A System of Applications for the Integration of BLE Beacons in Museums

by Kristin Nicole Asmus

S.B., Massachusetts Institute of Technology (2015)

Submitted to the
Department of Electrical Engineering and Computer Science
in Partial Fulfillment of the Requirements for the Degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

Massachusetts Institute of Technology

September 2016

Author: ______________________________
Department of Electrical Engineering and Computer Science
August 12, 2016

Certified by: ______________________________
Harold Abelson, Class of 1922 Professor of C.S. and Engineering, Thesis Supervisor
August 12, 2016

Accepted by: ______________________________
Dr. Christopher J. Terman, Chairman, Masters of Engineering Thesis Committee
August 12, 2016

A System of Applications for the Integration of BLE Beacons in Museums

by Kristin Nicole Asmus
Submitted to the Department of Electrical Engineering and Computer Science

August 12, 2016
In Partial Fulfillment of the Requirements for the Degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

There is great potential for BLE beacon technology to be incorporated into museum exhibits to provide more enriching content to visitors and gather feedback to improve the museum overall. I developed a system of three applications to accomplish these goals: one for museum staff to easily deploy and update beacons, another for visitors to engage with the supplemental material transmitted by beacons at various exhibits, and a third for museum staff to understand visitors' interactions and feedback provided by the beacons.

I implemented two versions of the system according to two different beacon protocols: Eddystone-UID, which broadcasts to a dedicated mobile app; and Eddystone-URL, which broadcasts to mobile browsers via the Physical Web.

Each system was tested in a real-world context through experiments hosted by the MIT Museum. I analyzed the success and potential of such systems based on observations, participant responses, and data gathered via the system during the experiments. The reliability of BLE connections remains an issue, and beacon content only appealed to a small subset of visitors; however, those who interacted with the system valued the experience highly and the MIT Museum expressed interest in providing a richer experience for these visitors.

Thesis Supervisor: Harold Abelson
Title: Class of 1922 Professor of C.S. and Engineering

Acknowledgments

First and foremost, I'd like to thank my advisor, Hal Abelson, without whom this project would not have been possible. From inspiring me to explore applications of BLE beacons to encouraging me to collaborate with the MIT Museum, Hal's vision helped me formulate a project idea that I was both confident and excited to tackle. Thank you also for your excellent advice along the way and the feedback to present my work compellingly.

Thank you as well to the App Inventor group at large; while my work was tangential to the App Inventor project this year, being a part of such an inspiring and driven group has been an amazing experience.

I owe an incredible amount of thanks to the phenomenal staff at the MIT Museum. Their willingness to collaborate and showcase student work encouraged me to take this project to the next level with live testing. Thank you for the time and effort that went into hosting and supporting my beacon experiment; especially to Anne, Cody, and Emma, who took a vested interest, and most importantly to Susan, who facilitated the experiment from the museum's side and made the whole thing possible. Thank you for the amazing opportunity!

Speaking of the experiments, I'd like to thank everyone who participated, from the anonymous museum visitors to the friends and colleagues who came out specifically to support my thesis. Thank you especially to Elliott and the troops from J Entry he rallied to come out and contribute when participation and morale were low. To all of you, your support meant the world.

Thank you to my family; you are the rock beneath my feet. I know I can always count on you for love and encouragement, whether I'm sharing the successes or the failures. With you at my back I can do anything. Thank you for everything you've done that's gotten me this far.

And finally, to MIT, and the friends near and far who made the past five years such an incredible experience: It's been the time of my life. Thank you.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The purpose of this thesis project was to create a system of applications to integrate BLE beacons into museums. BLE Beacons are small devices that can send short messages to nearby receivers such as smartphones using a low-power version of Bluetooth. In a museum, these beacons could be attached to various displays so that visitors passing by could receive additional related content on their smartphones. Furthermore, museums could benefit from information collected about visitors' exchanges with the system.

To achieve this goal, I developed a system of three components. The first, the Configuration Component, helps museum staff to set up or update their fleet of beacons to broadcast the desired content. Next, the Interaction Component provides a framework for how visitors receive and interact with the content beacons transmit to their smartphones. Finally, the Report Component enables museum staff to view live aggregate statistics on visitors' interactions.

As an additional contribution of this thesis, I developed and compared the described system according to two different frame types, or data formats, that beacons can use to broadcast: Eddystone-UID and URL. The difference is most notable in the Interaction Components, where visitors either download a dedicated museum app to view beacon content, or receive it in the form of Physical Web notifications linking to websites they can open in certain mobile browsers.

I then tested both complete systems in a real-world environment at the MIT Museum, gathered data on their effectiveness, and analyzed the results to evaluate the future of such applications. Findings suggested that only a minority of museum visitors would be interested in engaging with additional content on their smartphones, but that this minority valued the experience very highly.

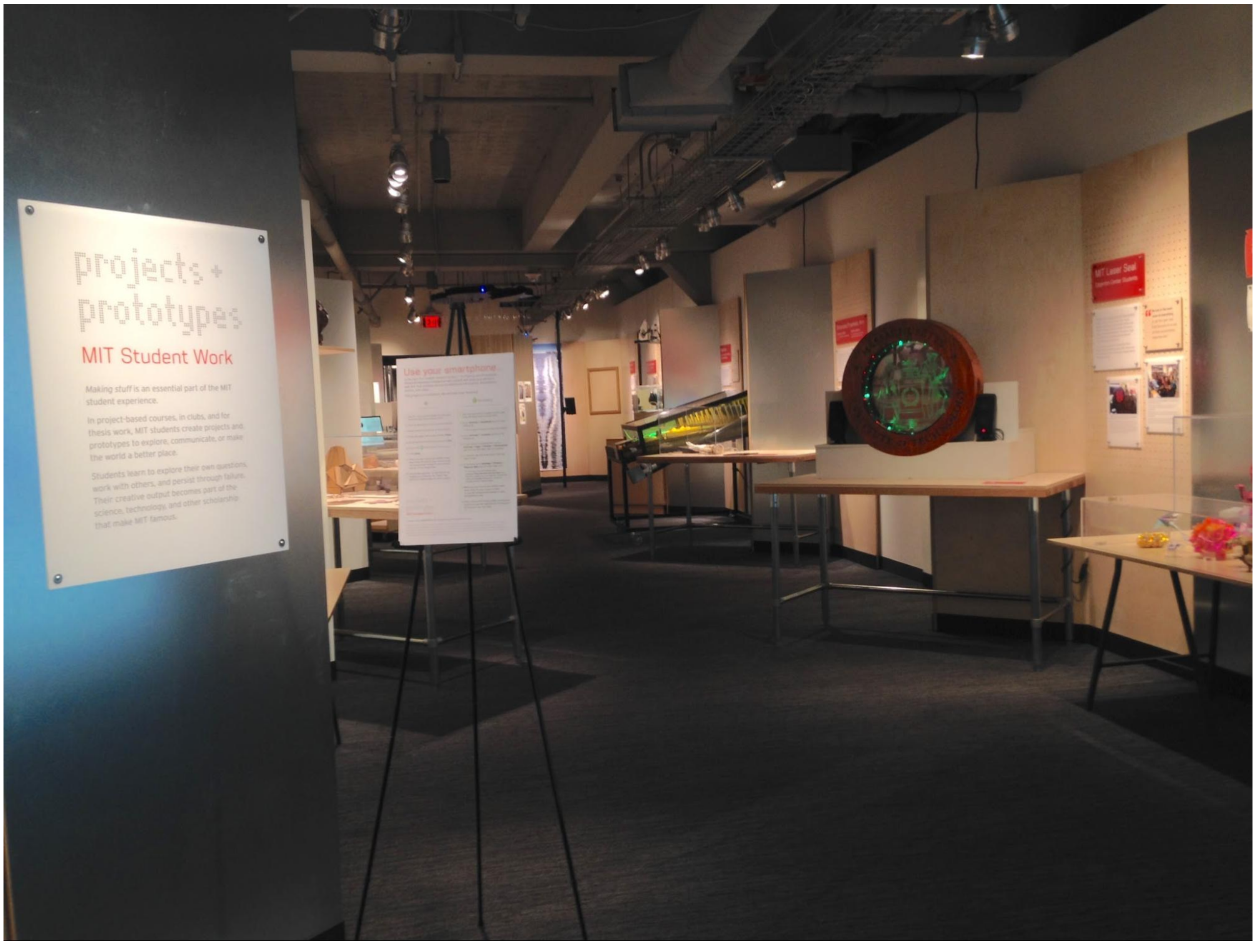

projects +
prototypes
MIT Student Work
Making stuff is an essential part of the MIT student experience.
In project-based courses, in clubs, and for thesis work, MIT students create projects and prototypes to explore, communicate, or make the world a better place.
Students learn to explore their own questions, work with others, and persist through failure. Their creative output becomes part of the science, technology, and other scholarship that make MIT famous.
Use your smartphone

Figure 1-1: A photograph of the MIT Museum Projects and Prototypes exhibit hall with a poster encouraging visitors to participate in the experiment by interacting with beacons via their smartphones.

In terms of the system's feasibility, the biggest obstacle was the unreliability of some devices, notably those running Android, in discovering the BLE transmissions sent by beacons.

In Chapter 2, I provide a brief background on BLE, beacons, and the Eddystone frame types used to develop the two versions of the system, followed by motivations for applying these to a museum setting. I explain how these systems were created in Chapter 3, beginning with a walkthrough from a user's perspective, then discussing in detail the development of the Eddystone-UID version, and wrapping up with the differences in the Eddystone-URL implementation. In Chapter 4, I describe the experiments performed in collaboration with the MIT Museum and the results I found. In Chapter 5, I draw conclusions about the effectiveness of the system as a whole. Chapter 6 is a discussion of future work, and Chapter 7 summarizes the contributions of this thesis.

Chapter 2

Background and Motivation

In this chapter, I will provide background information on BLE, beacons, and the Eddystone beacon format. I then motivate the application of these technologies to museum settings.

2.1 About BLE and Beacons

The advent of Bluetooth Low Energy beacons has opened up new possibilities in the past few years for applications to take advantage of microlocation capabilities. Bluetooth Low Energy (or BLE) is an inexpensive and energy-efficient version of the Bluetooth communication standard that has made communication between nearby devices more feasible than ever before, greatly empowering advances in the Internet of Things [1]. Small "beacon" devices make use of this technology to broadcast tiny amounts of data (usually just the beacon ID and signal power information used to estimate distance) periodically over an extended lifetime [2]. These signals are picked up by other devices, typically mobile smartphones, which relay the ID over a network and receive in return the potentially much larger set of content associated with that beacon [3]. Dozens of beacon manufacturers have entered the scene, and beacons are gradually reaching a price point that will make feasible much larger scale applications and likely foster a burst of development in the space.

Major manufacturers of beacons currently include Estimote, Kontakt.io, Radius Networks, BlueCats, Gimbal by Qualcomm, Access Systems, and more. Most beacons last a year or longer before requiring new batteries or replacement, and can broadcast from within a few meters to tens of meters away. The most popular commercial application of beacons presently is for marketing within retail businesses, which are able to advertise more directly to customers based on when they enter a physical store, or even where within the store they are currently shopping. Dozens of other organizations are taking advantage of the microlocation capabilities offered by beacons, including zoos, libraries, theme parks, public transportation agencies, conferences, and home automation companies, to name a few [4].

2.2 About Google's Eddystone Protocol

Introduced in 2013, Apple's iBeacon [5] was the first major player in establishing a protocol for these beacons; but more recently, Google's Eddystone format [6] has entered the ring, touting its robustness, cross-platform compatibility, and open-source availability. While iBeacon is compatible only with specific devices, Eddystone is designed to work with any BLE-enabled device. Open-sourcing also makes the protocol very friendly to the developer community, and support for multiple frame types, or data payloads, means that Eddystone can transmit generic URLs that can be opened via mobile browsers [7], as opposed to the UUIDs of iBeacon which are tied to specific applications.

With the support of Google and adoption by dozens of major hardware manufacturers of beacons already, Eddystone is well positioned to make the beacon environment more compatible and developer-friendly than ever [8] by opening up the market for different hardware manufacturers, smartphone platforms, and developers to all collaborate.

Eddystone includes several frame type options that can be broadcast by beacons [9]. In this thesis, I will compare the usage of two of them: Eddystone-UID and URL. The frame types are GATT services which determine the format of the information that the beacon advertises to nearby devices. These advertisements are a one-to-many broadcast as opposed to a one-to-one connection; any listening device can receive the information these beacons send out. The format of this data affects how it can be viewed by users.

2.2.1 Eddystone-UID and APIs

The Eddystone-UID (Unique Identification Number) frame type consists of 16-byte beacon ID with a 10-byte namespace and 6-byte instance identifier [10]. Because this is the only information broadcasted out by a beacon, Google has several APIs that handle associating that ID with additional data. The Proximity Beacon API handles maintaining a global registry mapping beacons' IDs to their attachment data (the Google Beacon Registry) [11], as well as monitoring the health and status of beacons [12]. Google's Nearby Messages API enables apps to subscribe to automatically receive messages from detected beacons [13].

A developer is therefore able to configure a beacon's UID to be associated with the data they want to send to users of their app. Their app is then programmed to look up the UID in Google's cloud when it finds one of their beacons, and pull down the relevant information for users to view in the app.

2.2.2 Eddystone-URL and the Physical Web

Eddystone-URLs are an alternate frame type that broadcasts only an 18-byte encoded URL, or web address, instead of an ID [14]. It doesn't have any additional data attached; it just sends that web address out to nearby smartphones. This address can be automatically received by mobile browsers that support the Physical Web project, and presented to users in the form of a simple notification that, when clicked, takes them to that webpage in the browser.

The Physical Web is an open project by Google that aims to enable "quick and seamless interactions with physical objects and locations" [15]. It enables mobile browsers to scan for nearby BLE devices. If these devices are broadcasting URLs via the Eddystone-URL protocol, then a simple click transports the user to that webpage. The Physical Web is currently supported by Google Chrome on iOS and Android, Opera and Firefox on Android, and several smaller players [16]. Because any object or location can have a beacon attached, the Physical Web touts endless possibilities for enabling the Internet of Things.

In this scenario, a developer creates a webpage with their desired content, then sets up a beacon broadcasting the address of that page. A user can enable a Physical Web browser like Google Chrome, and then upon coming into range of the beacon will instantly receive a non-intrusive notification that can take them to the webpage if they desire.

2.3 Motivation for Application in Museums

One setting that is particularly well suited to the integration of such microlocation technology is an indoor location where context changes depending on an individual's exact position, and more information regarding that context could be of use. In particular, beacon technology could be of huge benefit in a museum, where beacons can tell which exhibit is closest to a person and provide them with interesting supplemental material to that display. This use case has caught the attention of many developers, from several beacon manufacturers who advertise it as an application [17, 18], to several museums that have tried it out [19, 20, 21, 22]. There is even a current startup in the market providing consultation to museums on integrating beacons to supplement their exhibits [23]. However, one aspect that none of these ventures had taken advantage of yet was the possibility of a two-way flow of information. Though the beacons themselves cannot receive information back from smartphones, the websites or apps they link to could.

While museums are an excellent application for beacons to showcase their ability to relay rich media content - video, audio, pictures, text, even interactive options - they also provide an excellent opportunity for an organization to learn from how their users interact with that content. If visitors' views of that content and interactions with it could be delivered back to the curators, the possibility of improving the museum experience expands beyond the addition of supplemental content to also incorporate valuable data about how visitors are experiencing and enjoying different exhibits. In addition to the benefits of quickly, cheaply, and conveniently being able to add more engaging and up-to-date content to exhibits, this potential for real-time feedback makes museums an ideal problem space to address with beacons.

Chapter 3

Developed System

In this chapter, I first provide an overview of the system I developed and its three components. Next, I walk through of the system from a user perspective, to provide a basis for understanding what the system does. The following sections describe the development of each of the system's components. For Android apps, I used Google's Android Studio IDE and coded in native Java. When developing for the web, I used the editor Sublime Text 2 and coded primarily in Javascript. The first version of the system was developed using the Eddystone-UID frame type, and will be the primary focus of this chapter. The differences in the Eddystone-URL implementation are reserved for discussion in the final section.

3.1 Overview

In order to create a system to incorporate beacons in museums and take advantage of a two-way flow of information, I developed a system of three interconnected applications: a Configuration Component, an Interaction Component, and a Report Component. The first tool, intended for use by museum staff, facilitates quickly and easily deploying or updating the contents of a fleet of beacons. The second provides a platform for visitors to view the content transmitted by those beacons and interact with these extensions of the exhibits. The final application, a live display of the aggregate data collected from user interactions and feedback, informs the museum on different metrics surrounding the beacons.

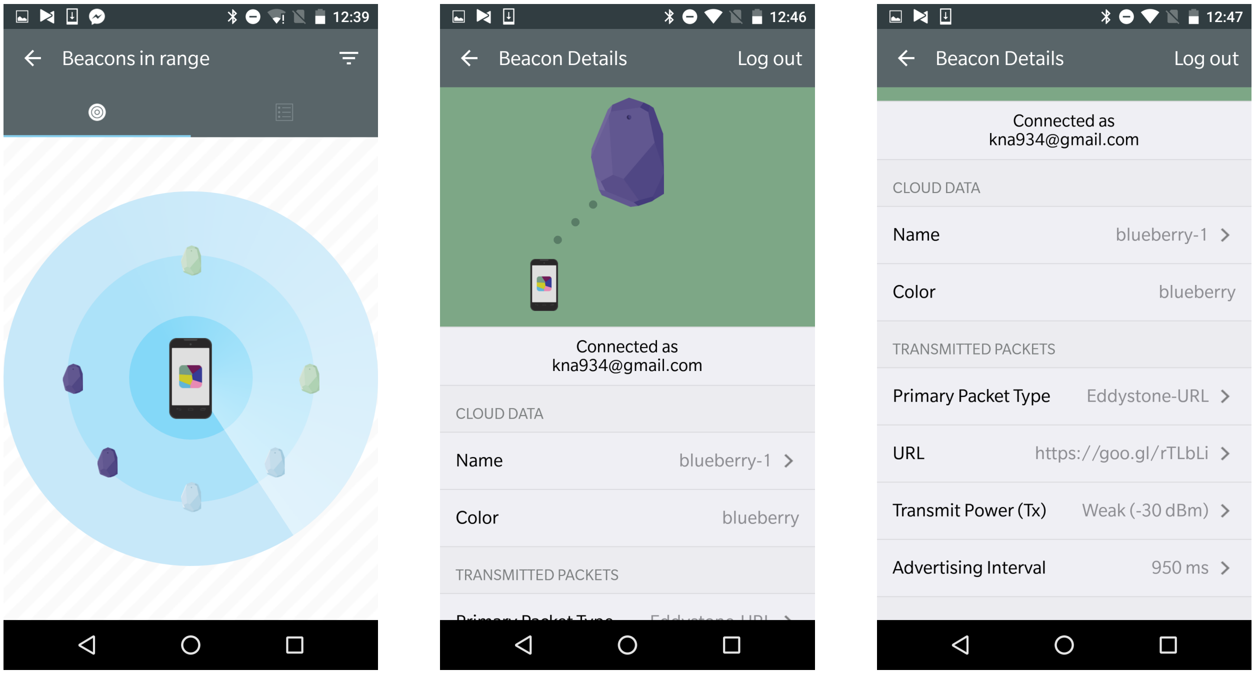
For the museum, this system comes together as a means for them to easily deploy and update a fleet of beacons, a way for their visitors to interact with them, and a useful source of data from these interactions. For visitors, it augments the standard museum experience with the opportunity to interact in more interesting ways with rich, up-to-date supplemental content to existing exhibits.

3.2 System Walkthrough

This section is intended to provide a very brief overview of the system from the user's perspective, whether that user is a museum staff member setting up their beacons or viewing the aggregate data they've gathered, or a museum visitor interacting with beacon content. Why these components behave the way they do and how they work under the hood will be explained in depth in later sections.

3.2.1 Configuration Walkthrough

To begin setting up their fleet of beacons, museum staff must first use the software developed by the beacon manufacturer to adjust hardware-specific settings. Because I opted to work with Estimote beacons for this thesis (a choice discussed further in section 3.3.1), their app is depicted below.



Connected as
kna934@gmail.com

CLOUD DATA

Name blueberry-1

Color blueberry

TRANSMITTED PACKETS

Primary Packet Type Eddystone-URL

Connected as
kna934@gmail.com

CLOUD DATA

Name blueberry-1

Color blueberry

TRANSMITTED PACKETS

Primary Packet Type Eddystone-URL

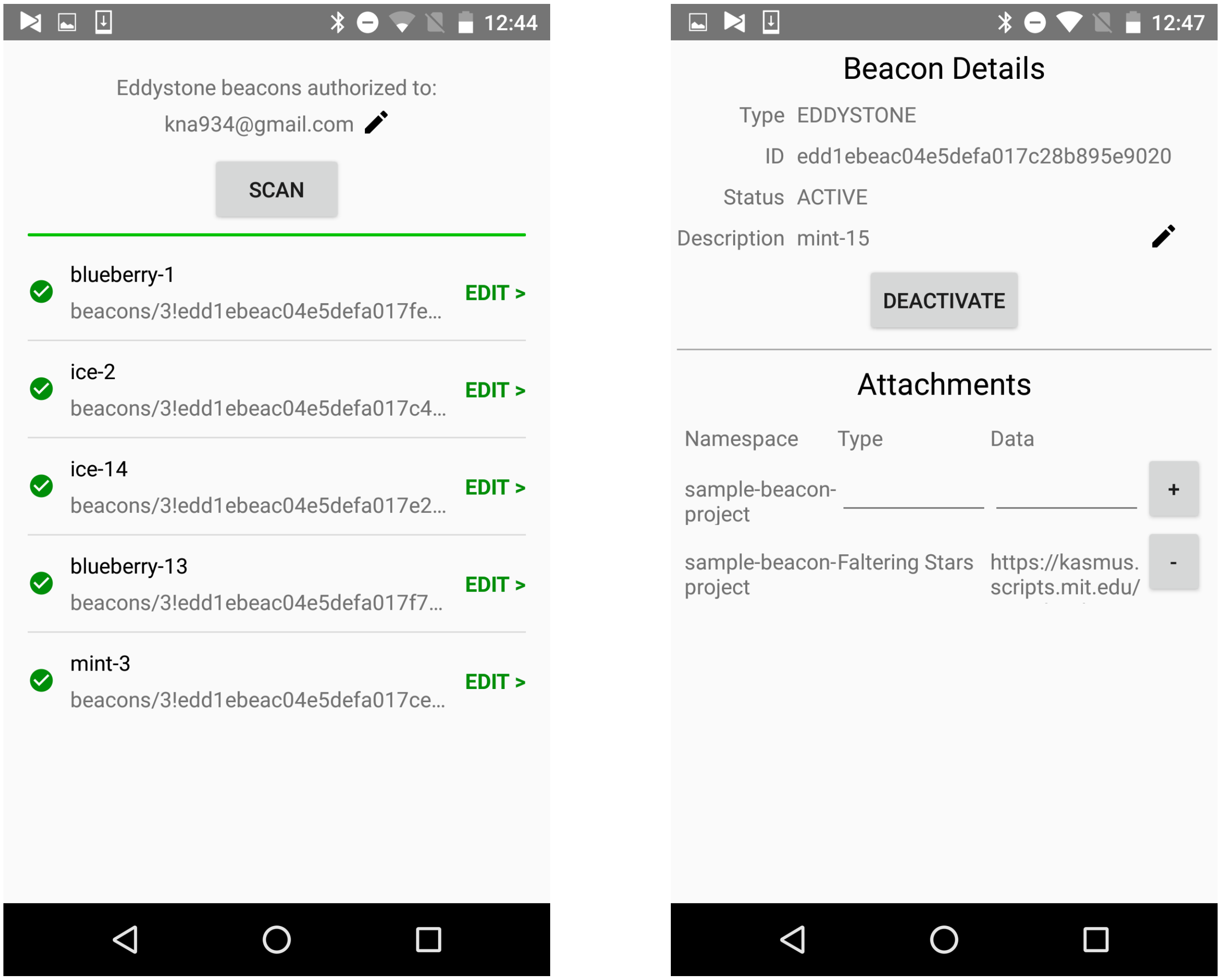
URL https://goo.gl/rTLbLi

Transmit Power (Tx) Weak (-30 dBm)

Advertising Interval 950 ms

Figure 3-1: The Estimote app scans for nearby beacons in the leftmost screenshot. The middle and right screenshots display the settings one can modify after selecting a beacon to configure.

If the Eddystone-URL frame type is being used, museum staff can set the desired URL within this software to complete the setup. However, for the Eddystone-UID frame type, an additional tool is necessary to register the beacon's ID and attach associated data.



12:44

Eddystone beacons authorized to:
kna934@gmail.com

SCAN

- ✓ blueberry-1 — beacons/3!edd1ebeac04e5defa017fe... — EDIT >
- ✓ ice-2 — beacons/3!edd1ebeac04e5defa017c4... — EDIT >
- ✓ ice-14 — beacons/3!edd1ebeac04e5defa017e2... — EDIT >
- ✓ blueberry-13 — beacons/3!edd1ebeac04e5defa017f7... — EDIT >
- ✓ mint-3 — beacons/3!edd1ebeac04e5defa017ce... — EDIT >

12:47

Beacon Details

Type	EDDYSTONE
ID	edd1ebeac04e5defa017c28b895e9020
Status	ACTIVE
Description	mint-15

DEACTIVATE

Attachments

Namespace	Type	Data	
sample-beacon-project			+
sample-beacon-project	Faltering Stars	https://kasmus.scripts.mit.edu/	-

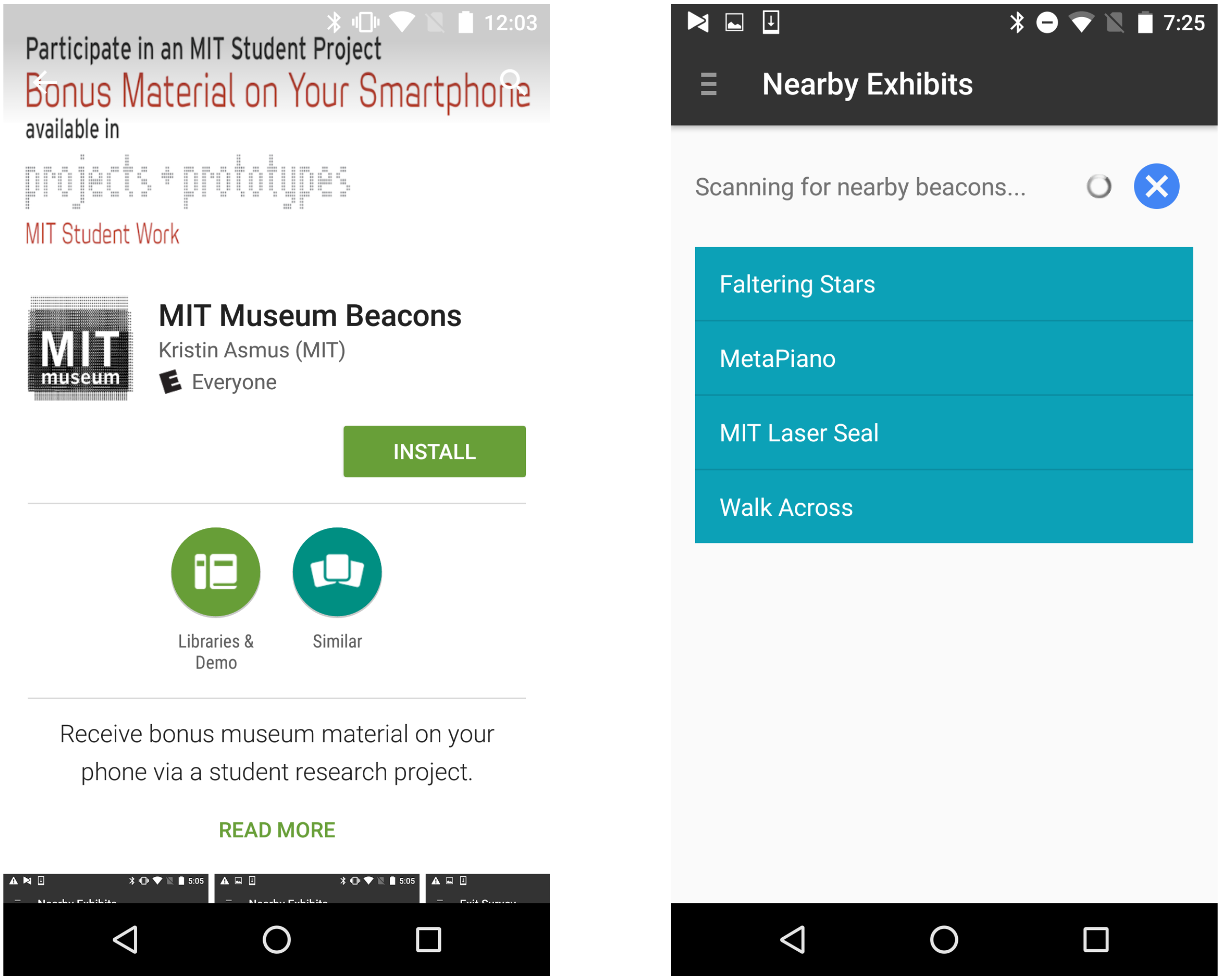
Figure 3-2: The left screenshot shows the Eddystone-UID Setup app after scanning and discovering several nearby beacons. Clicking on a beacon from the list brings the user to the left screenshot, where they can add or remove data attachments associated with that beacon.

3.2.2 Interaction Walkthrough

The way that museum visitors interact with beacon content depends on the frame type the beacons are set to broadcast.

3.2.2.1 Via a Dedicated App

When using the Eddystone-UID frame type, beacons communicate their content via a dedicated smartphone application: for the purposes of this thesis, an Android app. A museum visitor simply downloads the app from the Play Store, turns on their device's Bluetooth setting, and begins scanning for beacons. Upon nearing a display and discovering its associated beacon in the app, visitors can click the beacon title to view additional information about that display.


12:03

Participate in an MIT Student Project

Bonus Material on Your Smartphone

available in

projects + prototypes

MIT Student Work

MIT Museum Beacons

Kristin Asmus (MIT)

Everyone

Receive bonus museum material on your phone via a student research project.

