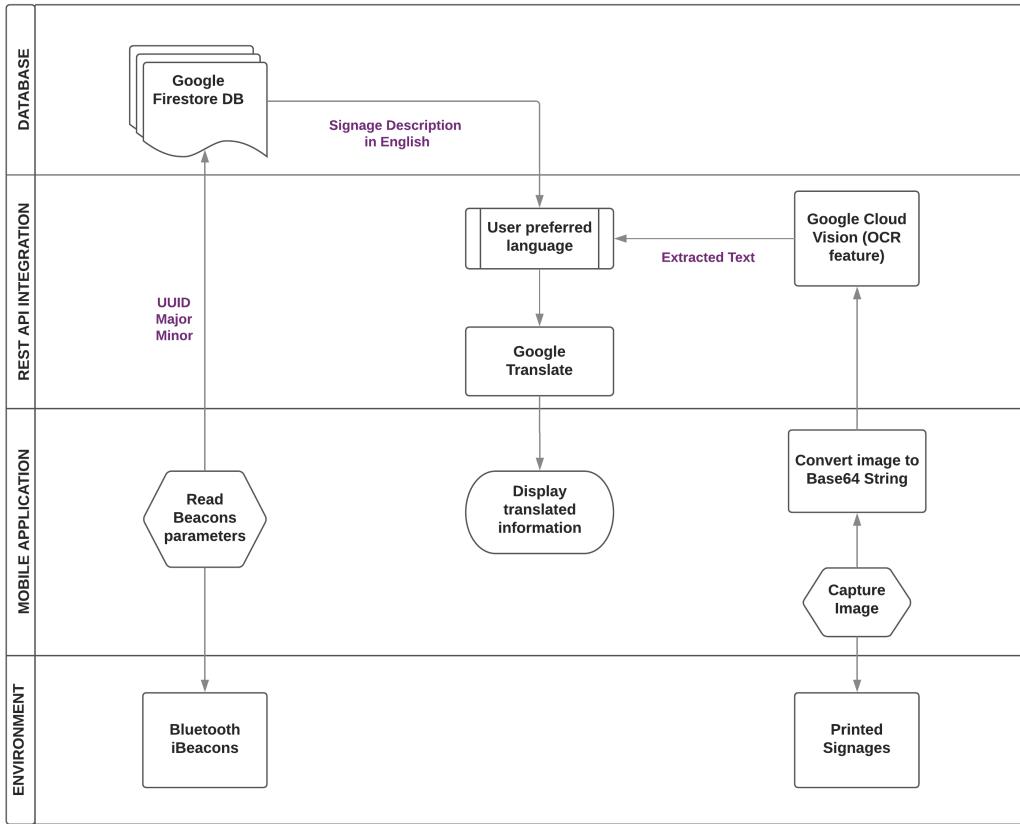


Figure 5: System design process flow diagram. The diagram shows all the individual components of the system and how it is integrated together to achieve the objective.

## 8.1 Environment setup

To evaluate the system, we constructed a combination of digital signages and printed signages were placed around the California State University, Channel Islands (CSUCI) campus. Each signages were placed at least 100 meters apart. For optimal working of the system, the beacons need to be placed 15 meters apart.

### **8.1.1 Digital Signages**

The digital signage was set up using the bluetooth iBeacons. Each iBeacon is mounted on individual signages in the campus. The beacons are powered by individual CR2032 lithium-ion coin batteries. The users used the iPhone 6s in which the application program that reacts with the iBeacon signals. The parameters of the beacons, which are discussed in the Section 3 can be configured only by the administrator of the mobile application. To evaluate the system, we use 3 beacons. Below are the configurations of each beacon used in the study.

UUID: E6FA79C7–804F–4742–863B–BD4D282ED9BA  
Major: 1111  
Minor: 9991

---

UUID: E6FA79C7–804F–4742–863B–BD4D282ED9BA  
Major: 1111  
Minor: 9992

---

UUID: E6FA79C7–804F–4742–863B–BD4D282ED9BA  
Major: 1111  
Minor: 9993

---

### **8.1.2 Printed Signages**

To evaluate our solution of translating the signages without the iBeacons, we printed out commonly used signages in Spanish language and placed them around the CSUCI campus.

## **8.2 Mobile application**

To discover the beacons from the smartphone, we have designed and developed a mobile application. This mobile application is built using React native and Javascript. The mobile application uses Google Cloud Vision to detect the printed text using the smartphone camera. The Google Cloud Vision API detects the printed information using OCR feature as mentioned in Section 5. This text can be converted to any language using Google Cloud

Translation API, discussed in Section 6. The mobile application interacts with these APIs using RESTful method. To improve the efficiency and user-experience, we used both iBeacon monitoring and ranging to discover and interact with the beacons.

### 8.2.1 Background Monitoring

Beacon monitoring is similar to geofencing, i.e., a virtual barrier that is usually defined using a set of geographic coordinates. In case of iBeacon, an area is defined by the range of one or more beacons. Moving and out of the area under the umbrella triggers 'enter' and 'exit' events, which the mobile application can listen and trigger events accordingly. The smartphone will keep scanning for the beacons at all times even if the application is not running in the foreground, and even if the smartphone is 'locked'. Once an 'enter' or 'exit' happens, iOS will launch the application into the background (if needed) and execute some code for a few seconds to handle the event.

### 8.2.2 Beacon Ranging

Ranging refers to the process of scanning for any nearby beacons on the foreground and delivers results as and when the environment changes. With monitoring, our application will be notified whenever the user enters and exits the terminal. Ranging for the exact same region, will provide us with the full list of matching beacons currently in range - with the UUID, major, and minor values.

### 8.2.3 Development stack

The mobile application was developed after evaluating all the major development tools and technology present in the current market. The main goal is to develop the system, keeping performance and scalability in mind. Also, we want to make sure the system works efficiently on both iOS and Android platforms. Below is the development stack we used to develop the mobile application:

- **React Native:** To develop the front end of the mobile application. The advantage of using React Native is it helps us to develop for both Android and iOS with the same codebase. Also, React Native is efficient in terms of performance.

