

Creation of a Library Tour Application for Mobile Equipment using iBeacon Technology

We describe the design, development, and deployment of a library tour application utilizing Bluetooth Low Energy devices known as iBeacons. The tour application will serve as library orientation for incoming students. The students visit stations in the library with mobile equipment running a special tour app. When the app detects a beacon nearby, it automatically plays a video that describes the current location. After the tour, students are assessed according to the defined learning objectives.

Special attention is given to issues encountered during development, deployment, content creation, and testing of this application that depend on functioning hardware, and the necessity of appointing a project manager to limit scope, define priorities, and create an actionable plan for the experiment.

by Jonathan Bradley, Neal Henshaw, Liz McVoy, Amanda French, Keith Gilbertson, Lisa Becksford, and Elisabeth Givens.

Background

QR Code audio tour

At Virginia Tech, guided tours of Newman Library, the main branch, were provided for incoming freshmen. The tours were approximately an hour long, and were led by librarians. As the size of incoming classes increased – now at over 6500 incoming students for the class of 2019 (Freshman Snapshot) – it was recognized that self-guided tours would allow more effective use of librarian time, as well as give each student more flexibility as to when to visit the library for her orientation.

The classroom manager and technology liaison for the library, Neal Henshaw, used instructional design techniques to create a self-guided tour for new students, based on QR code technology and iPad 2 tablets. The original self-guided tour was designed primarily as an audio tour. Students began at the circulation desk, and were shown by circulation staff how to scan a QR code which plays a video that gives an introduction to the tour. The introductory video described the tour, explained procedures for ending the video at each stop and moving to the next stop, and detailed a workaround for problems with unreliable wi-fi. Though the original tour was primarily designed for audio, most of the scan targets on the tour were actually video files. The video was used only occasionally, to help the student identify a tour stop by showing an image of the area, or to demonstrate computer search techniques.

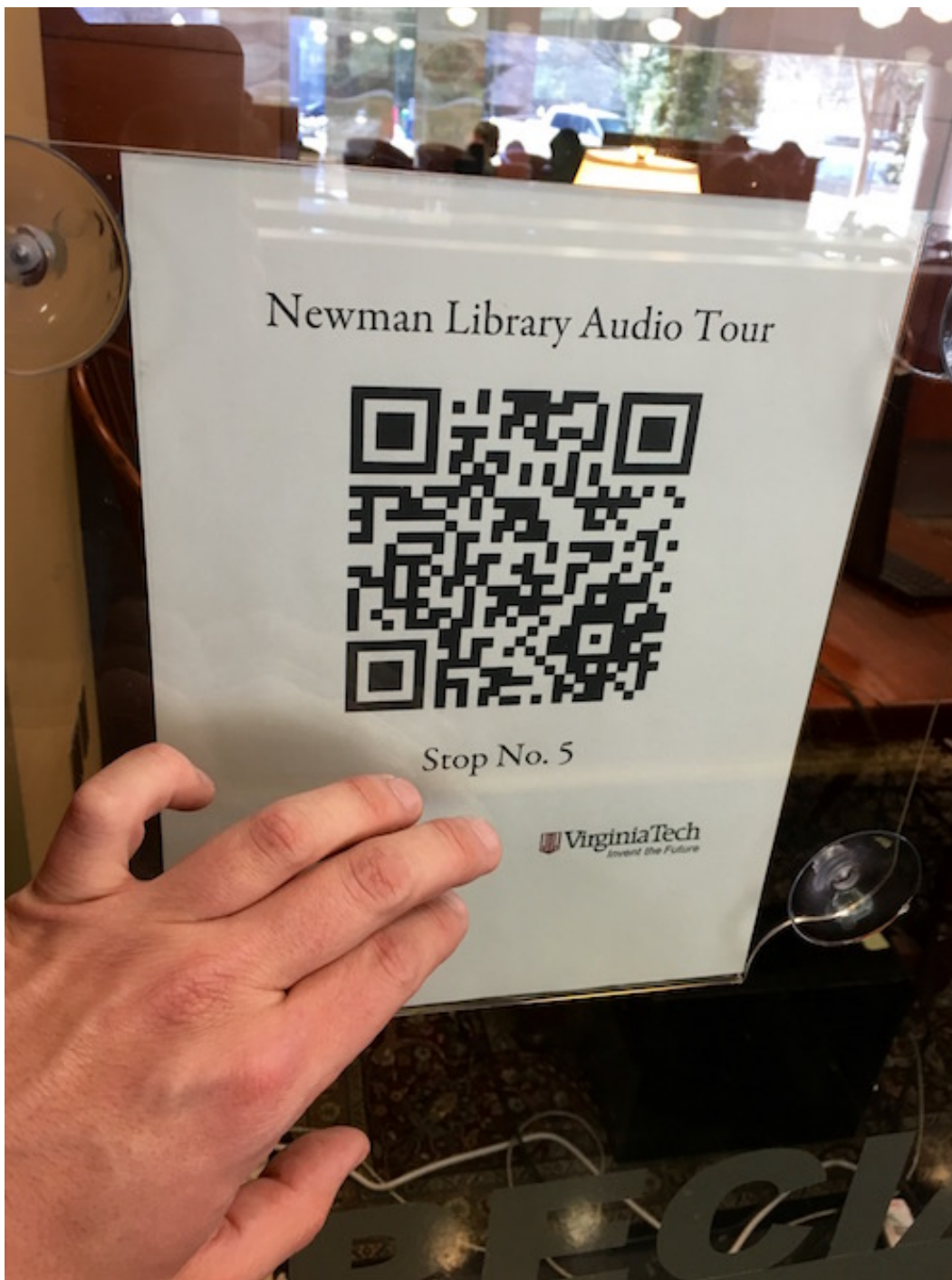


Figure 1. A QR code placard from the current library audio tour

In the original, QR code based self-guided tour, information about the stops was carefully written into a script for each of nine stops. The audio on the tour was narrated directly by Neal, who has a perfect radio voice. The tour introduces the students to quiet and group study areas, circulation and reserves, reference, and collaborative projects with other university groups, including a writing center and a communications lab. Students are also shown basic instructions for accessing the library web site and performing searches using Summon. This self-guided library tour has been required for the standard English course for the Freshman Class, *Introduction to Writing*. Students taking the tour for the purposes of this course are required to take a quiz after completing the tour.