7:25

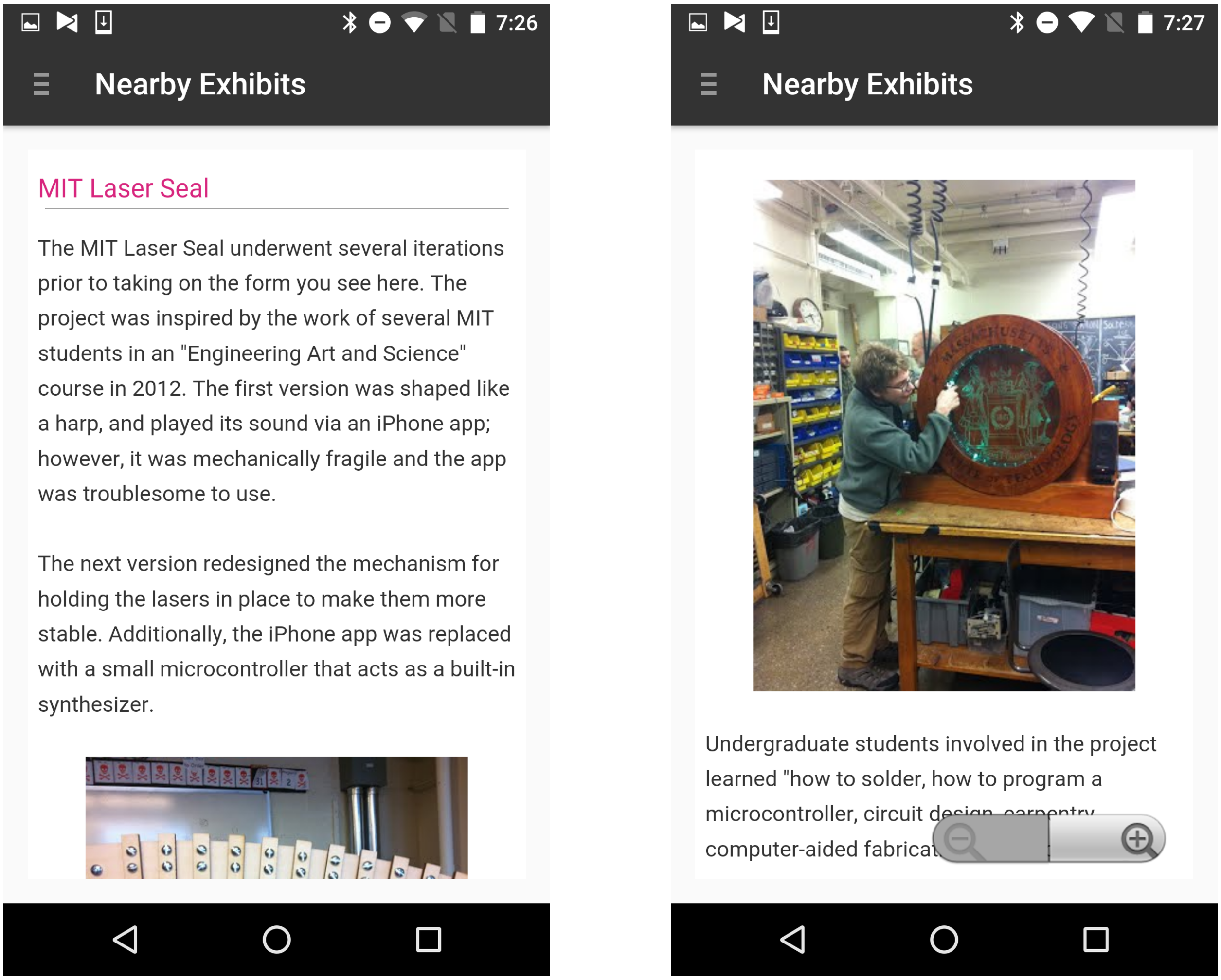
Nearby Exhibits

Scanning for nearby beacons...

- Faltering Stars
- MetaPiano
- MIT Laser Seal
- Walk Across

Figure 3-3: The left screenshot displays the app in the Google Play Store, where visitors can download it to their Android devices. The right screenshot depicts the app in action scanning for nearby beacons and displaying its findings in the blue list.

Additionally from within the dedicated app, visitors can view a map of the museum, check hours and admissions info, find out more about the beacon project, or take a survey about their experience interacting with beacons. These features and others considered are discussed at length



Nearby Exhibits

MIT Laser Seal

The MIT Laser Seal underwent several iterations prior to taking on the form you see here. The project was inspired by the work of several MIT students in an "Engineering Art and Science" course in 2012. The first version was shaped like a harp, and played its sound via an iPhone app; however, it was mechanically fragile and the app was troublesome to use.

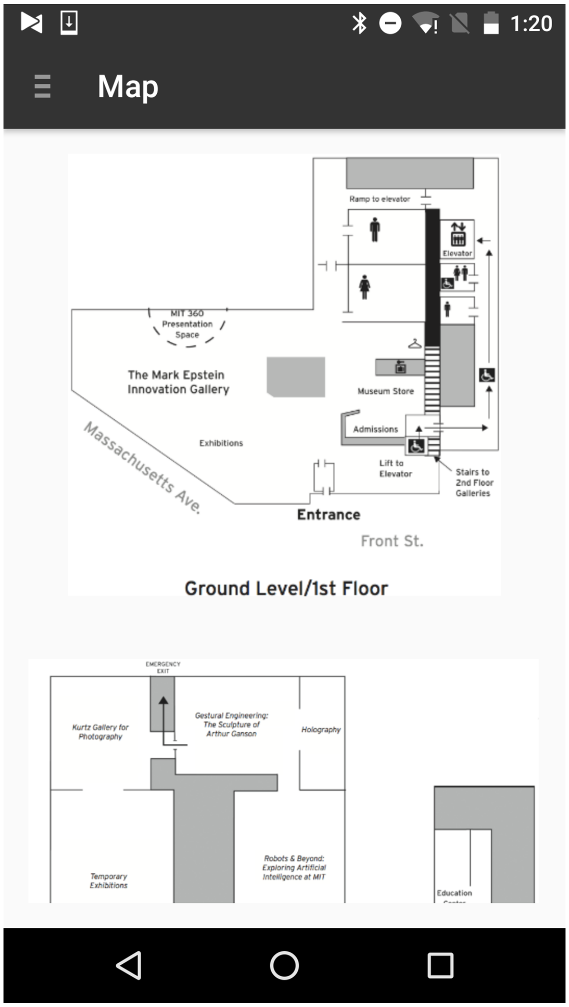
The next version redesigned the mechanism for holding the lasers in place to make them more stable. Additionally, the iPhone app was replaced with a small microcontroller that acts as a built-in synthesizer.

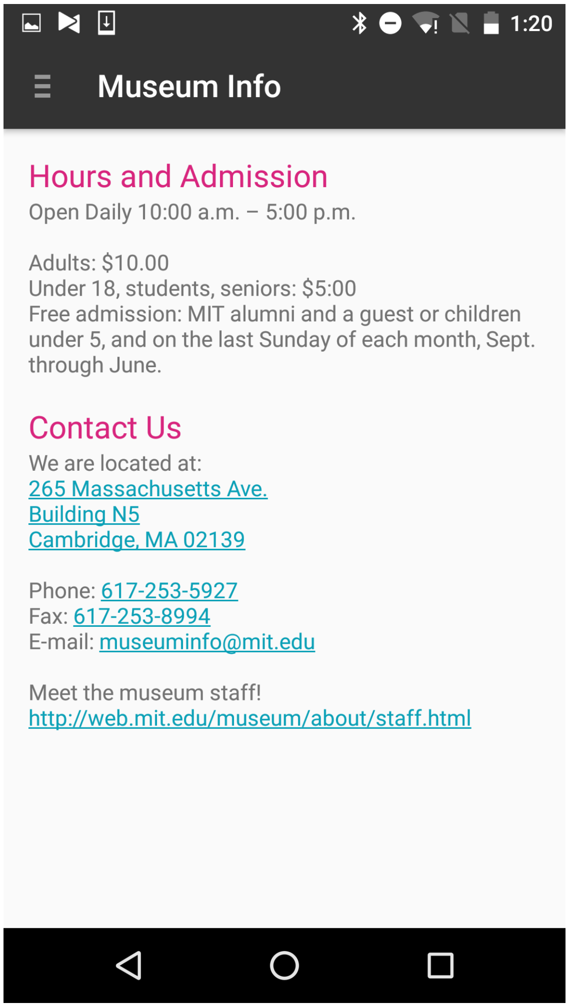
Nearby Exhibits

Undergraduate students involved in the project learned "how to solder, how to program a microcontroller, circuit design, carpentry, computer-aided fabricat

Figure 3-4: Clicking one of the entries of the blue list in the previous screenshot brings a visitor to a page containing the content that beacon is broadcasting. In this case, the content is more information about the MIT Laser Seal exhibit.

in section 3.4.4.





Museum Info

Hours and Admission

Open Daily 10:00 a.m. – 5:00 p.m.

Adults: $10.00
Under 18, students, seniors: $5:00
Free admission: MIT alumni and a guest or children under 5, and on the last Sunday of each month, Sept. through June.

Contact Us

We are located at:
265 Massachusetts Ave.
Building N5
Cambridge, MA 02139

Phone: 617-253-5927
Fax: 617-253-8994
E-mail: museuminfo@mit.edu

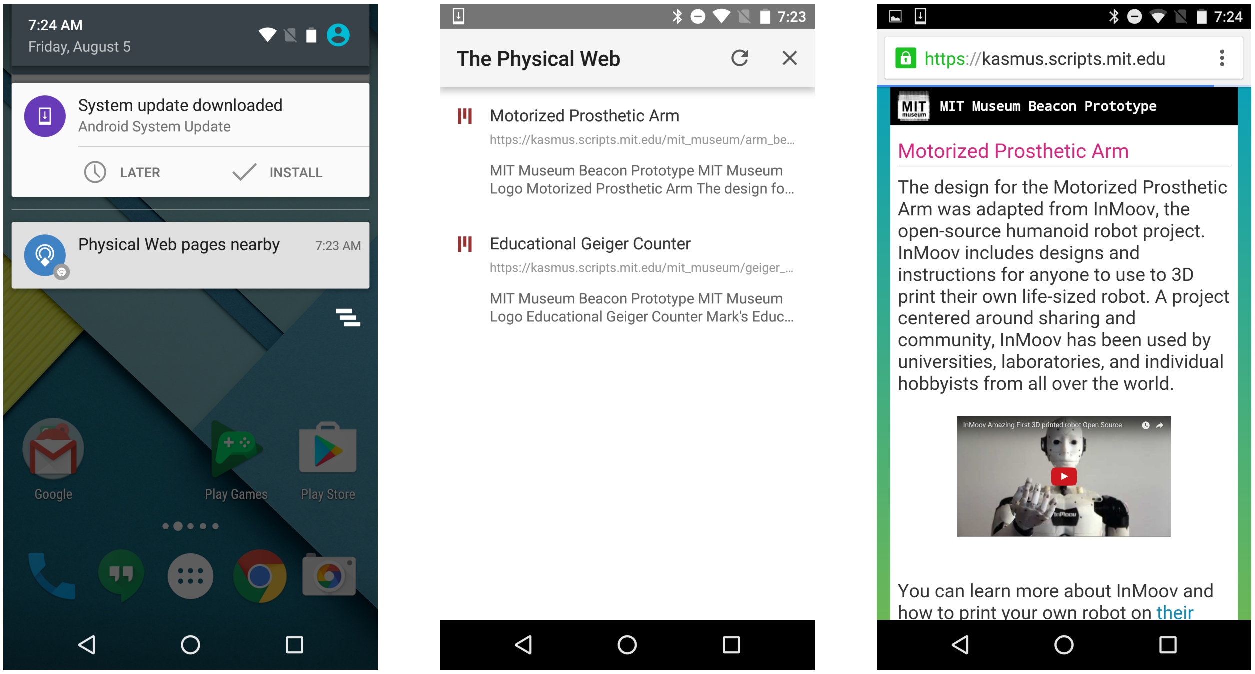
Meet the museum staff!
http://web.mit.edu/museum/about/staff.html

Figure 3-5: The left screenshot depicts the tab in the app that shows the visitor a map of the museum; the right shows them hours and admissions information.

3.2.2.2 Via the Physical Web

When beacons are broadcasting with the Eddystone-URL frame type, smartphones use the Physical Web to present content to visitors. Visitors must first ensure they have a mobile browser installed that supports the Physical Web project, such as Google Chrome, and that their Bluetooth setting is on. Then, on Androids, approaching a museum display with a beacon will prompt a notification informing the visitor of Physical Web pages nearby. Clicking this notification will bring up a list of all the pages being broadcasted nearby, and selecting one opens that page in the mobile browser.

On iPhones, Chrome's Physical Web interface takes the form of a widget in iOS's Today View. With Bluetooth on, a list of nearby Physical Web pages will appear within this widget when the visitor passes by displays with beacons. Clicking on any page in the list will open it in Chrome.



System update downloaded
Android System Update

Physical Web pages nearby

The Physical Web

Motorized Prosthetic Arm
https://kasmus.scripts.mit.edu/mit_museum/arm_be...
MIT Museum Beacon Prototype MIT Museum Logo Motorized Prosthetic Arm The design fo...

Educational Geiger Counter
https://kasmus.scripts.mit.edu/mit_museum/geiger_...
MIT Museum Beacon Prototype MIT Museum Logo Educational Geiger Counter Mark's Educ...

https://kasmus.scripts.mit.edu

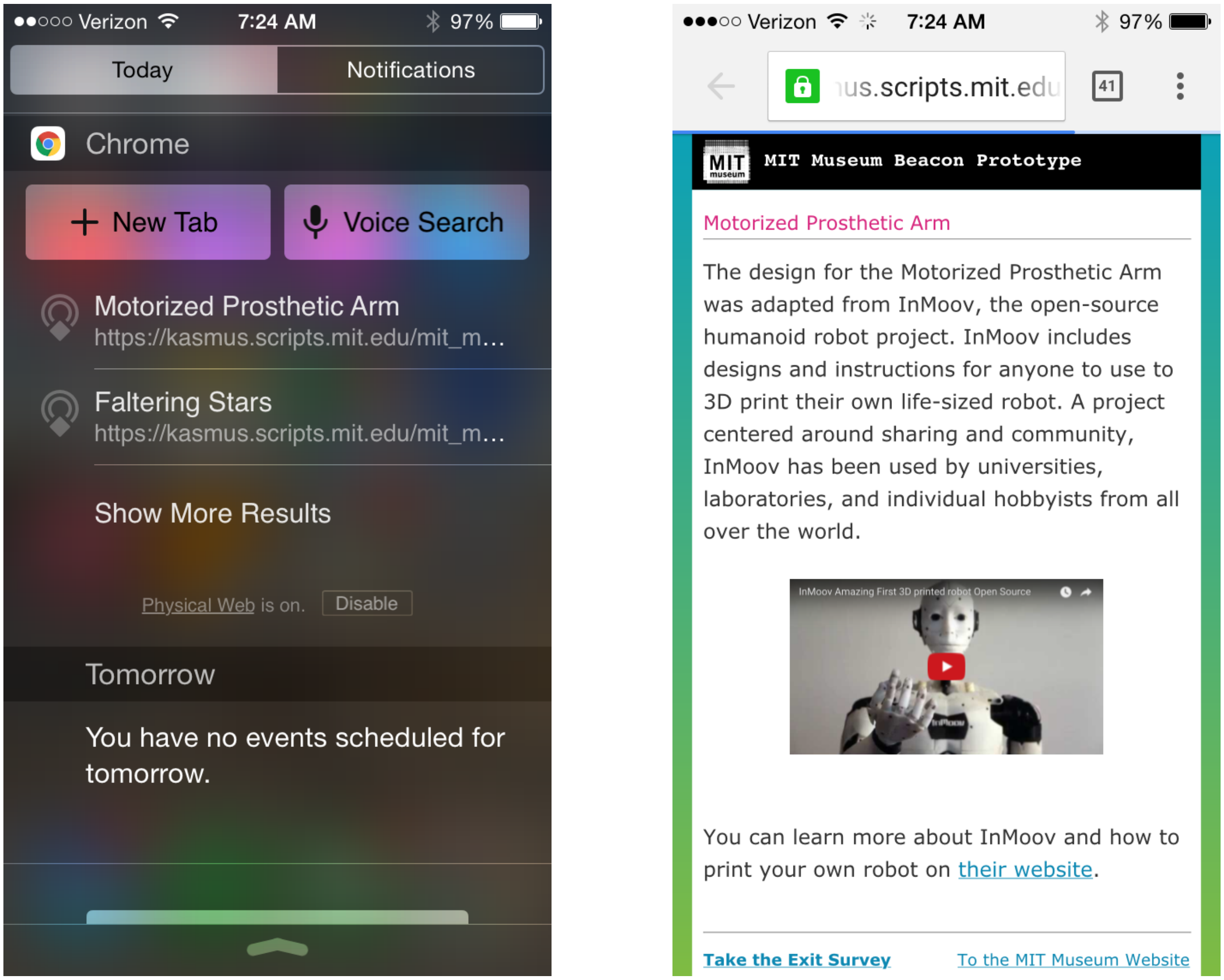
MIT Museum Beacon Prototype

Motorized Prosthetic Arm

The design for the Motorized Prosthetic Arm was adapted from InMoov, the open-source humanoid robot project. InMoov includes designs and instructions for anyone to use to 3D print their own life-sized robot. A project centered around sharing and community, InMoov has been used by universities, laboratories, and individual hobbyists from all over the world.

You can learn more about InMoov and how to print your own robot on their

Figure 3-6: In the left-most screenshot, a visitor on an Android device has received a Physical Web notification upon nearing a display with a beacon. Clicking this notification brings the visitor to the center screenshot, where they can view the complete list of Eddystone-URL beacons nearby. Choosing a beacon from the list opens that Physical Web page in the browser, as displayed by the third image.



Motorized Prosthetic Arm
https://kasmus.scripts.mit.edu/mit_m…

Faltering Stars
https://kasmus.scripts.mit.edu/mit_m…

Physical Web is on.

You have no events scheduled for tomorrow.

MIT Museum Beacon Prototype

Motorized Prosthetic Arm

The design for the Motorized Prosthetic Arm was adapted from InMoov, the open-source humanoid robot project. InMoov includes designs and instructions for anyone to use to 3D print their own life-sized robot. A project centered around sharing and community, InMoov has been used by universities, laboratories, and individual hobbyists from all over the world.

You can learn more about InMoov and how to print your own robot on their website.

Take the Exit Survey

To the MIT Museum Website

Figure 3-7: The screenshot on the left displays the Chrome widget in iOS's Today view listing several nearby Physical Web pages. Clicking on the "Motorized Prosthetic Arm" entry opens the associated URL in the Chrome browser to present the webpage about that display.

3.2.3 Report Walkthrough

Museum staff can view a live graphical display of visitors' interactions with the beacons online. Different web addresses correspond to visualizations of different metrics I queried from interaction data.

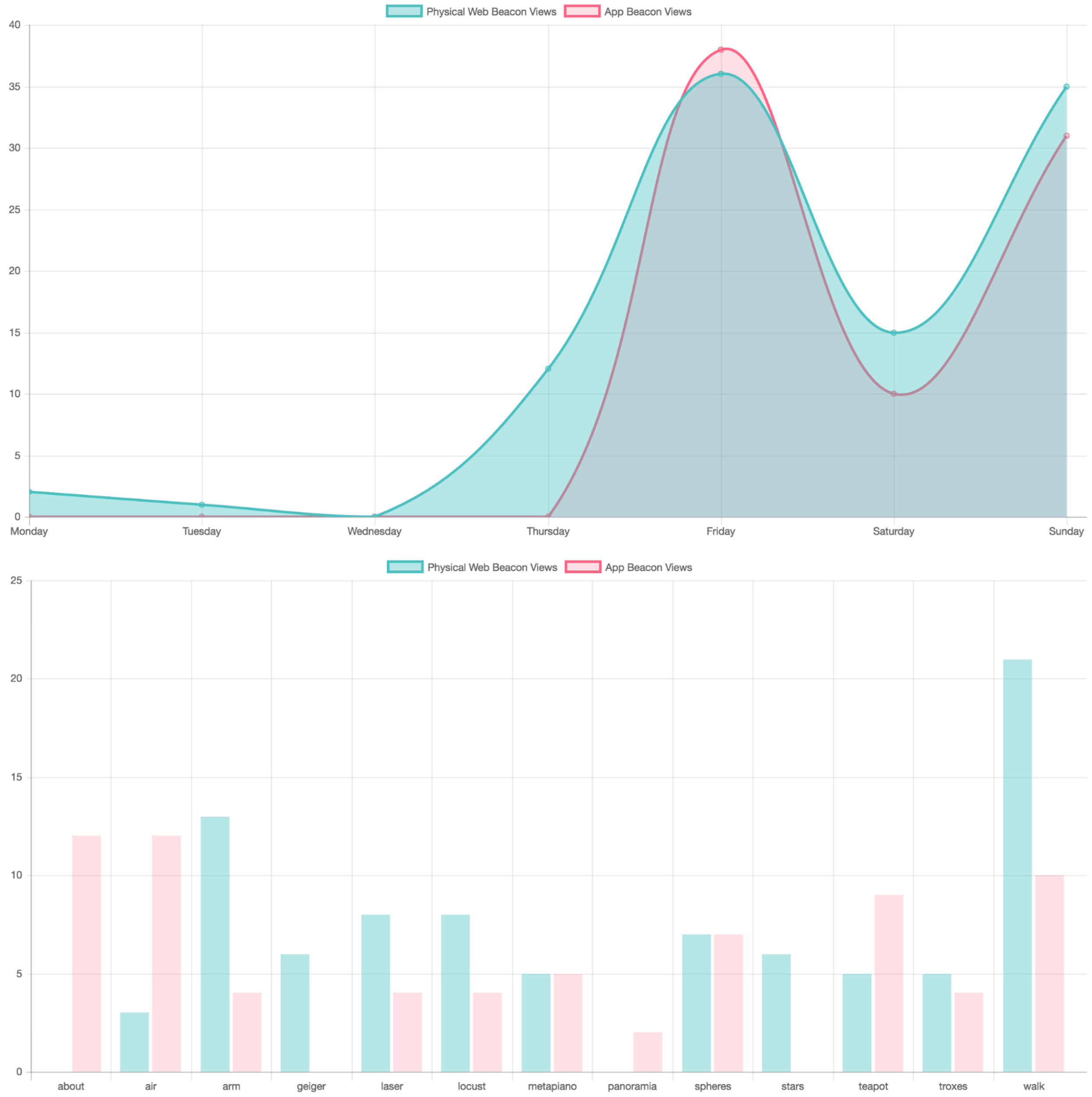


Figure 3-8: The charts above are examples of the data visualizations that can be viewed in a browser with the Report Component.

3.3 Configuration Component

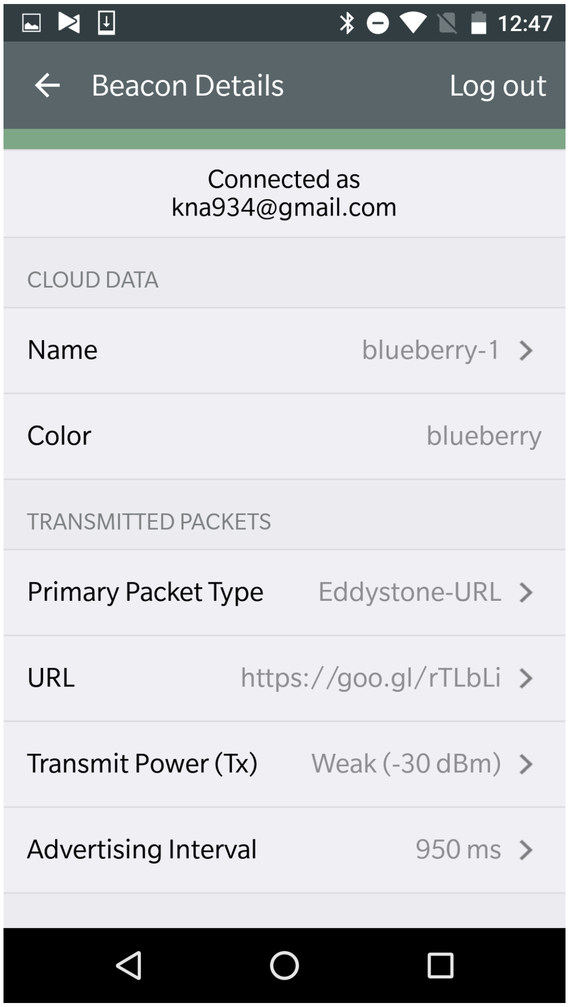
After purchasing a fleet of beacons, museums need to set them up to broadcast the desired protocol, to the correct area, with the relevant content attached. Initial configuration steps require software provided by the manufacturer of the beacon hardware. Then, Google's Proximity Beacon API can be used to register the beacons and attach content to them [24]. Google provides a sample Android app demonstrating how to set up a new beacon with this API [25]; I extended this app with an improved user experience and additional features to make it easier for museum staff to add and update the content their beacons broadcast. In this section, I will describe the configuration steps involving the manufacturer's software, then detail how my first app uses the Proximity Beacon API to facilitate management of beacon content.

3.3.1 Configure Beacon

Because the hardware implementations of beacons manufactured by different companies vary, initial configuration must be completed using software provided by the specific beacon manufacturer. For this thesis, I used Estimote Proximity Beacons [26]. I opted to purchase from Estimote due to their support for multiple protocols, affordable pricing, convenient developer tools, and well-supported, active developer forums. Therefore, I configured my beacons using the Estimote App for Android and the Estimote Cloud web application [27, 28]. However, software provided by many other beacon manufacturers advertises identical capabilities, so these configuration steps would remain consistent across most hardware [29, 30].

The most important configuration steps to perform via the manufacturer's software are setting the beacon's frame type, or communication protocol, and transmitting power (often abbreviated as Tx). This first implementation of the system is based on the Eddystone-UID frame type described in Section 2.2.1, so beacons must be set to utilize this protocol. Transmitting power determines how strongly the BLE signal from the beacon is sent, affecting the range where devices will be able to detect the beacon. Museums should modify this setting to reflect how close to a display's beacon a visitor should be in order to view that broadcasted content. Transmitting at larger distances consumes more power, so adjusting this setting will also affect the beacon's projected battery life. Once these two steps are completed, a beacon placed at a museum display will broadcast its

Eddystone-UID to devices within the specified range.



Connected as
kna934@gmail.com

CLOUD DATA

Name blueberry-1

Color blueberry

TRANSMITTED PACKETS

Primary Packet Type Eddystone-URL

URL https://goo.gl/rTLbLi

Transmit Power (Tx) Weak (-30 dBm)

Advertising Interval 950 ms

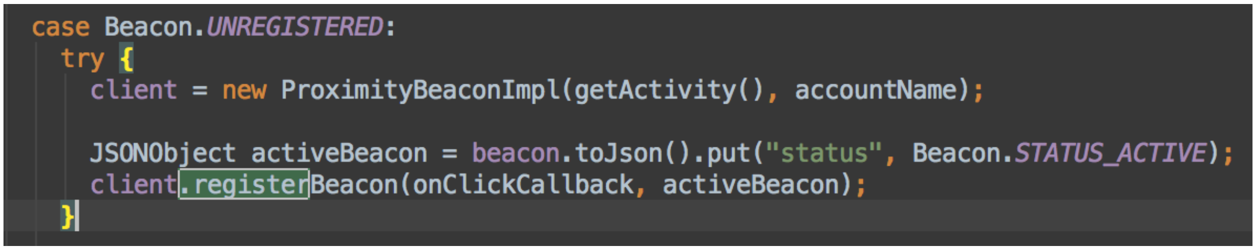
Figure 3-9: A screenshot of Estimote's configuration app displaying where to change the beacon's packet (frame) type and transmit power.

Optionally, most manufacturers' software provides additional settings that may be adjusted. For Estimote, these include metadata about the beacon - name, description, and location - that are helpful when managing a large fleet of beacons. Advertising interval, a setting that determines how frequently a beacon sends out a signal, can be increased to conserve beacon battery life or decreased to make beacon discovery faster. Other hardware specific settings, such as accelerometer options or power saving modes, are less relevant for this system.

3.3.2 Add Attachments with the Proximity Beacon API

The next steps involve Google's Proximity Beacon API to associate a beacon's Eddystone-UID with meaningful content [31]. First, each beacon must be registered to the Google Beacon Registry, which enables that beacon's UID to be associated with additional information - most importantly, the content that beacon should broadcast. It also assigns each beacon to a Google Developer Project to determine who is authorized to view the beacon's information. Next, key-value data pairs can be attached to beacons using a POST request to the API. These are the pieces of information that will be received by the app on visitors' smartphones to view information about an exhibit. The information format used by this system is discussed later in Section 3.4.3.

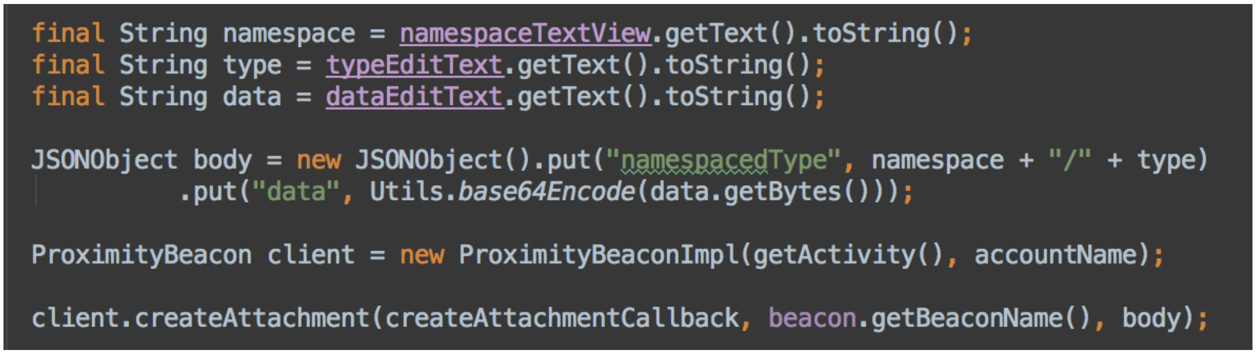
In order to accomplish these steps, I expanded on Google's code samples available on Github. The Beacon Platform sample included the necessary API calls to first register, then attach information to beacons. The app searches for nearby beacons broadcasting Eddystone-UIDs using an asynchronous Android BluetoothLeScanner object, and lists its findings. Any new beacons owned by the user will appear with a "Register" button, which makes a call to the Proximity Beacon API's register method to add this beacon to the Google Beacon Registry. The user's registered beacons can then be viewed to add or delete key-value pairs via the API's createAttachment and deleteAttachment methods. These calls provide the basic functionality necessary to provision a beacon with the desired contents.



```
case Beacon.UNREGISTERED:
  try {
    client = new ProximityBeaconImpl(getActivity(), accountName);

    JSONObject activeBeacon = beacon.toJson().put("status", Beacon.STATUS_ACTIVE);
    client.registerBeacon(onClickCallback, activeBeacon);
  }
```

Figure 3-10: The above code shows the Proximity Beacon API call for registering a currently unregistered beacon.

