

Indoor Positioning Services and Location-Based Recommendations

This chapter on indoor positioning systems (or IPS) is a case study on a location-based service developed at the University of Illinois Urbana-Champaign. The case involves location-based recommendation through mobile technology paired with commercially available Bluetooth low energy (BLE) beacons. Mobile technologies are a well-suited entry point into Internet of Things (IoT) hardware since mobile phones and tablets by way of their Bluetooth radios offer a way to both communicate with IoT tools and leverage existing mobile services for providing new services. The purpose of this chapter is to provide an example implementation of IoT technologies that responds to a real service need within an academic library setting. Portions of this example will apply to public libraries as well, since larger and more complex book stacks may require additional directional support to items regardless of library type. The example presented here goes beyond simple directional support, however, since it shows a way to recommend relevant digital resources and popular print items from within the print book stacks. As a final preface to this case, note that some portions of this chapter are technical, but no more technical than required in order to explain foundational interactions among Bluetooth beacons, mobile devices, web services, and the custom databases that provide recommendations.

With beacon technology, real-time turn-by-turn directions and real-time recommendations in the book stacks can be provided to a user's mobile device. With the infrastructure and research trajectory developed for an augmented reality experiment, researchers from the undergraduate library at the University of Illinois undertook an experimental project to incorporate Estimote beacons into the undergraduate library book stacks so that students new to the environment

could see the location of their mobile device within the library building through an interactive map on their phone, providing directional support to items and discovery of like items with location-based recommendations.¹ This chapter demonstrates the distributed computing processes and workflows necessary to integrate beacons into collections-based wayfinding and explores the key components for the recommendation algorithm used for “topic spaces” in collections. Distributed systems are common in mobile app development since usually there is a need to serve data from a remote API (Application Program Interface) to a mobile device, like an Android phone. If a remote API is used, this usually comprises at least one remote virtual server communicating with the mobile phone. Sometimes these remote APIs also interact with database data.

These components are listed here as a way of illustrating that components of the distributed system can become multipart. Before moving on to more detail, a comment about the environment within which this API operates: the technology stack used for this case study is heavily Java-based. The Android app is written in Java, the APIs were developed using Java, and specifically the Spring framework was used to generate JSON that the phone would utilize in displaying recommendations.

The experimental location-based recommendation service utilized in this case is grounded in the advantages of collocation that support information discovery and are supplemented with existing ILS data—for example, total circulation of a particular item. The research and development leading up to a popular algorithm was the result of several innovative idea-generating projects that included focus groups with students, user studies of mobile applications in the

library, and interdisciplinary app design competition on the University of Illinois campus.²

Wayfinder App Functionality

This case study uses the Wayfinder module from the Minrva app to illustrate IoT functionality. The Wayfinder module is a part of a larger mobile app platform that utilizes modules to separate different functions of the app for library services. As an example, the Minrva platform includes a catalog search, journal search, account modules for easy renewal, and several branch location custom modules. As an example of a custom module, several library units circulate technology (i.e., loanable technology), such as USB mics and graphic calculators, and since the pool of available devices changes often throughout the day, the technology module was developed so that users of the Minrva app can simply tap the technology module to see a list of currently available technology. From our user studies and follow-on focus groups, the library was able to learn that students are interested in knowing what the popularly circulating technology devices are.³ The request for popularity filters led us to sort the technology items with the most popular items at the top of the list. We use the technology module as an example to illustrate how recommendations are now commonplace as a feature request to add to library services. There are several other custom modules, which can be viewed on the Minrva project catalog list, but the key point to remember is that the custom modules are usually location-specific in order to offer value-added services.

Minrva Project
<http://minrvaproject.org>

Minrva Technology Module
http://minrvaproject.org/modules_tech.php

Minrva Catalog
<http://minrvaproject.org/catalog.php>

Wayfinder is a custom module. Each module is dependent on a unique bibliographic identifier (like the unique key for objects in a database) and is not directly coupled with the larger system of modules. Modules communicate only indirectly.⁴ After a user searches for an item in the catalog search module, they can tap the Wayfinder module to be directed to the location of an item. During user testing of recent Minrva modules, researchers uncovered a user preference of being able to locate their mobile device on the map, not just the location of the selected item.⁵

The focus of our previous study inquired as to how users wished to receive recommendations from stacks-based browsing. Thus, the goals and objectives for using Bluetooth beacons in the book stacks included the following: (a) locating a user's device in the library book stacks, (b) giving a user a recommendation for popularly circulating items, and (c) showing them relevant e-content based on their current location.