Technology update: mobile app with iBeacons

What's an iBeacon?

An iBeacon is a Bluetooth Low Energy (BLE) device that broadcasts a unique identifier, repeatedly. Mobile applications can be

created to listen for these identifiers, and perform location or context aware activities when the identifier is detected. (iBeacon 2016)

The iBeacon identifiers are set up in three parts: a UUID, a major number, and a minor number. The beacon identifiers are configurable by the beacon owners; the method for doing this is configurable according to the brand.

The most common example given for the use of iBeacons is in retail. The iBeacons can be placed near retail store locations, and the retail store brand's app running on a user device receives a notification when a user is near one of the stores. The app can alert the user to sales or special events at the store, and attempt to entice the user to enter.

The iBeacon aware apps can also detect specific locations within a store, with the use of major and minor numbers. The Apple developer documentation for iBeacon cites an example where all beacons belonging to a retail store with multiple locations are assigned one UUID, while a differing major number is assigned to each location of the store, and a different minor number to each department in the store. (Getting Started ... 2014) The app for the retail store can now perform context-sensitive actions when a person enters each department in each specific store.



Figure 2. Estimote beacons. Photo courtesy of Estimote, Inc.

Updating the tour

We chose to update the tour primarily because this saves students the step of having to fumble with QR codes at each stop. Converting the tour to a native app will also give us the opportunity to improve the experience in the future, through better presentation and assessment of learning content.

This has been an especially challenging project, primarily because it requires time and coordination from several teams, all of whom are busy on several other projects. Over the last few years technology units within the library have been better on completing technology projects within units specifically created for that task. This project has been different because graphics designers, video editors, marketers, script authors, education and learning specialists, and developers have all been needed.

During planning and implementation of this project we have faced library reorganizations, the loss of the initial designer to other employment opportunities, a deficiency of test hardware capable of working with the technology, and the loss of a talented project manager because the project was not within her primary area of responsibility.

We're excited to finally be implementing and deploying this application, and to tell you how we have been able to scrounge resources in order to pull this off.

User Roles

During the design of a technology application, it is sometimes useful to think about the roles of each user. So, let's begin with the primary role, that of the student.

Student

Liqin is a first year student that has just moved to Blacksburg to study computer engineering. Liqin was assigned to take the self-guided tour of the library by an English instructor, but does not understand why this is necessary. During the tour given to high school students and parents wishing to choose a university, it was clearly stated by the tour guide that there's no need to actually go to the library; everything is online. However, the student tour guide could be wrong; he also stated that the library was open 24/7, when in fact it is actually open 24/5, which in real life means that there are only 4 days in the week when the library is truly open 24 hours. Whatever the hours of the library are, Liqin has only one and a half hours between the end of Quidditch and the beginning of Aerial Robotics Club to eat and take the self-guided tour of the library.

In addition, there are some other use cases for minor roles that we had in mind when we designed the app.

Instructor

Dr. Jones is a professor associated with the English department, and teaches four sections of the introductory writing course. Dr. Jones needs to make sure the students know how to use the library for research, and wants to be able to follow up with each student that might be having difficulties understanding how to use library resources.

Learning Services and Education Technologies

Devon is a librarian responsible for helping students learn how to use the library, in both face-to-face and technology based scenarios. As the library undergoes incremental remodeling and reorganization, new spaces and services are introduced each year. This year, a studio of 3D printers has been added to the library. Devon needs a way to update the tour that incoming students are shown so that it explains the 3D printing studio, preferably without waiting for a software developer to become available.

Requirements

Thinking about the three roles in the system helped to determine requirements for the application.

Requirement	Supports Student	Supports Instructor	Supports Library	Not in this version
Load specific video file when near a beacon	X	X	X	
Ask questions at every stop	X	X		
Display important information on screen	X			
Give students the correct answer when they miss a question	X			
Subtitles for the Deaf and Hard of Hearing	X	X		
Content needs to be easy to update			X	
Includes maps	X			
Runs on library-owned equipment			X	
Records pass/fail grade for each student	X	X		
Records name of student and class, and stops visited	X			
Introductory screen	X			
Offer help for wi-fi problems	X			
Students visit stops in order			X	
Additional Languages				X
Run on student-owned equipment				X
Use indoor mapping				X

Using the roles to determine the system requirements was a useful exercise. As you can see from the chart, some requirements support multiple roles. Some requirements were determined not to be core requirements at all, but have been deferred instead as wishes for future versions.

This type of chart doesn't reflect that some requirements are basic capabilities for this type of app, regardless of the roles that we defined.

Interface Design

A basic understanding of the requirements for the application gave Liz McVoy, the Digital Media Specialist, the ability to design the interface using a largely visual approach. Liz designed screens that reflect the requirements.

Introductory Screen



Figure 3. The introductory screen for the new tour app [click to enlarge](#)

The introductory screen is designed to entice the student to explore the library. There is a single button, labeled "BEGIN", to start the tour. The color scheme, referenced throughout the interface, is that of a fantastic sunset behind Hokie stone.