While this sample app was serviceable, it lacked the explanation and instruction to make the necessary steps obvious to a user without requiring additional research. My primary modifications to the app were therefore the addition of new screens and text displays walking the user through the first-time setup of their beacons to make the process more user-friendly. I simplified parts of the user interface, limiting the options to those museum staff setting up their own fleet of beacons

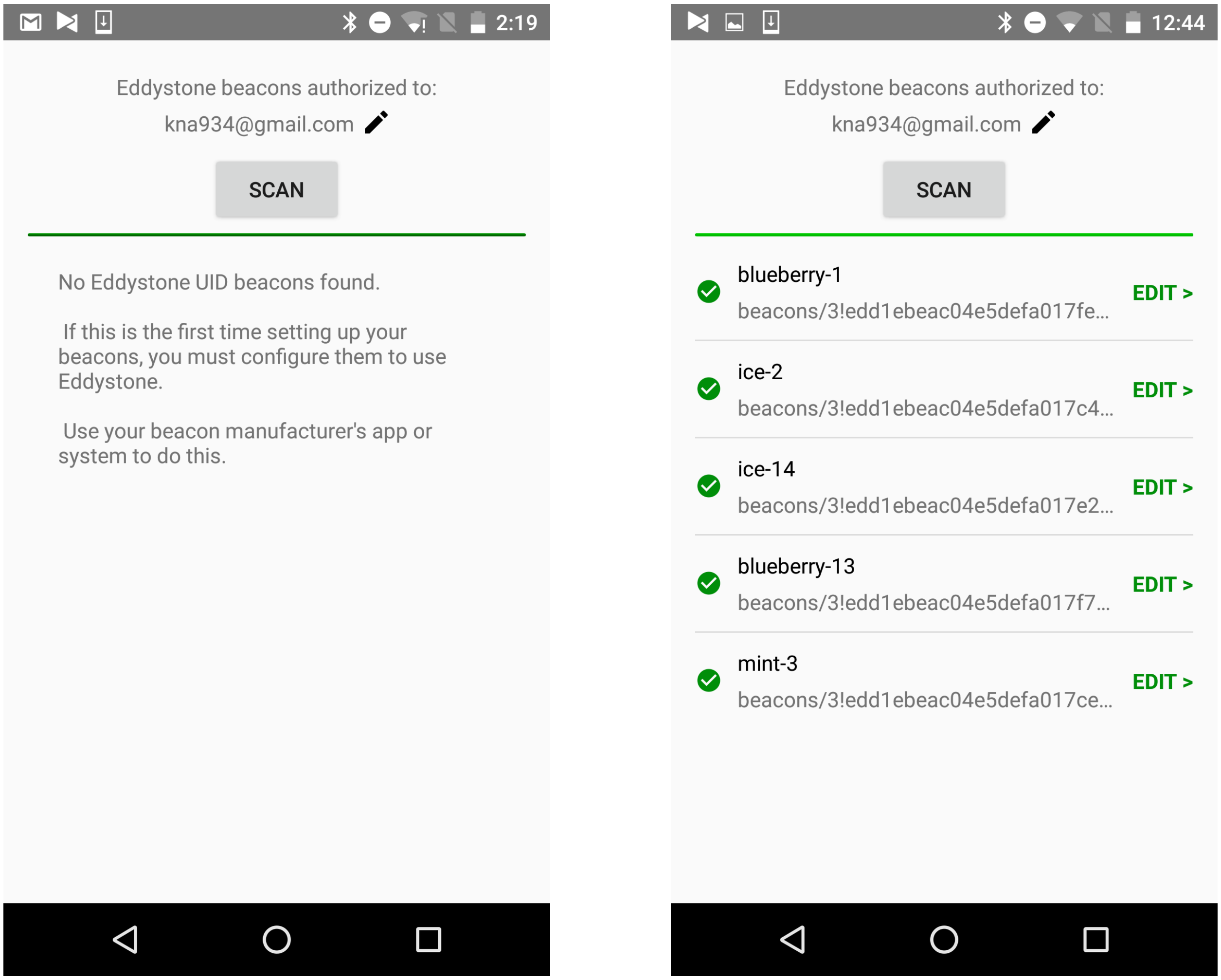


```
final String namespace = namespaceTextView.getText().toString();
final String type = typeEditText.getText().toString();
final String data = dataEditText.getText().toString();

JSONObject body = new JSONObject().put("namespacedType", namespace + "/" + type)
        .put("data", Utils.base64Encode(data.getBytes()));

ProximityBeacon client = new ProximityBeaconImpl(getActivity(), accountName);

client.createAttachment(createAttachmentCallback, beacon.getBeaconName(), body);
```

Figure 3-11: This code snippet demonstrates how to add a new attachment to a beacon. It records the beacon's namespace, followed by the type and data Strings that make up the key-value pair to be stored. It then makes a call to the Proximity Beacon API's createAttachment method.

would need to interact with. I had originally proposed integrating further features into this app to monitor the battery life and active status of beacons via the Proximity Beacon API; however, I found that this functionality was already present in the manufacturer software necessary for the previous steps, and would have been redundant if encountered in this app as well.

Since the time of development, Google has made publicly available a polished Beacon Tools app [32] which exposes much of the Proximity Beacon API's functionality via a convenient user interface very similarly to the app described in this section, and which could be used as an alternative moving forward.



Eddystone beacons authorized to:
kna934@gmail.com

No Eddystone UID beacons found.

If this is the first time setting up your beacons, you must configure them to use Eddystone.

Use your beacon manufacturer's app or system to do this.

Eddystone beacons authorized to:
kna934@gmail.com

✓	blueberry-1 beacons/3!edd1ebeac04e5defa017fe...
✓	ice-2 beacons/3!edd1ebeac04e5defa017c4...
✓	ice-14 beacons/3!edd1ebeac04e5defa017e2...
✓	blueberry-13 beacons/3!edd1ebeac04e5defa017f7...
✓	mint-3 beacons/3!edd1ebeac04e5defa017ce...

Figure 3-12: UI improvements included additional instructions and explanations as shown in the screenshot to the left, as well as a cleaner UI as demonstrated by the color coded and neatly organized list in the screenshot to the right.

3.4 Interaction Component

The primary function of the second application, the Interaction Component, is to display the content transmitted by a beacon to a visitor on their device, and enable some anonymous data to be collected in return. Beyond beacon functionality, the app contains some additional features to provide a more cohesive mobile museum experience. I interviewed members of the MIT museum staff in order to determine which additional features would be most valuable. In this section, I will discuss the Android app built for this purpose, including how the app communicates with beacons and design decisions made during development. The next section will detail how this app gathers data, the organization of that data, and its uses.

3.4.1 Nearby Messages vs Proximity Beacon API

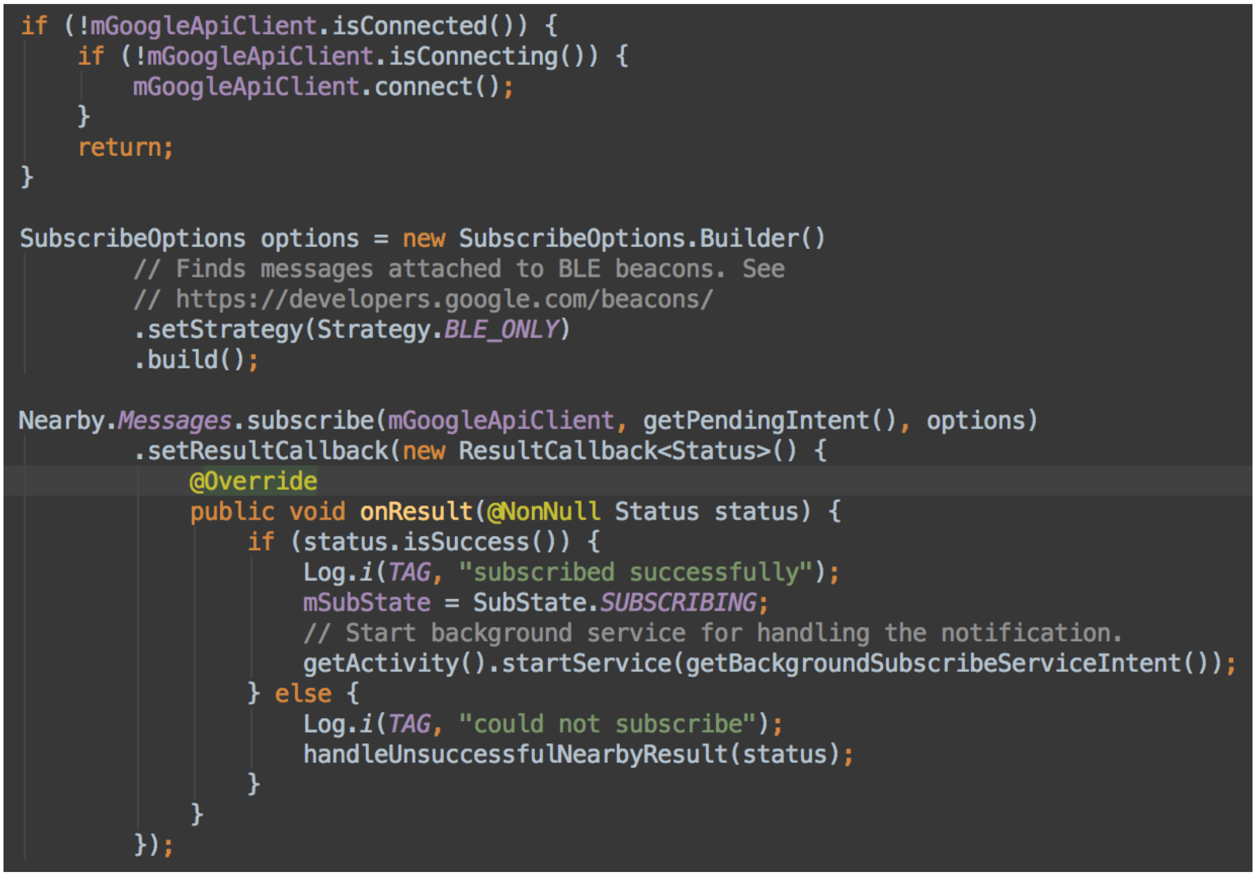
The first major design decision I faced when building this app was how to receive incoming BLE signals. This can be done in two distinct ways via either Google's Nearby Messages API or the Proximity Beacon API used in the Configuration Component. The Proximity Beacon API provides a beacons.attachments.list method that made for a simple first-draft implementation of the app where a visitor could press a button to fetch data from any nearby broadcasting beacons [33]. However, this mode of interaction requires a visitor to press the button to make that API call every time they wish to view a new beacon's content. This is less than ideal both in its repetitive nature and due to the fact that users may not always know when they are near a beacon and should request to view its information.

Fortunately, the Nearby Messages API provides alternatives that allow the app to do more of the work for the visitor. It is a publish-subscribe API, which allows the app to be constantly listening for new beacon messages without the user having to submit a request each time they'd like to check [34]. This can be done either in the foreground, exclusively when the app is actively being used, or in the background, which sends notifications alerting users to nearby beacons even when the app is closed. This presents a significant tradeoff: foreground scanning is lower latency and more reliable, but uses more battery power and requires the user to be actively engaged with the app at all times in order to work; background scanning can be less reliable, but requires less power and commitment from the user. I chose to use background scanning so that users would be notified of new beacons and therefore be less likely to miss out on content. I revisit this decision in my discussion of testing results in Section 5.2.

3.4.2 Using the Nearby Messages API

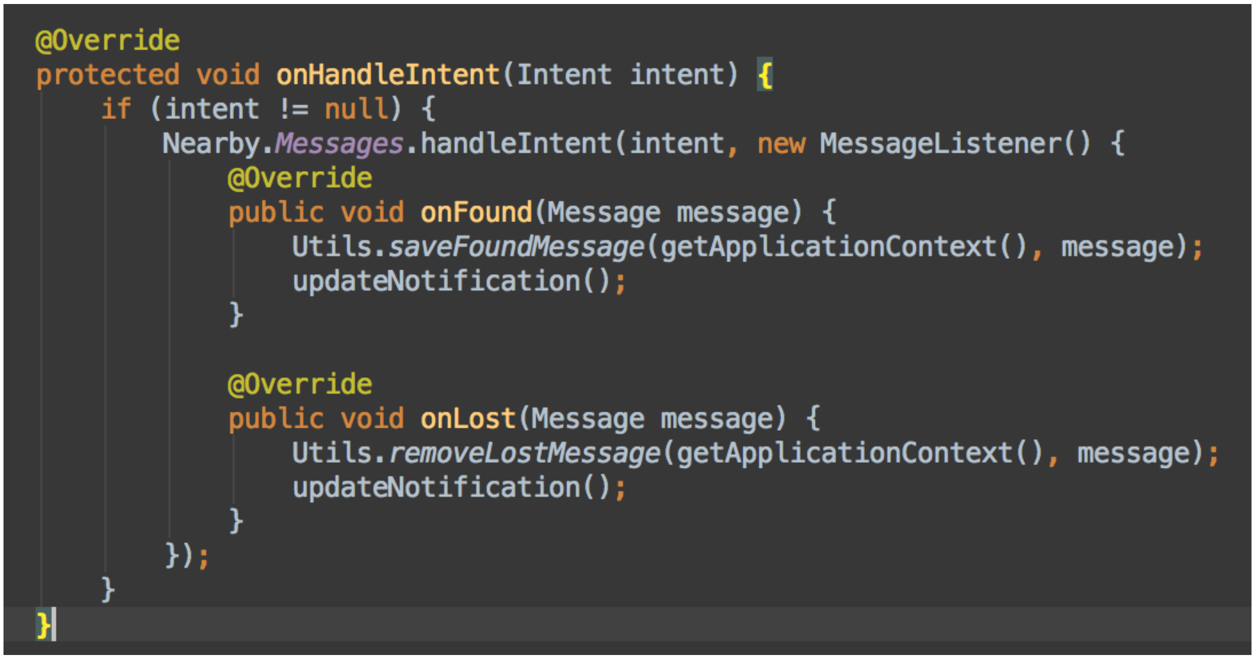
Development using the Nearby Messages API requires several preconditions [35]. First, the app must be set up to use Google Play Services as a gradle dependency. An API key must then be added to the app's Manifest file to authorize its use of the API. This API key must be generated from the same Google Developers Project as the one used to register beacons with the Configuration Component in order for the app to be authorized to view those beacons. Next, as subscribing to nearby messages can be resource intensive, Google advises requesting explicit user permission for the app to begin scanning. I included their code snippet for requesting this permission the first time the app connects to the Nearby Messages API in the app.

To actually subscribe to receive BLE messages in the background, the Nearby.Messages.subscribe method is called. Within the app, I incorporated code from Google's sample Nearby Background Beacons project to achieve this [36]. The app is then able to listen for nearby beacons and add their contents to Android's SharedPreferences storage when found - whether the app is currently active or not. If the app is not active, a notification appears informing the user of how many beacons are broadcasting messages nearby, with a preview of those messages. If it is active, it displays a list of previews, which updates automatically anytime the Nearby Messages subscription adds or removes a beacon entry from the SharedPreferences storage.



```
if (!mGoogleApiClient.isConnected()) {
    if (!mGoogleApiClient.isConnecting()) {
        mGoogleApiClient.connect();
    }
    return;
}

SubscribeOptions options = new SubscribeOptions.Builder()
        // Finds messages attached to BLE beacons. See
        // https://developers.google.com/beacons/
        .setStrategy(Strategy.BLE_ONLY)
        .build();

Nearby.Messages.subscribe(mGoogleApiClient, getPendingIntent(), options)
        .setResultCallback(new ResultCallback<Status>() {
            @Override
            public void onResult(@NonNull Status status) {
                if (status.isSuccess()) {
                    Log.i(TAG, "subscribed successfully");
                    mSubState = SubState.SUBSCRIBING;
                    // Start background service for handling the notification.
                    getActivity().startService(getBackgroundSubscribeServiceIntent());
                } else {
                    Log.i(TAG, "could not subscribe");
                    handleUnsuccessfulNearbyResult(status);
                }
            }
        });
```

Figure 3-13: Within the main activity fragment, clicking the button to begin discovering beacons calls the following code. It connects a GoogleApiClient object, creates SubscribeOptions so that this listener will only search for BLE broadcasts, and then subscribes via a call to the Nearby Messages API. If the subscription is successful, the app's BackgroundSubscribeServiceIntent is started to listen for communications.



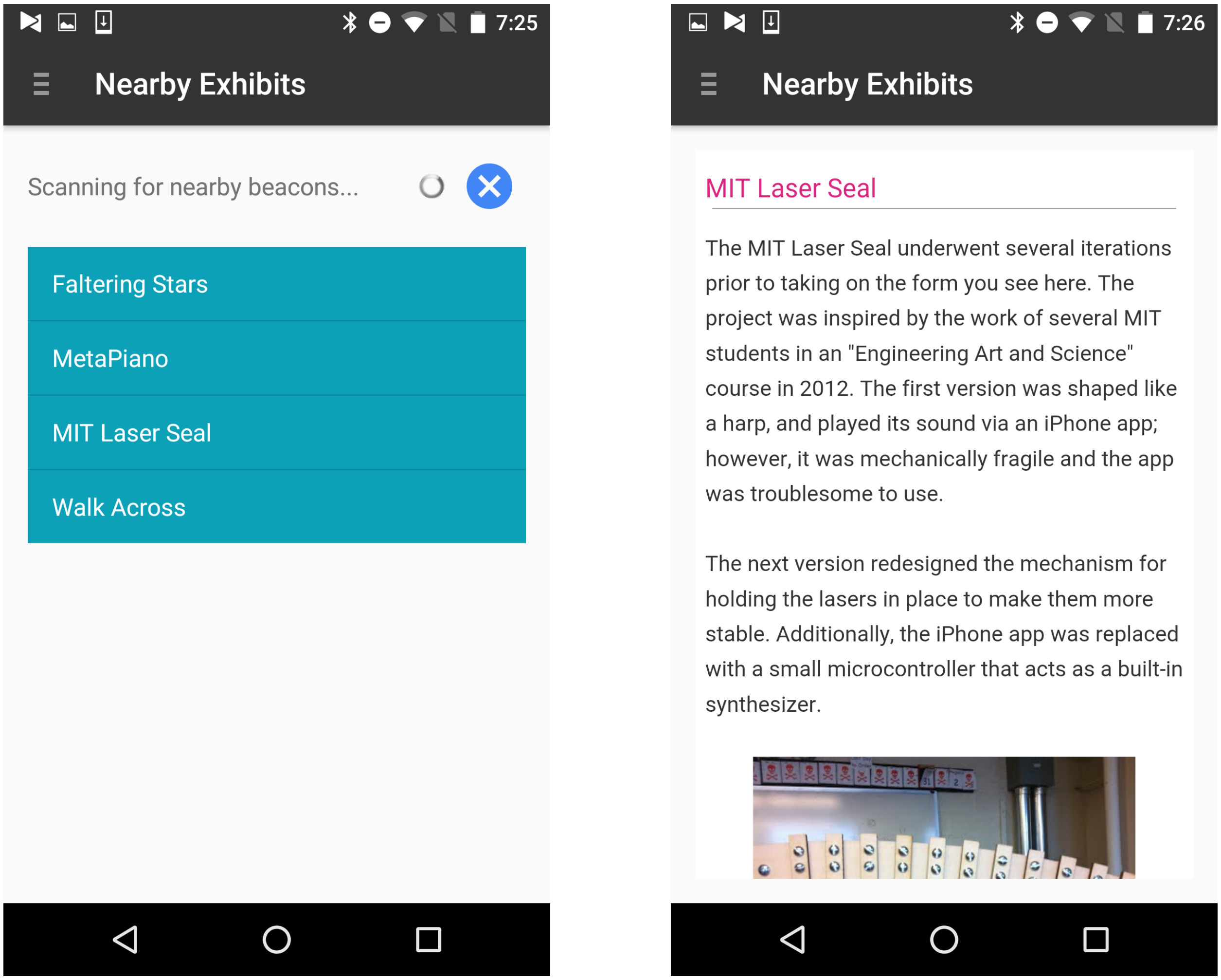
```java
@Override
protected void onHandleIntent(Intent intent) {
    if (intent != null) {
        Nearby.Messages.handleIntent(intent, new MessageListener() {
            @Override
            public void onFound(Message message) {
                Utils.saveFoundMessage(getApplicationContext(), message);
                updateNotification();
            }

            @Override
            public void onLost(Message message) {
                Utils.removeLostMessage(getApplicationContext(), message);
                updateNotification();
            }
        });
    }
}
```

Figure 3-14: From the BackgroundSubscribeIntentService, a listener is created to handle incoming Nearby Messages. When found, a utility function saves them to the application's Shared Preferences storage, which updates the list of nearby beacons in the app. It also updates the notification sent to the user to reflect the current nearby beacons. When a beacon message is lost, the opposite actions are taken.

3.4.3 Displaying Beacon Content

There are many options for how to organize and display the key-value pairs of data attached to a beacon; I considered two for the purposes of the Interaction Component. The first would have involved a more complicated data format, but possibly more simplicity in creating content for museum exhibits. It would have consisted of keys indicating the type of Android view to use for this element and the numerically ordered position where it should appear among the other elements on the page, and URL values pointing to the image, text, or other resource to populate the view. This would allow museum staff easily create beacon content out of a collection of different resources without requiring much overly technical knowledge - just how to use the Configuration Component to add these attachments and the ability to host the resources on the web. However, it also involves lots of manual entry of attachments and complicates the design of the app significantly - it must be programmed to handle each of the different types of views the museum wishes to incorporate. Instead, I opted for a streamlined approach which only attaches the name of the beacon's exhibit and a URL for a website, developed separately, which contains all of the desired content resources.

Because this only involves two pieces of data, it can be simplified to one attachment where the name is the key and the URL is the value. The name is then used where the app displays previews of a beacon's content - in notifications and the list of nearby beacons. Upon clicking a beacon in the list, users are taken to a new screen which displays the beacon's content at the associated URL within an Android WebView. This format, while requiring more technical knowledge to create content, allows greater flexibility, as any content included in a website can be automatically displayed.



MIT Laser Seal

The MIT Laser Seal underwent several iterations prior to taking on the form you see here. The project was inspired by the work of several MIT students in an "Engineering Art and Science" course in 2012. The first version was shaped like a harp, and played its sound via an iPhone app; however, it was mechanically fragile and the app was troublesome to use.

The next version redesigned the mechanism for holding the lasers in place to make them more stable. Additionally, the iPhone app was replaced with a small microcontroller that acts as a built-in synthesizer.

Figure 3-15: Screenshots demonstrating using the app to discover nearby beacons and then view their content.

3.4.4 Additional Features

I consulted with staff of the MIT Museum to evaluate several other features which could be added to the app to enhance beacon interactions or the museum experience at large. Here is a table

Proposed Feature	**Included?**	**Why Not?**
Text, Pictures, Videos, and Links to additional information on display	Y	
Interactive content relating to display (games, quizzes, etc.)	N	Concerns about time required to create content
Favoriting or Rating system for display	N	Time Constraints
Visitor Comments on display	N	Concerns about moderation of visitor comments outweighed benefits
"Not Working" button to inform museum staff when a display is down	N	Might give negative impression that exhibits are often faulty
Share display on social media	N	MIT Museum's displays change frequently; don't want to advertise
Map of Museum	Y	
Hours/admissions/contact info	Y	
List of Exhibits	N	Low-Priority
Calendar/events page	N	Time Constraints
Survey About Beacons	Y	
Link to Info about Beacon Project	Y	

Table 3.1: Additional features considered for inclusion in the Interaction Component app.

of proposed features indicating which were included and, for those that weren't, the reasons why. Some of these ideas were not relevant for the MIT Museum's use case but could be for other museums. Others were suitable but low-priority and ran into time constraints during development. These features were not implemented for the current version of the system, but are mentioned as an area of future work in Section 6.3.

3.5 Database Layer

The database layer connects anonymous statistics about visitors' usage of the Interaction Component with the Report Component so museum staff can learn from aggregate data and trends. This section will explain what data is gathered, how it stored in Google's Firebase real-time cloud database, how it is obtained from the Interaction Component, and additional metrics considered but not implemented in the current system. The next section will explain how this data is visualized to provide useful information to museum staff.

3.5.1 Firebase

Firebase [37] is a cloud database tool owned by Google. It uses a NoSQL representation to store data as simple JSON, a convenient format for writing and parsing. In addition to maintaining a well-supported developer community, Firebase is designed to integrate very easily with different client platforms - a potential advantage given that data for this system could involve both Android and web. Its real-time capabilities also facilitated making the Report component a live dashboard of events as opposed to static activity graphs that require manual refreshing. I also had previous personal experience developing with Firebase on Android and web and knew its functionality would be well-suited to the system at hand.

Firebase provides a free plan for developers, called "Spark," offering up to 1 GB of data storage and 100 simultaneous connections - more than enough for small-scale testing of the museum system [38].

3.5.2 Designing the Schema

Although Firebase is a NoSQL database, I chose to organize the system's data according to relational database paradigms to favor space efficiency over time efficiency, as Firebase's free plan provides a limited amount of storage. The data is organized into two tables, one representing users and the other representing logs of interactions that users made with the system.

The users table is represented as a list of JSON objects where each object correlates with a distinct visitor. Each visitor is assigned a unique ID. The table also stores information about the device the visitor is using: the brand, model, and OS of the phone; the browser name and version; and whether the browser has cookies enabled (discussed further in the next section).

The logs table records every trackable action users make within the Interaction Component. This currently includes viewing beacon content, clicking on videos to play them or links to discover additional information, and exiting the beacon page, but could be expanded to include additional actions. Each action is stored as a JSON object in a list, where that object includes fields for the ID of the visitor who made the action, the name of the beacon with which the user interacted, the type of action (view, clickVideo, clickLink, or exit), the click target (if applicable), and the timestamp when the action was taken. Click targets are strings describing a video or link, so that

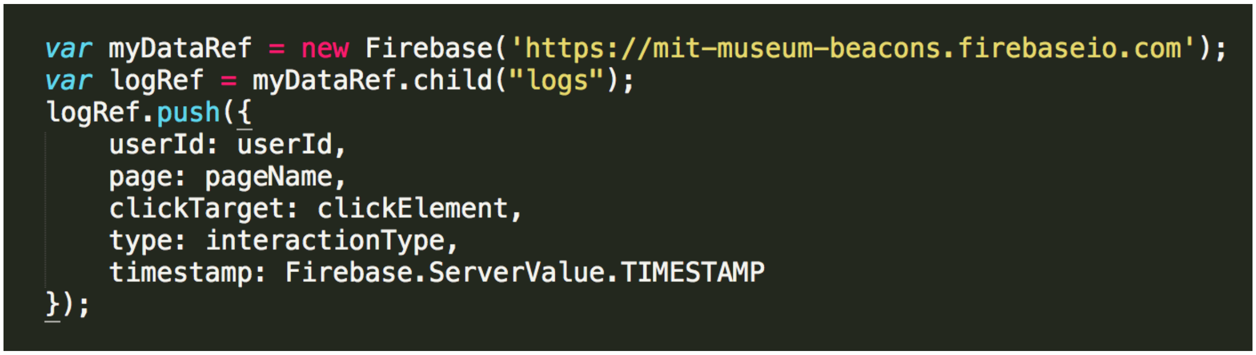
USERS	LOGS
userID	*userID*
device	page
browser	type
cookiesEnabled	timestamp
	clickTarget

Table 3.2: The above table represents the format in which interaction data is stored. Note that every log contains a userID which matches up with the userID of a single record in the users table.

these can be distinguished since a beacon's content can contain more than one.

3.5.3 Storing Data

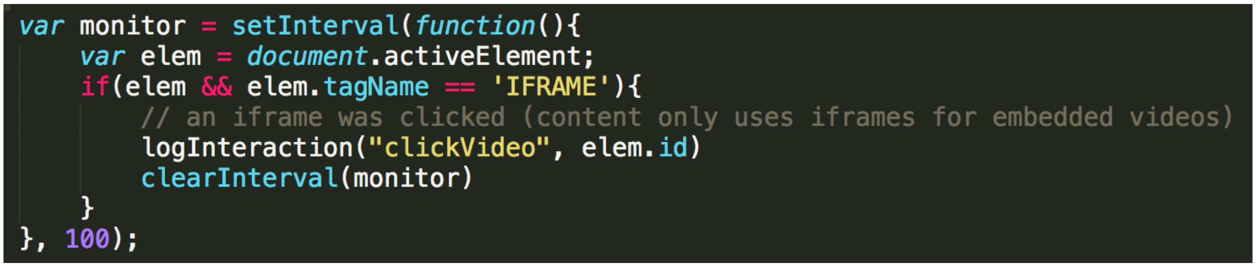
Whenever a user interacts with beacon content within the Interaction Component, the Firebase library is used to send a log of the interaction to be stored in the database [39]. Because beacon content was developed as websites, I made these Firebase push calls in Javascript that runs on the sites. A "view" action is logged immediately after the beacon content loads. Links to external sites all contain click listeners that log "clickLink" actions via Firebase. An HTML "onbeforeunload" listener on the page triggers an "exit" log.



```
var myDataRef = new Firebase('https://mit-museum-beacons.firebaseio.com');
var logRef = myDataRef.child("logs");
logRef.push({
    userId: userId,
    page: pageName,
    clickTarget: clickElement,
    type: interactionType,
    timestamp: Firebase.ServerValue.TIMESTAMP
});
```

Figure 3-16: The above code creates a reference to where the Firebase data is stored, then pushes a new record to the "logs" list. This new record is a JSON object containing a userId, page, clickTarget, type, and timestamp.