- **JavaScript:** We used JavaScript to interact with the web-services such as Google Vision and Google Translate. It makes it efficient to convert the captured images into Base64 encoded string and pass it in the headers of the request. Also, JavaScript makes it easier to process/-parse the JSON response from the web-service and display it on the front-end using React Native. We use JavaScript also to communicate with the Google Firestore database to query and retrieve data.
- **Camera:** To access the smartphone camera, we included a library called 'react-native-camera'. This extends the functionality of camera in terms of fragments and hence makes it easier to listen to the camera events and implement our logic where necessary.
- **Bluetooth:** Our system needs access smartphone's bluetooth module to discover the iBeacons. To check the state of Bluetooth, we included 'react-native-bluetooth-state' library. This helps us determine the bluetooth status of the smartphone, whether 'on' or 'off' and display appropriate messages to the user.
- **Beacon Manager:** To discover iBeacons from the smartphone, we integrated 'react-native-beacons-manager' library into React Native. This library helps us scan for the beacons and listen to changes of the beacon signal strength. The library also supports 'background ranging' and 'foreground monitoring' of the beacons, meaning it helps us scan for the beacons even when the application is running in background.
- **Xcode:** To build and run the application on the iOS device, we need an Integrated Development Environment (IDE). Xcode is an IDE that can be used to run the developed mobile application on the iOS device. The source code for the implementation could be found at [20].

#### 8.2.4 Screenshots

The below figures show the screenshots from the constructed mobile application. Figure 6 shows the Landing screen of the application. The user sees this screen on opening the application. The Home page has the following functionalities:

- Capture image of a physical signage

- Scan for digital signages around. If found, display the image, title and the distance of the digital signage from the user.
- The user can also upload an image from camera roll of the smartphone and extract text from it.
- The user can choose to type a text and convert it into any language



Figure 6: Home page



Figure 7: Text extraction

Figure 7 shows the text extracted from the image captured on Figure 6. This screen is presented to the user right after capturing an image of a physical signage. We convert the captured image into Base64 string and pass it to Google Cloud Vision using REST and extract the text from the image. The screen on Figure 7 has the below functionalities:

- Display the extracted text from the image
- The user can choose to translate the extracted text to any of the supported language.

Figure 8 shows the text translated into Spanish. This is the same text extracted from Figure 6. The extracted text is shown on Figure 7. This screen is presented to the user after the user chooses a language and taps on the 'GO' button. We listen to the event, capture the language selected and send a request to Google Translation API using web-services. The screen on figure 8 has the below functionalities:

- Display the translated text in user's preferred language.
- Share the translated text using Share button.
- Capture another image of a physical signage.
- Go back to landing page.

An important design consideration was to allow the option to share the translated text in any form. Figure 9 shows the Share feature we integrated so the user can record the translated text and share them with friends and family who might need the information. In our system, the user can choose from the multiple options to share the translated text. The default options of the iOS applications include e-mail application, messaging application, notes application etc.

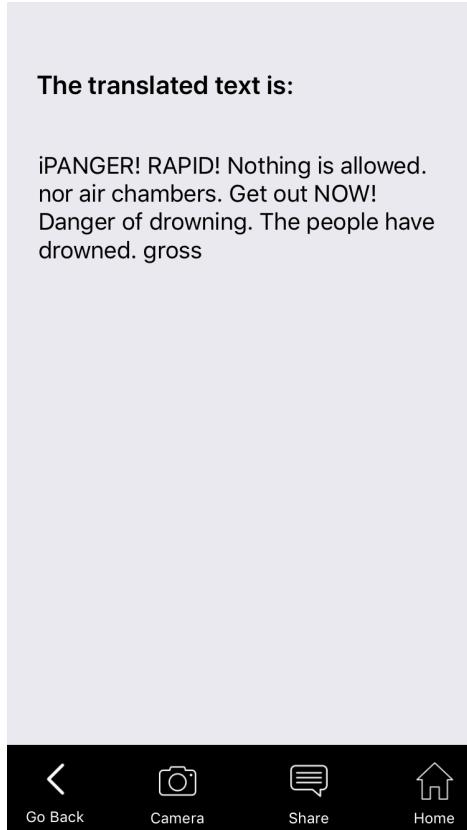


Figure 8: Translated text

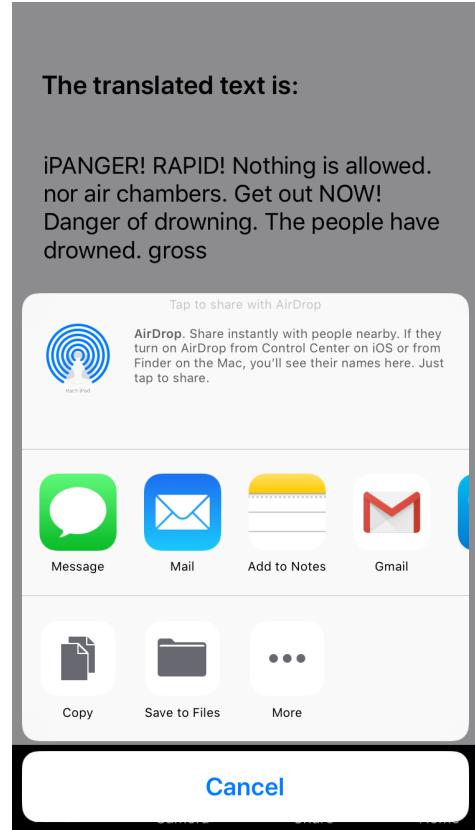


Figure 9: Share text feature

Since our primary users of our application are non-native visitors, we feel there is a strong need for them to communicate day-to-day for critical information in their native language. To achieve this in our system, we implemented a 'type and translate' feature. Figure 10 shows the screenshot of the 'type and translate' feature. The user can type in text in any language and choose to convert it into any of the supported languages.

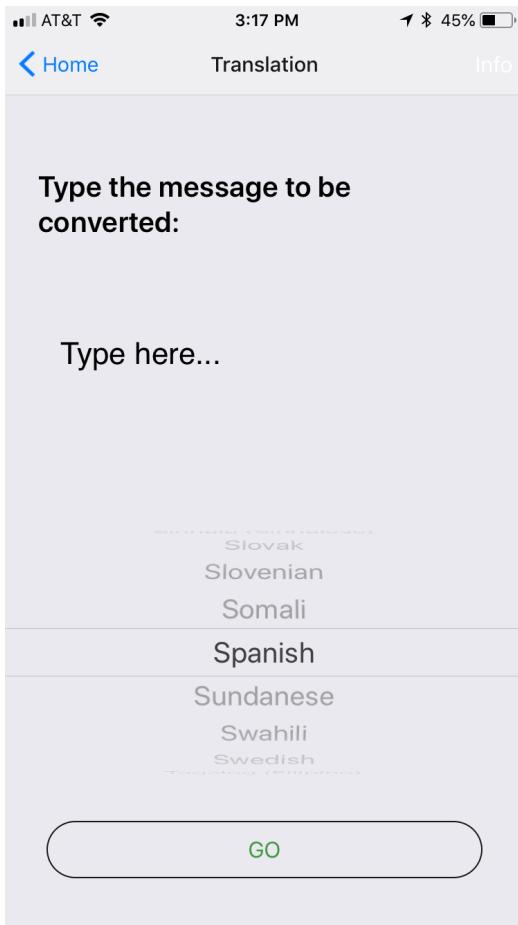


Figure 10: Type and translate text feature

The user lands on the screen shown in Figure 11 after tapping on any of the digital signage thumbnail. When the user taps on a particular signage, the system screen fetches the pre-configured information and media from the database and displays them on this screen. This screen displays the image of the signage, the title and the description of the signage in English. This component has the below functionalities:

- Fetch and display the appropriate data (media, title and description) from the database.
- Provide the user with the option of translating the information into supported languages.