The mockup in figure 2.1 demonstrates features of a mobile app that relies on the Estimote beacons to infer its location.

The mockup on the right of the figure demonstrates the recommendations component. There are highly circulating books in this location, but without the assistance of this app the user isn't readily able to discern this, since the information exists only in the integrated library system's reporting database. Researchers designed the business logic in the server so that the service can run queries related to the user location in the book stacks. The popularity query returns print books that are highly circulating in the subject area of this user.

As an additional IoT service, the query by the user's location is returning a variety of e-content that is relevant to the subject area. The e-content that may be suggested includes e-books, online journals, and journal databases that are relevant to the location. It is possible to generate this recommendation by using the subject metadata associated with the book stack range that the user is nearest. These e-resources are not typically brought into a user's navigation of the physical space in the library.

The undergraduate library at the University of Illinois measures 50.8 meters by 56.7 meters and was built underground. The planners for the campus did not want to disturb the oldest operating soil research plot in the country, and they feared a building with a shadow could interfere with the experiment. New undergraduate students must navigate to the lower level of the building in order to locate print items they are interested in. The new and unique building can be intimidating to navigate, and users might suffer from library anxiety. An indoor positioning service that utilizes beacons in order to support navigation of a new, unfamiliar environment is a welcome addition to undergraduate spaces.

The IoT technology that makes locating a user device in the building possible is Bluetooth proximity beacons. The case study at the Illinois site utilized the beacons from a company called Estimote. Fifty-two Estimote proximity beacons were placed above the tiles in the undergrad library.⁶ While IoT technologies are going to vary, the specifics of the Estimote beacons are helpful in illustrating the connectedness of many smaller Bluetooth beacon devices within a native mobile app.