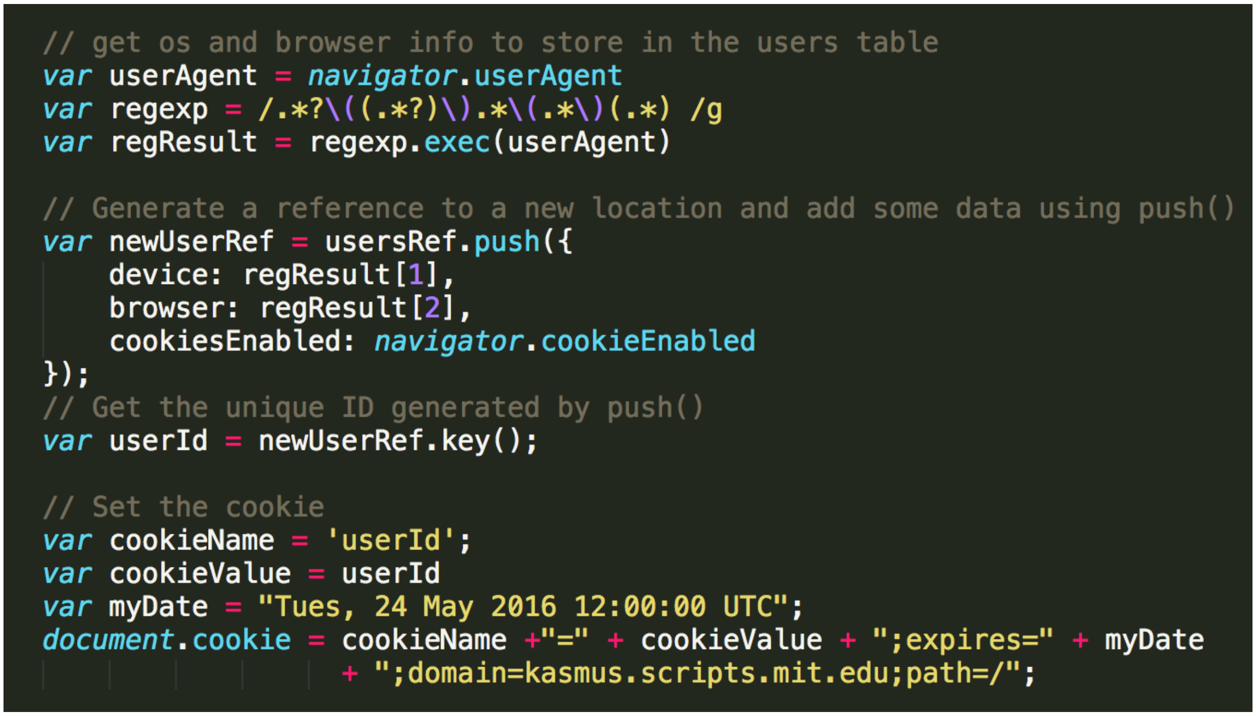
Because videos are typically embedded on the webpages as iframes, logging "clickVideo" actions was tricky. Since the video content was from a different domain than my scripts, the Same Origin security policy [40] prevents my scripts from interacting with that content to receive click events. To work around this, I discovered a script that periodically checks whether an iframe is the active element on a page, and logs a "clickVideo" event if and when it is [41]. In order to prevent additional logs from being created the entire time the video is the active element, I unsubscribe the checks after the first log. This means I wouldn't log a second event if a user watched a video twice in one sitting, but that use case seemed unlikely enough to dismiss.



```
var monitor = setInterval(function(){
    var elem = document.activeElement;
    if(elem && elem.tagName == 'IFRAME'){
        // an iframe was clicked (content only uses iframes for embedded videos)
        logInteraction("clickVideo", elem.id)
        clearInterval(monitor)
    }
}, 100);
```

Figure 3-17: The above code sets an anonymous function to be called every 100 milliseconds. The function checks the active element, and if it is an iframe (which would contain an embedded video), logs a clickVideo interaction on that element. It then cancels any further checks.

Creating an entry in the database for each unique user was also more involved. Because different beacons' content is viewed on different webpages, I needed a way to keep track of individual users as they went from one site to the next. Therefore, I set a domain level Javascript cookie the first time a new user (without a cookie) visited one of the beacon websites. I create an entry for the new user via Firebase, which passes back a new unique ID associated with that entry. This ID is stored in the cookie, which is then able to be looked up across any of the beacon websites in my domain, kasmus.scripts.mit.edu/, that the user visits later. When actions are logged, the ID is simply retrieved from the cookie and attached to the JSON object to be pushed to Firebase to identify which unique user made the action. Since some users may disable cookies in their browser, I kept track of this setting in the users table as well, so that any data analysis on distinct users could discount those whom I was unable to track.



```
// get os and browser info to store in the users table
var userAgent = navigator.userAgent
var regexp = /.*?\((.*?)\).*\(.*\)(.*) /g
var regResult = regexp.exec(userAgent)

// Generate a reference to a new location and add some data using push()
var newUserRef = usersRef.push({
    device: regResult[1],
    browser: regResult[2],
    cookiesEnabled: navigator.cookieEnabled
});
// Get the unique ID generated by push()
var userId = newUserRef.key();

// Set the cookie
var cookieName = 'userId';
var cookieValue = userId
var myDate = "Tues, 24 May 2016 12:00:00 UTC";
document.cookie = cookieName +"=" + cookieValue + ";expires=" + myDate
                + ";domain=kasmus.scripts.mit.edu;path=/";
```

Figure 3-18: The above code handles setting a new userID cookie, if one is not already found for this user. The first three lines compute information about how the visitor is accessing the content. Next, a new users record is pushed to the Firebase, since we have no cookie indicating this user has visited before. The unique ID created by Firebase when the new record is added is then fetched, and a new cookie is assigned to the project domain to store that userID so this visitor can be identified the next time they view a beacon content page.

3.5.4 Additional Metrics

One additional metric I considered tracking and storing directly is the elapsed time a user spent viewing the content for a given beacon. While an approximation of this value can be determined by taking the difference of the timestamps for “view” and “exit” logs by the same user at the same beacon, it would be resource intensive to perform this calculation across all logs in the database. Additionally, there turned out to be sources of error in that approximation which are not easily resolved, including when the app is in the background, the phone screen off, or the page left open when the user is done viewing it. For these reasons, this data was not stored directly in this implementation.

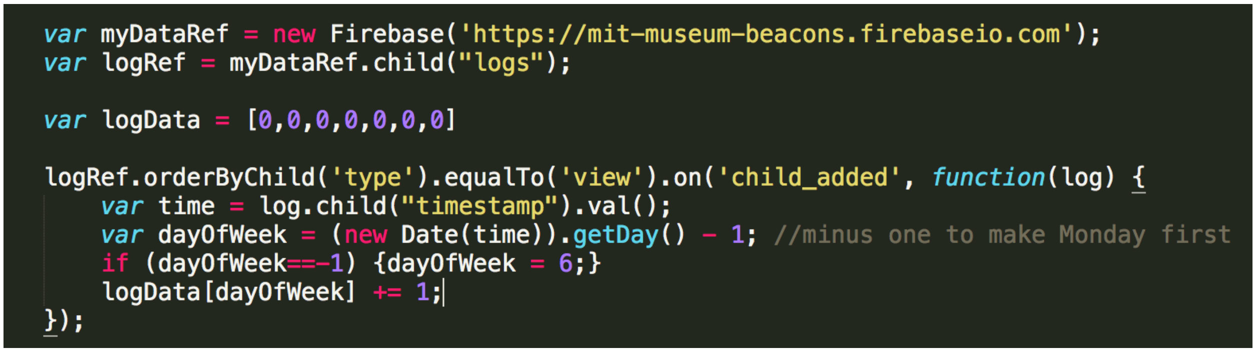
Another metric that didn’t make it into the database is records of when a visitor passed into and out of the range of each beacon. While this information was of interest to the MIT Museum, it was lower priority and faced privacy concerns. Though visitors would have to explicitly grant permission for the Interaction Component to scan for beacons in the background, tracking user location based on events occurring when they aren't engaging with the app is a non-obvious extension of that consent and could be unwanted by visitors. In fact, the Physical Web (an alternate means of connecting with beacons explored in Section 3.7) explicitly prevents developers from receiving any information at all from users until they have actively opted to interact with a beacon for this exact reason [42]. Therefore, we decided to exclude this tracking from the system.

3.6 Report Component

The third component of the system provides a web portal that enables museum staff to track various live visualizations of the data collected by the Interaction Component. These reports can help staff answer questions like how often beacons are being viewed, which are the most popular, or what type of content visitors engage with most. This information could then be used to help the museum better understand its audience and improve its future offerings. Because the visualizations automatically update as new data is logged, they can also be used as dashboard to monitor traffic in real time. In this chapter I will describe the development of the Report Component web app and demonstrate how the information in the database can be visualized to answer some example queries.

3.6.1 Retrieving Data

The data used to create the graphs and charts displayed by the Report Component come from the Firebase database described in the previous chapter. I used Firebase's Javascript library to retrieve the relevant data for each visualization. The onChildAdded event listener is triggered when a new log or user is added to one of the top-level database lists, and passes along the data added [43]. I then gathered the relevant information from this new object and applied it to the data structures used to create the graphs, and called for the charts to be redrawn. I used Firebase's orderByChild and equalTo methods to filter the events, so that only relevant updates to the database are considered. This pattern of listening for new data and redrawing graphs when new information is logged keeps the visualizations up to date, displaying a live representation of current interactions with beacons.

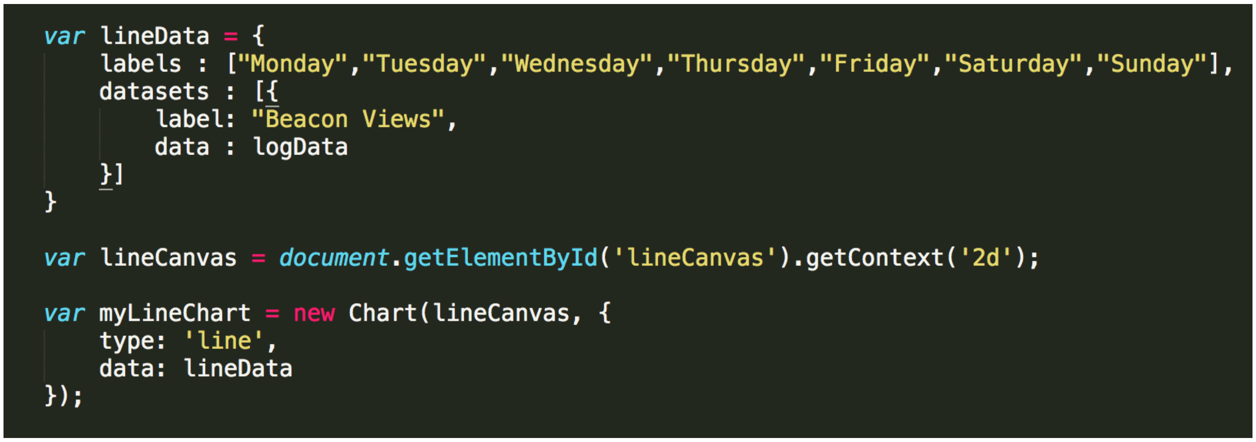


```
var myDataRef = new Firebase('https://mit-museum-beacons.firebaseio.com');
var logRef = myDataRef.child("logs");

var logData = [0,0,0,0,0,0,0]

logRef.orderByChild('type').equalTo('view').on('child_added', function(log) {
    var time = log.child("timestamp").val();
    var dayOfWeek = (new Date(time)).getDay() - 1; //minus one to make Monday first
    if (dayOfWeek==-1) {dayOfWeek = 6;}
    logData[dayOfWeek] += 1;
});
```

Figure 3-19: The above code demonstrates fetching data from the Firebase and manipulating it into a form that can be graphed. Here, when a log is added whose type is equal to "view," the timestamp attribute of that log is noted. Then, the day of the week in which that log took place is figured from the timestamp, and a list of counters representing the number of logs in each day is incremented accordingly.

3.6.2 Chart.js Library

I used the Chart.js Javascript library [44] to transform the retrieved data into neat, interactive bar, pie, and line graphs. The type of graph used was determined manually for each of the queries displayed. Chart.js graphs are represented by JSON objects containing lists of labels and the corresponding data, in addition to other style options. To create the visualizations, I applied the information pulled from Firebase to build the graph's data list. For example, a histogram showing beacon views by days of the week requires gathering all the "view" action logs from Firebase, transforming the associated timestamps into days of the week, and tallying up the total number of views for each day to build the data list that will create the bar chart.



```
var lineData = {
    labels : ["Monday","Tuesday","Wednesday","Thursday","Friday","Saturday","Sunday"],
    datasets : [{
        label: "Beacon Views",
        data : logData
    }]
}

var lineCanvas = document.getElementById('lineCanvas').getContext('2d');

var myLineChart = new Chart(lineCanvas, {
    type: 'line',
    data: lineData
});
```

Figure 3-20: Here, the code shows how to draw a line chart with Chart.js. A line data object is created with a list of labels for the horizontal axis and a dataset object to plot with a label and the list of points (the variable logData, which was built in the previous code snippet). A canvas from the webpage HTML is selected, and the Chart constructor is used to draw a new chart there with these given parameters.

3.6.3 Sample Queries

In addition to the example just mentioned, I will explain three more visualizations created with the Report Component to demonstrate how the gathered data can be transformed into useful metrics for a museum. First, I created a pie graph that breaks down the types of devices that visitors are using to interact with beacons. Pulling this data from Firebase is fairly straightforward; upon a new user being recorded, that user's device type is noted. I build up a map in Javascript associating each device type with the number of visitors using that type of device. So if a visitor's device type is already included in the map, I add one to the tally; if not, I make a new map entry with a tally of one. Then, I pass the keys of that map to the Chart.js pie graph constructor as labels, and the values of the map as data, to complete the visualization. This information can then be used to estimate, for example, what percentage of visitors have a phone new enough to take advantage of some new feature the museum could incorporate into an exhibit.

Another visualization displays the total number of clicks on each video or link included in beacon content. To gather this data, I listen for new logs being added where the action type is equal to "clickLink" or "clickVideo." Then, I identify the click target of that action and add one to its tally in a data list. Using that data list and a corresponding list of the click target names, Chart.js builds a bar graph to display the relative popularity of these elements. Museums could

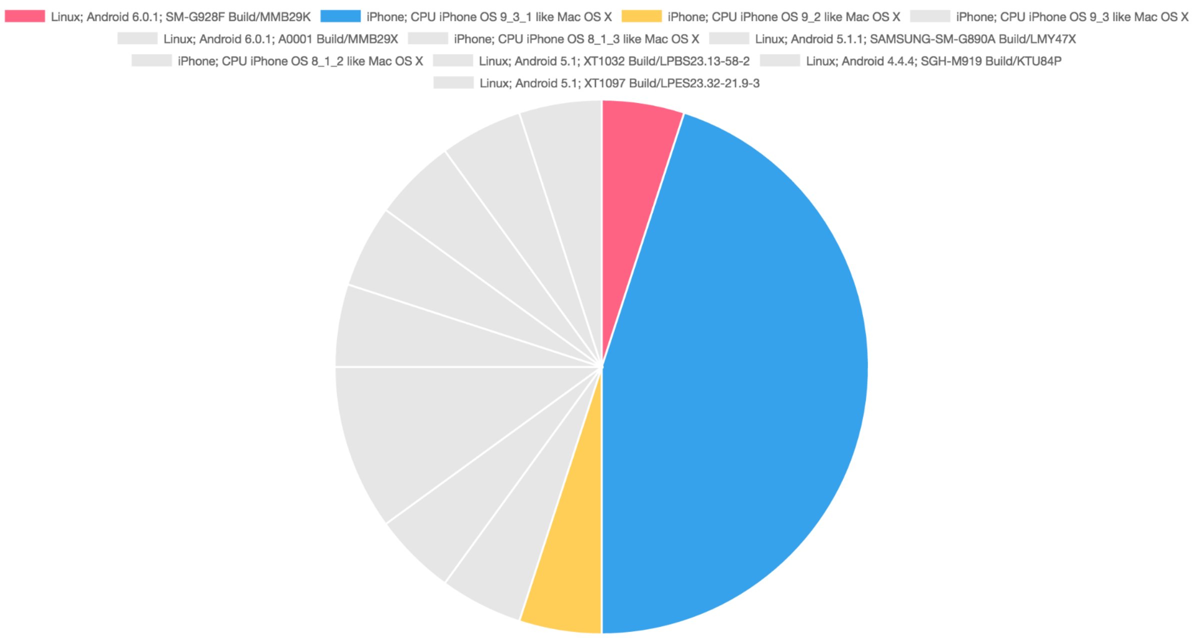


Figure 3-21: The resulting pie graph representing what kinds of devices are being used to interact with beacon content.

use this data to evaluate which types of content are most engaging to users, and incorporate these findings when creating new content.

A final example plots a bar graph indicating the number of interactions per unique visitor. I store a map associating user IDs with a tally of how many interactions they've made, and update that map accordingly every time a new action log is recorded. I then pass those keys and values to the Chart.js bar graph constructor. This visualization hints at how well users enjoy the beacon feature. A low average number of interactions per visitor (relative to the total number of unique interactions one could make with the beacon content available) would likely indicate that visitors got bored with the feature and opted not to continue using it to explore more of the museum. A high average would indicate that visitors enjoyed the feature well enough to continue using it at length.

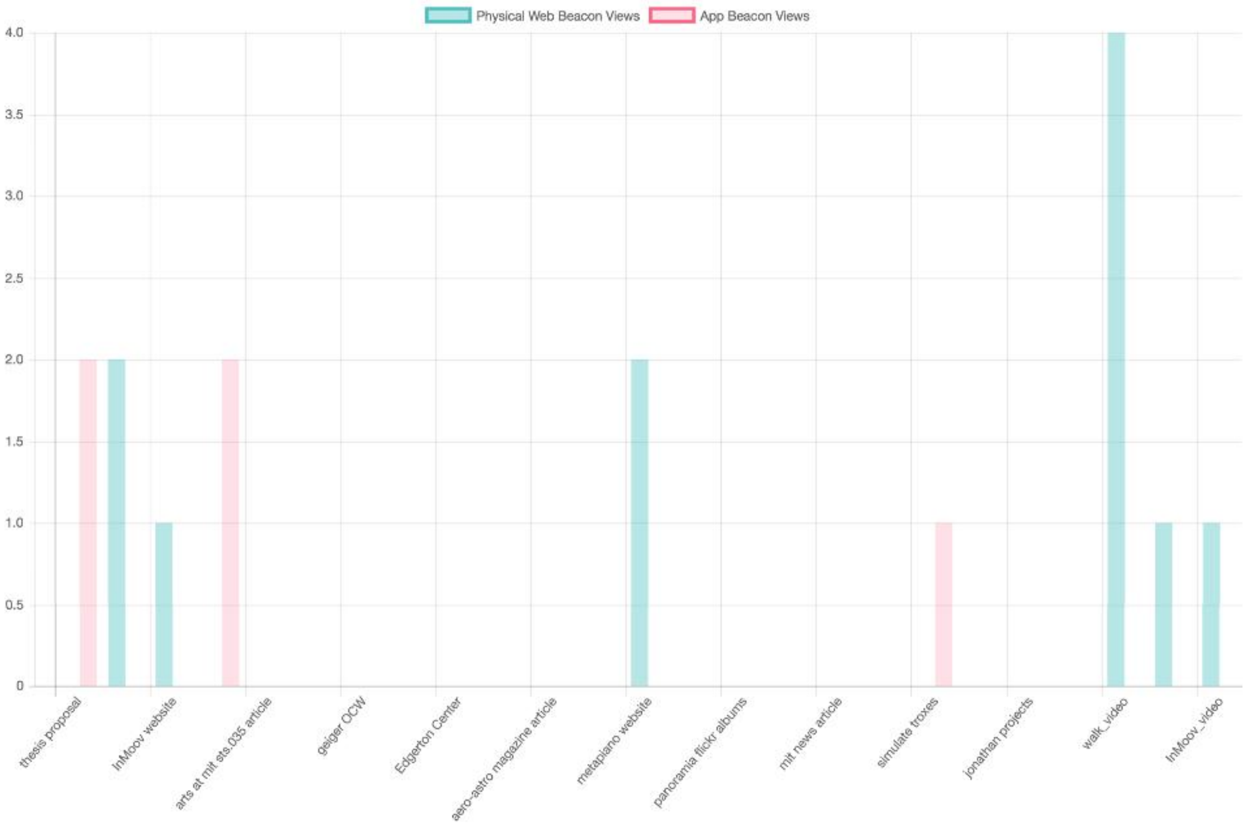


Figure 3-22: The resulting bar graph showing the number of clicks made on each link or video within beacon content.

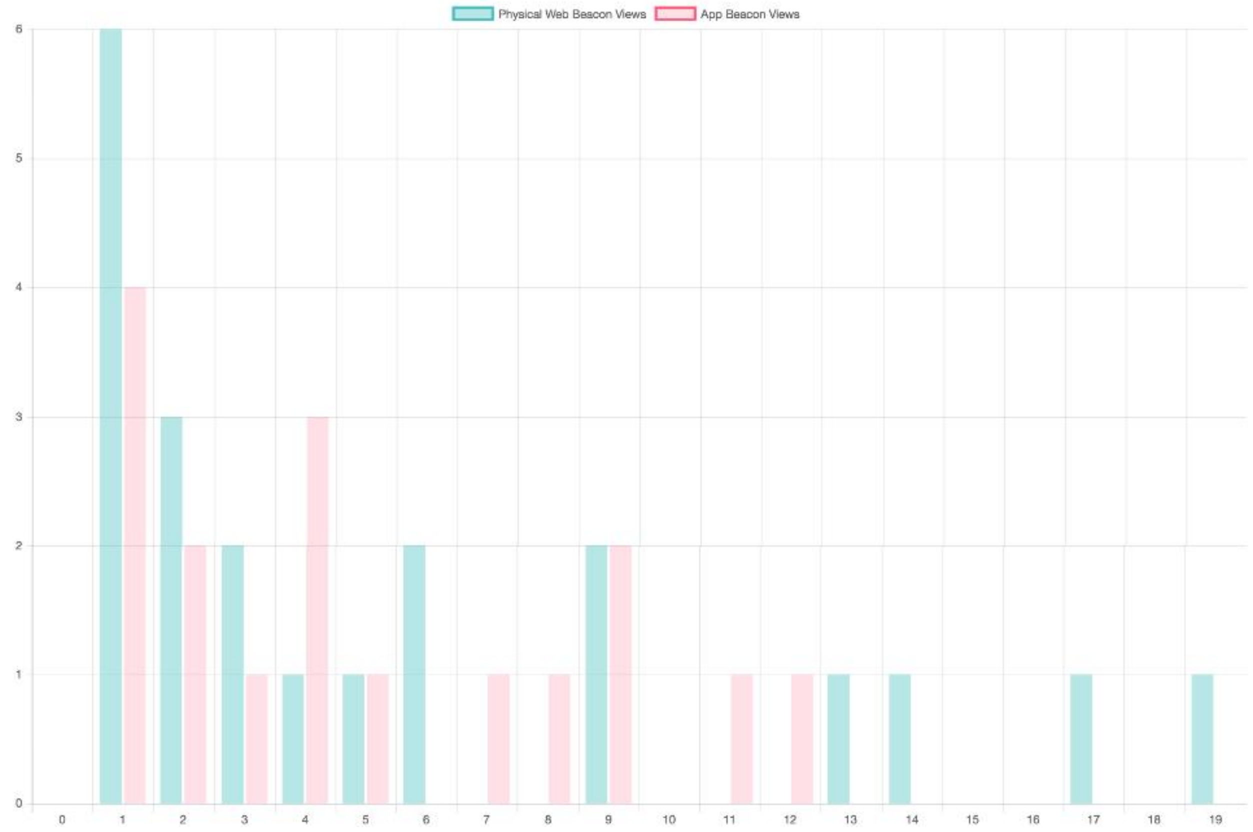


Figure 3-23: The resulting histogram depicting the number of interactions each unique user made with beacon content.

3.7 Alternate Version: Eddystone-URL and Physical Web

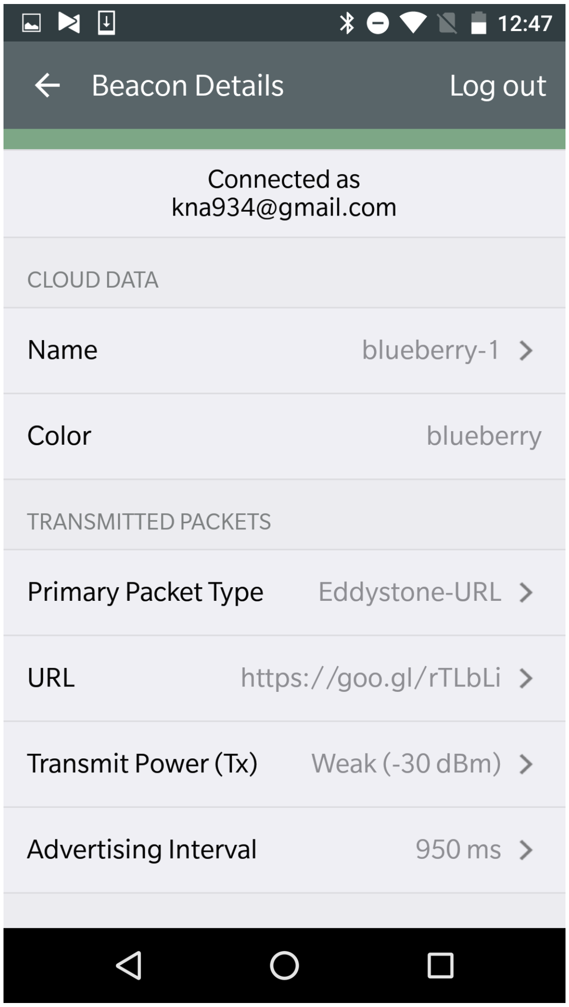
Originally, this system was intended to make use solely of Eddystone's UID protocol to communicate from beacons directly to a dedicated Android app. At the time, the Physical Web was less widely supported, so the URL protocol would have reached a very limited fraction of visitors. However, Google's release of version 49 of Chrome for Android in the spring opened new doors by enabling Physical Web support on the default browser of most relatively new Android devices (those running Android 4.3+ and supporting BLE) [45]. With this advancement, I expanded the goals of my thesis to include a comparison of the two Eddystone protocols. The results of this comparison are discussed in Section 5.2.

This section will revisit the system established in the previous sections, and explain how I reworked it to incorporate the Eddystone-URL frame type in place of Eddystone-UIDs. The biggest change is the replacement of a dedicated Android app for the Interaction Component with websites viewed in a Physical Web supported mobile browser.

3.7.1 Configuration Component with Eddystone-URL

Configuring a beacon to broadcast a single URL in place of completely arbitrary amount and types of attachments is unsurprisingly much simpler. First, within the manufacturer's software, the broadcasted frame type must be set to Eddystone-URL as opposed to UID. The desired URL is then also set within the manufacturer's tool, eliminating the need for a second app interfacing with Google's Beacon Registry and Proximity Beacon API altogether, as there are no attachments to manage.

The URL broadcasted must adhere to a few rules. The length is limited to 17 bytes to reduce the size of the advertisement packet, so the use of URL shorteners is widely recommended [46]. This also enables developers to change the website associated with the beacon by redirecting the shortened URL instead of having to return to this step to update the URL the beacon is configured to broadcast. Additionally, the website must use HTTPS in order to be viewed via the Physical Web in Chrome (one of the most popular supporting browsers).



12:47

Beacon Details

Log out

Connected as
kna934@gmail.com

CLOUD DATA

Name	blueberry-1
Color	blueberry

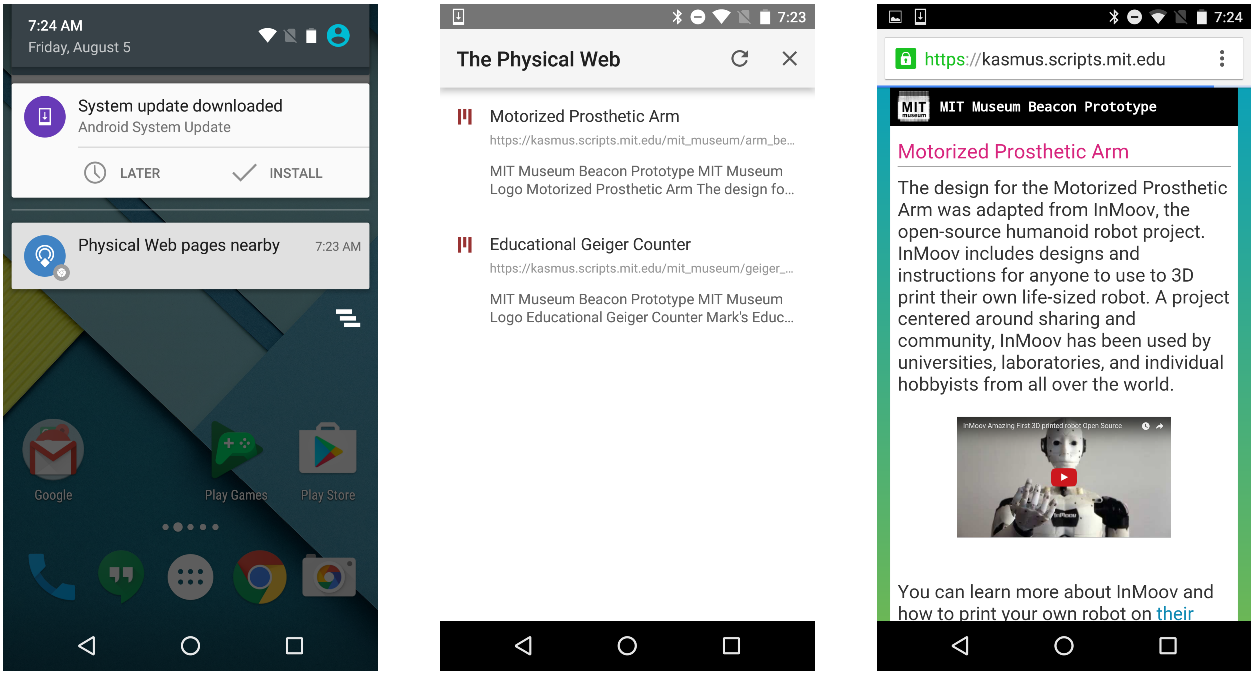
TRANSMITTED PACKETS

Primary Packet Type	Eddystone-URL
URL	https://goo.gl/rTLbLi
Transmit Power (Tx)	Weak (-30 dBm)
Advertising Interval	950 ms

Figure 3-24: This screenshot of the Estimote app shows where a user can adjust the packet type and URL associated with a beacon.

3.7.2 Interaction Component with Physical Web

The Interaction Component entirely changed with the switch to Eddystone-URLs. Instead of downloading a museum app to view beacon content, visitors now only have to turn on the settings necessary to enable the Physical Web on their devices with supported browsers. Rather than appearing within the app, new beacons now appear as silent notifications - in the Today view of iPhones and the notifications drawer on Android devices. Because the Physical Web is now supported by Chrome for iOS and Android, the content is immediately available for either platform; with the Eddystone-UID approach, separate apps would need to be developed for each operating system. Upon clicking on a Physical Web notification, users are immediately taken to the broadcasted URL in their browser.



Motorized Prosthetic Arm
https://kasmus.scripts.mit.edu/mit_museum/arm_be...
MIT Museum Beacon Prototype MIT Museum Logo Motorized Prosthetic Arm The design fo...

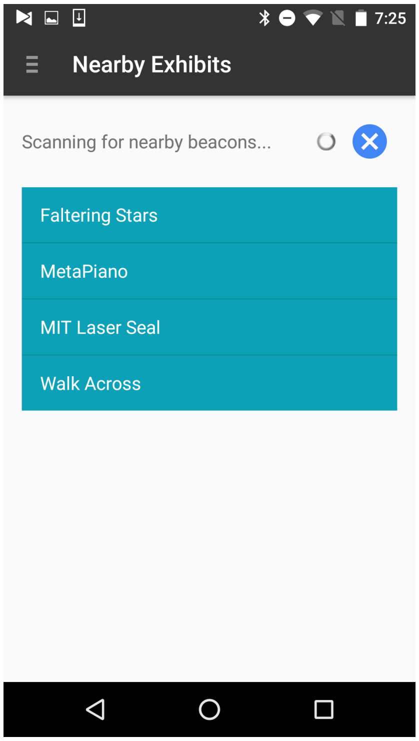
Educational Geiger Counter
https://kasmus.scripts.mit.edu/mit_museum/geiger_...
MIT Museum Beacon Prototype MIT Museum Logo Educational Geiger Counter Mark's Educ...