The screen in Figure 12 is shown when the user taps on 'Translate' button from screen in Figure 11. The information of the digital signage is passed to this screen. Now, the user can translate the text into any supported language. This component is capable of the below functionalities:

- Display the beacon information.
- Provide the user with the option of translating the information into supported languages.



Figure 11: Digital Signage

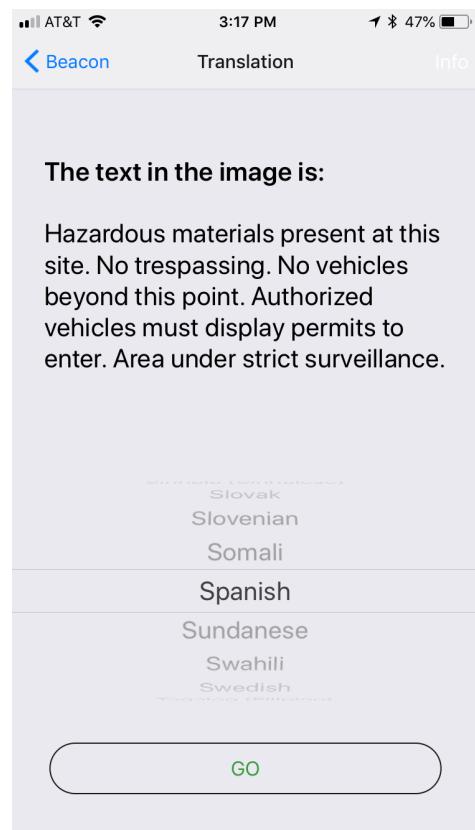


Figure 12: Translate feature

Figure 13 is shown when the user selects a language and taps on 'Go' button from Figure 12. The information of the digital signage is passed to this screen. Now, the user can translate the text into any supported language. This component is capable of the below functionalities:

- Display the translated text in user's preferred language.
- Share the translated information.
- Go back to landing page.

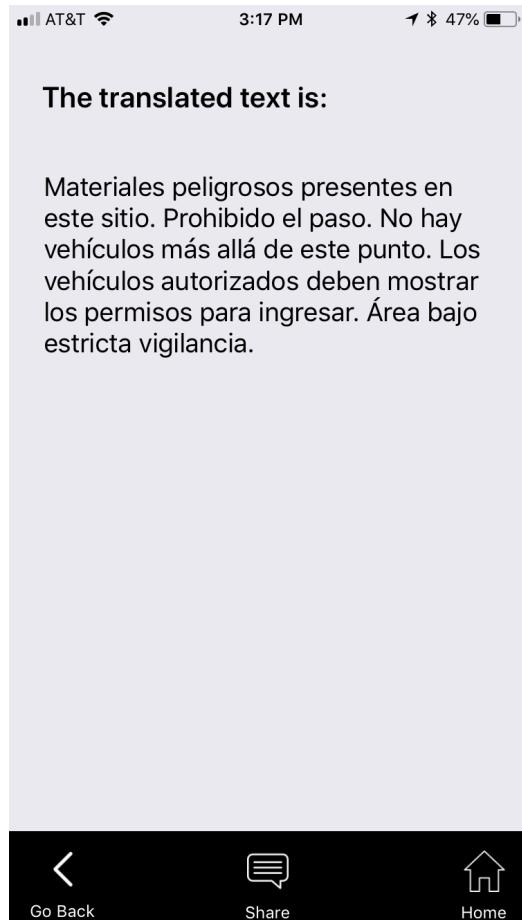


Figure 13: Text translated to Spanish language

### 8.3 Database

To store the information of each signage in a database, we chose Google Cloud Firestore. Cloud Firestore is a flexible, scalable database for mobile, web, and server development from Firebase and Google Cloud Platform [21].

It keeps the data in sync across applications through realtime listeners and offers offline support for mobile and web so we can build applications that work regardless of network latency or Internet connectivity [21]. Google Firestore is free of cost for less than 50 simultaneous connections to the database, which is a right fit for the study.

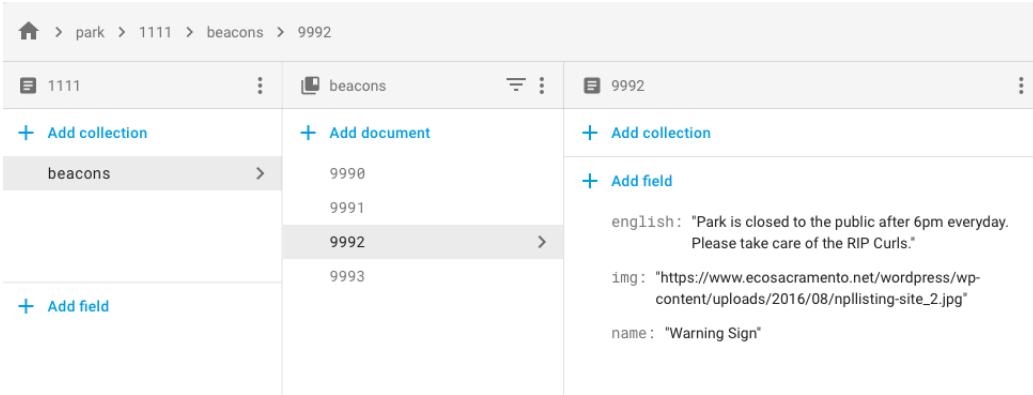


Figure 14: Database Schema

Database schema has been designed in a way that it is very easy to scale and use the system for any physical environment. The UUID is on top level of the schema and the Major and Minor number follows the UUID.