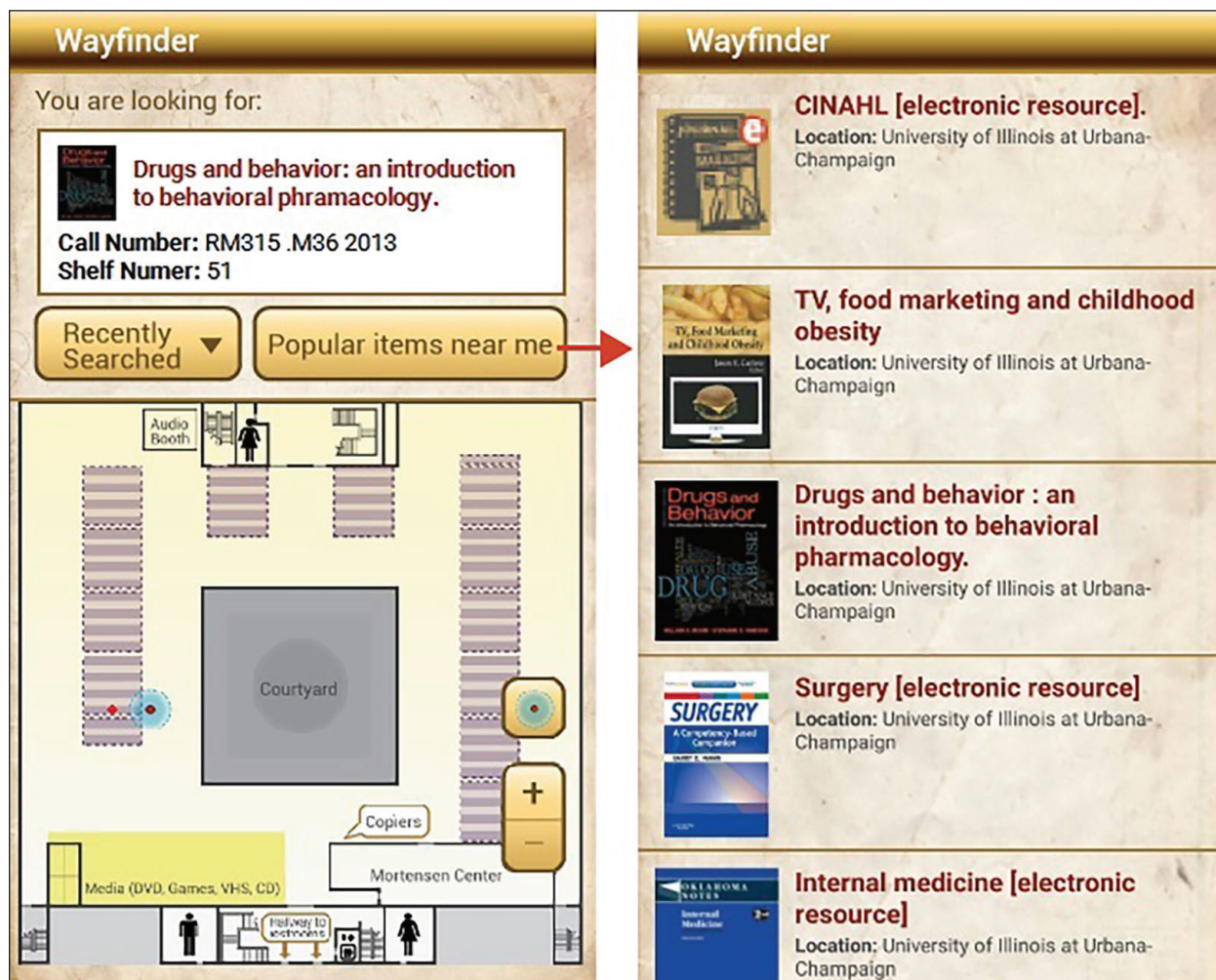


Figure 2.1
The dot with a circle around it is the location of the user's device. The dot within the book stacks indicates the location of the item that the user searched for from the catalog module.

Estimote proximity beacons are comprised of a 2.4 GHz radio using Bluetooth 4.0 Smart, also known as BLE. The Bluetooth beacon does not produce a continuous signal, but rather blinks on and off to alert other devices of its presence. The developer documentation also utilizes an easily understandable analogy for how Estimote Bluetooth beacons function—"You can think about the beacon as a small lighthouse. But instead of light, it uses radio waves, and instead of ships, it alerts smartphones of its presence."⁷ While the Bluetooth beacons from Estimote do feature small, lightweight computers to broadcast a signal, there is not a direct data collection component from the beacons into the Estimote Cloud server. In order to adjust beacon settings—like frequency strength—the user needs to make the changes in the web interface (logging in to the Estimote Cloud) and then sign in to the Estimote app to configure the beacons once the phone with the new configuration is in range of the Bluetooth

beacons. Then the new settings are applied. Both of these steps require authentication as the owner of these beacons. The ownership of beacons is set by Estimote when the order is placed, and it utilizes the customer's e-mail address for ownership at the time the beacons are ordered. We have verified that ownership can be transferred from one e-mail account to another if needed after the beacons have been ordered. This is especially helpful in case one e-mail address is used for a shipping and receiving department and another e-mail address is used by developers for authentication and provisioning purposes. According to the Estimote developer documentation, the Estimote proximity beacons actually have a real-world range of forty to fifty meters.⁸

Once the beacons have been installed in a library building, there is still the step of integrating beacons into mobile apps.