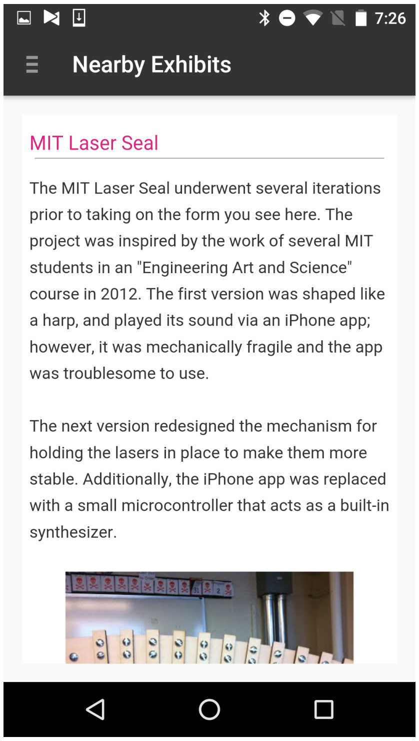
Motorized Prosthetic Arm

The design for the Motorized Prosthetic Arm was adapted from InMoov, the open-source humanoid robot project. InMoov includes designs and instructions for anyone to use to 3D print their own life-sized robot. A project centered around sharing and community, InMoov has been used by universities, laboratories, and individual hobbyists from all over the world.

You can learn more about InMoov and how to print your own robot on their

Figure 3-25: These screenshots show the different user flow on Android when using the Physical Web as opposed to a dedicated app - from being notified of a nearby Physical Web page to opening and viewing that content in a browser. Compare these with the Eddystone-UID and dedicated app flow from the first version displayed below.





MIT Laser Seal

The MIT Laser Seal underwent several iterations prior to taking on the form you see here. The project was inspired by the work of several MIT students in an "Engineering Art and Science" course in 2012. The first version was shaped like a harp, and played its sound via an iPhone app; however, it was mechanically fragile and the app was troublesome to use.

The next version redesigned the mechanism for holding the lasers in place to make them more stable. Additionally, the iPhone app was replaced with a small microcontroller that acts as a built-in synthesizer.

Though the user experience from one protocol to the other differs, it didn't require much additional development to adopt the Eddystone-URL frame type because I had already opted to work with websites to view content within the dedicated Eddystone-UID app. Therefore, with a few adjustments to ensure the URLs met the requirements mentioned above, I was able to present the same content websites via the Physical Web as were displayed in the Android WebViews of the original app. I hosted the websites with MIT's SIPB Web Script Service as it provided automatic HTTPS access [47]. I also made a few stylistic changes to make them appear more cohesive outside of the app, and added links to the museum website and survey in a footer to maintain some of the continuity provided by additional features within the app version.

3.7.3 Database Layer and Report Component Changes

No major changes to the database layer or report component would have been necessary to make the system work with Eddystone-URLs instead of UIDs. However, as part of the new goal was to compare the two protocols, I decided to separate out data gathered by the URL and UID implementations into separate tables in the database layer. This enabled the Report Component to pull data from the two systems independently. I then modified the Report Component to draw two overlaid, semi-transparent graphs to represent the metrics for each of the two systems.

While these modifications would be unnecessary in a finalized version of either system, it enabled me to draw comparisons when testing the system in the experiments described in the next section.

Chapter 4

MIT Museum Experiments

This chapter will describe the real-world testing of the system performed in collaboration with the MIT Museum. I discuss the preparations for running this experiment, then break down the actual testing into four phases. For each phase, I will describe the procedures used to test the components, my observations during these tests, and any additional data collected via surveys or the system itself.

4.1 Setup and Procedure

In order to test my system in a real-world environment, I collaborated with staff at the MIT Museum. The primary purpose of the experiment was to observe visitors using the Interaction Component to evaluate the level of interest, usability, and enjoyment of interacting with beacon content via smartphones - both via Eddystone-UID/dedicated Android app and Eddystone-URL/Physical Web. It also provided an opportunity to assess the Configuration and Report Components in small-scale tests with Museum staff.

The proposed experiment required evaluation by COUHES because it involves testing with human participants. As the testing posed minimal risk to subjects and would record all data anonymously, COUHES ruled it exempt from requiring a full review and granted approval.

4.1.1 Meetings with Museum Staff

I met with MIT Museum staff several times leading up to the experiment, both to consult on the design of the system and to ensure all aspects of the experiment would be up to the Museum's standards for visitor experience.

Our initial meeting on April 1st focused around planning and scheduling the experiment. When developing the Interaction Component, I met again with a small group of Museum staff to demonstrate a prototype and discuss useful features to be included (April 15th). On April 25th, a week before the beacons went live in the museum for visitors to use, I ran a demo of the Interaction Component on site to the full staff to receive feedback for last minute improvements.

4.1.2 About the Experiments

The experiment at the MIT Museum consisted of four phases, each of which involved testing of a component of the system by real-world users, making observations, and gathering anonymous responses. The first phase consisted of a small, informal meeting with Museum staff on April 14 to explain the beacon setup process, have them try out the Configuration Component, and gather feedback about the process. The next two phases, from May 2 through 15, each involved a week of observing visitors using the Interaction Component, getting their responses via an exit survey, and collecting anonymous statistics on usage with the database layer. The first week tested the Eddystone-URL protocol where visitors interact via the Physical Web; the second used the dedicated Android app communicating with Eddystone-UIDs. In the final phase, on May 19, I demonstrated the Report Component to a group of Museum staff to present the aggregate data gathered and survey their thoughts on the usefulness of that information. Altogether, the responses collected during these phases provides insight about the effectiveness of the overall system.

4.2 Phase One: Beacon Configuration

The first phase of testing consisted of having Museum staff try out the Configuration Component and provide feedback on the usability of the tools involved.

4.2.1 Description of Tasks

Two MIT Museum staff members were each given two beacons to configure, one with the Eddystone-UID frame type and attachments, and one with the Eddystone-URL frame type and a web address. I explained verbally the difference between the manufacturer's software - in this case, the Eddystone Android app - and the Proximity Beacon API app used to add attachments. I then provided a list of tasks for the volunteers to complete in order to configure each beacon, including setting the frame type, attachments or URL, and transmitting power. This list described the goals of the setup, such as, "Set the beacon to transmit to a maximum of 12 feet," - not direct instructions on how to perform these tasks within the app. I asked the staff members to speak aloud their thought processes, ask questions if anything was unclear, and provide verbal feedback for my notes. These results are discussed in the next section.

4.2.2 Responses

The volunteers found the Eddystone-URL beacons, which they set up first, "pretty straightforward" to configure, and appreciated that the entire setup could be completed from a single app. They raised some questions about transferring ownership of beacons and optional configuration fields, but had no trouble adjusting the settings necessary to configure the Eddystone-URL beacon.

The next beacon was to be configured via the Eddystone-UID protocol. The staff members found this setup process to be the more difficult of the two as it required the use of both apps. They also had more questions about how to use the second app, despite the additional instructions I had added to the app. We faced some issues establishing a Bluetooth connection with the beacons to be registered. While they were able to successfully add the desired attachments to the beacon, why they were entering the information in the provided format (key-value) was unclear. One participant suggested providing a more explicit description within the app of what attachments should be added for the museum use case.

Two overall comments provided useful insight as well. The first came as a question: "How can we tell the difference between the physical beacons?" As it turns out, the Estimote beacons we were setting up have no distinct physical differences (other than being provided in three different colors), so the only way to tell which one is which would be to place them in different locations, then scan to identify the differences in what they broadcast - a very inconvenient process. To remedy this, I physically labeled each of the beacons with a unique number, which streamlined the actual deployment process described in Section 4.3.2. The second pair of comments indicated further improvements could be made to the Proximity Beacon API app. The volunteers mentioned at different times both that it would be useful to have more written instructions in the app, and that the user interface would be more friendly if it displayed less information at once. I would propose alleviating these conflicting issues by adding more explicit instruction, but also breaking out steps into separate screens. In this way, users could experience a more "step-by-step" flow of screens as they worked through setup, clicking "next" to view a new screen with the next step when they finished the current one, as opposed to the more dashboard-style current UI, which allows users to modify multiple settings from the same screen. This change would slow down the setup for power users, but make it drastically easier to follow for everyone else.

4.2.3 Conclusions

The Configuration Component, while it could use further improvements to make it more user friendly, was able to be used successfully by MIT Museum staff to set up beacons. They favored the Eddystone-URL configuration highly over the UID setup, whose attachments required the use of a second app and also took more time to enter manually.

4.3 Preparing for Visitors

The next phases of the experiment would test the Interaction Component by deploying beacons in the MIT Museum and encouraging real visitors to engage with them on their smartphones. Such a public presentation required considerable preparation before the experiment went live from May 2 through 15.

4.3.1 Exhibit Hall and Supplementary Content

For a reasonable testing scope, the Museum staff recommended deploying beacons within an exhibition hall reserved for featuring MIT student work: Projects and Prototypes [48]. This was an ideal exhibit for testing the system for several reasons. First and foremost, my experiment fit the

Table 4.1: Below is a chart listing the displays in the Projects and Prototypes gallery that had associated beacons, and the links to the corresponding content pages.

Display Name	**Beacon Content URL**
Panoramia	https://kasmus.scripts.mit.edu/mit_museum/panoramia_beacon.html
Educational Geiger Counter	https://kasmus.scripts.mit.edu/mit_museum/geiger_beacon.html
Await	https://kasmus.scripts.mit.edu/mit_museum/await_beacon.html
Faltering Stars	https://kasmus.scripts.mit.edu/mit_museum/stars_beacon.html
Troxes	https://kasmus.scripts.mit.edu/mit_museum/troxes_beacon.html
Teapot	https://kasmus.scripts.mit.edu/mit_museum/teapot_beacon.html
Motorized Prosthetic Arm	https://kasmus.scripts.mit.edu/mit_museum/arm_beacon.html
SPHERES Interact	https://kasmus.scripts.mit.edu/mit_museum/spheres_beacon.html
Walk Across	https://kasmus.scripts.mit.edu/mit_museum/walk_beacon.html
Locust Micro-UAV	https://kasmus.scripts.mit.edu/mit_museum/locust_beacon.html
MIT Air Quality Network	https://kasmus.scripts.mit.edu/mit_museum/air_beacon.html
MetaPiano	https://kasmus.scripts.mit.edu/mit_museum/metapiano_beacon.html
MIT Laser Seal	https://kasmus.scripts.mit.edu/mit_museum/laser_beacon.html

theme of student projects, which also granted the Museum a degree of immunity in case the system were to fail - it was being presented as an in-progress prototype developed by a student instead of content the Museum was solely professionally responsible for producing. Furthermore, the number and nature of the displays worked well for testing: at thirteen, there were few enough displays that I could deploy a beacon at each, and the student projects on display could be feasibly augmented with interesting additional content showcasing the students involved, how they made the projects, and their further or more recent developments. The fact that the exhibition is laid out as a single hallway was favorable as well; with only two entrance points, I would be able to advertise the experiment at both ends in hopes of attracting participants.

For the purposes of the experiment, I developed the supplemental content broadcasted by beacons myself to reduce time demands on the Museum staff graciously helping to host the experiment. I found additional information about the displayed student projects online, and compiled pictures, videos, text, and links to further details about each. I created websites integrating this information, themed to match the MIT Museum's branding, and attached the scripts described in Section 3.5.3 to collect data on user interactions with the content. MIT Museum staff provided guidance and feedback and ultimately approved these webpages. In a full-scale museum implementation, further resources could be dedicated to develop more interactive content during this step to potentially create an even more engaging experience for visitors.

4.3.2 Deployment of Beacons

I deployed a total of fifteen beacons in the Projects and Prototypes exhibit hall: one at each display, and one at either end to broadcast out an exit survey asking about the beacon experience. The display beacons were adhered out of plain sight to discourage visitors from tampering with the hardware. I deployed the beacons approximately one week before the experiment was scheduled to begin, with their URLs redirecting to a "Coming Soon" page so that they could be tested without any tech-savvy visitors discovering them before the content was finalized.

Because the displays in this exhibition are in close quarters, the Museum staff recommended beacons broadcast to a range of five feet. This would ideally avoid most intersections of ranges, so that a visitor's smartphone directly in front of a display would find only the beacon associated with that display, and not a list with all of the neighboring displays' beacons as well. While this should have been possible, there was unfortunately a confirmed bug in Estimote's software for adjusting beacon ranges during this time [49]. I attempted to limit the beacons' default range of 40 meters manually as recommended in the Estimote forums by layering the beacons with signal-deflecting aluminum foil. This proved somewhat effective, reducing the ranges to approximately 10-12 feet as measured, but this higher range still caused more overlap than desirable. Estimote was looking into a fix for this bug which would resolve the issue for future implementations.

4.3.3 Passively vs Actively Recruited Participants

The original means of acquiring participants was solely passive, with posters advertising the experiment at the entrances to the student work exhibit encouraging visitors to interact with the displays using their smartphones. This opt-in measure does provide some selection bias, but is most representative of how the system would function in a true museum setting. These participants provided a diverse set of demographics and degrees of technical knowledge, making them an interesting sample. When I was available, I monitored the experiment, offering encouragement to participate and assistance where necessary. However, as the results in the following sections will discuss, participation via this channel was much lower than anticipated. In order to still gather a reasonable amount of data during the weeks the beacons were actively deployed, I additionally recruited volunteers from my circles at MIT - friends, housemates, and members of the App Inventor team - to visit the exhibit and participate in the experiment. All survey responses are anonymous so these volunteers could report their feedback honestly.

4.3.4 Instructions Posters

With the help of MIT Museum staff, I created posters to be displayed at either end of the Projects and Prototypes gallery to inform visitors about the experiment, encourage them to participate, and explain how to connect their smartphones to the beacons. I made separate sets for each week: Physical Web instructions for the first week and dedicated Android app instructions for the second week. The Physical Web instructions were split up into separate steps for iOS or Android devices. I chose to make those instructions specific to Google Chrome, as it is by far the largest Physical Web supported browser currently available on both platforms.

Because these instruction posters were intended for a non-technical audience, the majority of the effort went into simplifying the steps and the language as much as possible. Even so, there are a number of settings that must be modified to enable BLE communications. Museum staff and friends provided feedback to ensure the steps were understandable and complete. The complete posters, as they were displayed during the experiments, can be found in Appendix A.

4.3.5 Visitor View

These images convey the beacon experiment experience for a visitor to the museum.

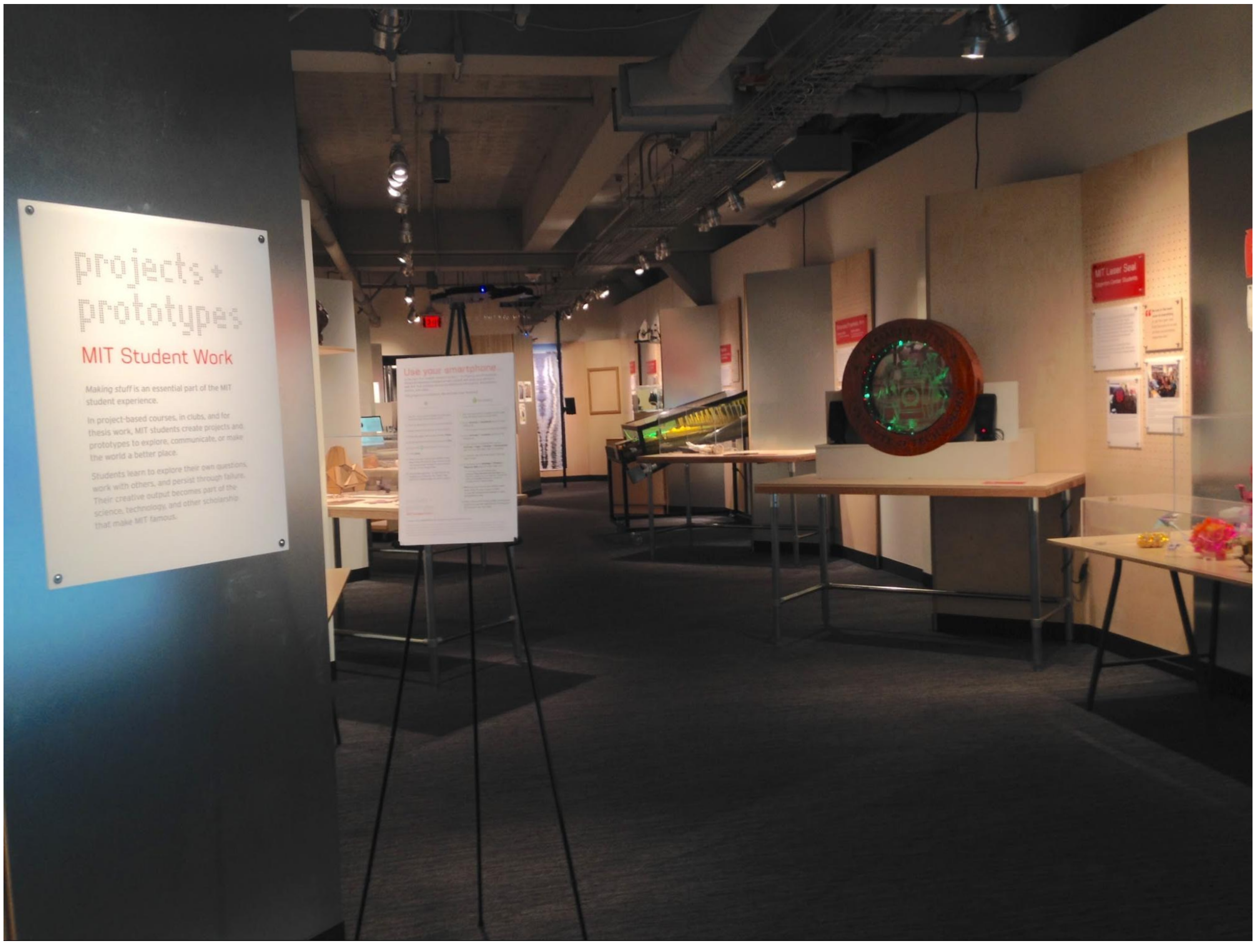
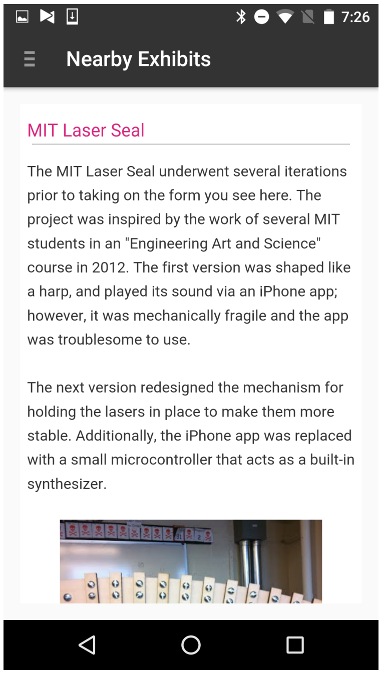

projects +
prototypes
MIT Student Work
Making stuff is an essential part of the MIT student experience.
In project-based courses, in clubs, and for thesis work, MIT students create projects and prototypes to explore, communicate, or make the world a better place.
Students learn to explore their own questions, work with others, and persist through failure. Their creative output becomes part of the science, technology, and other scholarship that make MIT famous.
Use your smartphone
MIT Laser Seal

Figure 4-1: This is a picture taken from one entrance to the Projects and Prototypes gallery. The poster on the tripod at left encourages visitors to "Use your smartphone" to get additional information on these displays, and provides instructions. An identical poster was placed at the opposite end of the hallway.



MIT Laser Seal

The MIT Laser Seal underwent several iterations prior to taking on the form you see here. The project was inspired by the work of several MIT students in an "Engineering Art and Science" course in 2012. The first version was shaped like a harp, and played its sound via an iPhone app; however, it was mechanically fragile and the app was troublesome to use.

The next version redesigned the mechanism for holding the lasers in place to make them more stable. Additionally, the iPhone app was replaced with a small microcontroller that acts as a built-in synthesizer.

Figure 4-2: A visitor standing in front of the MIT Laser Seal display as shown on left would be able to access the additional content shown on right via the corresponding beacon.

4.4 Phase Two: Physical Web Interactions with Eddystone-URL

During the first week of testing, beacons at the MIT Museum's Projects and Prototypes exhibition were configured to broadcast an Eddystone-URL that visitors could view and interact with on their smartphones via the Physical Web. The first three subsections provide assessments of that experience from my observations of visitors' behavior and comments, their anonymous survey responses, and the interaction metrics provided by the Report Component. Section 4.4.4 then offers an evaluation of these findings.

4.4.1 Observations

Here is a condensed list of observations about the system made by watching or conversing with visitors as they interacted with the system or opted out, and my thoughts on their implications.

- The vast majority of visitors did not notice or pay mind to the instructions poster at all - only 20 participated over the course of the week. Many seemed to skim most of the textual material in the exhibit in favor of interacting with more eye-catching or interactive physical displays, which likely indicates they would not be interested in reading additional content about the displays regardless of whether they had noticed the poster. However, it is possible that a more noticeable poster, or more notices than just at the ends of the exhibit, could have garnered more participation.

- Others took the time to look at the poster but declined to participate because the instructions looked too complicated. Unfortunately this is difficult to mitigate, as significant efforts had already gone into simplifying the necessary steps as much as possible.

- Several iOS users who did not have the Google Chrome app installed declined to participate as they did not wish to download the app.

- Many visitors seemed surprised that the Physical Web experience did not involve them downloading an "MIT Museum" app of some sort. Some considered this favorably and were glad they did not have to download another app and would just receive notifications from Chrome instead. Another, when faced with multiple setup steps, asked if there wasn't an app they could just download instead that would do that work for him. This feedback is interesting to consider in terms of the comparison between Eddystone-URL and UID protocols; the UID version of the system tested the following week does use a dedicated app.

- Participants using iPhones typically reported no trouble with finding nearby beacons on their phones. Participants using Android devices had mixed results, with some devices locating them fine, some finding them after a significant delay, and in one case never finding them at all. The source of this buggy behavior is difficult to pin down due to its inconsistency, but likely stems from some hardware receiving BLE signals more reliably than others, as Android devices are produced by a variety of manufacturers.

- Several participants reported strange behavior where the list of beacons discovered did not accurately reflect what was nearby: sometimes the closest beacon would be missing from the list; other times, a faraway beacon would mysteriously appear. This is almost certainly a consequence of the overlap of beacon ranges caused by not being able to accurately adjust transmitting power; however, it is possible that interference caused by various surfaces or configurations of the exhibit hall could cause similar issues.

- Several insightful comments were made about the user experience of interacting with Chrome's Physical Web feature:

    - After clicking through to view a webpage, it is non-intuitive how to get back to the list of Physical Web pages nearby.

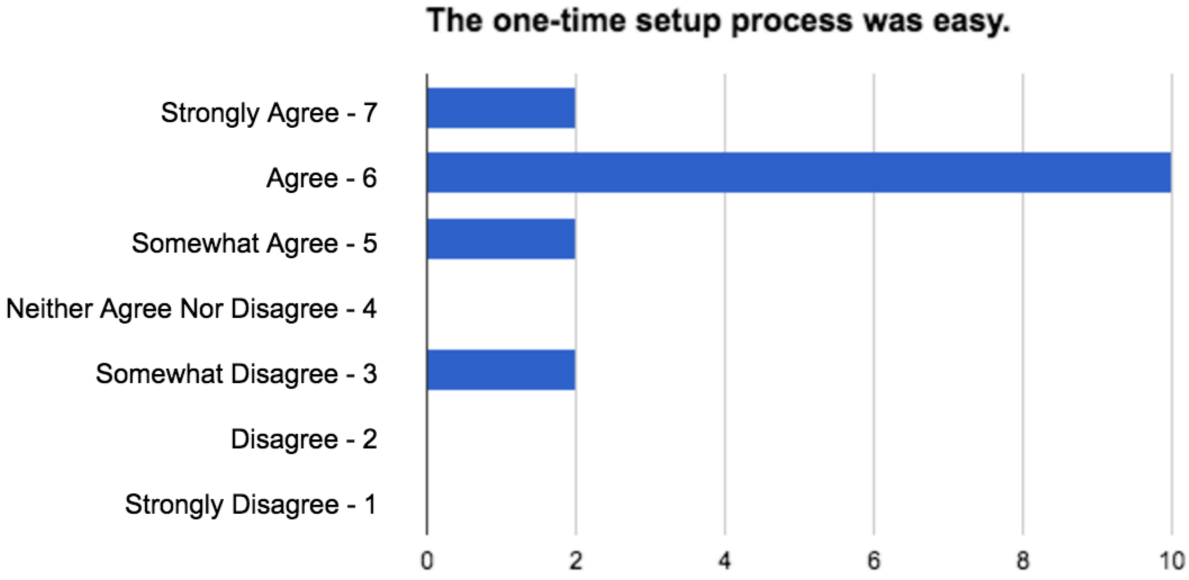
    - One participant commented favorably on the fact that a new tab is opened for every Physical Web page viewed, stating that they might like to be able to go back to them later. Another found it annoying that they would have to go back and close all of them.

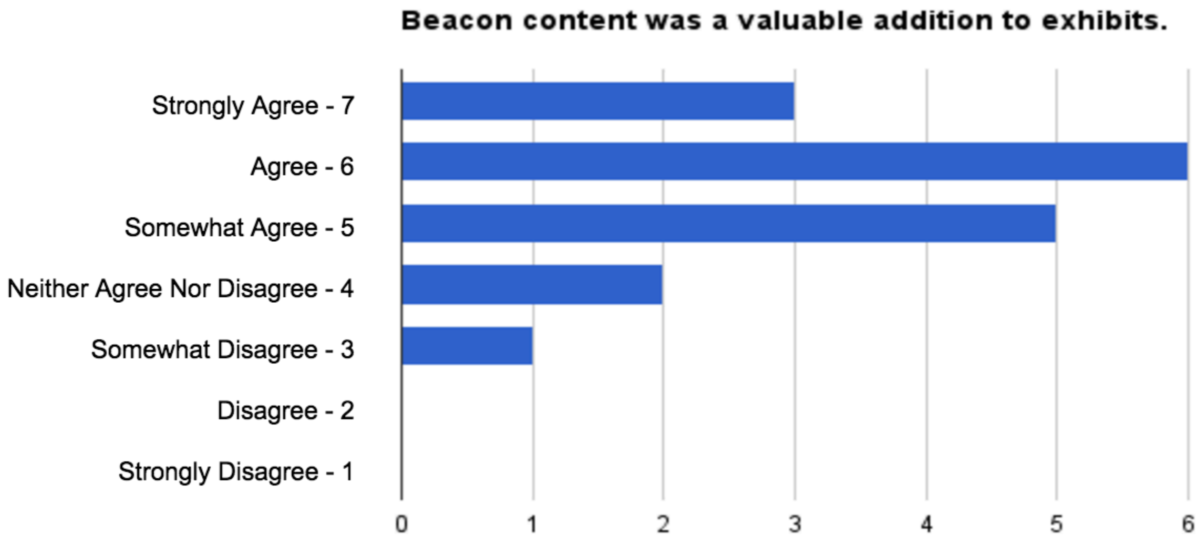
    - One Android user disliked the silent notifications when Physical Web pages are found nearby, finding them easy to miss. These are a deliberate choice in the Android implementation to not interrupt a user, but only present the information when it is deliberately sought out.

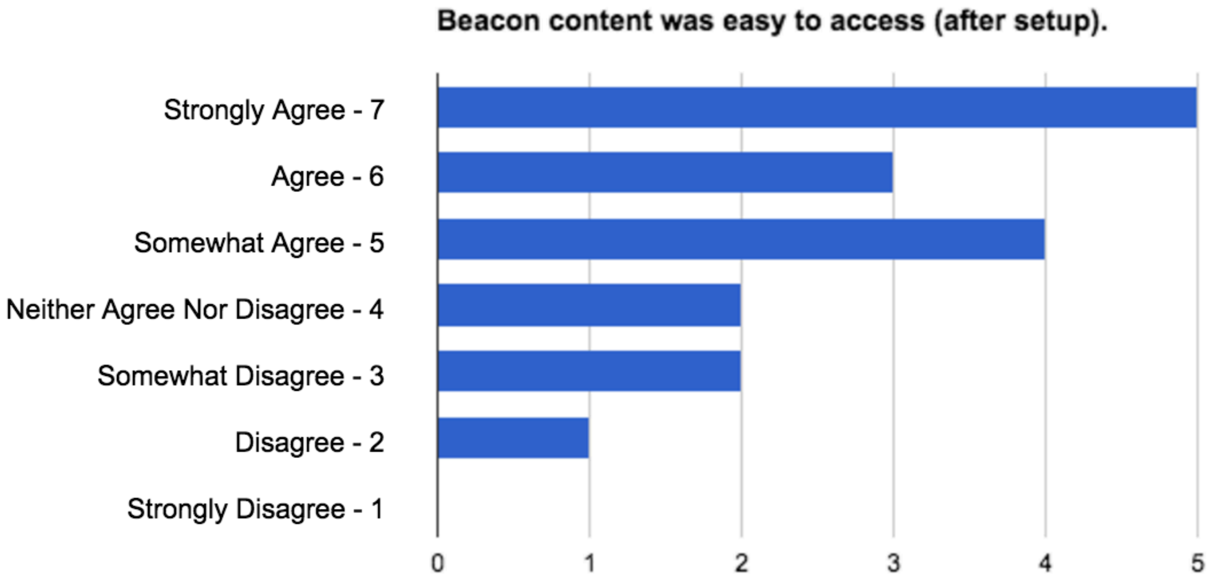
4.4.2 Survey Results

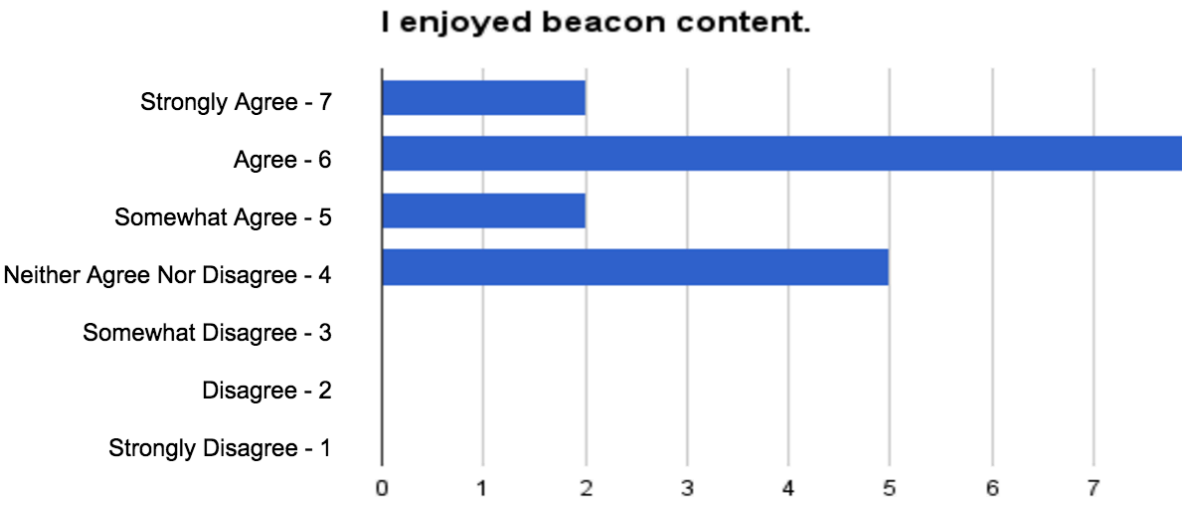
Participants were prompted to complete an exit survey reporting on the quality of their beacon experience upon exiting the exhibit, either verbally, by encountering a beacon broadcasting the survey to their smartphones, or in the footer of every beacon website. Paper copies were also available at the information desk just outside of the exhibit. Because those not interacting with beacons were less likely to encounter the survey, there is a selection bias skewed towards those who were interested in trying out the beacon technology. Due to the small sample size of respondents (seventeen), the data from these surveys is evaluated qualitatively instead of performing statistical analysis.