We designed the database schema with a goal to keep it as simple as possible so it can be scalable in the future. This makes it easy to segregate data and query it without having to compromise for the application performance. We organize the database schema as follows:

- **Park:** Root of the database. All the beacons with the specified UUID falls under this bucket.
- **Major:** This is the level 1 of the database. This is essentially the 'major' parameter of the iBeacons. Since all the 3 beacons we configured have the same 'major' value, we just have one collection of level 1. This level of database makes it possible for our mobile application to differentiate the sections of the park. This allows the administrator to configure beacon parameters according to sections of the park.
- **beacons:** This is the level 2 of the database schema. We now have identified which park the beacons are present in and which section of the

park the beacons are present in using the above two levels. ‘beacons’ section of the database serves as the collection of beacons under a particular section of the park. They contain all the information about the individual beacons present in discovered section.

- **Minor:** This is the last level of the designed database schema. ‘9992’ refers to the ‘minor’ value of the iBeacon. Each iBeacon will have its own unique ‘minor’ value. This helps us to query for the exact beacon data. Each entry at this level have the below fields and values:

‘name’: The name of the signage.

‘img’: Contains the URL of a picture of the signage.

‘english’ : This field contains the description of the signage, in English.

## 9 User Experience Design (UX Design)

Incorporating User Interface (UI) Design principles and guidelines are very important for developing any mobile-based application. By observing good UI design practices and implementing them in the system, it can be helpful for users to improve usability and grasp critical information as displayed in the application.

For this study, a set of User Interface Design (UID) principles is considered [22] for designing and developing the mobile-based multi-lingual signage application. Below are the set of UIDs adopted:

- Navigation needs be simple and clear from page to page or to any particular section. It needs to be consistent throughout all pages in the application.
- Complex and nested navigation needs to be avoided in the application.
- Limit scrolling frequency.
- Application should be user-friendly and allow users to understand how the application works intuitively.
- Similar actions and call-to-actions need to be located consistently. For instance, buttons must be found in similar positions for all the pages of the application.
- Providing necessary information only. Unnecessary information confuses the learners and decreases usability.
- Decreasing textual information and increasing information which is provided in graphical media to minimize learners' cognitive load and motivate them to keep using the application.

We are going to carefully design the User Interface (UI) and User Experience (UX) based on the best practices for Human Computer Interaction (HCI). The section below describe the concepts behind designing the UI and UX.

## 9.1 Context Analysis

In this section, we discuss where the system in study can be used. We also discuss the social and environmental conditions. Below are the environments our system could be used:

- In parks and exhibits that involve multiple signages.
- Museums, where there is a need for foreign visitors to understand the signages in their native language.
- In shopping malls, so visitors can understand critical information and navigate easily.

## 9.2 Task Analysis

In this section, we identify what the user is trying to achieve using our system. We also identify the tasks that need to be supported in order for the user to achieve his goal. Below are the list of tasks the system needs to support.

- The system needs to support taking pictures of the signage and extract the printed text.
- The system needs to translate text into the user's preferred language.
- The application should remember the selected language and display the translated text into the preferred language.
- Once the application discover the digital signages, it should display the distance of the user from the signage.
- The application should fetch the appropriate information from the database and display the same.
- The administrator of the system should be able to configure the beacons and change the fields of the database as well.

## 9.3 User Analysis

Below are the different types of users we expect to use our system:

- **Primary Users:** People who visit foreign countries either for tourism or business. Usually, the signages are displayed in local languages. This is a hindrance to visitors. They would want to understand their surroundings.
- **Secondary Users:** Visitors of parks and exhibits who would like to know more about their surroundings in their native language. The application will scan for digital signages and present relevant information once the user is in proximity.
- **Tertiary Users:** Tertiary users are the users who interact with the translated information. They may be the native people interacting with the visitors.

### 9.3.1 Example Persona

This section describes an example persona of a primary user:

Rebecca is a freshman in the Environmental Sciences program. She is 22 years old, from Arlington, VA. Throughout her high school years, she has been interested in preserving the environment, and has been involved in numerous recycling and conservation activities. She has had various international travel experiences, including a trip to Japan for a youth group ministry project.

Rebecca lives on campus in the dorm she shares with two other college students. She is very comfortable with technology and uses Twitter, Facebook, and email to communicate with friends and family. Rebecca believes in technology to reach out to people and spread knowledge about Amazon rainforests through images and icons. Rebecca is motivated to participate in a Peruvian conservation project during the summer following her high school graduation. She is the current Vice President of the Environmental Society.

During her foreign travels with the society, especially to the national parks, Rebecca likes to understand more about the environment and always makes sure there is enough signages so people are aware of the rules and do not pollute the environment.

## 9.4 Design Principles

The user experience was designed according to the design principles of Human Computer Interaction (HCI). In this section, we discuss the various design principles incorporated in designing the UX.

### 9.4.1 Affordance

According to Donal Norman (1988) an affordance is the design aspect of an object which suggest how the object should be used; a visual clue to its function and use [24]. Affordance is a property in which the physical characteristics of an object influence its function.

In the mobile application developed, the UI of the Landing page (shown on opening the application) is divided into 2 sections. About 3/4th of the screen shows the camera view, which means the application is ready to capture a picture of printed signages and extract text from it using Google Cloud Vision. The rest of the screen shows the nearby digital signages. The application also shows the distance between the digital signage and the user.

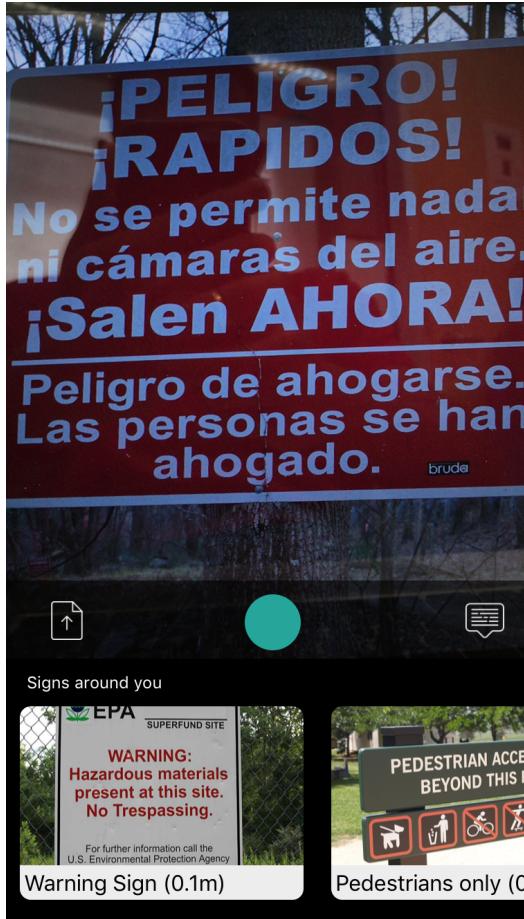


Figure 15: Affordance design principle

This is the concept in which the physical characteristics of an object influence its function. We have clearly separate individual functions using primary colors and appropriate icons. The main features of the application is loaded by default without the user having to make any clicks or taps.