Main Interface

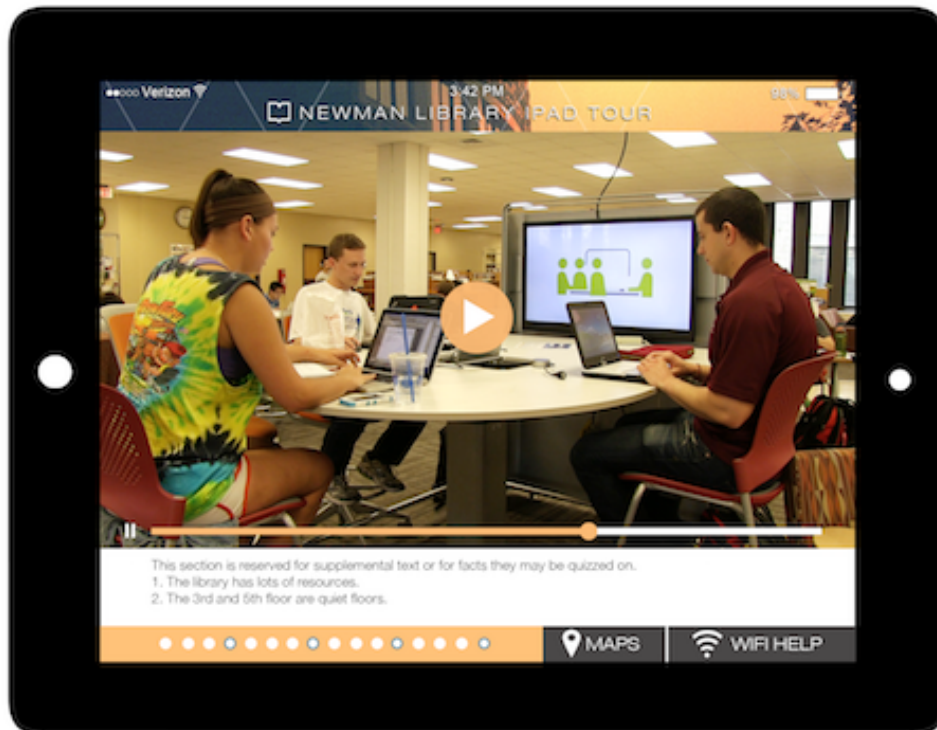


Figure 4. The main user interface for the tour app [click to enlarge](#)

The main interface is the primary screen that students will see while taking the tour. This main screen has several elements taken from the requirements document:

- a large window that displays videos for each tour stop
- a “MAPS” button that displays maps of the library
- a “WIFI HELP” button to help the student with connectivity issues
- a display area to show important library facts

There is also a progress bar to indicate the remaining stops on the tour.

Explore Prompt

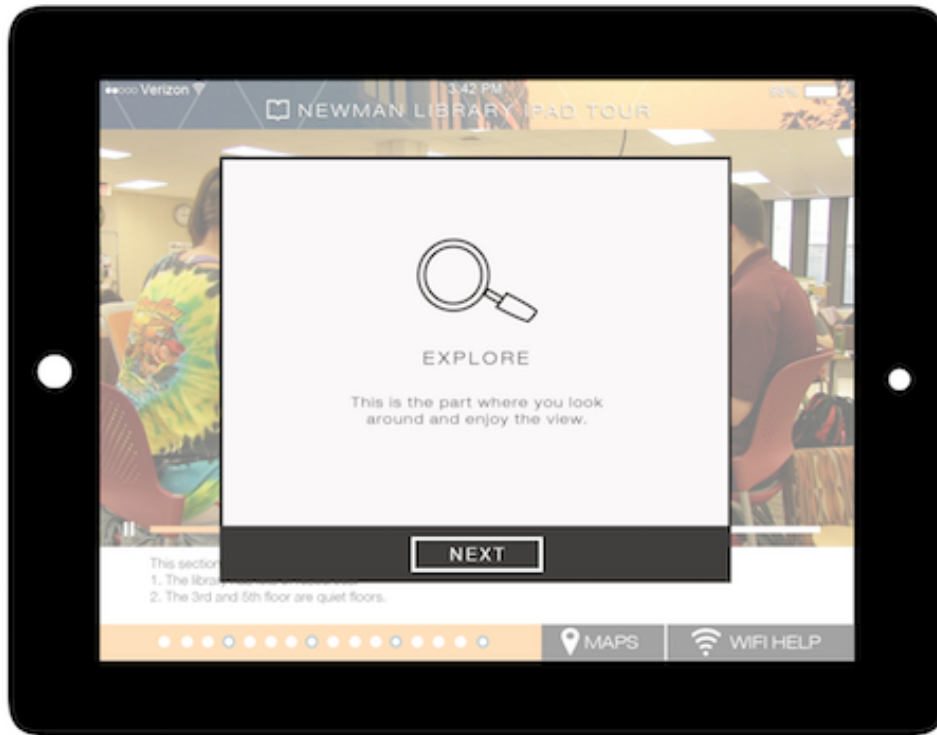


Figure 5. An example of a prompt in the tour interface click to enlarge

This prompt appears during the tour when students are asked to look at specific details in the environment.