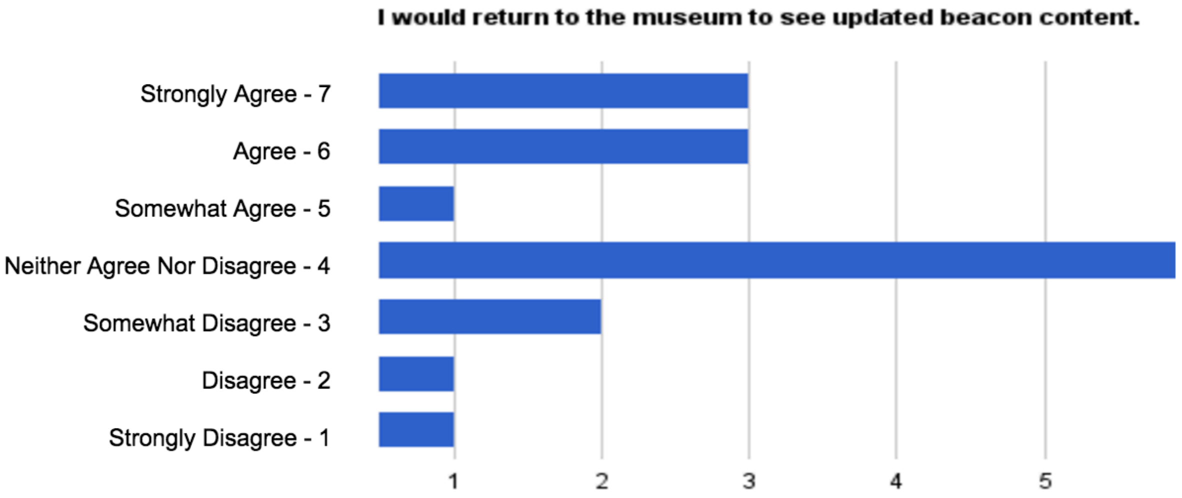
Of seventeen respondents, sixteen visitors were able to successfully set up their device and interact with beacon content. Of those successful, eleven accessed beacon content from iPhones and five from Android devices; the unsuccessful attempt was made from an Android device. Participants were then asked to rank their agreement or disagreement, on a 1-7 Likert scale, with the following statements to assess the user-friendliness of the system, their enjoyment of the beacon experience, and the value added by the beacon system. The results of these questions are charted here and analyzed below.




Beacon content was a valuable addition to exhibits.
Strongly Agree - 7
Agree - 6
Somewhat Agree - 5
Neither Agree Nor Disagree - 4
Somewhat Disagree - 3
Disagree - 2
Strongly Disagree - 1
0
1
2
3
4
5
6


Beacon content was easy to access (after setup).
Strongly Agree - 7
Agree - 6
Somewhat Agree - 5
Neither Agree Nor Disagree - 4
Somewhat Disagree - 3
Disagree - 2
Strongly Disagree - 1
0
1
2
3
4
5


I enjoyed beacon content.
Strongly Agree - 7
Agree - 6
Somewhat Agree - 5
Neither Agree Nor Disagree - 4
Somewhat Disagree - 3
Disagree - 2
Strongly Disagree - 1
0
1
2
3
4
5
6
7


I would return to the museum to see updated beacon content.
Strongly Agree - 7
Agree - 6
Somewhat Agree - 5
Neither Agree Nor Disagree - 4
Somewhat Disagree - 3
Disagree - 2
Strongly Disagree - 1
1
2
3
4
5

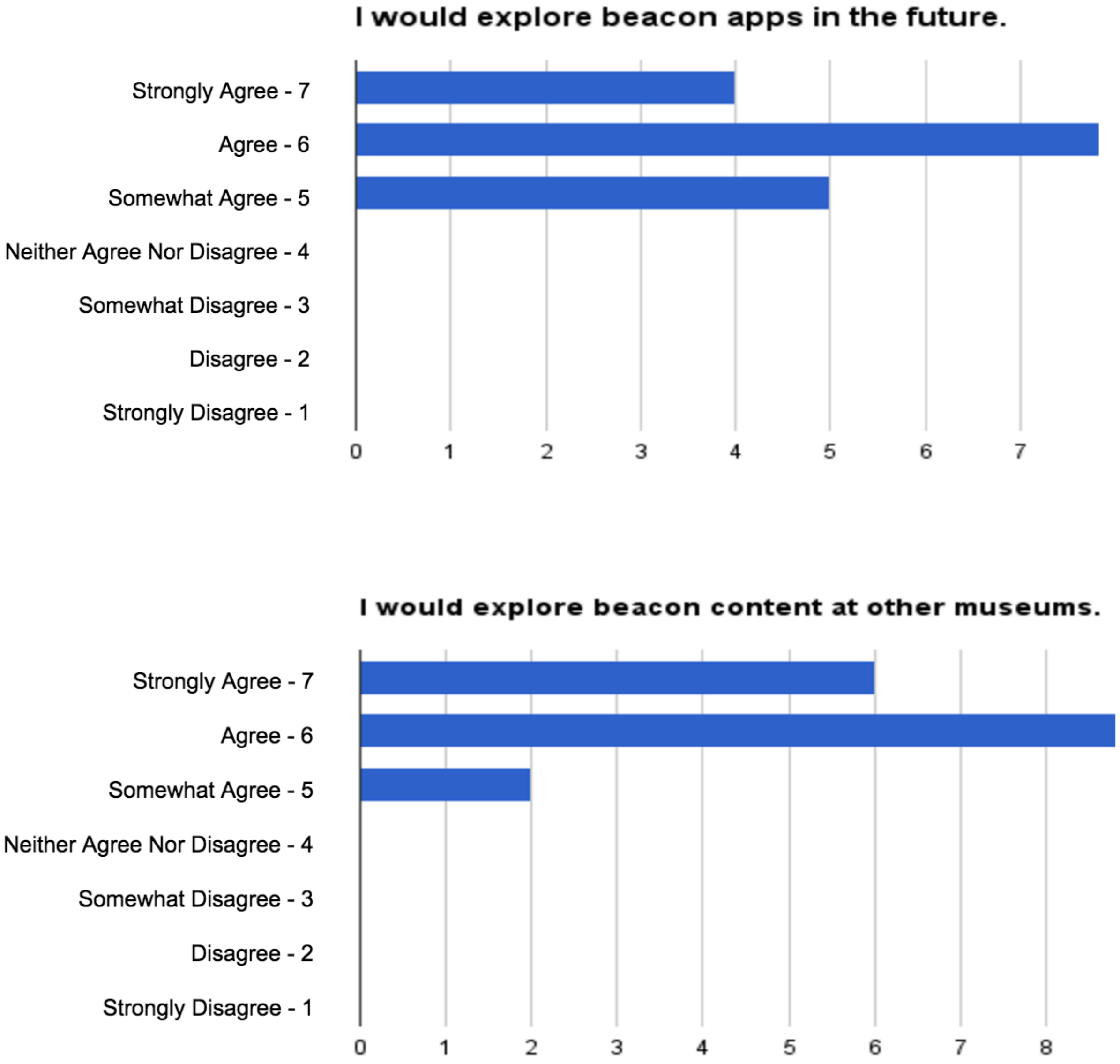


Figure 4-3: The above charts represent the anonymous responses of participants interacting with the Physical Web version of the system.

Overall, most participants agreed that setting up their devices to enable the Physical Web was fairly easy, as was accessing Physical Web pages with beacon content thereafter - with a couple voices of dissent. They enjoyed or were ambivalent about beacon content, but found it to be a valuable addition to the museum. Interest in returning to the museum to see updated beacon content was mixed, but participants agreed they would explore beacon content at other museums or at large in the future. This data represents a very small sample of visitors who opted in to the experiment, but it shows promise that most visitors found the experience viable, valuable, and enjoyable. While museums may not gain much added value from keeping beacon content fresh, its addition alone is something visitors were interested in exploring. Quotes taken from the free response portion of the survey provide further insight:

- “Some website content was better than others. The pages that had links allowing you to dig deeper into the research or connected you to the creator’s personal project page were more interesting.”
- “This was a cool way to learn more about the exhibits! I enjoyed that I could get more info and be able to continue learning afterwards.”
- “It was a very good idea - I think the interactive information to your phone seems a natural next step for museums. I think the refresh could be a little faster (sometimes it took a while to locate the exhibits). If the info could be on an app it might be a bit more streamlined.”
- “Would be easier without having to download Chrome - can it be done on other browsers?”
- “Had problems finding websites - only found [two]. Kept force closing application and refreshing - nothing happened.”
- “I thought the Beacons added valuable content to the exhibits, but was somewhat difficult to access.”
- “I liked interacting with the content on my phone, but I worry that I would read it more than the things on the wall. And sometimes I go to museums to get away from my phone.”

4.4.3 Interaction Data

Data gathered from the Interaction Component and displayed via the Report Component reveals some statistics on how participants interacted with beacon websites. This data shows 20 unique visitors to the set of beacon websites - slightly higher than the number of survey respondents. They interacted with beacons 120 times, for an average of 6 interactions per visitor. This would seem to indicate that most visitors view only a handful of the beacon websites, as opposed to exploring thoroughly all of the available content. A deeper look into this data proves this hypothesis true, as 70 percent make 6 or fewer interactions, and the other 30 percent make an average of 14 interactions (on 13 beacon websites). Six visitors viewed only one beacon and did not revisit the system. The vast majority of the views came in Friday through Sunday as opposed to during the week, which reflects the MIT Museum's usual visitation, though this statistic is likely skewed according to when I actively staffed the experiment and recruited participants. The most-viewed beacon by far was the Walk Across display, which makes sense as it is located at one of the far ends of the exhibit hall by an instructions poster. In terms of types of interactions, watching videos and exploring links to further information were about equally popular, with six and five clicks respectively.

While the survey responses seemed to indicate a generally positive view towards the beacon experience, the metrics revealed by interactions provide a slightly more nuanced picture. Most visitors who interacted with beacons opted to only look at a few. This would be the case for those who had trouble getting others to load, but could also indicate fading interest in the additional information provided. However, a significant handful did opt to explore the content more thoroughly, interacting with the majority of beacons, which would seem to indicate a significant proportion of users that found the material engaging enough to continue looking for them.

4.4.4 Conclusions

There were several interesting takeaways from this phase of the experiment. Firstly, I was surprised by how few visitors opted to try out the beacon experiment at all. I recognize that not everyone would have the time or want to make the effort to undergo the setup, but thought there would be more interest in a museum setting to receive additional content. Not only were most visitors uninterested, but 70 percent of those who were viewed less than half of the beacons available. I predict that more engaging content could improve these figures, but clearly it is a feature that not everyone needs in their museum experience. However, the 30 percent that interacted more did interact thoroughly, and survey responses indicated that most participants enjoyed the beacon experience and found it a valuable addition to the museum.

Another interesting aspect is the usability of the system. Observations showed that many visitors declined to participate after viewing the instructions poster, but those who did go through the process found it fairly easy to complete. iPhone users had a consistently excellent experience using the system, while difficulties connecting to beacons hampered some Android users, but not others. While this is a reasonable success rate for an experiment with a new technology, it would be more problematic were this system to be offered as a commercial service to museum patrons - visitors expect everything they pay for to work.

4.5 Phase Three: App Interactions with Eddystone-UID

For the second week, I reconfigured the beacons to broadcast with the Eddystone-UID frame type. Updated instructions posters informed visitors how to interact with beacons via a dedicated app for Android devices.

4.5.1 Observations

As in the first week, I monitored the experiment and made notes on how visitors were interacting with the system and any comments they made to me. Here are the most relevant findings:

- Two participants commented that they wished the beacon content was more interactive, as opposed to "just static information." Personally, I think this desire is very aligned with the most interesting possibilities for a system like this in the future, should a museum have the resources to create such content.

- Several Android users were once again stuck finding only a subset of the deployed beacons. Turning off the background scan and restarting it as a sort of "manual refresh" alleviated the issue for some users.

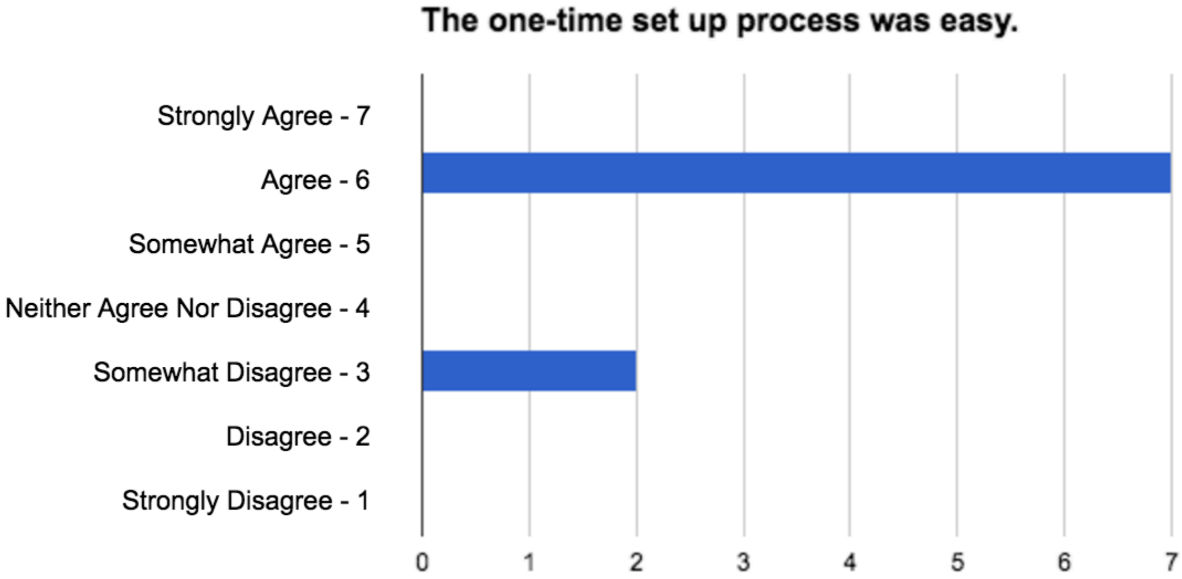
- One participant commented that the app should be able to check and notify the user if their device is not BLE compatible. I agree that this would be a useful feature. Furthermore, the app could also be made to check that the Bluetooth setting was enabled at the time a user attempts to scan for beacons. Both of these preconditions were mentioned in the instructions for the experiment, but could be usefully integrated for future applications.

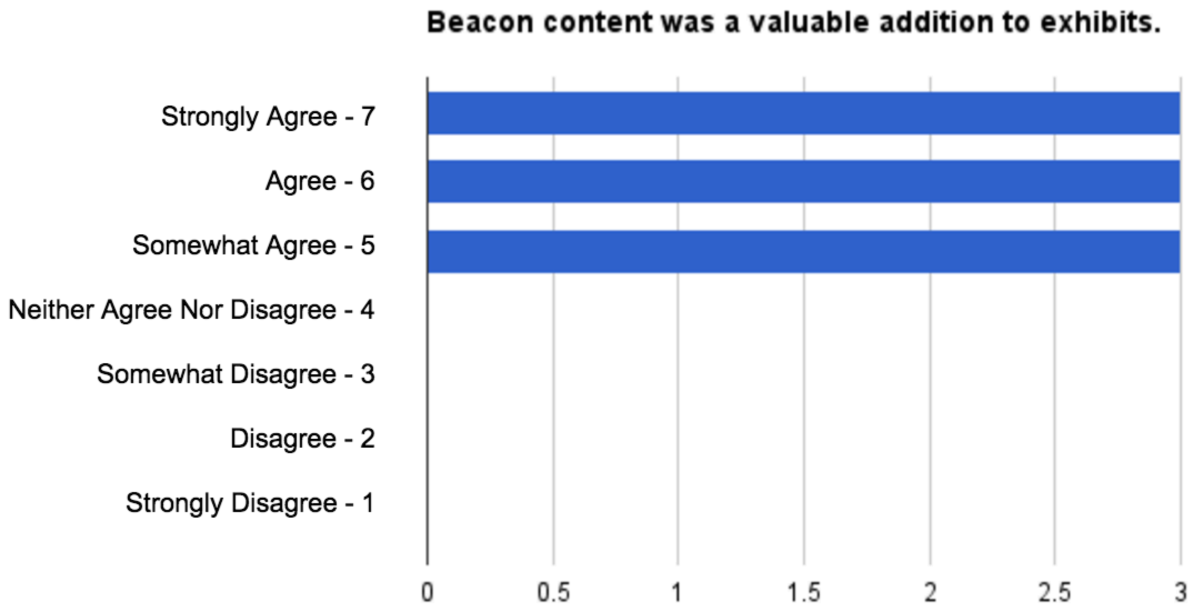
- The inconsistent and overdrawn beacon ranges continued to cause minor issues. During this week, it manifested as users often finding beacons other than those directly in front of them. Therefore, they were being guided through the exhibit by the app, as opposed to the app tracking them accurately as they made their own way through from display to display. One user exclaimed, “Oh, here it is!” upon finally coming across the physical display of the beacon his app was displaying. The ideal behavior here, when the Estimote range bug is fixed, should be much more seamless.

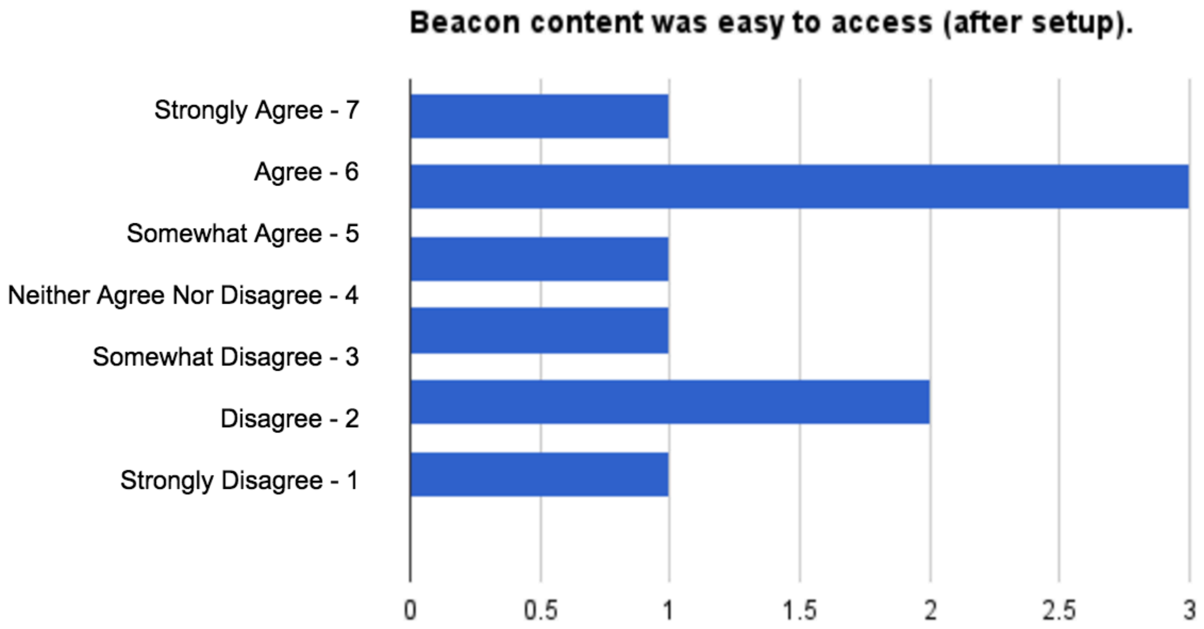
4.5.2 Survey Results

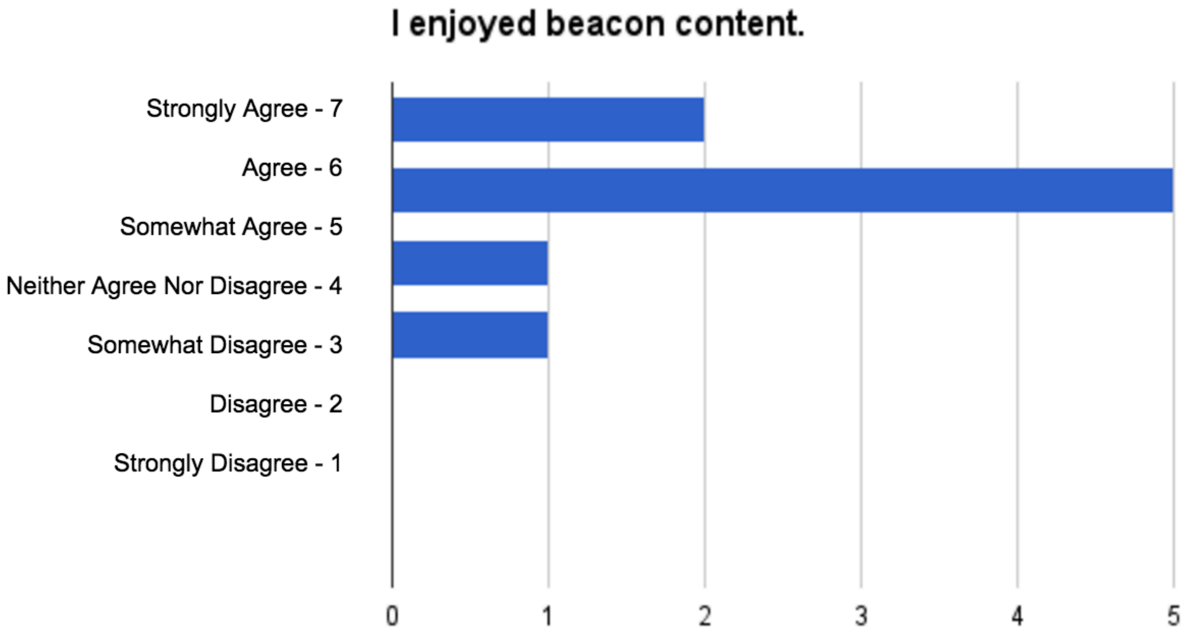
Visitors were once again encouraged to complete an exit survey after interacting with beacons. This week, the exit survey was additionally available as a tab in the Android app, in order to gain more responses. The survey was identical to the one distributed the previous week, with the exception of some minor language changes to reflect that participants were now interacting with an app instead of the Physical Web. The question about which OS a user’s smartphone was running was also removed, as the app was only available for Android users at the scope of this thesis project. Once again, the results are considered qualitatively and with acknowledgement of selection bias.

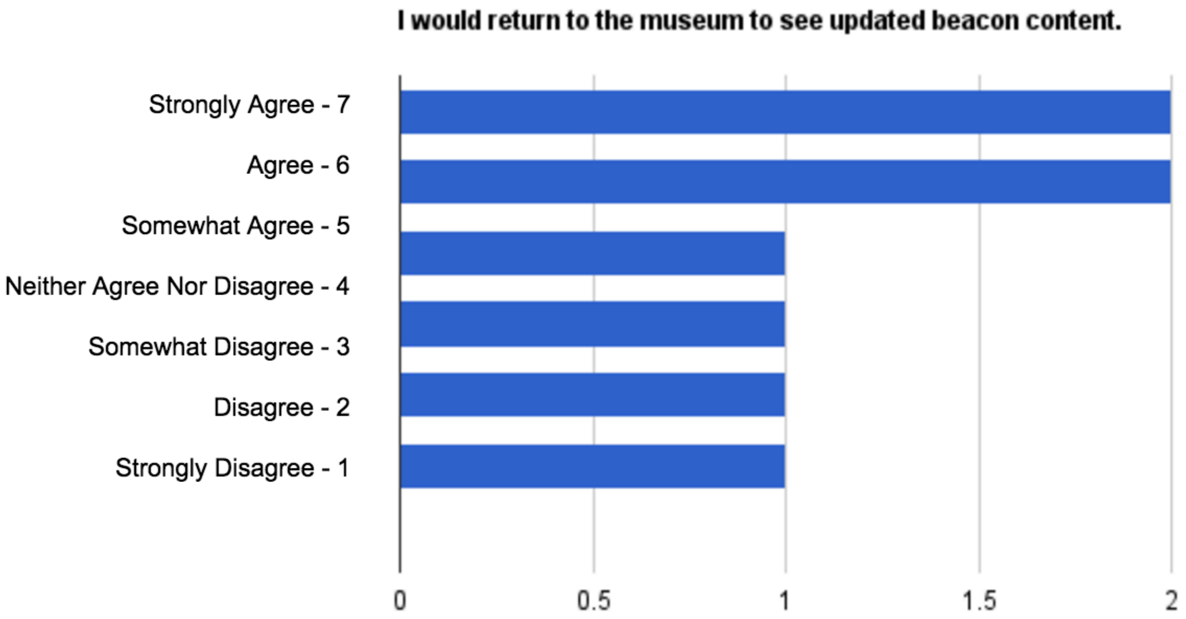
Ten participants took the survey, and of these, nine answered all questions and nine successfully viewed beacon content via the Android app. As in the previous section, I will present their responses to the seven Likert-scale questions and then draw conclusions from the results.




Beacon content was a valuable addition to exhibits.
Strongly Agree - 7
Agree - 6
Somewhat Agree - 5
Neither Agree Nor Disagree - 4
Somewhat Disagree - 3
Disagree - 2
Strongly Disagree - 1
0
0.5
1
1.5
2
2.5
3


Beacon content was easy to access (after setup).
Strongly Agree - 7
Agree - 6
Somewhat Agree - 5
Neither Agree Nor Disagree - 4
Somewhat Disagree - 3
Disagree - 2
Strongly Disagree - 1
0
0.5
1
1.5
2
2.5
3


I enjoyed beacon content.
Strongly Agree - 7
Agree - 6
Somewhat Agree - 5
Neither Agree Nor Disagree - 4
Somewhat Disagree - 3
Disagree - 2
Strongly Disagree - 1
0
1
2
3
4
5


I would return to the museum to see updated beacon content.
Strongly Agree - 7
Agree - 6
Somewhat Agree - 5
Neither Agree Nor Disagree - 4
Somewhat Disagree - 3
Disagree - 2
Strongly Disagree - 1
0
0.5
1
1.5
2

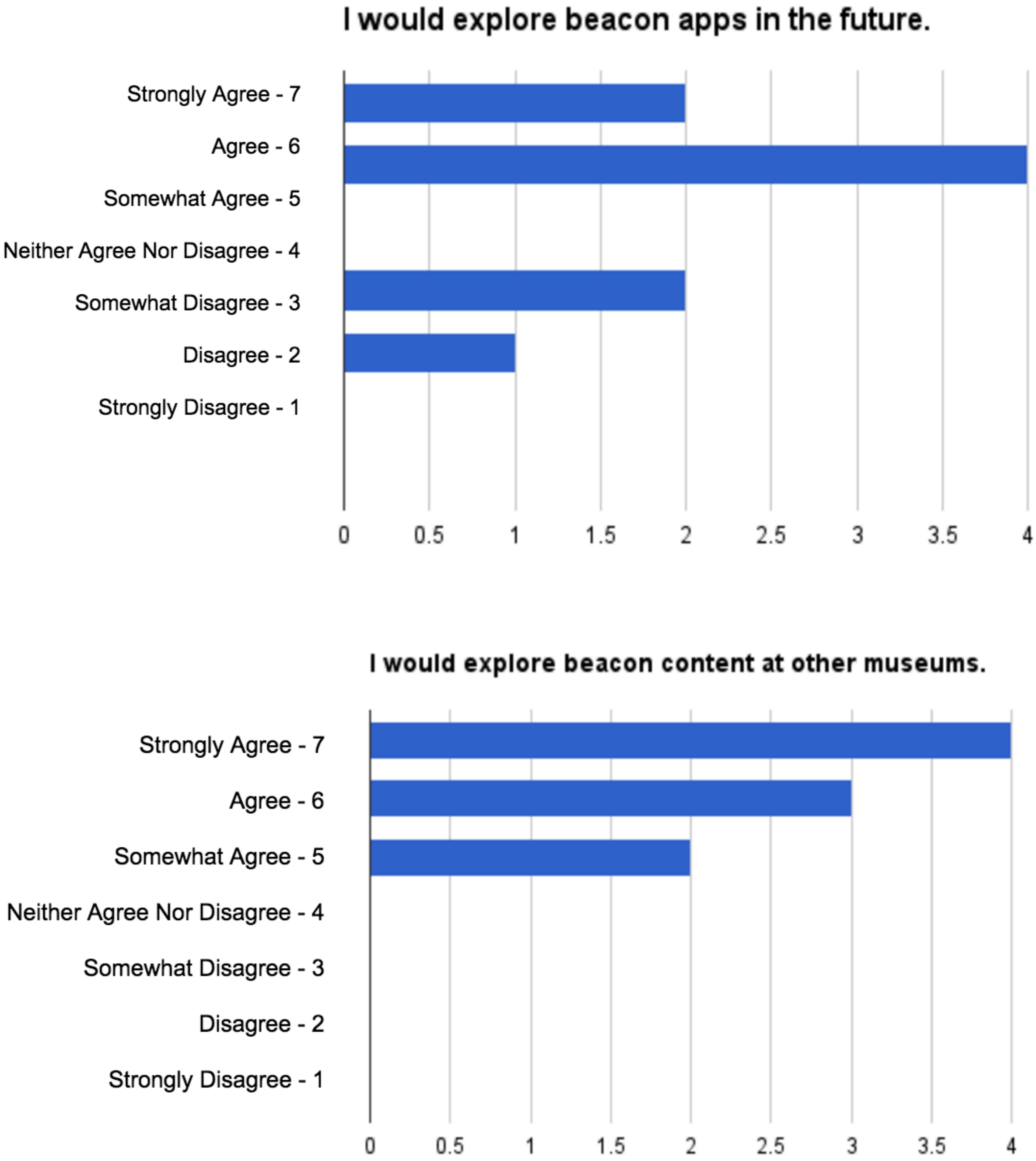


Figure 4-4: The above charts represent the anonymous responses of participants interacting with the dedicated app version of the system.