#### 9.4.2 Visibility

The usability is improved when its status and methods of how to use are clearly visible, intuitively. The more visible functions are, the more likely users will be able to understand how the system works [24]. In contrast, when functions are 'out of sight,' it makes them more difficult to find and understand the application.

In the mobile application developed, the capture picture button has a green color, so as to invite the user to click the button. The camera functionality also has an 'upload image' icon and 'type and translate' icon present upfront. There is no ambiguity in the user interface. The digital signage section also lists the nearby digital signages along with the title and distance from the user. This improves the visibility of all the features of the app without the user having to scroll down.

#### 9.4.3 Constraint

The design concept of constraining refers to determining ways of restricting the kind of user interaction that can take place at a given moment [24]. This helps the application to do what it is exactly supposed to do and eliminates many error scenarios. This is helpful for the stable functioning of the application.

In the system we built, we establish constraints by only listing the digital signages around. If there are no digital signages around, 'No digital signages around' message is shown to the user. Also, the system is constrained to capture/upload only one image and use web-services to extract text from it. This helps the application function without many errors.

#### 9.4.4 Consistency

This refers to designing interfaces to have similar operations and use similar elements for achieving similar tasks [24]. In particular, a consistent interface is one that follows rules, such as using the same operation to select all objects. This helps the user intuitively learn and use the application without having to spend more time on it [24].

In our study, we have designed the UI using the modules approach. That is, whether the user chooses to capture an image or upload from the camera roll, the layout is the same. In the digital signages section, all the signages are listed consistently and the whole thumbnail is clickable. Each click takes the user to a screen with more details about the signage. This page is designed consistently. This helps the user to understand our application faster.

#### **9.4.5 Feedback**

Feedback is about informing the user about what action has been done and what has been accomplished, allowing the user to continue with the activity [24]. This helps the user to understand there is a background process running and at the same time, the user is constrained from starting another process or quitting the current process.

In the application developed, the user is shown 'loading' icon when a background process is running, for example, when an image is being uploaded through web-services. This helps the user understand that there is a process running in the background and also helps the user to understand the application better. Also, appropriate error messages are shown at the appropriate places to help user understand about the action being carried out [24].

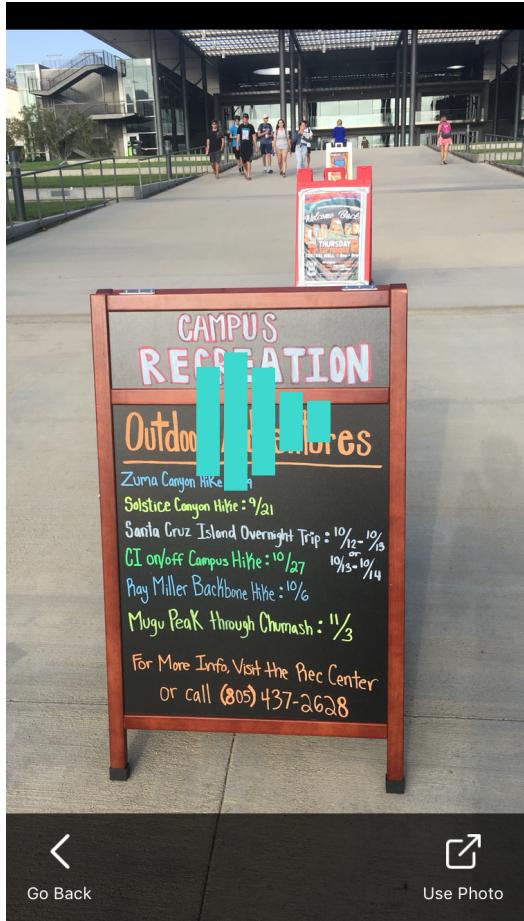


Figure 16: Feedback design principle  
We added a 'loading' animation to let the user know that there is a process running on the background.

	Data
<b>Smartphone</b>	iPhone 6s
<b>iOS version</b>	11.3
<b>Internet</b>	LTE and Wi-Fi
<b>iBeacons</b>	Gimbal Series 10

Table 2: Testing environment

## 10 Case study

We conducted a small scale evaluation of the multilingual digital signage system as mentioned in Section 8. The purpose of this evaluation is to study the feasibility of using the inexpensive iBeacons along with computer vision features of the system. We set up a combination of digital and printed signages in a room. We set up three iBeacons along with two printed signages and measuring points in a room, approximately 45m x 35m.

The iBeacons used in this experiment were Gimbal Series 10 beacons, distributed by Gimbal. For reception, we used installed the developed mobile application on iPhone 6s, running iOS version 11.3. We also setup two printed signages with information in Spanish language and placed them around the library, at least 30 meters apart.

### 10.1 Testing environment

Three iBeacons were configured as digital signages and setup along with two printed signages in a room, approximately 45m x 35m. The developed mobile application was installed on iPhone 6s, running iOS version 11.3 with a 4G network. Table 2 shows the testing environment for case study.

### 10.2 Digital Signages

These include the signages with iBeacons installed. There are two interactions during the discovery of nearby digital signages from the mobile application. The first part is interaction between iBeacons and the user. iBeacon will broadcast signals at a certain interval, if the smartphone detects and locates the iBeacons around it, the application can handle the iBeacon event

based on proximity location information, including major ID, minor ID and distance. The second part is interaction between the mobile application and database. During this interaction, the application should fetch respective information and display on the mobile application.

#### **10.2.1 Test Case 1**

With the above mentioned setup of the signages inside the campus library, we use the mobile application to evaluate the feasibility of the proposed system. Each signage is at least 20 meters away from each other. We downloaded the developed mobile application on to the iPhone 6s and walked towards the room.

The mobile application picked up the closest digital signage when the user was 17.4 meters away. As and when the user walked towards the signage, the distance decreased appropriately. The other digital signages were listed at the distance of 22.3 meters and 27 meters away. On tapping each thumbnail of signage from the application, it fetched and displayed relevant signage information from the database. We were also able to translate the signage description from English to Spanish and the translation was 100% correct.

