Using Beacons for Attendance Tracking

D. Deugo¹

¹The School of Computer Science, Carleton University, Ottawa, Ontario, Canada

Abstract – Students experiencing difficulty with their studies don’t often identify themselves as students in need of support. The Science Student Success Centre at Carleton University actively seeks out these science students to give them the help they require. One indicator of need is a student’s grades. Another indicator is attendance in class. In this paper, we describe how we use beacon technology to help track attendance. While our existing online card swiping and QR Code scanning attendance tracking system was suitable for smaller class sizes, our beacon-based attendance tracking system is now suitable for classes and events with over 1000 attendees.

Keywords: Attendance, Tracking, Android, Beacon

1 Introduction

In [1, 2, 3], we described different variants of our attendance tracking system and its usage in classes to examine its performance and usability. Our motivation for building the system came from The Science Student Success Centre (SSSC) at Carleton University. The SSSC collaborates with other on-campus departments to help develop a student’s academic and professional skills. The SSSC also takes an active role in helping science students struggling in computer science, math, and science courses with the following mission:

• Increase the engagement and retention of students in the Faculty of Science
• Foster the growth and achievements of high performing students
• Identify and support students who may be experiencing difficulty with their studies
• Inform students of professional and academic development opportunities
• Aid in the recruitment of outstanding students to the Faculty of Science

It was our belief that identifying students with less than 60% attendance records and having the SSSC talk with and mentor these students would help with their overall success in the faculty of Science. We had already been successful mentoring students with grades less than 60% by discussing with them the issues they face with their courses and suggesting possible techniques and actions they can take to overcome the issues.

Taking attendance using paper and pen was one approach we could have used, but we knew it was slow and prone to errors. In [2, 3] we developed a student card scanning based system for tracking attendance in classes containing less than 50 students. However, we needed to expand that capacity. In [1] we added the ability for students to generate custom QR codes that were scanned by users or instructors using a custom application that ran on a smartphone or tablet. The scanned information was sent back to our system’s server and a student’s attendance at a class or event was recorded. The approach doubled our ability to check students into a class within a required timeframe. However, we needed to expand that capacity by at least an order of magnitude to handle the potential for very large classes or event attendance tracking.

1.1 Goals

Reiterating our goals and objectives from [3]. Our main goal is to provide a fast and efficient attendance tracking system. In addition, the system must work in any and all classrooms at Carleton, including its electronic classrooms – those with computers and projectors – and those classrooms containing no computers. A further goal is to provide a system that requires minimum hardware, and can be maintained at minimum cost. Our new goal is to be able to track attendance of classes or events with over 1000 students or participants.

1.2 Objectives

To meet the goals, we have the following objectives:

• Use easily found, inexpensive hardware for the system.
• Make use of mobile devices to help with the attendance tracking process but not specifically require the use of student owned mobile devices
• Use open source software to minimize development and maintenance costs
• Installation should be fast and simple
• Attendance reports must provide end users with the greatest flexibility for manipulating the collected attendance data
• Support both Mac and PC platforms, and Android and iOS mobile devices
1.3 Outline

In section 2, we survey beacon technology and beacon-based attendance tracking systems. Section 3 describes our system and how it uses beacons in classes to take attendance. Section 4 describes our results. Section 5 provides our conclusion and reviews our system’s feature set.

2 Background

There are numerous beacon vendors providing many different styles of beacons for different needs. A 2015 report by Aislabs [5] examined 26 different vendors, including the beacons we use from Radius Networks [6]. Beacons are generally very small devices that are often powered by a battery such as the 3-volt CR2032 battery. Some beacons, like the ones from Radius Networks, can be powered from a wall outlet or a USB outlet. Beacons emit a Bluetooth Low Energy (BLE) signal that can be detected by a BLE enabled mobile device. Beacons have a transmit power and advertising interval. The former dictates how beacon’s signal can travel (up to 50 meters) and the latter indicates how often the signal is sent, e.g. 100 to 500 milliseconds. Both the power and advertising setting impact the battery life of a beacon. Depending on one’s settings, and a beacon’s battery, the battery life can be as short as 0.6 of a month to upwards of 56.3 months.

Each hardware beacon can support one or more of the following beacon specifications:

- iBeacon [7]
- AltBeacon [8]
- EddyStone once called UriBeacon [9]

[10] provides an overview of the first two specifications. In summary the iBeacon and AltBeacon specifications are similar. In short both support beacons transmitting a universally unique identifier (UUID) and a few bytes of configurable data specific to the device that is static once configured. Eddystone beacons are configured with arbitrary blobs of data that are served back as messages to Android and iOS apps. One can update attachments remotely. One of the important features of the Eddystone specification is the following:

Eddystone is an open beacon format from Google that works with Android and iOS. Eddystone includes a number of broadcast frame types, suitable for different types of deployment. [11]

As hardware and specifications stabilize, so too do the number of uses for beacon technology. [12] describes over 100 different use cases for examples of iBeacon technology. Retail, Hospitality, Tourism, Education, Healthcare, Entertainment, and the Travel industries are all prime areas for use of beacons. In addition, the Corporate, Automotive, Real Estate and Advertising industries are areas that can benefit from the technology. In this paper, we are interest in using beacons to help track attendance at events or classes.