Once again, the survey results show that almost every participant enjoyed beacon content and thought it was a valuable addition to the museum. Most found the dedicated app setup easy, but a larger minority this time disagreed that finding beacons was easy after the setup. It's possible that this is due to the app being Android-only, if the correlation between Android hardware and a less reliable Bluetooth connection experience hypothesized the previous week is indeed a causal relationship. Participants were similarly mixed on whether they'd return to see updated beacon content at the MIT Museum, but all agreed that they would explore beacon content at other museums. In this week's testing though, participants were less certain they would want to try out other beacon apps in the future; likely another consequence of the unreliable Bluetooth connection issues.

Notable quotes from the open response question follow:

- "Once understood I enjoyed the added information."
- "This application definitely enhanced my museum viewing experience! It was really cool having the app point me to interesting exhibits nearby. I almost missed one of the cooler exhibits in the area but then the app pointed it out!"
- "Only two exhibits showed up."
- "I was able initially only to see [one beacon]. No other beacons showed up until I closed and opened app again then [more] showed up. Interaction from there was good."
- "The range was short enough that just the visible exhibits showed. It might be nice to change the listing as I walked around, without [manually] refreshing."

4.5.3 Interaction Data

The Report Component was again used to collect information and shed light on how experiment participants interacted with beacons, this time through the Android app. The database recorded 16 unique visitors making 83 total interactions. Originally, there were far more app users recorded than logs of activity, which upon investigation turned out to be people or bots downloading the app from locations other than the MIT Museum. Because the entries were only relevant for users who were actually able to interact with beacons, I filtered these users down to only include those who had at least one beacon interaction log associated with their ID, resulting in the 16 reported. That makes an average of 5.2 interactions per user, which is fairly consistent with Week 1's results. The graph is slightly less skewed, but similarly so, with most users interacting with relatively few beacons, and few users interacting with relatively many beacons. Links received the same number of views within the app as in the Physical Web (five); however, videos saw no views at all within the app. Interactions plotted over time once again occurred most heavily from Friday through the weekend, and there was little pattern to which beacons received more or less traffic.

4.5.4 Conclusions

Week 2 of the experiment provided its own interesting data as well as a comparison against the results of Week 1. This comparison will be postponed for discussion in Section 5.2. Standing alone, the dedicated app approach was easy to set up but faced connectivity issues that made it difficult for several participants to view more than a couple of beacons. Low participation can be partially attributed to the fact that this phase of the experiment was only available to Android users, and largely to the same factors that affected Week 1. Participants did find beacon content valuable and enjoyable when they were able to access it, and unanimously agreed that they would explore such content at other museums, though they were less enthusiastic about beacon apps on the whole.

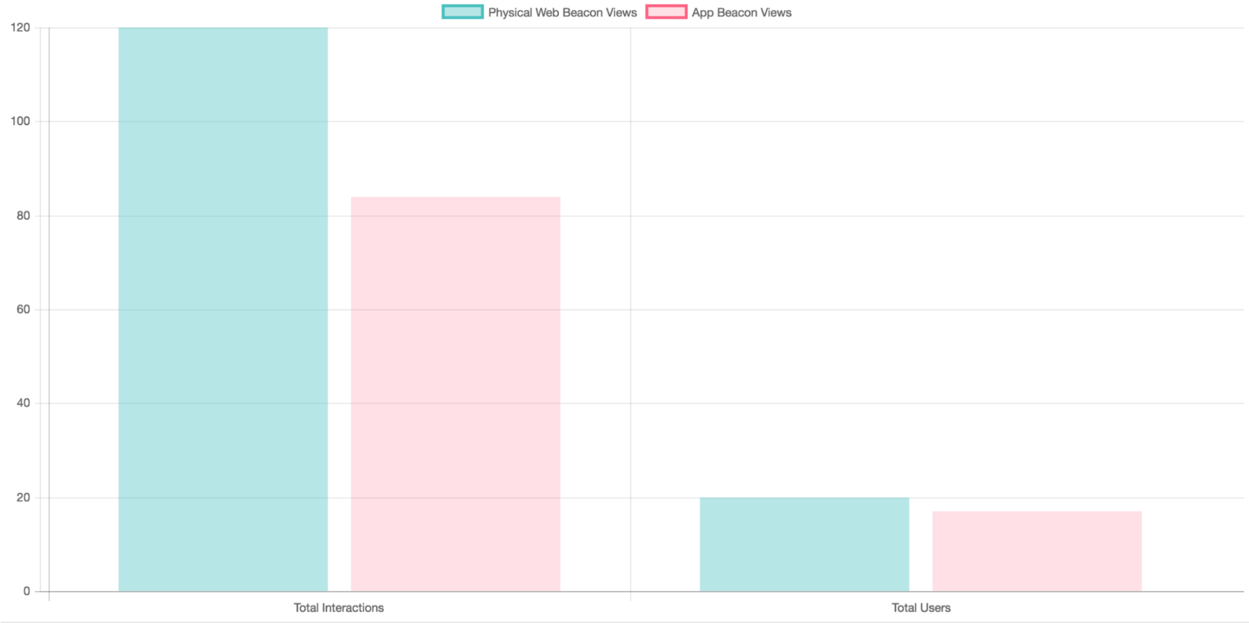
4.6 Phase Four: Aggregate Data

For the final phase of the experiment, I demonstrated the Report Component to a group of MIT Museum staff, displaying the real data gathered during the two weeks of testing the Interaction Component with museum visitors. I then asked the staff members to respond to an anonymous survey to indicate how useful they thought such statistics could be for the museum in making future decisions.

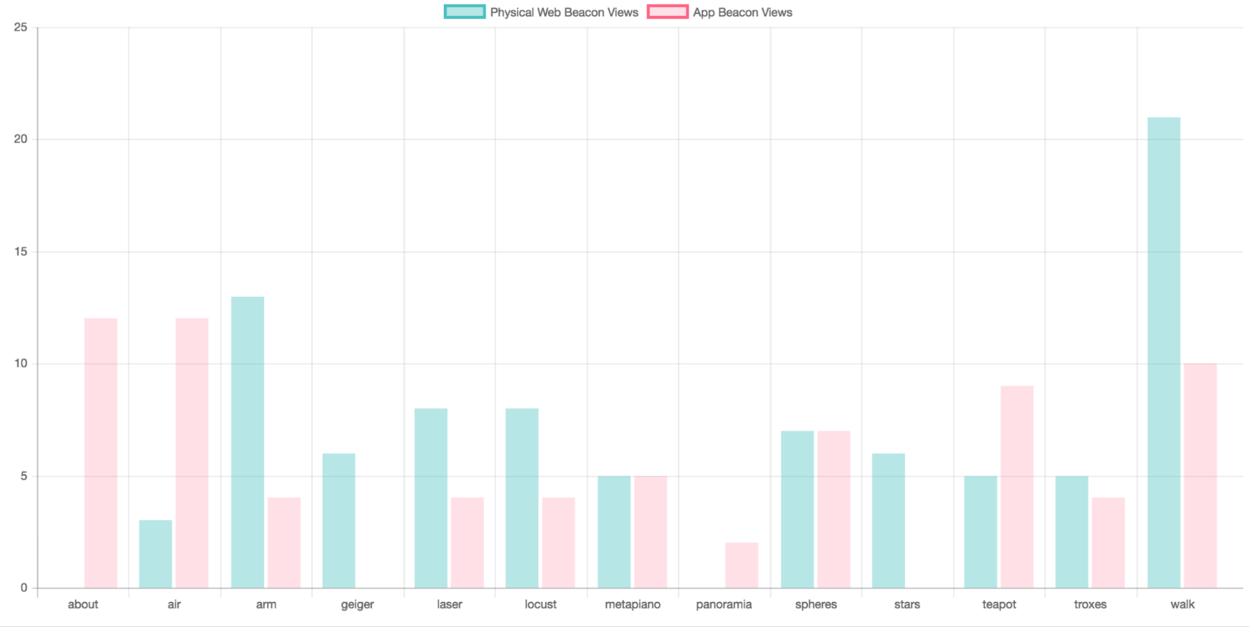
4.6.1 Description of Presentation

To demonstrate the capabilities of the Report Component, I created graphs of several metrics calculated from the Interaction Component's data to present to MIT Museum staff. For each of these, I overlaid the Week 1 and Week 2 data so they could be taken in together. Though the beacons were removed from the exhibit at the time of this meeting, I explained that these visualizations would update their contents live to reflect new interactions as visitors made them. The visualizations I graphed and presented are as follows, with some examples displayed:

- A simple bar graph depicting the total number of interactions and unique users for each week of the experiment



- A histogram showing the total number of interactions with each of the 13 beacons (plus the "about" page linked by the exit beacons)

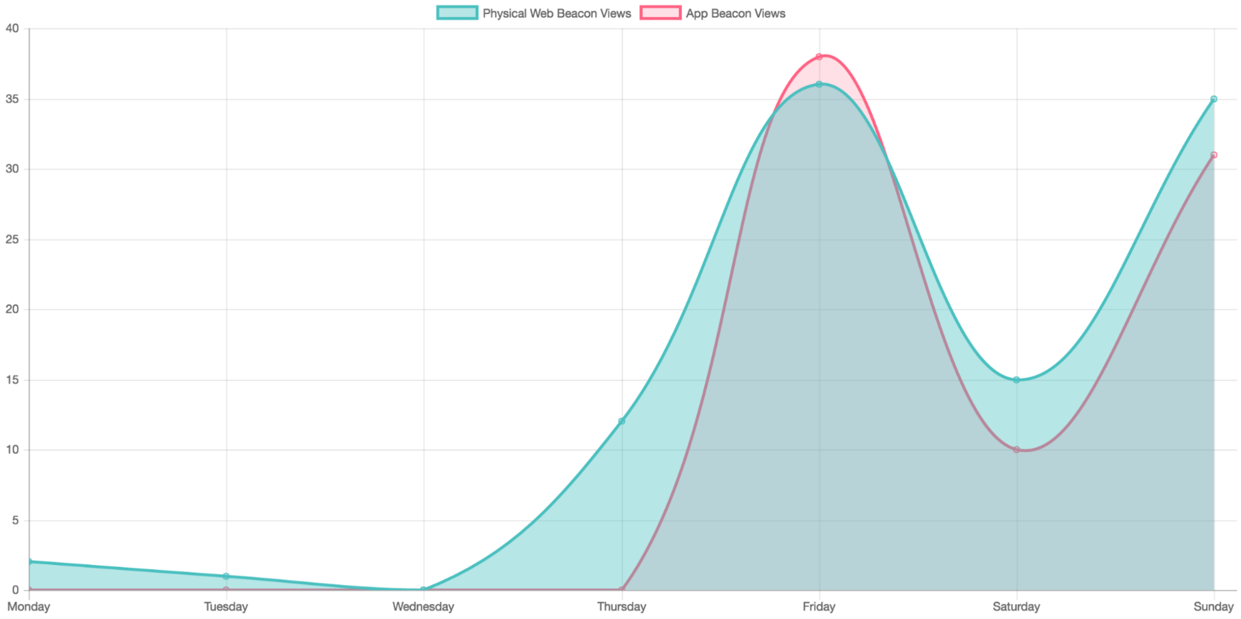


- A pie chart displaying the types of devices used by participants

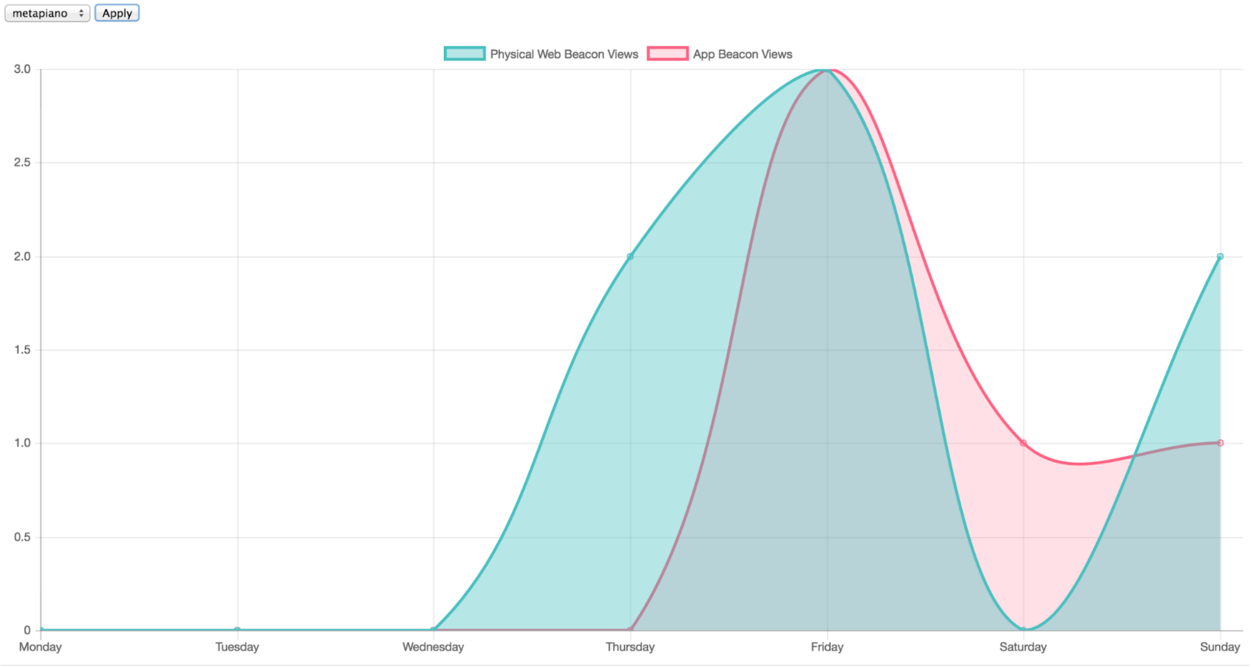
- A histogram illustrating how many interactions with beacon content each unique visitor made

- A histogram displaying the total number of interactions users made with each external link and embedded video included in beacon content

- A line graph tracking the total number of interactions made with all beacons over time



- A page showing a line graph for the total number of interactions made with a selected beacon over time, with a dropdown box to select the given beacon



4.6.2 Survey Results

With the rise of "big data" as a major factor in how modern large or tech-savvy businesses make decisions, the goal of the Report Component was to provide similarly useful information to museum staff. After presenting the data above, I surveyed a group of MIT Museum staff to assess its success.

Of the questions asked, the staff's responses indicated that the information could be useful in some ways I predicted, but not in others. For example, the staff unanimously agreed at least somewhat that beacon data could inform the museum about the popularity of exhibits. However, they disagreed collectively that it would be used in making decisions about what new types of displays to add or which displays to retire. There were mixed responses as to whether beacon data could be a good indicator of which content was well-liked or not. Asked overall whether beacon data is useful to the museum, all of the staff members surveyed "somewhat agreed" (a response of 5 out of 7 on the Likert scale).

One staff member provided an insightful rationale for this hesitant approval in a response to the open-ended prompt for additional feedback:

> "Although I think that additional data pushed by beacons is an interesting and fun way to add to the visitor experience in the museum I would be hesitant to rely overmuch on beacon data when evaluating the relative popularity of museum exhibits. Exhibits

> are created with different reactions and interactions in mind; some visitors may prefer a more passive experience in the museum and find that pushed content interferes with their enjoyment. I definitely think that beacon data correlates to a certain type of visitor's interest in a given exhibit but not necessarily to lack of interest."

This comment reflects some of the conclusions I drew in the previous sections regarding lower visitor participation with beacons than expected, and the different extents to which participants opted to explore available content. It makes sense that museums should consider this limited window of perspective when analyzing the data for their purposes as well.

Additionally, I asked staff members what other information they might like to see collected by the system. Two asked to see a more clear representation of the interactions an individual user was taking across the museum. Duration of interaction was suggested, a metric considered and discussed in Section 3.5.4. Another recommendation that would be very useful, but difficult to track, was an indication of visitors' failed attempts to use the system. One staff member mentioned the potential for social moments to be tied in with the Interaction Component; while this feature was determined too resource intensive for this thesis project as mentioned in Section 3.4.4, it would be interesting to revisit for a larger-scale future implementation.

4.6.3 Conclusions

Overall, MIT Museum staff found the Report Component to be potentially useful, but were careful to take into account that this data would not reflect all visitors' behaviors and therefore should not impact certain types of decisions. They had thoughtful ideas about how else data could be used in this system, indicating interest in the possibilities provided to them by such a system.

Chapter 5

Results

In this chapter I will address several big-picture conclusions drawn from the experiment results. First I will discuss beacon technology as a whole, then I will compare the Eddystone-UID and Eddystone-URL versions of the system, and finally I will explore the potential of beacons in museum contexts.

5.1 On Beacon Technology

My experience working on this thesis has confirmed my belief that BLE, beacon technology, and microlocation applications will continue to grow and reach more consumers in the near future. Bluetooth Low Energy has paved the way for the Internet of Things by creating a resource friendly avenue for communications between nearby devices. Beacon technology will continue to benefit from competition between different manufacturers; in just the few months between when I purchased my first small development set and and the several packages of beacons required for the MIT Museum experiment, the price of Estimote beacons dropped by a third. However, the beacon environment will benefit from the open projects being established, like Eddystone and the Physical Web, which will keep a well-supported community of developers collaborating to build exciting new applications. The biggest barrier I encountered that must be overcome for these applications to succeed in the real world is connection reliability issues. As the hardware in both smartphone and beacons continues to improve, I have no doubt these connections will be dependable enough to support consumer applications in the near future.

5.2 On Eddystone-UID vs URL

After running live experiments with Eddystone-UID and Eddystone-URL protocols each for a week, I see benefits and drawbacks to either system. Currently, the Physical Web functionality of Chrome is hidden away behind many settings, and has a higher barrier to entry to set up. This isn't to mention the fact that it lacks support on Apple's Safari mobile browser, the default browser used by the vast majority of iPhone users. Using a dedicated app with Eddystone-UID beacons eliminates much of the setup process for users, but is more complicated and time consuming to configure the attachments for each beacon. Additionally, "app fatigue" is a current trend among users: people are tired of having to download yet another app for each venue they visit. The Physical Web avoids this problem for Chrome users; once it is enabled on a user's device, Eddystone-URLs from any venue will automatically appear as silent notifications to open in Chrome without further effort. The Physical Web on Chrome for iOS had the additional benefit of making very reliable connections, whereas both frame types suffered on Android devices. One possible remedy for this in a dedicated Android app would be to attempt lower latency foreground scanning instead of lower power background scanning, however, this also forgoes the benefits of notifications when the app is not in use.

Overall, my assessment is that the Physical Web shows the most potential for widespread use and excellent user experience in the future. I started making conversation with one visitor during the first week to see if he'd be interested in participating in the experiment, when he became distracted by a notification on his phone. He made to apologize, but then, realizing the notification was for the nearest beacon of the exact Physical Web project I was pitching, exclaimed, "Oh! That's what this is - how cool!" This goes to show how seamless and delightful the Physical Web experience can be, once the pains of setup are past. I believe that if support for the project extends to other major browsers, and if the setup in Chrome can be streamlined and better advertised, enough users could get on board that the Eddystone-URL protocol gains real traction. Until then, however, I believe the dedicated app solution is the simpler way to reach the most consumers, and will therefore be the protocol of choice for businesses and institutions in the immediate future.

5.3 On Museum Application

When faced with such low participation numbers in the visitor phases of the experiment, I began to question whether beacon technology in museums was something visitors really wanted. However, after reviewing survey responses and discussing results with the MIT Museum staff, it is clear that there is place for beacons in museums - they will simply only appeal to a subset of visitors. Those visitors that did try out the technology reported an enjoyable and valuable experience, despite some technical hiccups with BLE connections. To quote one staff member, “Visitors are a very diverse group. Clearly, those who were interested and able to interact with the beacon technology enjoyed it and found the content useful and informative.” In order to help beacons reach a wide enough audience, they would need to be advertised more apparently throughout the museum. The three-application approach to configuring beacons, enabling visitors to interact with them, and gathering useful data was a successful design, with MIT Museum staff and visitors alike benefitting from the additional information.

I surveyed MIT Museum staff at the conclusion of the experiment to seek their opinions on the technology as a whole, and found myself largely in agreement with their assessments. They unanimously agreed that beacons would be a valuable addition to museums and that the MIT Museum should continue considering the technology, but were neutral on whether the MIT Museum should incorporate them at full scale in the near future. They agreed that users enjoyed and valued the beacon experience, but had mixed opinions on whether it was straightforward for visitors to interact with them. They reported overall having learned useful information via the experiment.

Chapter 6

Future Work

In order for this system such as this to be successful at scale in the future, there are several aspects which could be improved: some at the level of this project and some in larger beacon ecosystem.

6.1 Improvements to Eddystone-UID Version

In the current implementation of the Eddystone-UID system, visitors use a dedicated Android app as the Interaction Component with beacons at the museum. The app was developed exclusively for Android due to my personal background as an Android developer, the ease of integration with Google's APIs, and the time restrictions of a one-year thesis. However, for such a system to be fully adopted by a museum, other platforms - most notably iOS - would need to be supported. Similarly, the Configuration Component I developed currently only exists as an Android app. For this tool, however, I believe a solution better than added platform support could exist: eliminating its necessity altogether. The configuration process for newly acquired beacons already requires using software created by the beacon manufacturer to adjust settings such as transmitting power and communication protocol. Next, when using the Eddystone-UID protocol with beacon attachments, an additional tool must be used to first register the beacons with Google and then add or modify attachments. The number of tools and steps makes this process at best unwieldy, and at worst inhibiting, for museum staff. Ideally, Google and beacon manufacturers could formulate a solution that enables the manufacturers' software to incorporate the registration and attachment steps, such that end users could complete the entire configuration process within a single user-friendly tool.

6.2 Improvements to Eddystone-URL Version

As mentioned in Section 5.2, the future of the Physical Web based system largely depends on the adoption of the technology. However, with Google's mobile versions of Chrome as the most popular browsers to support the Physical Web, some changes could increase the number of users engaging with the feature. On both iOS and Android, Physical Web pages will never even appear to users without them first undergoing a series of steps to activate the functionality. Users must have the knowledge and initiative to seek out the feature - and instructions - for themselves if they are interested. Even then, especially on Android, the setup requires many steps to ensure all of the necessary settings are enabled. This is understandable in that a feature utilizing the user's location should absolutely require their expressed permission; however, I would encourage a simplification of the process and consideration of an active notification informing users of the Physical Web feature's existence.

6.3 Extensions to the Interaction Component

As discussed in Section 4.4.4, there were many additional features considered to include in the Interaction Component. Some were deemed unnecessary or undesirable by the MIT Museum staff for their purposes, but could suit another museum's goals. Others didn't make the cut due to the limited scale of testing or the time constraints of the project, but would be advantageous to include in a full-scale version. Adding any of these features could improve the system according to a given museum's vision.

6.4 Improvements to the Report Component

The current implementation of the Report Component displays charts with several different metrics of visitor interactions with beacons (see Section 3.6). This tool could be improved by enabling it to display a chart representing the results of any specific query entered by a user. The user would have to be aware of the underlying database structure in order to format such a query, but would gain the flexibility of automatically creating a chart for any metric they desire. Were a museum to add additional features to their app as mentioned in the previous section, they could also add logs to track those new interactions and, with an improved Report Component, query metrics that don't even exist in the current tool.

6.5 Alternative Museum Considerations

While discussing issues with the reliability of BLE connections with some visitors' smartphones, museum staff suggested a way to circumvent that problem: provide reliable, museum-owned devices for visitors to check out and use instead of encouraging them to use their own, untested devices. This idea merits further consideration. If a specific device could be identified that was both affordable and located BLE signals consistently, museums could employ a system similar to current audio tours where visitors borrow a device to guide them through the museum. This would drastically reduce issues with struggling to find beacons, as well as eliminate the obstacles involved with visitors setting up their own devices. More research would be required to test different devices and determine whether such a solution would be financially feasible.

Chapter 7

Contributions

I made three main contributions in this Master's thesis.

1. I conceived a three-component system of applications to enable both consumers and museums to benefit from beacon interactions by reporting aggregate usage data.

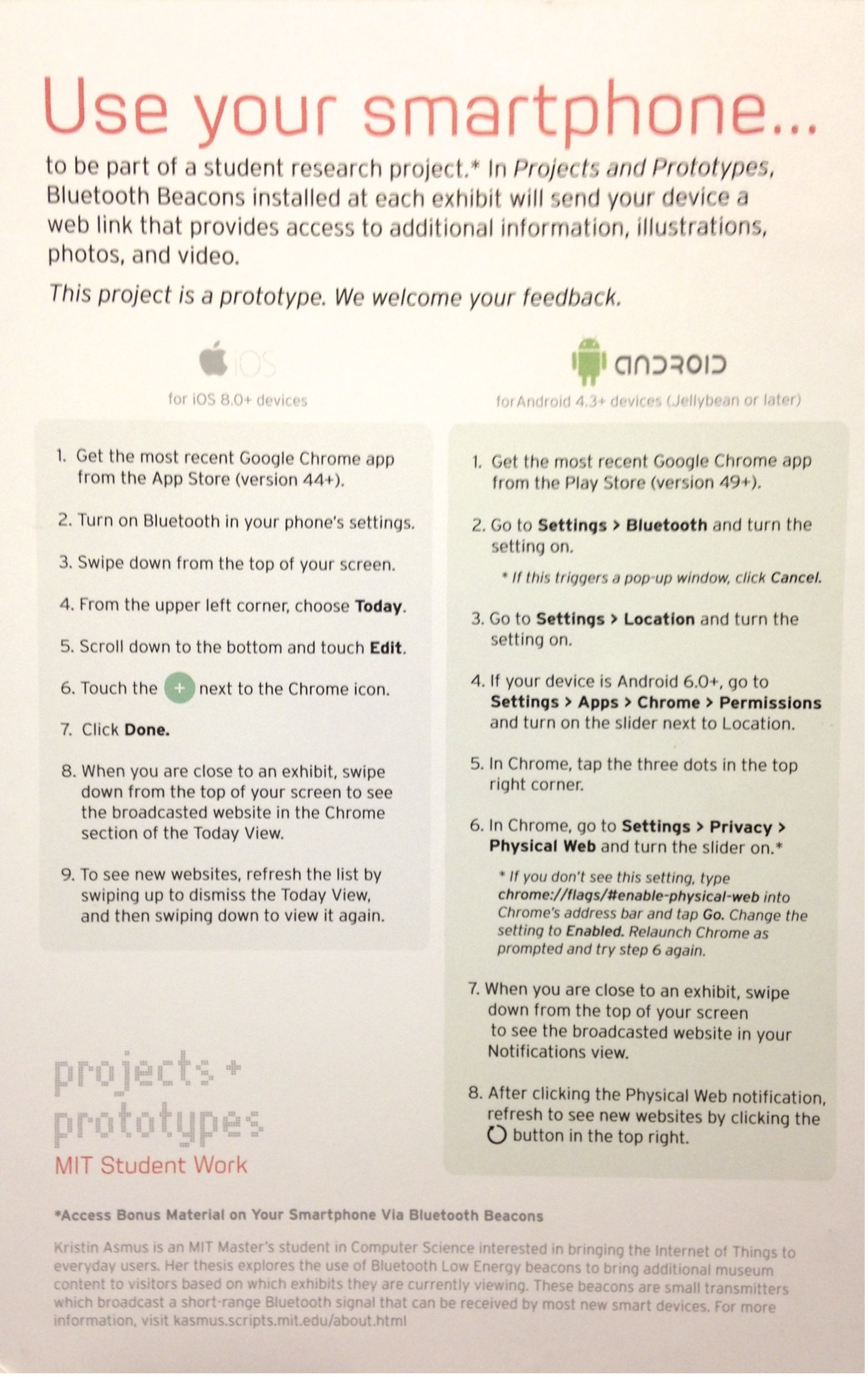
2. I developed functional implementations of this system and applied them in a real-world museum environment. I gathered both quantitative and qualitative data from real users of each application to assess the effectiveness of this experiment. This evaluation included a comparison of two versions of the system: one utilizing Eddystone-UID protocol and a dedicated app; the other, the Eddystone-URL protocol and a mobile browser via the Physical Web.

3. I presented my conclusions on the feasibility of this application of beacons at large scale, and explored what progress is necessary to make beacons a more pertinent technology in the near future.

Appendix A

Experiment Materials

Figure A-1: The instructions poster placed at either end of the Projects and Prototypes exhibit hall during Week One of the experiment, when beacons were broadcasting the Eddystone-URL frame type to Physical Web browsers.



Use your smartphone...

to be part of a student research project.* In *Projects and Prototypes,* Bluetooth Beacons installed at each exhibit will send your device a web link that provides access to additional information, illustrations, photos, and video.

This project is a prototype. We welcome your feedback.

iOS

for iOS 8.0+ devices

1. Get the most recent Google Chrome app from the App Store (version 44+).
2. Turn on Bluetooth in your phone's settings.
3. Swipe down from the top of your screen.
4. From the upper left corner, choose **Today**.
5. Scroll down to the bottom and touch **Edit**.
6. Touch the + next to the Chrome icon.
7. Click **Done.**
8. When you are close to an exhibit, swipe down from the top of your screen to see the broadcasted website in the Chrome section of the Today View.
9. To see new websites, refresh the list by swiping up to dismiss the Today View, and then swiping down to view it again.

android

forAndroid 4.3+ devices (Jellybean or later)

1. Get the most recent Google Chrome app from the Play Store (version 49+).
2. Go to **Settings > Bluetooth** and turn the setting on.

   ** If this triggers a pop-up window, click **Cancel.***
3. Go to **Settings > Location** and turn the setting on.
4. If your device is Android 6.0+, go to **Settings > Apps > Chrome > Permissions** and turn on the slider next to Location.
5. In Chrome, tap the three dots in the top right corner.
6. In Chrome, go to **Settings > Privacy > Physical Web** and turn the slider on.*

   ** If you don't see this setting, type **chrome://flags/#enable-physical-web** into Chrome's address bar and tap **Go.** Change the setting to **Enabled.** Relaunch Chrome as prompted and try step 6 again.*
7. When you are close to an exhibit, swipe down from the top of your screen to see the broadcasted website in your Notifications view.
8. After clicking the Physical Web notification, refresh to see new websites by clicking the ↻ button in the top right.