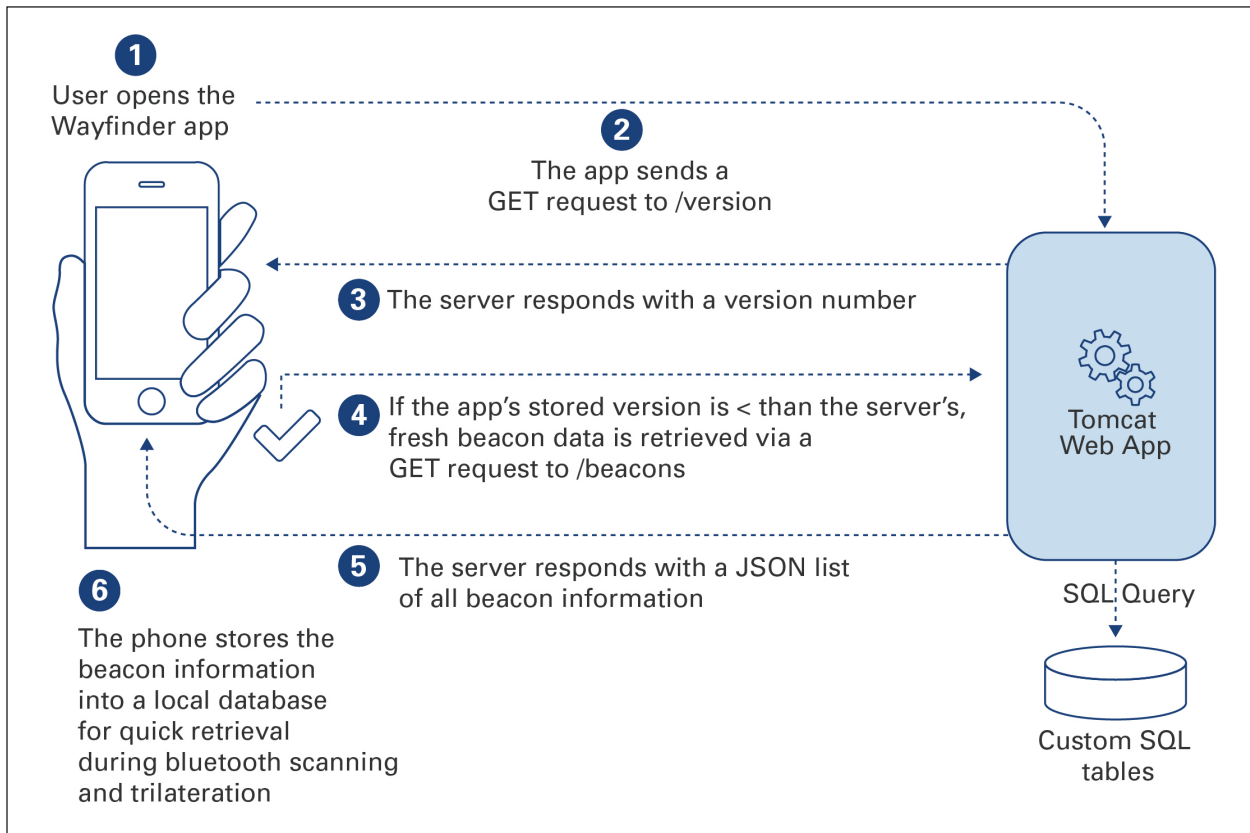


Figure 2.2
An illustration of the data flow from the database of the beacon locations in the grid to the user's smartphone.

Integrating Beacons into Mobile Apps

In our case study, we installed beacons in the ceiling of the library, above ceiling tiles, and adhered to a grid layout of our own design, which divided the library into a 16 by 18 grid. Each beacon was recorded as an x and y coordinate on that grid; device locations were then stored in a small database. The database is a Microsoft SQL: Structured Query Language table that corresponds to the grid system developed over the building's library map. Microsoft SQL is a common relational database tool used in digital library applications. The Wayfinder module loads after a user taps the Wayfinder icon in the larger Minrva app. Upon startup, it checks a web service (in this case, the Minrva Location API) to ensure it has the most recent table of beacon locations (figure 2.2).

Once beacon locations are downloaded to the user's device, there are at least two possible ways for the phone to infer where it is in the building. These are outlined as a way of illustrating design choices and the relative advantages and drawbacks for both. One way developers can make the phone aware of its location is by using the Estimote Indoor SDK. The Indoor SDK has the advantage of including several

novel noise-cancellation algorithms and may be optimized for the device. As an alternative to the SDK, the research team at Illinois has found that a math library for trilateration has provided more reliable indoor location results than the Indoor SDK for Estimote proximity beacons.⁹ This may be due to the underground location or unique environment of many shelves of books, obstacles that likely cause some unexpected interference for the Estimote Indoor SDK. According to the Estimote developer documentation, Estimote engineering believes that the trilateration method may be good enough in cases when the accuracy can fluctuate up to five meters.¹⁰ The reason Estimote developed its own SDK for indoor locating is that it believes that it is possible with several additional methods to get granularity much more precise. Since we have saturated our undergraduate library stack location with the Estimote beacons, we are achieving a better result than five meters.