	Data
<b>Discovered</b>	27.4 meters away
<b>Fetched signage data</b>	"Hazardous materials present at this site. No trespassing."
<b>Text translated</b>	"Materiales peligrosos presentes en este sitio. Prohibido el paso. No hay vehículos más allá de este punto. Los vehículos autorizados deben exhibir permisos para ingresar. Zona bajo estricta vigilancia."
<b>Error in text translated</b>	0%

Table 3: Observations from test case 1

We find that the digital signage was discovered 27.4 meters away from the user and the appropriate message is translated to Spanish with 0% error

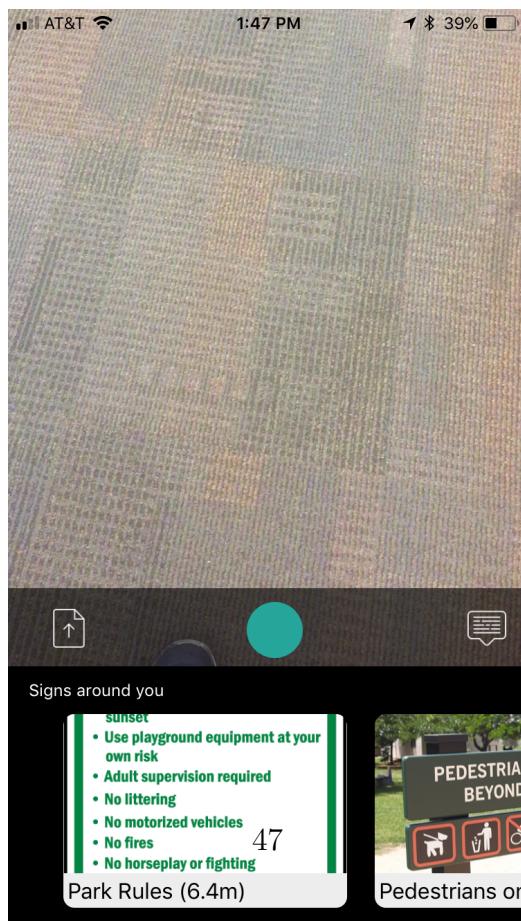


Figure 17: Digital signage 1

We notice that the digital signage called 'Park Rules' is discovered from the mobile application at a distance of 6.4 meters away

### **10.2.2 Test Case 2**

For this test case, we setup the digital signages closer than the previous test case. The aim of this test case is to evaluate for interference in beacon signals. Each signage was installed 15 meters away from each other. We ran the developed mobile application on the iPhone 6s and walked towards the room.

The mobile application picked up the closest digital signage when the user was 23.4 meters away. The other digital signages were listed at the distance of 26.3 meters and 35.7 meters away. On tapping each thumbnail of signage from the application, it fetched and displayed relevant signage information from the database. We were also able to translate the signage description from English to Spanish and the translation was 100% correct.

This proves that the digital signage system can be set up in close proximity to each other and the system will not face any interference from the crowded iBeacon signals.

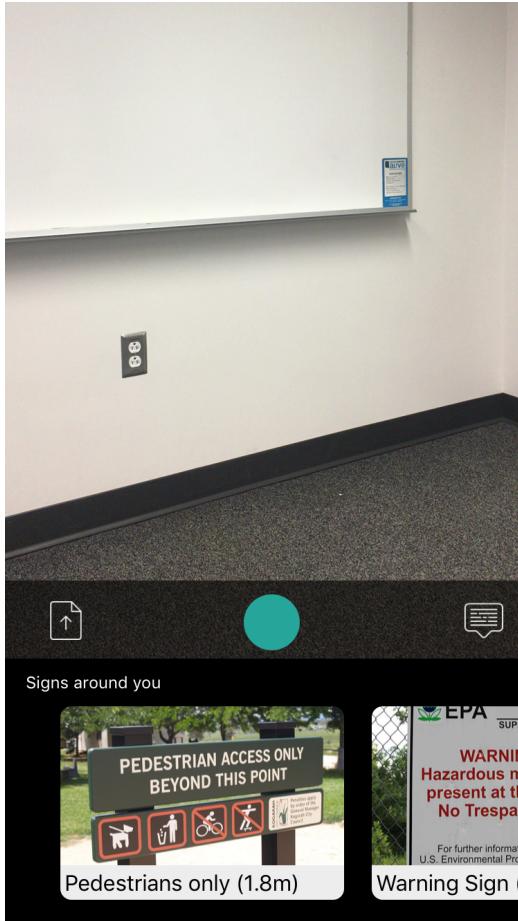


Figure 18: Digital signage 2

When the digital signages are placed in close proximity to each other, the mobile application does not interfere with the crowded signals. Hence, the mobile application works when we setup the signages in close proximity.

### 10.3 Printed Signages

The other scenario we need to evaluate is the multilingual translation of printed signages (signages without iBeacons installed). When the user clicks an image of a signage, the mobile application should be able to extract text and translate it to the user's preferred language without missing critical information. We evaluate the feasibility of the system in the following scenarios.

	Data
<b>Discovered</b>	19.4 meters away
<b>Fetched signage data</b>	”Pedestrian access only beyond this point.”
<b>Text translated</b>	”Acceso peatonal solo más allá de este punto.”
<b>Error in text translated</b>	0%

Table 4: Observations from test case 2

We notice that another printed signage was discovered at a distance of 19.4 meters away from the user. On tapping the thumbnail on the mobile application, the system fetched the relevant details of the signage. We could also translate the fetched information into Spanish with 0% error.

#### 10.3.1 Test Case 3

We use the same mobile application used in test cases 1 and 2 to evaluate the feasibility of the system to extract text and translate it from Spanish to English. We positioned the mobile application at a distance of 35 feet from the signage.

To evaluate the text extraction and translation from printed signage, we clicked a picture of the printed signage. The application extracted relevant text from the printed signage. The application also supported the translation of the extracted text to Spanish language and the translation was 100% correct.

We noticed an 0% error in the text extraction and text translation from Spanish to English. Below figures 20 and 21 show the flow of the text extraction and translation for printed signage 1.



Figure 19: Printed Signage 1  
The screenshot shows the mobile application about to capture the image of a printed signage in Spanish language.

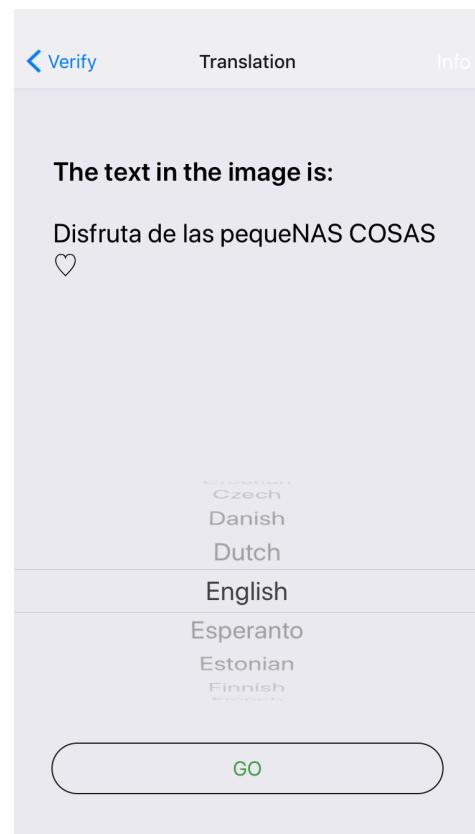


Figure 20: OCR feature  
This screenshot shows the text extracted from the printed signage shown in Figure 19. We used OCR feature to extract the text.