Maps

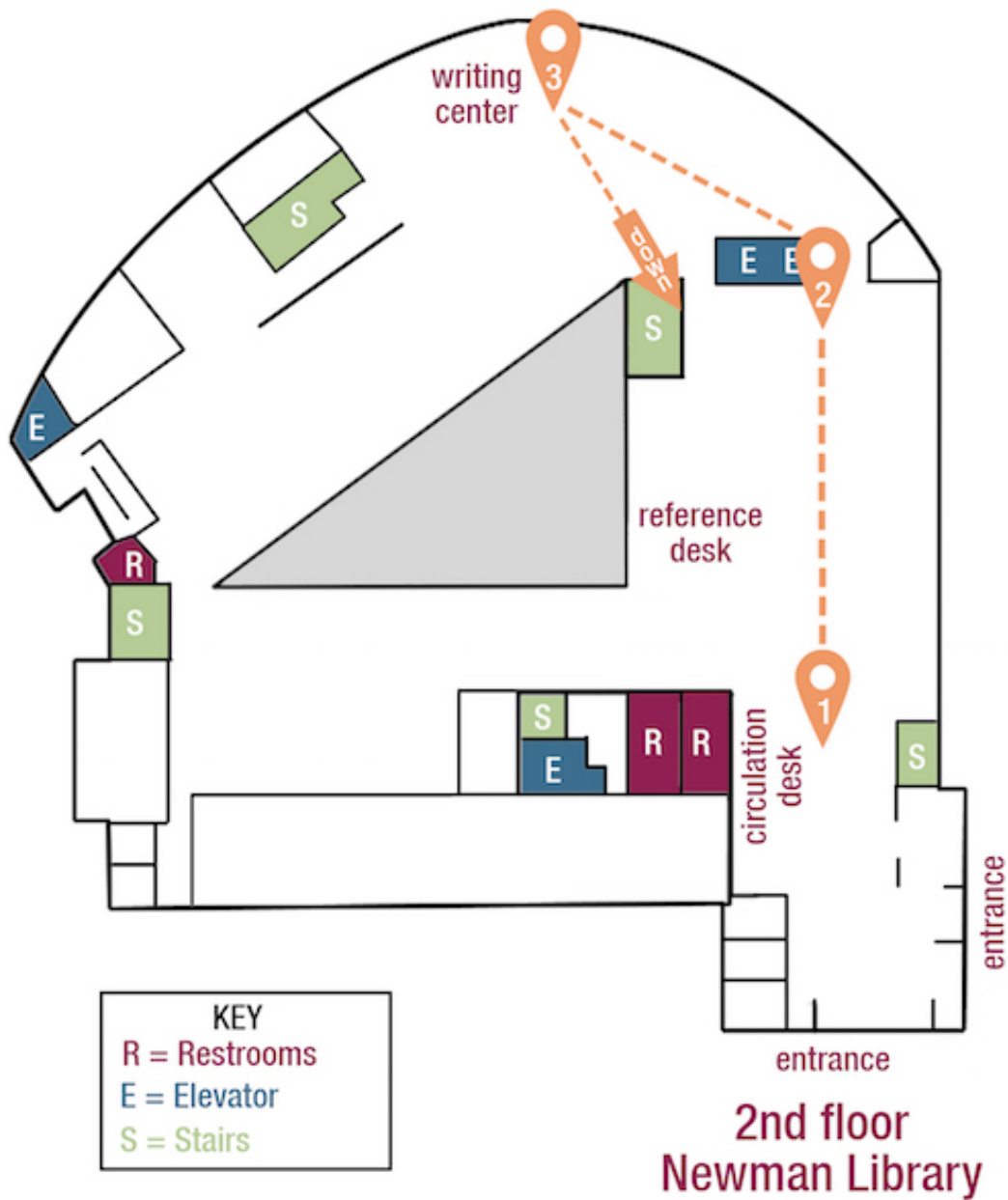


Figure 6. An in-app map of the second floor of Newman library, with tour locations marked [click to enlarge](#)

The app has maps of each library floor on the tour, with paths and numbered markers showing how to get to each tour stop. Maps are potentially important, because even people who work in the library occasionally get lost. The iBeacon technology allows for the possibility of indoor mapping using signal triangulation to detect location more precisely, and to give directions. The current version of the app doesn't use this ability; instead, the beacons associated with each tour stop location are stored in a list. The correct video plays when the student is near the correct beacon for the next stop.

Wi-fi Help Screen



Figure 7. The wi-fi help screen

This screen was included in design sessions because in the past, the library has often had problems with wi-fi connectivity, especially when moving throughout the library. The QR code version of the tour has a video that shows students the steps to perform on the device when there are wi-fi problems. It might be possible to eliminate this screen in the deployed implementation, because the videos are now stored on the device instead of streamed via wi-fi.

An Internet connection is still required in order for a student to sign-in and take the quiz, but this is done at the end of the tour in an area with reliable wi-fi coverage.

Sign-in Screen

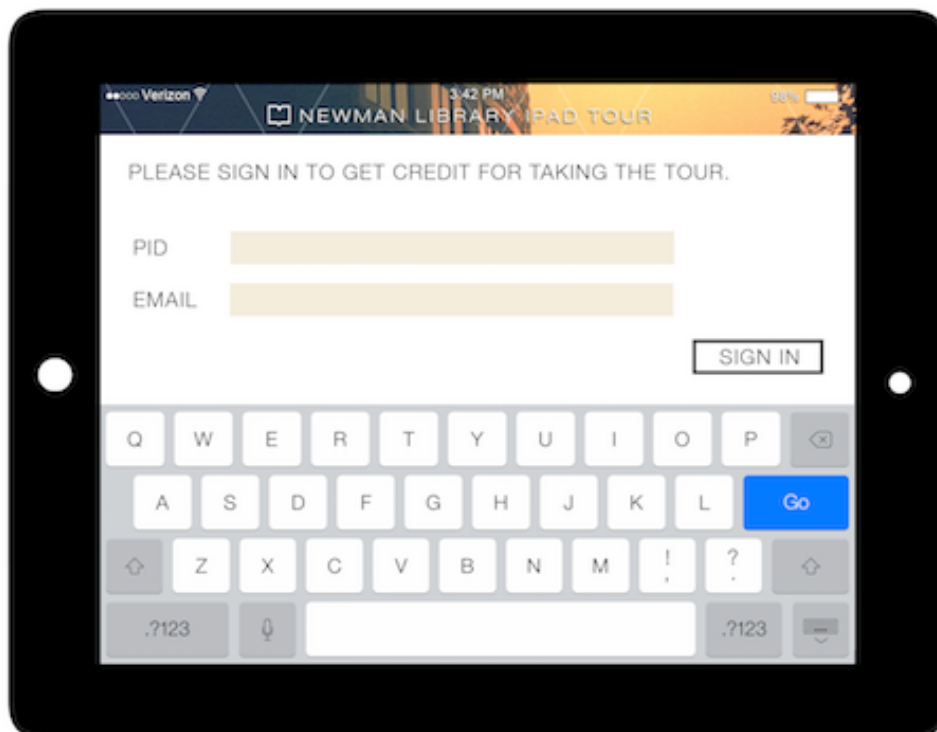


Figure 8. Student sign-in is required for students to get credit for the quiz

This screen is used by students to “sign-in” to the tour. After the tour is complete, the tour will record the email address of each student so that instructors can keep track. After signing in, students will take the quiz.