For the library directional Wayfinder case, the precision needs are not to the exact location of the book on the bookshelf, but are simply to guide a student to a row of books (the shelf of books). The system recommendations will depend on which shelf the user is nearest when they tap the recommendation button on the app. Therefore, precision beyond several

meters is not necessary. Though not utilized in this case, it should be noted that radio-frequency identification (RFID) type systems will provide a unique area of service capabilities by actually guiding users to exact items. This technology will be explored in more depth in chapter 3, which details alternative approaches to location services in libraries.

Modular APIs

As previously mentioned in the introduction to this chapter, the Minrva app is designed as a modular app. Each component part or module of the application uses a specific RESTful (Representational State Transfer) API. RESTful APIs are designed as concise, specifically formatted data produced by programs to be consumed by other programs accessing data over the web. A list of publicly viewable APIs can be viewed on the Services page of the Minrva Project website.

Minrva Services

<http://minrvaproject.org/services.php>

The API that was developed for the Wayfinder module upgrade provides data based on three sources (see figure 2.3): basic call number layout (custom table we curate), searching and filtering (VuFind), and ranking popularity (CARLI reports server).

Examining figure 2.3, we can see first the database labelled as the CARLI Reports Server refers to the Oracle database that is the backend of the integrated library system (ILS). Researchers currently pull all of the data at once through one dynamic query. For production, a service such as this would necessarily require some amount of caching—for example, not directly connecting to a reporting server to get popularity rankings dynamically. In a production system, the technologies may require an immediate response, and this being the case, researchers recommend pulling the data from a prerun report. Reports that store data that would otherwise need to be dynamically

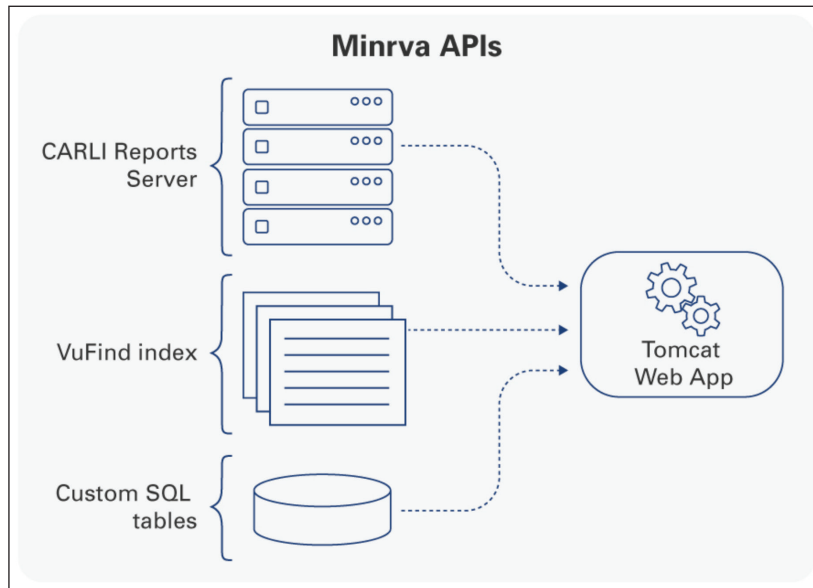


Figure 2.3

Illustration of the databases and data sources that a location-based recommendation service requires.

```
// Get the patron's physical coordinates
PatronLocation location = getLocation(locationInfo);

// Get shelf ranges from Minrva DB, based off of patron location
ArrayList<String> ranges = getRanges(location);

// Get relevant books/ebooks based off of the shelf ranges
BooksDB bookdb = getEBooks(ranges, context);

// Filter and add popular books that are near the patron
List<RecBooksModel> books = bookdb.bookList;

// Filter and add relevant ebooks that would be shelved
// near the user, if the ebook was a physical item
List<RecBooksModel> bookEBook = getPopularBooks(ranges, books);

// Add relevant databases, based off of the patron's location,
// to the list
List<RecBooksModel> bookEBookDB = getDBSuggestions(bookdb.DBList,
bookEBook);
```