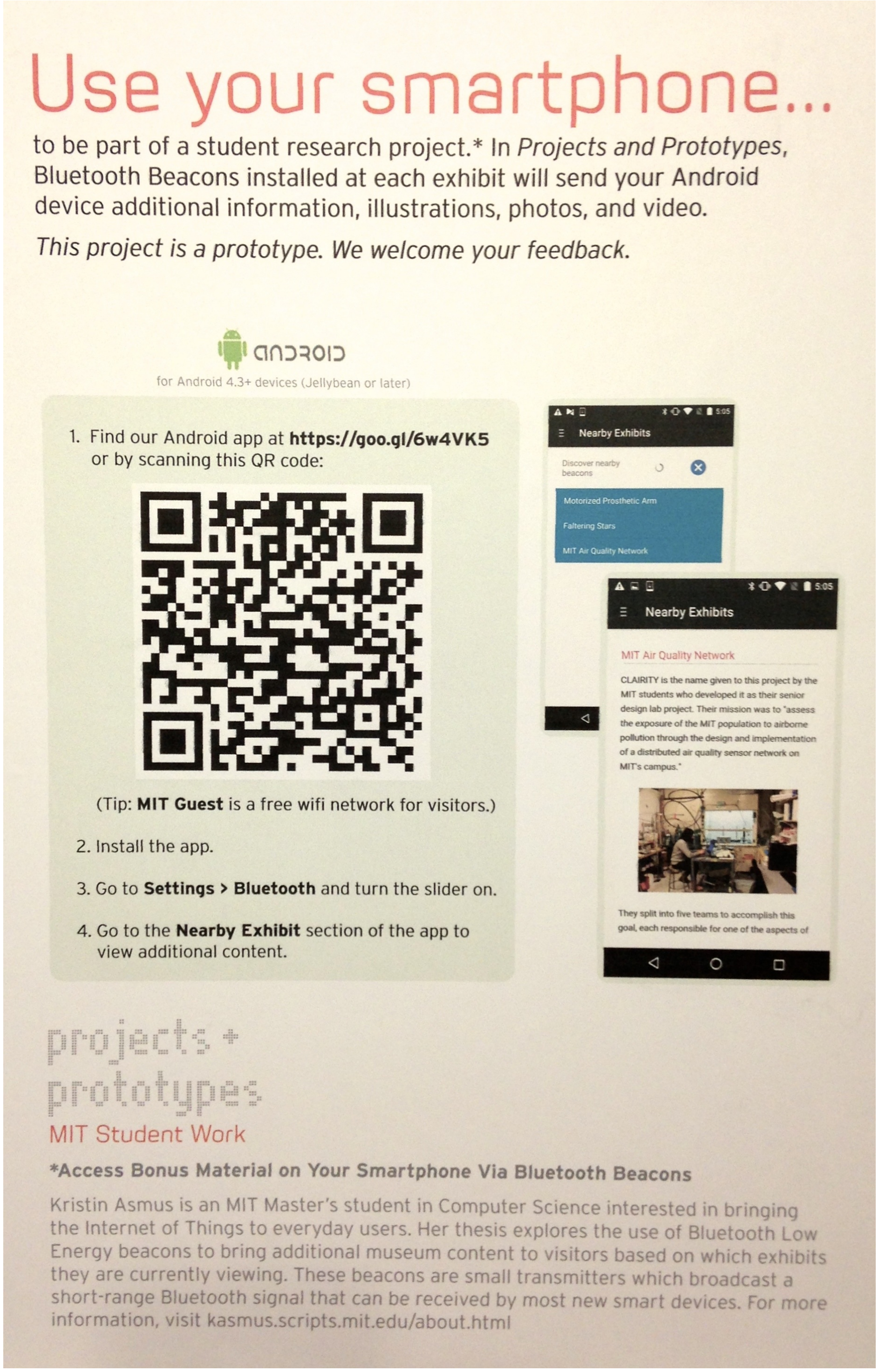
projects + prototypes

MIT Student Work

***Access Bonus Material on Your Smartphone Via Bluetooth Beacons**

Kristin Asmus is an MIT Master's student in Computer Science interested in bringing the Internet of Things to everyday users. Her thesis explores the use of Bluetooth Low Energy beacons to bring additional museum content to visitors based on which exhibits they are currently viewing. These beacons are small transmitters which broadcast a short-range Bluetooth signal that can be received by most new smart devices. For more information, visit kasmus.scripts.mit.edu/about.html

Figure A-2: The instructions poster placed at either end of the Projects and Prototypes exhibit hall during Week Two of the experiment, when beacons were broadcasting the Eddystone-UID frame type to a dedicated app.



Use your smartphone...

to be part of a student research project.* In *Projects and Prototypes*, Bluetooth Beacons installed at each exhibit will send your Android device additional information, illustrations, photos, and video.

This project is a prototype. We welcome your feedback.

for Android 4.3+ devices (Jellybean or later)

1. Find our Android app at **https://goo.gl/6w4VK5** or by scanning this QR code:

   (Tip: **MIT Guest** is a free wifi network for visitors.)

2. Install the app.
3. Go to **Settings > Bluetooth** and turn the slider on.
4. Go to the **Nearby Exhibit** section of the app to view additional content.

Motorized Prosthetic Arm

Faltering Stars

MIT Air Quality Network

MIT Air Quality Network

CLAIRITY is the name given to this project by the MIT students who developed it as their senior design lab project. Their mission was to "assess the exposure of the MIT population to airborne pollution through the design and implementation of a distributed air quality sensor network on MIT's campus."

They split into five teams to accomplish this goal, each responsible for one of the aspects of

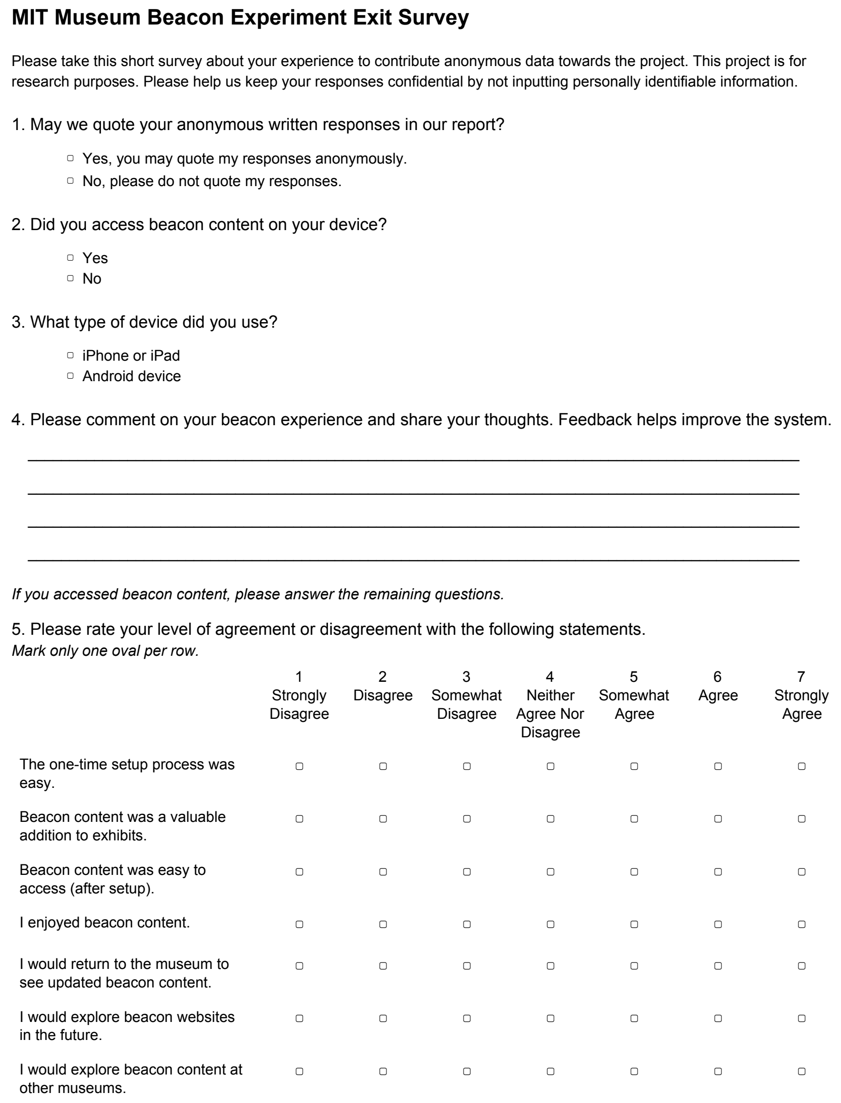
projects + prototypes

MIT Student Work

***Access Bonus Material on Your Smartphone Via Bluetooth Beacons**

Kristin Asmus is an MIT Master's student in Computer Science interested in bringing the Internet of Things to everyday users. Her thesis explores the use of Bluetooth Low Energy beacons to bring additional museum content to visitors based on which exhibits they are currently viewing. These beacons are small transmitters which broadcast a short-range Bluetooth signal that can be received by most new smart devices. For more information, visit kasmus.scripts.mit.edu/about.html

Figure A-3: The survey offered to MIT Museum visitors participating in the experiment by interacting with beacon content on their smartphones.



MIT Museum Beacon Experiment Exit Survey

Please take this short survey about your experience to contribute anonymous data towards the project. This project is for research purposes. Please help us keep your responses confidential by not inputting personally identifiable information.

1. May we quote your anonymous written responses in our report?

   □ Yes, you may quote my responses anonymously.
   □ No, please do not quote my responses.

2. Did you access beacon content on your device?

   □ Yes
   □ No

3. What type of device did you use?

   □ iPhone or iPad
   □ Android device

4. Please comment on your beacon experience and share your thoughts. Feedback helps improve the system.

____

____

____

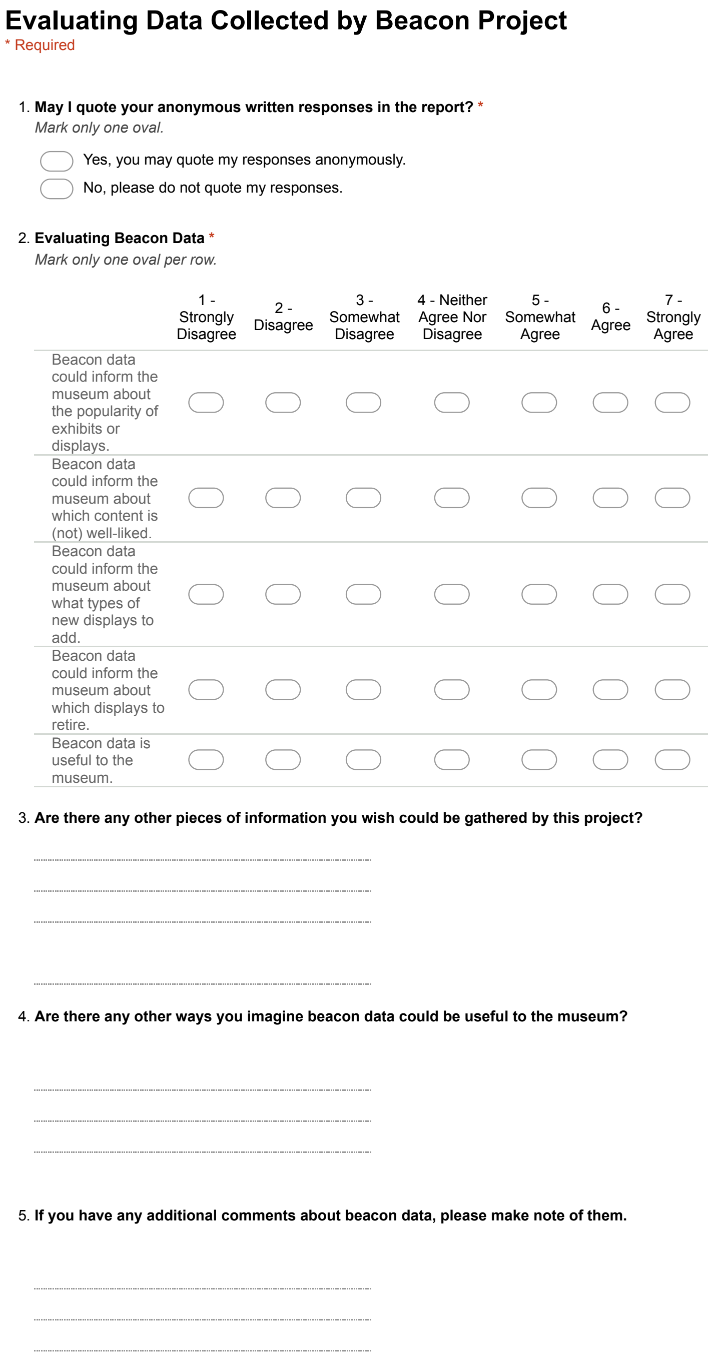
____

If you accessed beacon content, please answer the remaining questions.

5. Please rate your level of agreement or disagreement with the following statements.
Mark only one oval per row.

	1 Strongly Disagree	2 Disagree	3 Somewhat Disagree	4 Neither Agree Nor Disagree	5 Somewhat Agree	6 Agree	7 Strongly Agree
The one-time setup process was easy.	□	□	□	□	□	□	□
Beacon content was a valuable addition to exhibits.	□	□	□	□	□	□	□
Beacon content was easy to access (after setup).	□	□	□	□	□	□	□
I enjoyed beacon content.	□	□	□	□	□	□	□
I would return to the museum to see updated beacon content.	□	□	□	□	□	□	□
I would explore beacon websites in the future.	□	□	□	□	□	□	□
I would explore beacon content at other museums.	□	□	□	□	□	□	□

Figure A-4: The survey given to MIT Museum staff after presenting visualizations via the Report Component of the aggregate visitor interaction data gathered during the experiment.



Evaluating Data Collected by Beacon Project

* Required

1. **May I quote your anonymous written responses in the report?** *
   Mark only one oval.

   ○ Yes, you may quote my responses anonymously.
   ○ No, please do not quote my responses.

2. **Evaluating Beacon Data** *
   Mark only one oval per row.

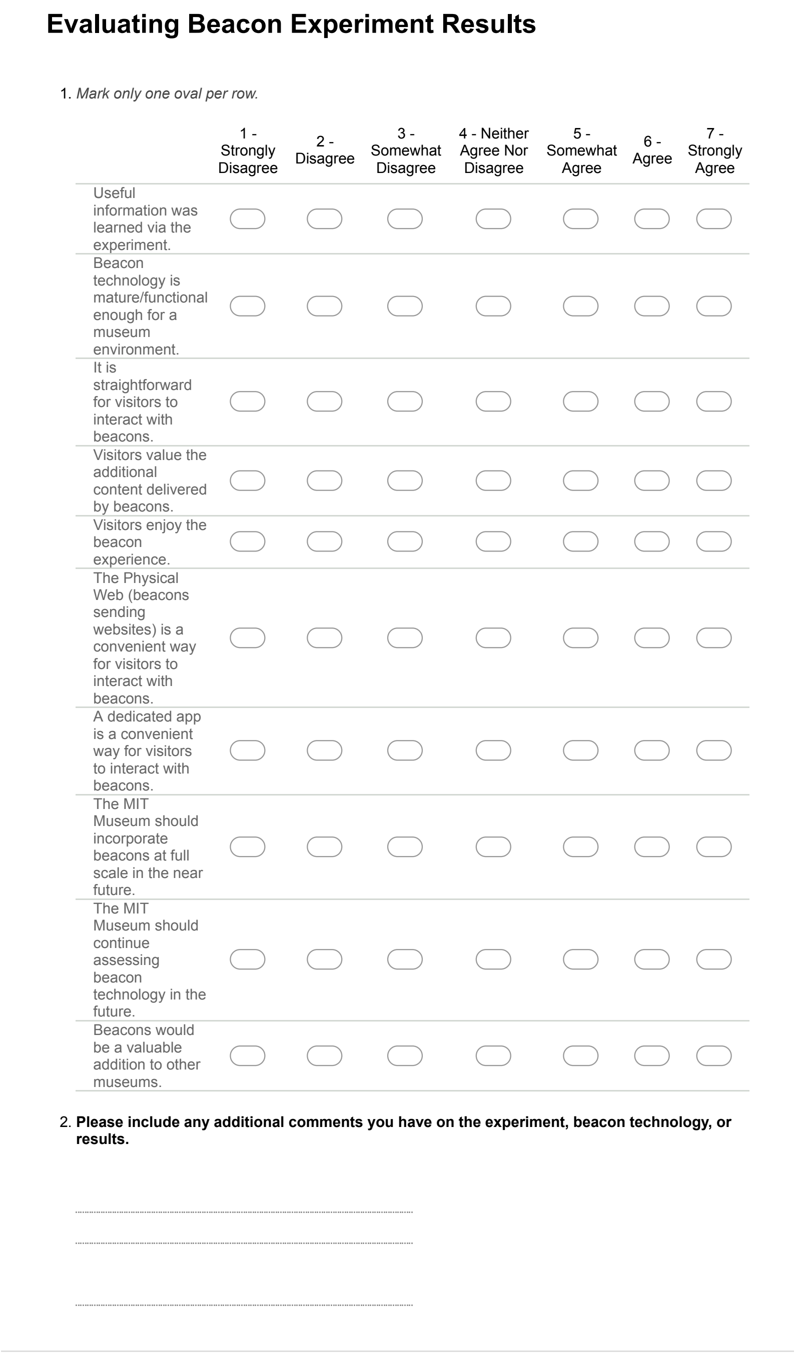
	1 - Strongly Disagree	2 - Disagree	3 - Somewhat Disagree	4 - Neither Agree Nor Disagree	5 - Somewhat Agree	6 - Agree	7 - Strongly Agree
Beacon data could inform the museum about the popularity of exhibits or displays.	○	○	○	○	○	○	○
Beacon data could inform the museum about which content is (not) well-liked.	○	○	○	○	○	○	○
Beacon data could inform the museum about what types of new displays to add.	○	○	○	○	○	○	○
Beacon data could inform the museum about which displays to retire.	○	○	○	○	○	○	○
Beacon data is useful to the museum.	○	○	○	○	○	○	○

3. **Are there any other pieces of information you wish could be gathered by this project?**

4. **Are there any other ways you imagine beacon data could be useful to the museum?**

5. **If you have any additional comments about beacon data, please make note of them.**

Figure A-5: The survey presented to MIT Museum staff at the conclusion of the experiment to evaluate their thoughts on different aspects of beacon technology, visitors' valuation of it, and the experiment as a whole.



Evaluating Beacon Experiment Results

1. *Mark only one oval per row.*

	1 - Strongly Disagree	2 - Disagree	3 - Somewhat Disagree	4 - Neither Agree Nor Disagree	5 - Somewhat Agree	6 - Agree	7 - Strongly Agree
Useful information was learned via the experiment.	○	○	○	○	○	○	○
Beacon technology is mature/functional enough for a museum environment.	○	○	○	○	○	○	○
It is straightforward for visitors to interact with beacons.	○	○	○	○	○	○	○
Visitors value the additional content delivered by beacons.	○	○	○	○	○	○	○
Visitors enjoy the beacon experience.	○	○	○	○	○	○	○
The Physical Web (beacons sending websites) is a convenient way for visitors to interact with beacons.	○	○	○	○	○	○	○
A dedicated app is a convenient way for visitors to interact with beacons.	○	○	○	○	○	○	○
The MIT Museum should incorporate beacons at full scale in the near future.	○	○	○	○	○	○	○
The MIT Museum should continue assessing beacon technology in the future.	○	○	○	○	○	○	○
Beacons would be a valuable addition to other museums.	○	○	○	○	○	○	○

2. **Please include any additional comments you have on the experiment, beacon technology, or results.**

Appendix B

Protocols, APIs, and Samples

The following appendix provides more information about the various resources powering the system described in Chapter 3. The first section links to the specifications for the Eddystone frame types used by beacons to transmit data. The second describes the various APIs and Samples used to create the apps.

B.1 Eddystone Protocols

Google published a guide on their Eddystone format here:
https://developers.google.com/beacons/eddystone.

The full protocol specification is published on Github (since Eddystone is an open format):
https://github.com/google/eddystone/blob/master/protocol-specification.md.

B.1.1 Eddystone-UID

Eddystone-UID is the frame type featured in the first version of the system with a dedicated app. It broadcasts a unique ID, which smartphone apps can look up via Google's APIs discussed below to find the attached data. The specification for the Eddystone frame is included in the Eddystone Github project here: https://github.com/google/eddystone/tree/master/eddystone-uid.

B.1.2 Eddystone-URL

The Eddystone-URL frame type was used in the second version of the system. It sends a short URL which can be received by mobile browsers supporting the Physical Web project. The Eddystone-URL specification is also published as part of the Eddystone Github project:
https://github.com/google/eddystone/tree/master/eddystone-url. The Physical Web is also an open project; for more information, resources, and open-sourced examples, see
https://google.github.io/physical-web/.

B.2 APIs and Samples

The following resources were utilized to implement the three components of the Eddystone-UID version of the system described in Chapter 3. Google's Proximity Beacon API and associated sample app were used for the Configuration Component. The Interaction Component was based on the Nearby Background Beacons sample project, which uses Google's Nearby Messages API.

The Interaction Component and Report Component use Firebase's Javascript library to store and retrieve visitor interaction data. Finally, the Chart.js library was used to draw the visualizations displayed in the Report Component.

B.2.1 Proximity Beacon API

Google's Proximity Beacon API enables management of beacon attachments and monitoring beacon health via a cloud service. Google provides guides for using the API at https://developers.google.com/beacons/proximity/guides, and the complete REST API documentation at https://developers.google.com/beacons/proximity/reference/rest/.

B.2.1.1 Beacon Platform Sample

Google's open-source Beacon Platform app for iOS and Android serves as an example of the Proximity Beacon API's use. It shows how to scan for nearby beacons, register them with Google's Beacon Registry, and add or delete data attachments that will be associated with that beacon's ID in the registry. The Android app from this project was extended for the Configuration Component of the system in this thesis. The project's code can be viewed in full on Github: https://github.com/google/beacon-platform.

B.2.2 Nearby Messages API

Google's Nearby Messages API provides a framework for publishing or subscribing to receive Bluetooth messages or other communications between nearby devices. A subset of this API can be used to receive incoming BLE messages from beacons. Google's published guides, including one detailing how to subscribe to beacon messages, can be found here:
https://developers.google.com/nearby/messages/overview.

The full Android API is documented at
https://developers.google.com/android/reference/com/google/android/gms/nearby/messages/package-summary.

B.2.2.1 Nearby Background Beacons Sample

Google has open-sourced several projects on Github demonstrating different capabilities of the Nearby Messages API, but the one most useful for developing this thesis was the Nearby Background Beacons project:
github.com/googlesamples/android-nearby/tree/master/messages/NearbyBackgroundBeacons.

This project highlights code for subscribing to receive BLE broadcasts sent by beacons, and was incorporated into the Interaction Component of the system.

B.2.3 Firebase

Firebase's Realtime Database is a cloud-based NoSQL database storing data as JSON. It provides convenient SDKs and API for storing and accessing data across multiple platforms including iOS, Android, and web. The Javascript SDK enabled web content viewed by visitors in the Interaction Component to store logs of activity, and enabled the Report Component to fetch that data to visualize. Documentation and guides are online at https://firebase.google.com/docs/database/.

B.2.4 Chart.js

Chart.js is an open-source API for visualizing data, used in creating the Report Component. The open-source project is located on Github: https://github.com/chartjs/Chart.js. Full documentation for the API can be found at http://www.chartjs.org/docs/.

Bibliography

[1] *Low Energy*, 2016. [Online]. Available: `http://www.bluetooth.com/what-is-bluetooth-technology/bluetooth-technology-basics/low-energy`.

[2] W. Borowicz, *How Do Beacons Work? The Physics of Beacon Tech*, 2015. [Online]. Available: `http://blog.estimote.com/post/106913675010/how-do-beacons-work-the-physics-of-beacon-tech`.

[3] *What Is iBeacon? A Guide to iBeacons*, 2014. [Online]. Available: `http://www.ibeacon.com/what-is-ibeacon-a-guide-to-beacons/`.

[4] W. Borowicz, *Apps and Projects with Estimote*, 2016. [Online]. Available: `https://community.estimote.com/hc/en-us/articles/207295257-Apps-and-projects-with-Estimote`.

[5] *iBeacon for Developers*, 2016. [Online]. Available: `https://developer.apple.com/ibeacon/`.

[6] R. Amadeo, *Meet Google's "Eddystone" – a flexible, open source iBeacon fighter*, 2015. [Online]. Available: `http://arstechnica.com/gadgets/2015/07/meet-googles-eddystone-a-flexible-open-source-ibeacon-fighter/`.

[7] *What Is Eddystone?* 2015. [Online]. Available: `http://developer.estimote.com/eddystone/`.

[8] *Beacons*, 2015. [Online]. Available: `https://developers.google.com/beacons/?hl=en`.

[9] Google, *Eddystone*, 2015. [Online]. Available: `https://github.com/google/eddystone`.

[10] ——, *Eddystone-UID*, 2016. [Online]. Available: `https://github.com/google/eddystone/tree/master/eddystone-uid`.

[11] *Register Beacons*, 2016. [Online]. Available: `https://developers.google.com/beacons/proximity/register`.

[12] *Proximity Beacon API*, 2015. [Online]. Available: `https://developers.google.com/beacons/proximity/guides`.

[13] *Nearby*, 2015. [Online]. Available: `https://developers.google.com/nearby/`.

[14] ——, *Eddystone-URL*, 2016. [Online]. Available: `https://github.com/google/eddystone/tree/master/eddystone-url`.

[15] ——, *Physical Web*, 2016. [Online]. Available: `https://google.github.io/physical-web/`.

[16] *Beacon browsers give you access to Physical Web content nearby*, 2016. [Online]. Available: `https://www.phy.net/physical-web-browsers`.

[17] *Beacons In Museums*, 2015. [Online]. Available: `http://www.beaconstac.com/museum`.

[18] *Culture, Museums, & Art*, 2015. [Online]. Available: `https://glimwormbeacons.com/about/culture-museums-art/`.

[19] *Using Beacons Effectively in Museums*, 2014. [Online]. Available: `http://innovation.mubaloo.com/news/using-beacons-effectively-museums/`.

[20] V. Doljenkova, *Beacons: Exploring Location-Based Technology in Museums*, 2015. [Online]. Available: http://www.metmuseum.org/about-the-museum/museum-departments/office-of-the-director/digital-media-department/digital-underground/2015/beacons.

[21] S. Bernstein, *The Realities of Installing iBeacon To Scale*, 2015. [Online]. Available: https://www.brooklynmuseum.org/community/blogosphere/2015/02/04/the-realities-of-installing-ibeacon-to-scale/.

[22] M. Cannell, *Museums Turn to Technology to Boost Attendance by Millennials*, 2105. [Online]. Available: http://www.nytimes.com/2015/03/19/arts/artsspecial/museums-turn-to-technology-to-boost-attendance-by-millennials.html.

[23] Cuseum, *About Cuseum*, 2015. [Online]. Available: http://cuseum.com/#about-cuseum.

[24] *Get Started with Beacons*, 2016. [Online]. Available: https://developers.google.com/beacons/get-started.

[25] Google, *Android Beacon Service Demo App*, 2015. [Online]. Available: https://github.com/google/beacon-platform/tree/master/samples/android.

[26] *Estimote Development Kit*, 2016. [Online]. Available: http://estimote.com/#jump-to-products.

[27] Estimote, *Estimote on Google Play*, 2016. [Online]. Available: https://play.google.com/store/apps/details?id=com.estimote.apps.main&hl=en.

[28] *Estimote Cloud*, 2016. [Online]. Available: https://cloud.estimote.com/.

[29] Beaconinside, *Beacon Manager on Google Play*, 2016. [Online]. Available: https://play.google.com/store/apps/details?id=com.beaconinside.android.

[30] Bluevision, *Specification Sheet*, 2016. [Online]. Available: http://bluvision.com/wp-content/uploads/2015/12/Specs-iBEEK1.6.pdf.

[31] *Google Proximity Beacon API*, 2016. [Online]. Available: https://developers.google.com/beacons/proximity/reference/rest/.

[32] Google, *Beacon Tools on Google Play*, 2016. [Online]. Available: https://play.google.com/store/apps/details?id=com.google.android.apps.location.beacon.beacontools.

[33] *Proximity Beacon API: Retrieve Attachments*, 2016. [Online]. Available: https://developers.google.com/beacons/proximity/proximity-retrieve.

[34] *Nearby Messages API: Get Beacon Messages*, 2016. [Online]. Available: https://developers.google.com/nearby/messages/android/get-beacon-messages.

[35] *Nearby Messages API: Get Started*, 2016. [Online]. Available: https://developers.google.com/nearby/messages/android/get-started.

[36] ——, *Nearby Background Beacons*, 2016. [Online]. Available: https://github.com/googlesamples/android-nearby/tree/master/messages/NearbyBackgroundBeacons.

[37] *Firebase Realtime Database*, 2016. [Online]. Available: https://firebase.google.com/docs/database/.

[38] *Firebase: Pricing*, 2016. [Online]. Available: https://firebase.google.com/pricing/.

[39] *Firebase: Save Data on the Web*, 2016. [Online]. Available: https://firebase.google.com/docs/database/web/save-data.

[40] I. Kantor, *The "Same Origin" Security Policy*, 2011. [Online]. Available: http://javascript.info/tutorial/same-origin-security-policy.

[41] D. Kochin, *Detect Click into Iframe using JavaScript*, 2015. [Online]. Available: http://stackoverflow.com/questions/2381336/detect-click-into-iframe-using-javascript/32138108#32138108.

[42] Google, *Physical Web: Can't the user be tracked?* 2016. [Online]. Available: https://github.com/google/physical-web/blob/master/documentation/introduction.md#7-cant-the-user-be-tracked.

[43] *Firebase: Retrieve Data on the Web*, 2016. [Online]. Available: https://firebase.google.com/docs/database/web/retrieve-data.

[44] *Chart.js*, 2016. [Online]. Available: http://www.chartjs.org/.

[45] *The Physical Web has arrived for Chrome on Android*, 2016. [Online]. Available: https://www.phy.net/blog/physical-web-arrived-chrome-android/.

[46] *Configure your beacons*, 2016. [Online]. Available: https://developers.google.com/beacons/get-started{\#}2-configure-your-beacons.

[47] *MIT SIPB Web Script Service*, 2013. [Online]. Available: https://scripts.mit.edu/web/.

[48] *Projects and Prototypes: MIT Student Work*, 2016. [Online]. Available: http://mitmuseum.mit.edu/exhibition/projects-and-prototypes-mit-student-work.

[49] *Cannot Save Transmit Power of Nearables*, 2016. [Online]. Available: https://forums.estimote.com/t/cannot-save-transmit-power-of-nearables/3642.