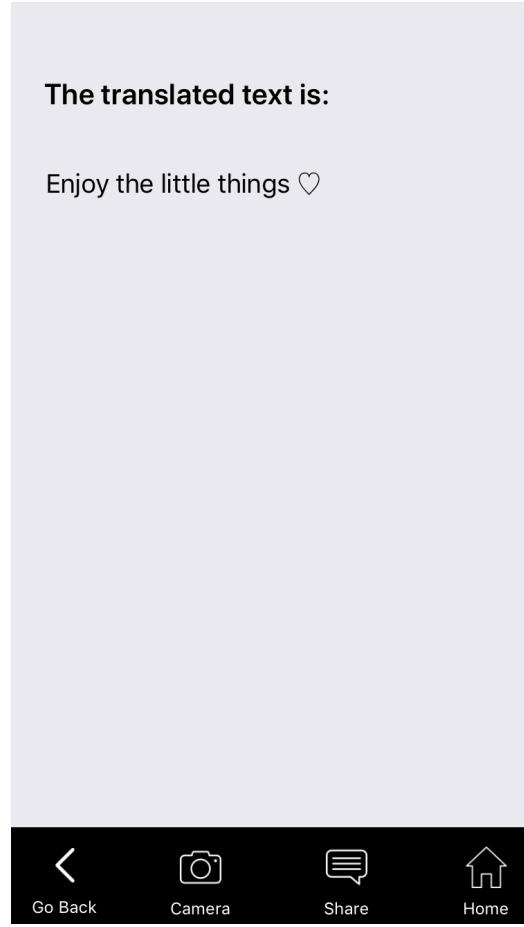


Figure 21: Text translated to English

The extracted text from figure 19 is then translated to English using the Google Translate API. We notice that the text is translated with 0% error, including the emoticon.

### 10.3.2 Test Case 4

For this test case, we use another printed signage with information in Spanish language and use our system to extract the text and translate it to English. The aim of this test case is to evaluate the text extraction and translation of different languages.

	Data
<b>Image captured from</b>	iPhone 6s
<b>Text extracted</b>	”Disfruta de las pequenas”
<b>Text translated</b>	”Enjoy the little things”
<b>Error in text extraction</b>	0%
<b>Error in text translated</b>	0%

Table 5: Observations from test case 3

The table shows the text extracted from the printed signage shown in Figure 19. We notice that the text is extracted from the Spanish printed signage and translated to English with 0% error in text extraction and translation

To evaluate the text extraction and translation from a Spanish printed signage, we clicked a picture of the printed signage. The application extracted relevant text from the printed signage. The application also supported the translation of the extracted Spanish text to English language and the translation was 91% correct.

We noticed an 8% error in the text extraction feature and thereby the same error while translating the text to English. Below Figure 22, 23 and 24 show the flow of the text extraction and translation for signage 2.

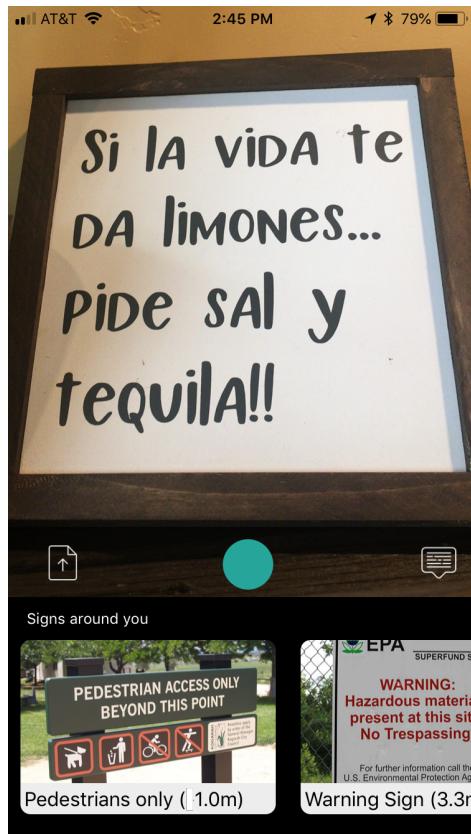


Figure 22: Printed Signage 2  
The screenshot shows the mobile application about to capture the image of a printed signage in Spanish language



Figure 23: OCR feature  
This screenshot shows the text extracted from the printed signage shown in Figure 22. We used OCR feature to extract the text.

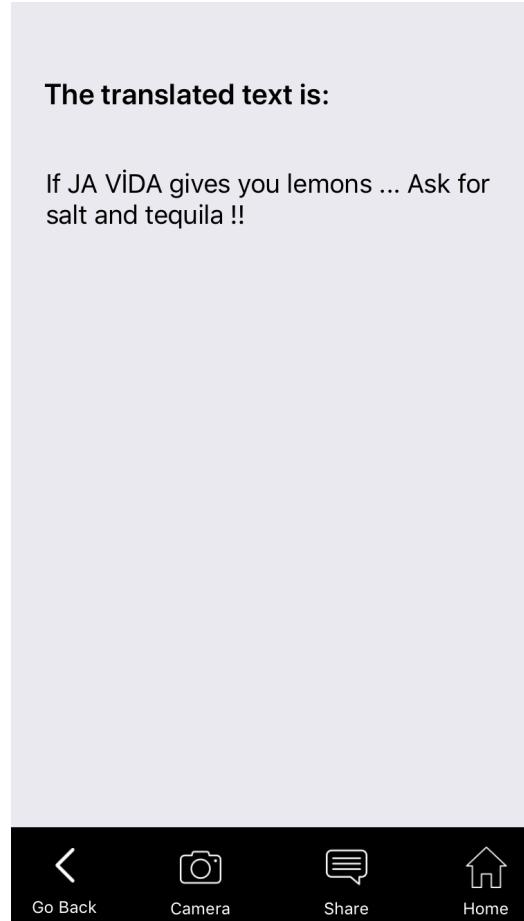


Figure 24: Text translated to English

The extracted text from Figure 22 is then translated to English using the Google Translate API. We notice that the text is extracted with with 4% error and 8% error in text translation.