Quiz Interface

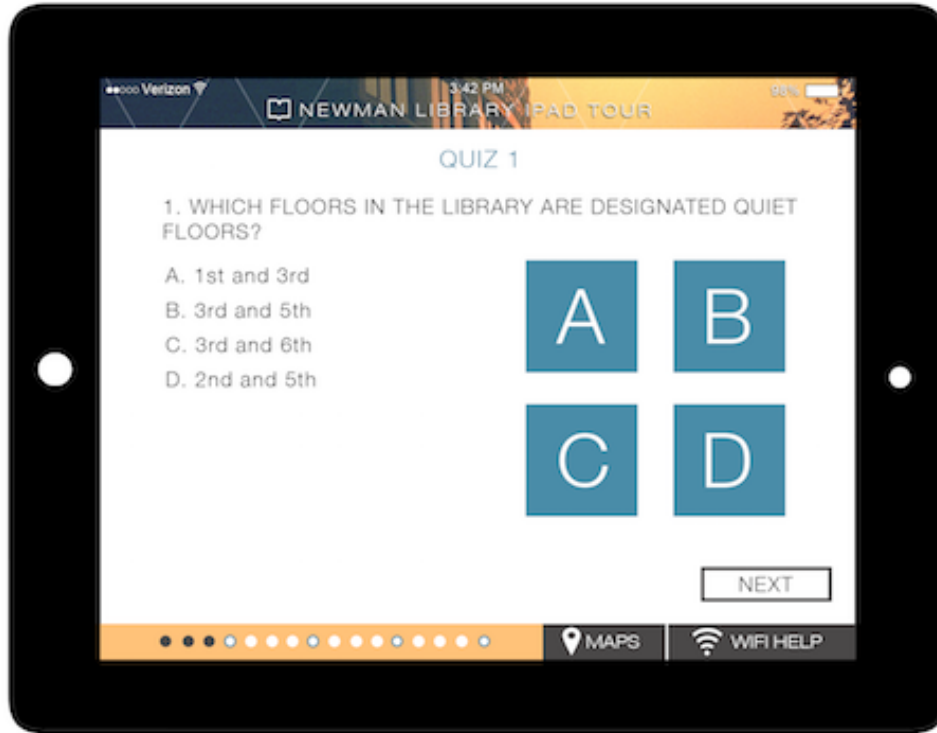


Figure 9. The multiple-choice quiz interface [click to enlarge](#)

The quiz interface presents students with a series of multiple choice questions. The questions that are asked were covered in the tour stop videos, and in many cases, the answers to these questions were shown in the display area of the main interface. The buttons to answer the questions are very large, to accommodate people with big fingers, while still allowing for use by people with small fingers. Completing the quiz leads to the final screen.

Final Screen

There has been intense debate about whether balloons should be used as the final image, or fireworks. Currently, the balloon proponents are winning the debate.

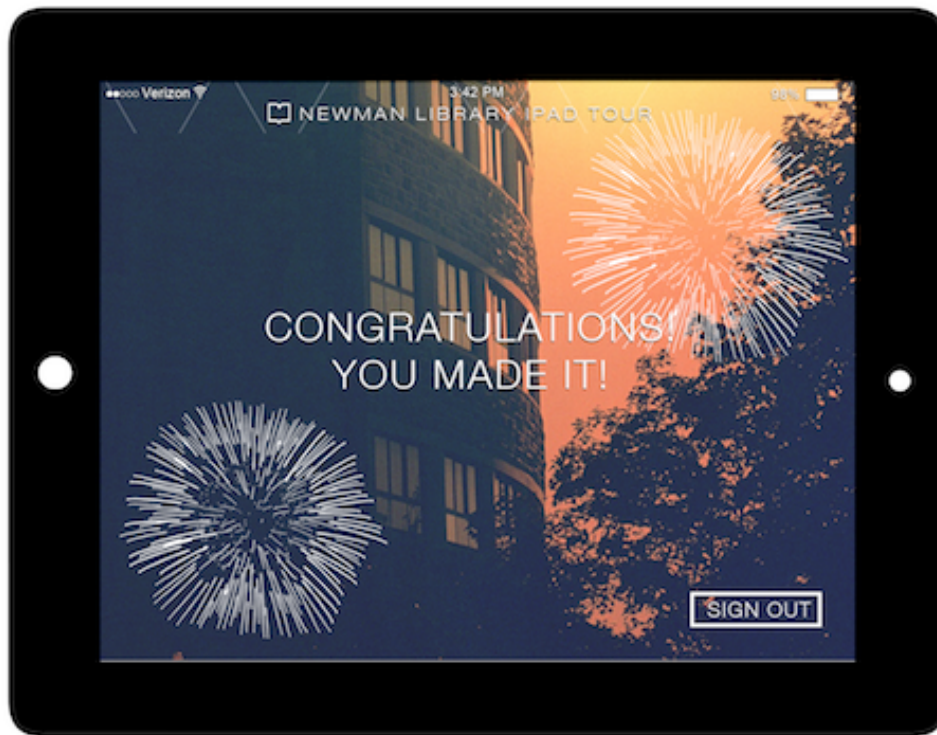


Figure 10. The final screen, shown after the tour is completed [click to enlarge](#)

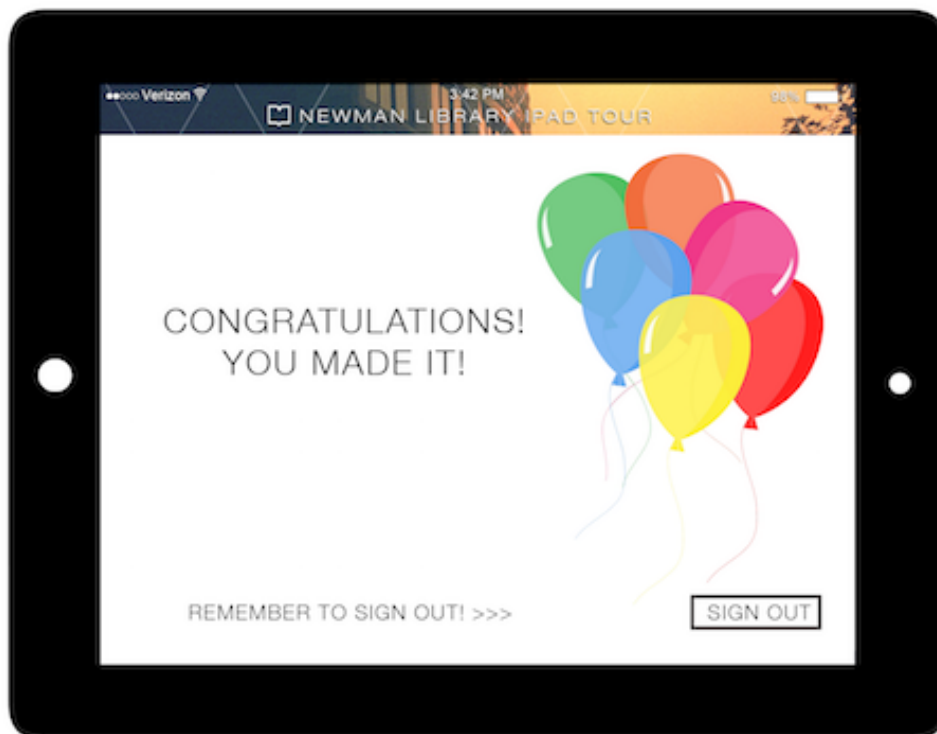


Figure 11. An alternate final screen, with balloons instead of fireworks [click to enlarge](#)