Figure 2.4

A demonstration of computing processes that researchers developed in order for the phone to receive the recommendations of print, e-books, and databases.

pulled could run overnight or twice a day as recommendation requirements merit. There may also be value in visualizing these recommendations since their change over time may be of interest to collection developers and system managers alike.

Recommendations Processing

The recommendations processing is the most technical component of the case. In this section, we walk

through the recommendation algorithm. An algorithm, at its most basic and essential parts, is simply a number of functions that represent the steps that the server is taking. These steps are gathering recommendation results defined by a predetermined criterion that will be shown to the user based on the location of their device when they tap the recommendation button (figure 2.4).

In order to process recommendations, the server will filter through functions beginning with the current location. Based off of patron location, we get the shelf ranges from our custom database of all call numbers. Once we know the ranges of a nearby stack, we get relevant books and e-books. We filter for popularity based on that range and add any e-books that would be relevant. Finally, we add any online journal databases that may be relevant to the patron's starting location and serialize this as a JSON response. In order to gather database journal recommendations, the service relies upon an EBSCO article search API. While recommending is not the main purpose of the EBSCO API, the conceptualization of a "recommendation" is considered broadly to encompass those items that are more relevant by arrangement. The API is returning a set of results to a subject query (gathered from the call number). In the case that a single journal title holds most of the results, the service then recommends that journal as being relevant to the subject area. The students still must search within the journal to get articles. This part of the algorithm, at its most basic, promotes the availability of topically relevant journal titles from the EBSCO API. Certainly there are limitations to this type of search since promoting some resources over others may leave out resources that are relevant but not found in the recommendation model.

Testing

Outside of the app, we use Estimote's configuration app to test where beacon sparseness may occur by locating beacons in range.¹¹ With Estimote's configuration app, we can locate areas in the library where beacon reception is sparse and where we might want to add one or more beacons so that there are at least three points to compute—the requirement of trilateration. The more points available to the phone for inferring location, the better the results of computing trilateration.

```

- {
  bibId: "uiu_6548355",
  title: "Encyclopedia of Buddhism",
  thumbnail: "http://secure.syndetics.com/index.aspx?type=xw12&isbn=9780028659107/SC.GIF&client",
  author: "",
  pubYear: "2004",
  location: "University of Illinois at Urbana-Champaign",
  format: "eBook"
},
- {
  bibId: "uiu_7010231",
  title: "Buddhism and transgression : the appropriation of Buddhism in the contemporary West",
  thumbnail: "http://secure.syndetics.com/index.aspx?type=xw12&isbn=9789047441106/SC.GIF&client",
  author: "Konik, Adrian",
  pubYear: "2009",
  location: "University of Illinois at Urbana-Champaign",
  format: "eBook"
},
- {
  bibId: "hat_819122",
  title: "Christianity and philosophy",
  thumbnail: "http://vufind.carli.illinois.edu/vf-uiu/images/1x1.gif",
  author: "Holmes, Arthur Frank",
  pubYear: "1960",
  location: "HathiTrust eResources (public domain)",
  format: "eBook"
}

```

Figure 2.5

The Android app consumes this JSON to create the location-based recommendations view.

Security

Beacon security for the location-based recommendation system described in this chapter will be primarily handled with the Estimote Cloud. The previously mentioned Estimote Cloud is one way for system designers to ensure that only the owner of the Estimote beacons can change settings. In a real-world environment, if beacon settings are changed by a third party acting to harm the library infrastructure, the beacon battery life could be run down by setting the power level of the beacons too high. As a corollary to this problem, the library's IoT service would also be in danger if a third party somehow managed to set all the power levels of the beacons very low or off. In that case, the system would not function. Therefore, the middleware components that make up some of the valuable connectors of the IoT represent a potential security weak point.