	Data
<b>Image captured from</b>	iPhone 6s
<b>Text extracted</b>	”Si la vida te Da limonoes... pide sal y tequila!!”
<b>Text translated</b>	”If JA VIDA gives you lemons... Ask for salt and tequila !!”
<b>Error in text extraction</b>	4%
<b>Error in text translated</b>	8%

Table 6: Observations from test case 4

The table shows the text extracted from the printed signage shown in Figure 19. We notice that the text is extracted from Spanish language with 4% error and 8% error in text translation from Spanish to English language.

## 11 Conclusion and future work

We studied the feasibility of using the combination of inexpensive iBeacons and computer vision to construct a system of multilingual digital signages. We conducted a preliminary experiment using an actual room to estimate the positioning accuracy and accuracy of language translation. When three beacons, placed on individual signages, we were able to discover them from the mobile application when in proximity. We were able to fetch respective data from the server and display signage information on the mobile application. We could also translate the information to any of the supported languages. Using the same mobile application, we could also extract text from printed signages and translate the same into different languages with acceptable context of translation. We believe this level of accuracy is acceptable for a system of multilingual digital signage. This system proves to be useful for non-native visitors to understand their surroundings better and obtain critical information using the mobile application.

Future work could be done on determining user's position in the network of iBeacons by triangulating the user's position using the RSSI of three beacons in the network . Further, the user experience of the mobile application could be enhanced by translating just the text in the signage while maintaining the graphics like color and font of the signage. The system in this research could also be extended to be used in museums and shopping malls.

## References

- [1] C. Takahashi and K. Kondo, “Indoor positioning method for augmented audio reality navigation systems using ibeacons,” in *2015 IEEE 4th Global Conference on Consumer Electronics (GCCE)*, pp. 451–452, Oct 2015.
- [2] T. Ogi, K. Ito, and S. Konita, “Multilingual digital signage using ibeacon communication,” in *2016 19th International Conference on Network-Based Information Systems (NBiS)*, pp. 387–392, Sept 2016.
- [3] P. Salunkhe, S. Bhaskaran, J. Amudha, and D. Gupta, “Recognition of multilingual text from signage boards,” in *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 977–982, Sept 2017.
- [4] A. S. Gudrun Gorlitz, “Digital signage: Informal learning in animal parks and zoos,” *World Conference on E-Learning in Corporate Government Healthcare and Higher Education 2008*, vol. pp. 841-846, pp. 841–846, 2008.
- [5] Y. Jeng, M. Hong, and H. Lu, “The development of the mobile situated learning application based on microlocation technology,” in *2017 6th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI)*, pp. 715–718, July 2017.
- [6] Y. Yang, Z. Li, and K. Pahlavan, “Using ibeacon for intelligent in-room presence detection,” in *2016 IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA)*, pp. 187–191, March 2016.
- [7] S. Rajkiran and C. Balakrishnan, “Rssi estimation using bluetooth in smart phones,” in *2015 2nd International Conference on Electronics and Communication Systems (ICECS)*, pp. 1431–1434, Feb 2015.
- [8] S. Sonth and J. S. Kallimani, “Ocr based facilitator for the visually challenged,” in *2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEEC-COT)*, pp. 1–7, Dec 2017.

- [9] M. Koöhne and J. Sieck, “Location-based services with ibeacon technology,” in *2014 2nd International Conference on Artificial Intelligence, Modelling and Simulation*, pp. 315–321, Nov 2014.
- [10] X. Li, D. Xu, X. Wang, and R. Muhammad, “Design and implementation of indoor positioning system based on ibeacon,” in *2016 International Conference on Audio, Language and Image Processing (ICALIP)*, pp. 126–130, July 2016.
- [11] T. Peng, X. Wang, C. Wang, and D. Shi, “Hybrid wireless indoor positioning with ibeacon and wi-fi,” in *11th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM 2015)*, pp. 1–5, Sept 2015.
- [12] H. Zou, Z. Chen, H. Jiang, L. Xie, and C. Spanos, “Accurate indoor localization and tracking using mobile phone inertial sensors, wifi and ibeacon,” in *2017 IEEE International Symposium on Inertial Sensors and Systems (INERTIAL)*, pp. 1–4, March 2017.
- [13] M. S. Aman, C. D. Quint, A. Abdalgawad, and K. Yelamarthi, “Sensing and classifying indoor environments: An iot based portable tour guide system,” in *2017 IEEE Sensors Applications Symposium (SAS)*, pp. 1–6, March 2017.
- [14] Z. He, B. Cui, W. Zhou, and S. Yokoi, “A proposal of interaction system between visitor and collection in museum hall by ibeacon,” in *2015 10th International Conference on Computer Science Education (ICCSE)*, pp. 427–430, July 2015.
- [15] S. Bai, C. Chiu, J. Hsu, and J. Leu, “Campus-wide wireless indoor positioning with hybrid ibeacon and wi-fi system,” in *2017 6th International Symposium on Next Generation Electronics (ISNE)*, pp. 1–2, May 2017.
- [16] P. Burzacca, M. Mircoli, S. Mitolo, and A. Polzonetti, ““ibeacon” technology that will make possible internet of things,” in *International Conference on Software Intelligence Technologies and Applications International Conference on Frontiers of Internet of Things 2014*, pp. 159–165, Dec 2014.

- [17] C. Takahashi and K. Kondo, “Accuracy evaluation of an indoor positioning method using ibeacons,” in *2016 IEEE 5th Global Conference on Consumer Electronics*, pp. 1–2, Oct 2016.
- [18] Wikipedia, “Representational state transfer (rest) - <https://en.wikipedia.org/wiki/.>”
- [19] M. F. M. V. Davide Mulfari, Antonio Celesti and A. Puliafito, “Using google cloud vision in assistive technology scenarios,” *IEEE Workshop on ICT solutions for eHealth 2016*, 2016.
- [20] S. Dwarakanath, “<https://github.com/suhasd/thesis>.”
- [21] G. Firebase, “<https://firebase.google.com/docs/>.”
- [22] M. Seraj and C. Y. Wong, “A study of user interface design principles and requirements for developing a mobile learning prototype,” in *2012 International Conference on Computer Information Science (ICCIS)*, vol. 2, pp. 1014–1019, June 2012.
- [23] X. Fu, “Mobile phone ui design principles in the design of human-machine interaction design,” in *2010 IEEE 11th International Conference on Computer-Aided Industrial Design Conceptual Design 1*, vol. 1, pp. 697–701, Nov 2010.
- [24] D. A. Norman, “Emotional design,” *House of Electronics Industry*, pp. 44–79, 2005.