Reboot and Mini-Hackathon

After the project had languished for some time due to lack of resources, the project was rebooted. Jonathan Bradley, the Web Learning Environments Application Developer, became the main developer on the project. After observing Amanda's success in managing and completing other projects with limited resources, Keith Gilbertson, another developer on the project, attempted to

get Amanda French involved in the project.

Amanda held a reboot meeting and clarified requirements on the app. From the meeting, each person was finally assigned a set of action items. She created a project in Virginia Tech's GitLab environment, documented the project, created an issue tracker, and organized a mini-hackathon and proof-of-concept demo.

During the hackathon, Jonathan and Keith worked on coding what we perceived as the most important pieces of the app: the code to detect the beacons, and the code to detect the appropriate video.

Shortly after the hackathon, the developers were able to present a demonstration to the rest of the team. Beacons were placed in several locations throughout the library. The team walked from beacon to beacon, and observed as the correct video played in each location.

During the hackathon and the resulting demonstration, we noticed three interesting issues. First, in the time it takes to walk from station to station, especially when the stops are between floors, the iPad screen would go blank. When the screen went blank, the app would stop searching for beacons. Our current workaround for this is a simple one. The iPads are library owned, and we have chosen to disable automatic screen blanking on the iPads.

Second, the beacons had been placed approximately one half hour before we started our demonstration. As we approached a stop on the first floor, where a beacon had been placed in an out of the way area, but not secured, we observed a student handling and examining the beacon. In recognition that brightly colored technology devices might be interesting to visitors at a university with a large number of technology students, secure, 3D printed mounting devices have been proposed. The mounts will also help with another issue – sometimes the adhesive isn't strong enough to keep the units attached directly to certain surfaces.

Third, we learned that the iPad units that were allocated for library circulation at the time would not work with our app. We initially tested the Estimote SDK on a newer, personal iPad Air 2 model successfully. The same test failed when used with the older iPad 2 models owned by the library. After researching the issue, we found that the iPad2 model is not compatible with the Bluetooth Low Energy specification, meaning that it will not work with the iBeacon technology. However, new iPad Air 2 units were being ordered for the library, and they arrived before we would finish work on the app.

After the hackathon, Amanda was no longer available to work on the project due to the workload in her primary areas of responsibility. However, her work was useful in organizing and defining the project, and helping us to make progress.

Implementation Choices

Jonathan Bradley became the lead developer on the project around the time of the Hackathon, and as evidenced by e-mail timestamps, at times had to work in the wee hours. Swift was chosen over Objective C as the programming language for the iOS app. This is primarily because iOS development in general tends to be shifting over to Swift, which has been open-sourced. Swift also has a reputation for being easier to read and easier to maintain. (Solt 2015). The Estimote SDK required an Objective C bridging header in order to be used with Swift. The procedure is documented at <http://developer.estimote.com/ibeacon/tutorial/part-1-setting-up/#swift-users-add-an-objective-c-bridging-header>.

We chose to use the latest, non-beta version of XCode as our development tool. XCode is the standard development tool from Apple for creating iOS applications, and requires a system with OS X to run. By choosing the latest release version, but avoiding beta we are able to use new features without sacrificing stability, and can test on devices with release versions of iOS. However, due to the unexpected timeline of the project, the beta versions of XCode eventually made it to release status, and this required changes to our code as the relatively new Swift language has evolved.

Source code for the project is stored in our GitLab repository, and will be open-sourced and placed on GitHub when proven.

We are using Estimote beacons, primarily because they were the first available when the project was envisioned, but we have decided to stick with them for the initial implementation because of a strong SDK and a large developer community.

This version of the app is loaded manually onto library-owned iPads, instead of being distributed through the app store.

Learning Objectives

While designing the new version of the app, Learning Services came up with a set of eight core learning objectives.

1. By the end of this tour, the student will identify the ways in which they can receive reference help (Ref Desk, College Librarians, Ask a Librarian Chat).
2. By the end of this tour, the student will differentiate how Special Collections is different from the rest of the library, by identifying its unique mission and collection specifics.
3. By the end of this tour, the student will list the type of items that can be checked out through Circulation Services.
4. By the end of this tour, the student will compare and contrast between different study spaces and the benefits of each.
5. By the end of this tour, the student will be able to find library materials using Summon.
6. By the end of this tour, the student will differentiate the services offered by the Comm Lab, Writing Center, and 3D printing studio.
7. By the end of this tour, the student will explain the basics of using the library print collection (reading call nos., library terminology, returning unshelved books, classifications).
8. By the end of this tour, the student will describe how to print from their devices and locate the printers in the library.

The learning objectives, along with the tour stops, were used to create a script for the new tour videos.

Videos

As part of the overall renewal of the self-guided library tour, new videos were desired. There has been some difficulty in making this happen. The library has video equipment and an event capture team that, upon request, records and edits events around the university. They've been booked solid. Additionally, creating new videos requires collaboration between educators, who determine learning objectives, scriptwriters, video recording staff, and editors. As mentioned earlier, collaboration across functional groups within the library is often difficult.

In order to make progress on the project, an outside consultant was contracted. The outside consultant is a neighbor of one of the project members, and was hired from money out of his own pocket. While this was useful in making progress on the project, we are not recommending it as a general practice for developing this type of application! Use library funds when possible.