From the example we describe in this chapter, a consumer entity is partially responsible for servers that control how the library administers the beacon settings. This consumer-facing service is not directly secured by the library. These types of IT services are often referred to as "shadow IT"—the tradeoff for management is that the control a department might normally have over infrastructure is lost. This is the tradeoff the library makes in order to utilize an off-the-shelf IoT consumer product like Estimote beacons. Without the infrastructure of the Estimote Cloud, Bluetooth devices that make this possible would have to be developed from scratch. And while there are certainly many hobbyists who experiment with Arduino boards with Bluetooth capabilities or small microcontrollers with Bluetooth broadcast affordances, these would not be a great basis for an enterprise or production service provided by the library. This illustrates

a compromise of several pillars of IT, including standardization and control in order to advance the state of the art in services. If these compromises are to be made by IT leaders in the organization, the author proposes a process of value analysis and understanding how employing strategic technology such as IoT would be meeting strategic service objectives.

After taking precautions to securely protect the Estimote beacons to the extent we are able, there are two additional databases that need to be secured in this case. These are custom databases that the library maintains. The Microsoft SQL database of beacon locations in the library requires security authentication to be put in front of the server. This beacon location database provides the grid coordinates of where the beacons are placed in the library. If the server were breached and the database compromised by a malicious third party that gained unauthorized access to the database, then that party could delete the tables of beacon coordinate data on which the system relies to function. The second SQL table, a call number range database, would also be at risk. Though databases may not technically be an IoT component of the stack, these data are fundamental to the proper functioning of the application. Without securing both the call number database and the beacon location database with authentication, the range lookup could potentially fail, and the system could not properly give the user suggestions based on their location in the building. As we noted in our walkthrough of the algorithm, location-based suggestions are themselves derived from subject metadata that is a component of constructing a call number.

Outside of this specific case, there are general themes to IoT security that are worth mentioning and unpacking with more detail than can be afforded in this case. We hinted at middleware, which is a significant part of the IoT. Connecting many small computers together and deriving service innovation requires a middle component that interfaces with devices like beacons and other tools (like a database) to store or retrieve very large amounts of data to provide value. IoT middleware security will be examined in more detail in chapter 4. As we have shown with the mobile app-based example, the IoT is by necessity a system that interacts with personal computing devices, and so a deeper treatment on security and privacy is necessary.

Statistical Analysis, Data Storage, and the Internet of Things

Assessment is an important component of new service development. Without an assessment plan, it can be difficult to know if a service is actually meeting its intended need. Previous studies of collection-based

wayfinding with mobile technologies utilized formative evaluation methods.¹² These formative methods are well suited for new services such as these, and for IoT technologies, which are new and offer many possibilities, it makes sense to gather feedback early in the design phase, with functional prototypes, so that as the functionality develops it can be vetted with real users in a real-world environment. Formative evaluation in previous mobile app studies was qualitative. However, with Estimote beacons and library server infrastructure, new possibilities for quantitative data analysis emerge.

For this case study, there are some potentially interesting and valuable ways to go beyond traditional assessment, into data analytics and also data visualization. Consider the data analytic possibilities derived from subject-specific and title-specific recommendations that are periodically extracted, stored, and then visualized in a dashboard. Each shelf will have recommendations that will change over time based on what is popularly circulating. Also, e-books are added to the collection over time; as the collection grows and is used, recommendations will also fluctuate. The data analytics component would be suited to collecting and describing this change over time and could provide insights into what types of items users are being recommended as popular, and additionally, if there are gaps in the suggestions over areas that perhaps require more e-content to recommend over time. Finally, if these data are appropriately de-anonymized, the analysis of these data could be made into a shareable dashboard.

Another possible data set that could form the basis for analytics includes storing and visualizing the grid locations where users are seeking recommendations as a heat map. This would help collection managers to understand which areas users are most interested in or which areas of the book stacks are being browsed—and as a visualization experiment, it could also showcase other users' parts of the collection that their peers are finding interesting. These heat maps of where users have browsed in the collection could be visualized on library displays. Display walls in particular would be a possible target of this data analytics and visualization.