Several different systems are using beacons for event tracking. Špica International, a company based out of Slovenia, has created an application named All Hours [4] that enables employees to clock in at work with their smartphones. Within a range of about 20 meters, employees use their smartphones as identification once an Estimote Bluetooth beacon is detected by their smartphone application. The application then registers an employee as in the premises. Their backend system provides various reports on when employees arrive and leave the premises and their system can also enforce a company’s working time policies by permitting a company to set its work hour rules.

SessionRader is a U.S. company that offers secure and verified local beacon check-ins at any location. The benefits they suggest [13] of their system are the following:

- No Badge Required
- Passive + Interactive
- No Scanning or Lines
- Integrates with SmartWatch
- Instant Automatic Attendance
- Wireless Check In

In their system, attendees are either running the associate iOS mobile application and are informed when they are in range of a beacon allowing them to check in, or the detection of the beacon is integrated with the device’s notification support, alerting an attendee of the possibility of checking in. A backend system with application support enables users to view attendance data in real time.

DoubleDutch, a company with offices in the U.S. Netherlands, Hong Kong, and the U.K., use beacon technology to support capturing who is attending a session using their Head Count application. Their application runs on both iOS and Android devices that support Bluetooth 4.0 or greater. Provided an iBeacon-enabled device is available for each session, attendees can conform their check in when in range of the beacon.

In contrast to using beacons, [14] uses a combination of QR codes that each attendee have, typically on their conference badge, and conference support staff having iOS and Android devices running their custom application to scan and track attendees. Their system provides real time reports and registration details to registered users. The disadvantage of this system is that it focuses in the conference domain, where the number of events and attendees are usually small. A mid-size university can have hundreds of classes and over 25,000 students to track on a daily bases. The sizes of the education domain make typical costing models using this approach with attendees and sessions expensive.
The above approaches suggest several important features for any attendance tracking system:

- Real time reports
- Fast Scanning
- Attendee identification
- Mobile device application
- Attendee id and registration

All of these features are outlined in our approach, described in the next section.

3 Approach

As indicated in [1, 2, 3], our system users are instructors or people that are taking attendance in classes or at different events. They are provided with a user id, which is their email address, and a unique password. After logging into the system using the Event Tracker website, shown in Figure 1, users can download their attendance real time reports for any class or event.

In [2, 3], users attached a USB card reader to users computers and swiped student cards in order to track attendance. We also had a mobile application that students could use on their devices to identify themselves as attending the class or event. Their location was taken from their current GPS location of the device. In [1] we added the ability for students to generated custom QR codes that were scanned by users or instructors using a custom application that ran on a smartphone or tablet. The scanned information was sent back to our system’s server and a person’s attendance at a class or event was record.

3.1 Beacons

Our system use beacons from Radius Networks [6]. Available from the Radius Network Store at $21.00 U.S. for one,

“RadBecon USB is a fully standalone Bluetooth Smart proximity beacon using iBeacon, AltBeacon and Eddystone™ technology, implemented in a tiny USB package.”

The RadBecon provides proximity services for iOS, Android and other mobile environments and is powered by a standard USB port. This means that these beacons can be plugged into a USB AC adaptor, car adaptor, or a computer with a USB port.

3.2 Attendance Setup

Tracking attendance using our system involves the following components: User, Event, Beacon, and mobile application.

The system’s administrator assigns a user’s email address, password, organization, and the events or classes they are able to track. The system administrator has access to the backend of the system and is outside the scope of this paper. After entering their email address and password into our web interface, a user can look at their attendance reports for their associated events. Reports are in the form of an Excel spreadsheet. Each entry in the Excel spreadsheet provides information on an individual checking in or out of an event. This information includes the following:

- First Name
- Last Name
- ID Number
- Email Address
- Event Name
- Organization
- Time
- Direction (In/Out)
- Latitude
- Longitude
The Device entry can be Android, iOS or web, identifying the three methods of data entry available for our system. This paper discusses the Android approach using our Android mobile application. In the case of Android entry, the accuracy is always 1 and the latitude, longitude and altitude are 0.

To track attendance, a user must ensure that the beacon that is registered to their event is available. Beacons are configured with three values: UUID, Major, and Minor. A UUID contains 32 hexadecimal digits, split into 5 groups, separated by dashes. Major and Minor values are numbers assigned to the beacon, in order to identify them with greater accuracy than using the UUID alone. They are unsigned integer values between 1 and 65535. This configuration is done by the system administrator and outside the scope of this paper. Once configured, a user brings the beacon to the corresponding event and plugs it in, or the beacon is permanently installed and powered.

The final step in the setup is that each attendee must install our EventTracker application on their Android device. The only device requirement for the application is that it supports BLE, also called Bluetooth Smart or Version 4.0. BLE is designed to provide significantly lower power consumption and is the Bluetooth specification that our beacons use.

3.3 Mobile Application

We had the goal for our EventTracker mobile application that it was easy to use. To meet that goal we wanted to minimize the number of touches an attendee makes in order to indicate their attendance at an event.