Fortunately, the consultant that was hired is an experienced videographer and has won competitions in creating short videos. Armed with the learning objectives, a list of tour stops, and scripts from the old videos, the videographer consultant has been recording new videos. While the original videos were meant to be primarily auditory in nature, the new videos are meant to have some visual appeal and humor.

Here's a draft of one of the new videos.



Figure 12. Old-timey lettering in a screenshot from one of the new tour videos

Note the young sounding voice in the video. Hypothetically, a younger voice will appeal to our “typical” college audience, but this is an accidental experiment. Our videographer is 15 years old! She has offered to have the videos narrated by her father, if the young voice doesn’t work out, or by Neal, the previous narrator, who has a professional-sounding, radio-quality voice.

Operational Challenges

We’ve struggled with many of the same issues in deploying a beacon app as the Brooklyn Museum – including lost beacons, and beacon confusion. (Bernstein 2015) While the availability of different beacon colors is not as much of an issue in our deployment as in that of an art museum, where bright colors may act as visual distractions from exhibits, we wished that the beacons would either blend into our walls, or come in our school colors of maroon and orange.

Battery Replacement

We have two different revision numbers of the Estimote beacon hardware, and the differing revisions use different battery types. The initial beacons, which were considered beta hardware, but will be used in our production application, use CR2477 coin batteries. The newer Estimote beacons use CR2450 batteries. We need to keep a supply of both types of batteries on hand.

The older, beta beacons require a minor surgery, and must be sliced open in order to replace the batteries. In the newer beacons, the adhesive backing can be separated from the main unit, and the battery is easily replaced.



Figure 13. A cross-section of the newer revision of Estimote beacons, showing the battery location. Photo courtesy of Estimote, Inc.

We don’t yet have statistics for how often the batteries will need to be changed, but an initial estimate is every four months. The devices broadcast the ID repeatedly after a specified advertising interval. Estimote initially claimed a battery life of two years, but this was based on an advertising interval of 900ms, while Apple recommends a much shorter, and energy expensive, advertising interval of 100ms. However, we can control power expenditure somewhat by adjusting the signal strength of the beacons, instead of just the advertising interval. We’re optimistic because of the current design of our application. We’re not using indoor location services. Each stop is far away from other beacons, and will be visited in a specific order. This combination of characteristics means that we are able to set the broadcast power quite low on the beacons, which will extend battery life.

Updating Videos at Existing Stops

In order to support the requirement that video content can be updated easily by the library, the app has some built-in functionality to check for new videos when it starts. Videos are stored in Google Drive.

When the library needs to update content for a specific stop, we will place a new video with the same name into the Google Drive folder. The library will then restart the app on the library-owned iPads, and the new video will be detected and downloaded.

As the videos are now pre-loaded, this reduces the need for dependable wi-fi at all stops. Videos are large, but the current version of the application runs only on library owned devices, so we don't yet need to worry about taking too much storage space on the device.

Adding New Stops

Adding a new stop to the tour is currently a heavyweight operation that requires many hands and many steps:

1. Write a narrative script for the new stop
2. Produce a new video for the stop
3. Deploy a new iBeacon device at the stop location
4. Add the stop information to the app code and recompile
5. Redeploy the app to the library iPads

All of the stops on the tour, along with information for the associated beacons and videos are currently hard-coded into the app.

```
1 // This represents a stop on the tour
2 struct TourStop {
3     let major: CLBeaconMajorValue
4     let minor: CLBeaconMinorValue
5     let identifier: String
6     let videoName: String // filename of video, without .mp4 extension
7 }
```

The stops are ordered according to their position in an array:

```
1 let tourBeacons = [
2     TourStop(major: 10, minor: 5, identifier: "reading_room", videoName: "04_readingroom"),
3     TourStop(major: 10, minor: 1, identifier: "special_collections", videoName: "05_spec"),
4     TourStop(major: 20, minor: 5, identifier: "reference_desk", videoName: "09_refdesk"),
5 ]
```

We anticipate that we will be able to eliminate the recompilation and redeployment steps by writing code that retrieves stop information from a network location at the start of the tour.

Collecting Student Data

The initial version of the app uses the the same method for collecting student data as the original self-guided tour. Students log into a web page and take a quiz.

This is where we hope that future versions of the app will have improvements that help the students, teacher, and library. The goal is to automatically record grades into Canvas, our new university-wide learning management system.

Tracking Beacon ID Numbers

While coding for the post-hackathon demonstration, one morning the app stopped working. After troubleshooting, Jonathan discovered that the ID numbers on the beacons were different from the ID numbers that had been coded into the app. Either the ID numbers had been reset when the batteries were replaced, or we accidentally mixed up our beacons.

We decided that, especially for deployment and operational purposes, we would need a way to track the ID that the app expects for each beacon location. We have a Google Drive document for this purpose, which looks like the following:

Location	Color	Real color	UUID	Major	Minor
Reference Desk	Mint Cocktail	Light green	library default	20	1
Special Collections	Icy Marshmallow	Light blue	library default	10	1
4th Floor	Blueberry Pie	Purple	library default	40	1
Learning Commons	Mint Cocktail	Light green	library default	20	5

The major number corresponds to a particular floor in the library. Major number 10 corresponds to the first floor, 20 corresponds to the second floor, 30 to the third floor, and so on. The minor number, (when taken along with the major number), corresponds to a specific stop on a specific floor. The beacon with major number 20 and minor number 1 corresponds to the reference desk on the second floor; the beacon with major number 20 and minor number 5 corresponds to the learning commons on the second floor. For the purposes of our current app, it isn't necessary for all of the beacons on the same floor to share the same major number, but the organization helps our mental model and will make things easier if we ever decide to create an app that uses the beacons for indoor mapping. We've also left large gaps in the sequence for minor numbers on each floor, in case we add more beacons later.

An oddity of the chart is the existence of both "Color" and "Real color" columns. The "Color" column displays the Estimote name for the beacon color. The Estimote names, such as "Icy Marshmallow", were confusing to us, so we added a "Real color" column. The "Real color" is the basic color of the beacon, to the eyes of a person that does not have color blindness. An Icy Marshmallow beacon is actually light blue. If we ever paint the beacons, we will update the "Real color" column. We kept the "Color" column, because the beacon reports this color information in Estimote apps that are used for beacon configuration.