Case Conclusion

This chapter has outlined an ongoing research project that details how to integrate IoT technologies into a real-world academic library environment. Utilizing Bluetooth beacons paired with a modularly designed mobile app, library researchers were able to locate the user's mobile device in the building and guide that user to items of interest. As an added benefit, system designers have chosen to implement a location-based

recommender that relies on subject classifications in call numbers to provide recommendations based on location.¹³ These recommendations are novel for library systems since they integrate digital content like e-books, e-journals, and journal databases within the book stacks browsing experience.

Look for future updates from the Minrva's Android or iOS app store presence that will include upgrade details of completed functionality for the indoor positioning service. The development group responsible for this service has posted updates to its project portfolio page, so those who are interested in more details can check back to the Prototyping Group website regularly for additional findings that come out after this work is published.¹⁴

Notes

1. Jim Hahn, Ben Ryckman, and Maria Lux, "Topic Space: Rapid Prototyping a Mobile Augmented Reality Recommendation App," *Code4Lib Journal*, no. 30 (October 15, 2015), <http://journal.code4lib.org/articles/10881>.
2. Jim Hahn and Hillary Bussell, "Curricular Use of the iPad 2 by a First-Year Undergraduate Learning Community," chap. 7 in "Rethinking Reference and Instruction with Tablets," *Library Technology Reports* 48, no. 4 (November 2012): 42–47; David Ward, Jim Hahn, and Lori S. Mestre, "Designing Mobile Technology to Enhance Library Space Use: Findings from an Undergraduate Student Competition," *Journal of Learning Spaces* 4, no. 1 (June 2015), <http://libjournal.uncg.edu/jls/article/view/876/812>.
3. Hahn and Bussell, "Curricular Use of the iPad 2."
4. For more information on Minrva's module system, see Jim Hahn and Nathaniel Ryckman, "Modular Mobile Application Design," *Code4Lib Journal*, no. 18 (October 3, 2012), <http://journal.code4lib.org/articles/7336>.
5. Ibid.
6. At the time of this study, Estimote made available "proximity beacons." However, more recently, after we installed these beacons, Estimote is shipping a new generation of Bluetooth beacons, "location beacons," which promise to be "the most robust" Bluetooth location beacons on the market. There is an all-new software developer kit (SDK) that ships with the new hardware. (Estimote, "Launching the Most Robust Location Beacons on the Market," *Reality Matters: The Estimote Team Blog*, February 24, 2016, <http://blog.estimote.com/post/139902664710/launching-the-most-robust-location-beacons-on-the>.) One would expect the upgraded SDK to provide better indoor location support than the generation launched in 2014.
7. Estimote, "Beacon Tech Overview," Developer Docs website, accessed July 25, 2016, <http://developer.estimote.com/>.
8. Estimote, Developer Docs website, accessed July 29, 2016, <http://developer.estimote.com/>.
9. Trilateration is a process of determining a location given distance measurements of circles. More information can be found on this geometric computation in Wikipedia, s.v. "Trilateration," last modified June 1, 2016, <https://en.wikipedia.org/wiki/Trilateration>.
10. Wojtek Borowicz, "How Do Beacons Work? The Physics of Beacon Tech," *Reality Matters: The Estimote Team Blog*, January 2, 2015, <http://blog.estimote.com/post/106913675010/how-do-beacons-work-the-physics-of-beacon-tech>.
11. "Estimote," Apple iTunes Store, last updated July 22, 2016, <https://itunes.apple.com/us/app/estimote/id686915066?mt=8>.
12. Jim Hahn and Alaina Morales, "Rapid Prototyping a Collections-Based Mobile Wayfinding Application," *Journal of Academic Librarianship* 37, no. 5 (September 2011): 16–22.
13. For a conceptual and technical dive into subject recommendations and location-based services in libraries, see Jim Hahn, "Location-Based Recommendation Services in Library Book Stacks," *Reference Services Review* 39, no. 4 (2011): 654–74.
14. For project next steps on this case study, please see the Technology Prototyping Service project page, University of Illinois at Urbana-Champaign University Library, last updated June 8, 2016, <http://sif.library.illinois.edu>, where technology prototyping project for the library are regularly documented.