After downloading and starting up our application, the attendee, and from their perspective a user, is presented with the view shown in Figure 2. It is in this view the attendees enters the information that they wish recorded when their attendance is taken, namely their first and last names, their email address, id and language preference. It is worth noting that an attendee does not indicate the events they are going to attend. Event identification is done on a discovery bases. After entering the information the Save Info button must be clicked in order to commit the information to the device. If not done, the attendee will see this screen and need to fill in their information each time the application starts up. However, at anytime the attendee can always come back to this screen by selecting the User tab to change and then save the information again.

By selecting the menu button (three dots in the top right corner of the view) a user can configure the backend server IP address the application will use. For most attendees, they will never need to change this value. However, there are scenarios where this value will need to change dynamically and we have configured this into our application. It is just a simple matter of selecting the menu, entering the new address and selecting the OK button. Figure 3 shows the corresponding IP Address configuration dialog.

Once the attendee information has been entered and the server IP address has been configure, an attendee will see the screen shown in Figure 4 when the application starts up. This is the screen where attendees initiate the scanning for events that they can register their attendance. The first step for an attendee is to click the Scan For Events button. This initiates a search by the application for any beacons that are within range of the device. As a result of the scan, the application will gather each detected beacon’s UUID and its Major and Minor values. These values are then sent to the server to gather information about the events that are associated with the values. This information is returned to the application and presented to the attendee. Figure 5 shows an example of the information that is presented to an attendee. In this example, three different events were indicated from the detected beacon(s).

![Figure 2: Attendee Information](image-url)

The last step in the process for an attendee is to select which event they are attending and then click the Attend Event button. Once done, the application sends the attendee’s information back to the server to indicate the attendee’s presence at the corresponding event. Once complete an attendee see the status line indicating “Success!”, as shown in Figure 5.
3.4 Report Generation

After logging in to the Event Tracker web front end, as shown in Figure 1, an end user can generate up-to-the-minute Excel reports of the events they are tracking. By selecting Reports in the left hand menu or selecting Login to Reports in the home screen a user is taken to the Reports Screen shown in Figure 6. All of a user’s events are listed in this screen and by selecting one and clicking the download button, an Excel spreadsheet of results is downloaded for the user’s review, as shown in Figure 7.
Results

In [3] our system used the swipe of a student card to indicate a student’s attendance in class. The issue we faced with this version of the system was that it required us to register each student once. As reported, it took an average of 50.34 seconds to register a student. In small class sizes, this one-off operation was acceptable, but in classes of over 50 students, this time requirement in the initial class(es) was too high. The other issues that we faced was that the time required to swipe in students for a class took anywhere between 8 and 16 minutes for class sizes of 50, either before or after class. Given the swipe timings found, we decided that the use of student cards just didn’t work in classes of 50 or more.

In [1] we added the ability for students to generated custom QR codes that are scanned by users or instructors using a custom application that ran on a smartphone or tablet. The scanned information was then sent back to our system’s server and a student’s attendance at a class or event was record. One improvement with this approach was that students only needed to generate their custom QR code once, and this activity could either be done offline or in parallel with other students, using our QR code generation website. A second improvement with this approach was that it took an average 5.3 seconds to scan an individual’s QR code. The implications was that in the first ten minutes before an event we could scan a little over 100 attendees, doubling the size of event or class our application could support over [3].

Our goal was to be able to handle events with over 1000 attendees and with our current approach and custom mobile application we have been able to reach that goal. This was achieved by distributing the workload. Now every attendee has a device that can record attendance, rather than having to go through a single device. Moreover, the number of attendance recording actions permitted is only limited by the number of simultaneous incoming connections our server can support. Our server is based on Tomcat [15]. As part of Tomcat’s default configuration,

\[
\text{The default maximum queue length for incoming connection requests when all possible request processing threads are in use is 100.}
\]

This number is only the default. As reported in [16], it is possible to have over 13000 concurrent connections in Tomcat by manipulating its Connector configuration and the Java heap size Tomcat is running on. This number, combined with the fact that the maximum time to execute a single connection request is 40 ms, means that we can support attendees in the 1000s, which meets our goal.

Conclusion

We have created a mobile application and server-based system that be used for track ing attendance for events or classes with well over 1000 people attending. Moreover, if attendees attend different events, they do not need register with subsequent events, as their personal information is only recorded once in their mobile application.

Another issue we have resolved is a technique to ensure that an attendee is actually at an event when being tracked. Having the mobile application detect beacons at the event and then only allowing an attendee to register for only the events associated with the beacons enables this feature.

Our working system has the following features. Ultimately, attendance tracking data is stored on our server. We support the creation of a user with corresponding events they wish to track attendance. Users can login to our system with their user id and password in order to retrieve real time Excel reports of who has been tracked at one of their events. We have a custom mobile device application that permits attendees to enter their personal information, detect beacons and register attendance for events associated with those beacons. We use beacons that support standard protocols requiring only to be plugged in for power.

We believe we have a system that is ready for product development and will be moving in that direction in the near future.

References


[16] Cool, Tomcat is able to handle more than 13,000 concurrent connections,