This document is useful in case a beacon needs to be swapped in temporarily for another that is having its battery replaced; it makes it easier to set the ID of the replacement beacon to the correct settings. Unfortunately, the app doesn't read directly from the document at this time. It serves only as an aid for humans to maintain the beacons.

Beacons can be easy to lose. One hectic morning, one of the developers who had a slight problem with insomnia had taken home a set of beacons and was working on the project at odd hours. While packing up to go to the office, he noticed that one of the beacons was missing. The "Icy Marshmallow" beacon was eventually found with the aid of a Bluetooth scanning application in a box of cereal on the pantry shelf.

Beacon Security

We are having some early problems with beacons staying on the wall, possibly due to the adhesive on the backs of the beacons. Two beacons went completely missing during testing. We have proposed, but not yet designed or printed custom mounts for the beacons using our 3D print lab.

Future Opportunities

Additional Languages

Our Assistant Director for International Outreach has proposed an idea to increase the usability of the tour for new international students. Some international students arrive at Virginia Tech fluent in English; others have difficulty listening in real time to fast conversations that have domain specific terminology, such as conversations about libraries. For example, think about the terms "stacks", "interlibrary loan", "course reserves", "circulation", and "journal database."

In order to facilitate the introduction of the library to these students, the director has suggested making the tour available in other languages, in particular Arabic, Spanish, and Chinese. This could be done incrementally. First, the Subtitles for the Deaf and Hard of Hearing could be translated into written representations of these other languages.

Students would hear the videos in English while seeing subtitles in a preferred language. As a next step, the audio for each stop could be recorded again in each language, and students would be asked at the beginning of the tour to select a preferred language so that the appropriate audio track could be used. The International Outreach Director has offered to recruit students to translate and record the videos.

Student-owned devices

This version of the tour was developed for the iPad Air 2 devices owned by the library. Having a single known device eased development, because we didn't need to be concerned with multiple screen sizes, multiple operating systems, or with differing video playback capabilities for each type of device. For future updates, we have in mind that students would be able to bring their own devices into the library and take the self-guided tour.

The first update would allow the tour to be taken on all iOS devices that students might bring, including iPhones. The next update

would allow the app to run on Android devices, including tablets and phones. The Android update would require significantly more developer time, as the Android APIs are based on an implementation of the Java language, and all of the code would be new.

Making the tour available to student owned devices would also mean that the app would be distributed by the app stores for each platform. Currently, the tour app is loaded manually onto the iPad Air 2 tablets that belong to the library.

References

Bernstein, Shelly. The Realities of Installing iBeacon to Scale. [Internet]. 2015 Feb 04. Available from <https://www.brooklynmuseum.org/community/blogosphere/2015/02/04/the-realities-of-installing-ibeacon-to-scale/>

Freshman Snapshot [Internet]. [Updated 2015 May 15]. Virginia Tech. Available from: <http://www.admiss.vt.edu/apply/freshman-snapshot/>

Getting Started with iBeacon [Internet]. 2014 June 02. Apple, Inc. Available from: <https://developer.apple.com/ibeacon/Getting-Started-with-iBeacon.pdf>

iBeacon [Internet]. [Updated 21 Feb 2016]. Wikipedia. Available from: <https://en.wikipedia.org/wiki/IBeacon>

Solt P. 2015 May 11. [Internet]. It's high time to make the switch to the more approachable, full-featured Swift for iOS and OS X app dev. InfoWorld. Available from: <http://www.infoworld.com/article/2920333/mobile-development/swift-vs-objective-c-10-reasons-the-future-favors-swift.html>

About the Authors

Dr. Jonathan Bradley is the Web Learning Environments Application Developer at Virginia Tech's Newman Library. Jonathan creates educational technologies and trains faculty and staff in their use, both inside and outside of the library. He is the technical admin for Virginia Tech's LibGuides instance, and most of his previous research has surrounded pedagogy at the university level.

Neal Henshaw is the Educational Technology and Instructional Designer at Virginia Tech's Newman Library. Neal designed and maintains the self-guided tour that freshman take, and also creates tutorials for use in online instruction. Neal has a background in education in both the public and private sectors, as well as experience as a server/network administrator.

Liz McVoy lends her keen design and film eye to Virginia Tech's University Libraries as the Digital Media Specialist. She has a strong background in video production and graphic design, and combines these skills to create impactful, visually exciting film and design pieces that break down the invisible "fourth wall" to reach students, faculty, staff, and other users around Virginia Tech and beyond. Part of her role is to manage the Event Capture Service, which offers the recording and storing of scholarly lectures in the University Repository, VTechWorks. This service invites people all around the University into a new relationship with the Libraries, highlighting just one of the ways we can partner with them to learn and be more successful.

Former CLIR Postdoctoral Fellow **Amanda French** is currently Director of Digital Research Services and Associate Professor at Virginia Tech University Libraries, where she is helping to build digital humanities infrastructure and is running the institutional repository, VTechWorks. From 2010-2014 she was THATCamp Coordinator and Research Assistant Professor at the Roy Rosenzweig Center for History and New Media at George Mason University. Her particular expertise consists of making humanities content (both cultural content and scholarly interpretation of that content) openly available online, as well as introducing scholars to the various methods of and issues with making humanities content openly available online.

Keith Gilbertson serves as a Digital Technologies Development Librarian at Virginia Tech.

Lisa Becksford is the Learning Services and Educational Technologies Librarian at Virginia Tech University Libraries, where she provides face-to-face and online learning experiences that help students across the university grow as researchers and scholars.

Elisabeth Givens is the Director of Digital Strategy and Outreach at Virginia Tech's University Libraries, where she oversees the development and implementation of marketing initiatives and outreach campaigns that serve to connect the Virginia Tech community with the services and resources provided by the libraries. Prior to Virginia Tech, Elisabeth was the Senior Social Catalyst at 919 Marketing, a national content marketing agency based in Raleigh, NC, where she worked with clients of all sizes, from startup companies to Fortune 500 corporations.

This work is licensed under a [Creative Commons Attribution 3.0 United States License](#).

